HAL
open science

Versatile and scalable cosparse methods for physics-driven inverse problems

Srđan Kitić, Siouar Bensaid, Laurent Albera, Nancy Bertin, Rémi Gribonval

To cite this version:

Srđan Kitić, Siouar Bensaid, Laurent Albera, Nancy Bertin, Rémi Gribonval. Versatile and scalable cosparse methods for physics-driven inverse problems. 2017. hal-01496767v1

HAL Id: hal-01496767

https://inria.hal.science/hal-01496767v1

Preprint submitted on 27 Mar 2017 (v1), last revised 16 Jul 2017 (v2)

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.

Versatile and scalable cosparse methods for physics-driven inverse problems

Srđan Kitić, Siouar Bensaid, Laurent Albera, Nancy Bertin and Rémi Gribonval

Abstract Solving an underdetermined inverse problem implies the use of a regularization hypothesis. Among possible regularizers, the so-called *sparsity* hypothesis, described as a *synthesis* model of the signal of interest from a low number of elementary signals taken in a dictionary, is now widely used. In many inverse problems of this kind, it happens that an alternative model, the *cosparsity* hypothesis (stating that the result of some linear *analysis* of the signal is sparse), offers advantageous properties over the classical synthesis model. A particular advantage is its ability to intrinsically integrate physical knowledge about the observed phenomenon, which arises naturally in the remote sensing contexts through some underlying partial differential equation. In this chapter, we illustrate on two running examples (acoustic source localization and brain source imaging) the power of a generic cosparse approach to a wide range of problems governed by physical laws, how it can be adapted to each of these problems in a very versatile fashion, and how it can scale up to large volumes of data typically arising in applications.

Srđan Kitić
Technicolor, Cesson-Sévigné, e-mail: srdan.kitic@technicolor.com

Siouar Bensaid
Université de Rennes 1 (LTSI), Rennes, e-mail: siouar.bensaid@univ-rennes1.fr

Laurent Albera
Université de Rennes 1 (LTSI), Rennes, e-mail: laurent.albera@univ-rennes1.fr

Nancy Bertin
IRISA, Rennes, e-mail: nancy.bertin@irisa.fr

Rémi Gribonval
Inria Rennes - Bretagne Atlantique, Rennes, e-mail: remi.gribonval@inria.fr

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

1 Introduction

Inverse source problems consist in inferring information from an object in an indirect manner, through the signals it emits or scatters. This covers in particular *remote sensing*, a term coined in Earth sciences [11], to indicate acquisition of shape or structure data of the Earth's crust, using signal processing techniques. More generally, remote sensing considers any method that collects distant observations from an object, with a variety of possible signal modalities and imaging modes (active or passive) that usually determine the applied processing technique.

Remote sensing and inverse problems encompass a wide range of practical applications, many of which play essential roles in various parts of modern lifestyle: medical ultrasound tomography, electro-encephalography (EEG), magnetoecephalography (MEG), radar, seismic imaging, radio astronomy...

To address inverse problems, an important issue is the apparent shortage of observed data compared to the ambient dimensionality of the objects of interest. A common thread to address this issue is to design low-dimensional models able to capture the intrinsic low-dimensionality of these objects, while allowing the design of scalable and efficient algorithms to infer them from partial observations.

The sparse data model has been particularly explored in this context [8, 74, 77]. It is essentially a generative *synthesis* model describing the object of interest as a sparse superposition of elementary objects (*atoms*) from a so-called *dictionary*. Designing the atoms in a given application scenario can be challenging. As documented in this chapter, exploiting dictionaries in the context of large-scale inverse problems can also raise serious computational issues.

An alternative model is the so-called *analysis* sparse model, or *cosparse* model, whereby the sparsity assumption is expressed on an analysis version of the object of interest, resulting from the application of a (differential) linear operator. As we will see, this alternative approach is natural in the context of many inverse problems where the objects of interest are physical quantities with properties driven by conservation or propagation laws. Indeed, the fact that such laws are expressed in terms of partial differential equations (PDEs) has several interesting consequences. First, using standard discretization schemes, the model (which is embodied by an *analysis operator* rather than a dictionary) can be directly deduced from the knowledge of these PDEs. Moreover, the resulting model description is often very concise and the associated linear analysis operator is very sparse, leading to efficient computations. The framework thus fits very well into iterative algorithms for sparse regularization and large-scale convex optimization. Finally, the framework is adaptable to difficult settings where, besides the object of interest, some other "nuisance" parameters are unknown: uncalibrated sensors, partially known impedances, etc.

To demonstrate the scalability and versatility of this framework, this chapter uses as running examples two practical scenarios involving two types of signals (namely acoustic and electroencephalographic) for illustration purposes. The generic modeling and algorithmic framework of physics-driven cosparse methods which is described here has however the potential to adapt to many other remote sensing situations as well.

2 Physics-driven inverse problems

Many quantities of interest that are measured directly or indirectly by sensors are intimately related to propagation phenomena governed by certain laws of physics. Linear *partial differential equations (PDEs)* are widely used to model such laws including sound propagation in gases (acoustic wave equation), electrodynamics (Maxwell equations), electrostatic fields (Poisson's equation), thermodynamics (heat equation) and even option pricing (Black-Scholes equation), among many others. When coupled with a sparsity assumption on the underlying sources, these lead to a number of feasible approaches to address physics-driven inverse problems.

2.1 Linear PDEs

Hereafter, $\boldsymbol{\omega}$ denotes the coordinate parameter (e.g., space r and/or time t) of an open *domain* $\boldsymbol{\Theta}$. Linear PDEs take the following form:

$$\mathfrak{D}\boldsymbol{x}(\boldsymbol{\omega}) := \sum_{|d|\leq\zeta} \boldsymbol{a}_d(\boldsymbol{\omega}) D^d \boldsymbol{x}(\boldsymbol{\omega}) = \boldsymbol{c}(\boldsymbol{\omega}), \quad \boldsymbol{\omega} \in \boldsymbol{\Theta}, \tag{1}$$

where $\boldsymbol{a}_d$, $\boldsymbol{x}$ and $\boldsymbol{c}$ are functionals of the variable $\boldsymbol{\omega}$. Typically one can think of $\boldsymbol{x}(\boldsymbol{\omega})$ as the propagated field and $\boldsymbol{c}(\boldsymbol{\omega})$ as the source contribution. The functional $\boldsymbol{a}_d$ denotes coefficients that may (or not) vary with $\boldsymbol{\omega}$.

Above, d is the multi-index variable with $|d| = \mathsf{d}_1 + \ldots + \mathsf{d}_\mathsf{l}$, $\mathsf{d}_\mathsf{i} \in \mathbb{N}_0$. For a given $d = (\mathsf{d}_1, \ldots, \mathsf{d}_\mathsf{l})$, $D^d \boldsymbol{x}(\boldsymbol{\omega})$ denotes the dth partial differential of $\boldsymbol{x}$ with respect to $\boldsymbol{\omega}$, defined as:

$$D^d \boldsymbol{x}(\boldsymbol{\omega}) := \frac{\partial^{|d|}\boldsymbol{x}}{\partial \omega_1^{\mathsf{d}_1} \partial \omega_2^{\mathsf{d}_2} \ldots \partial \omega_\mathsf{l}^{\mathsf{d}_\mathsf{l}}}.$$

In order for continuous $D^d \boldsymbol{x}(\boldsymbol{\omega})$ to exist, one needs to restrict the class of functions to which $\boldsymbol{x}(\boldsymbol{\omega})$, $\boldsymbol{\omega} \in \boldsymbol{\Theta}$ belongs. Such functional spaces are called *Sobolev spaces*. In this chapter, functionals are denoted by boldface italic lowercase (*e.g.* $\boldsymbol{f}$), while linear operators acting on these are denoted by uppercase fraktur font (*e.g.* $\mathfrak{D}, \mathfrak{L}$).

For linear PDEs, linear *initial conditions* and/or *boundary conditions* are also considered, and we denote them as $\mathfrak{B}\boldsymbol{x} = \boldsymbol{b}$. Finally, we compactly write (1), equipped with appropriate boundary/initial conditions, in linear operator form:

$$\mathfrak{L}\boldsymbol{x} = \boldsymbol{c}, \tag{2}$$

where $\mathfrak{L} := (\mathfrak{D}, \mathfrak{B})$ and $\boldsymbol{c} := (\boldsymbol{c}, \boldsymbol{b})$, by abuse of notation. For simplicity, we consider only *self-adjoint* operators[1] $\mathfrak{L}$. With regards to remote sensing, $\mathfrak{L}$, $\boldsymbol{x}$ and $\boldsymbol{c}$ represent the "channel", the propagated, and the "source" signal, respectively.

[1] The operators for which $\langle \mathfrak{L}p_1, p_2 \rangle = \langle p_1, \mathfrak{L}p_2 \rangle$ holds. Otherwise, setting the adjoint boundary conditions would be required.

2.2 Green's functions

While our final goal is to find $\boldsymbol{c}$ given partial observations of $\boldsymbol{x}$, let us assume, for now, that $\boldsymbol{c}$ is given, and that we want to infer the solution $\boldsymbol{x}$.

The existence and uniqueness of solutions $\boldsymbol{x}$ of PDEs, in general, is an open problem. These are subject to certain boundary and/or initial conditions, which constrain the behavior of the solution at the "edge" $\partial\Theta$ of the domain Θ. Fortunately, for many types of PDEs, the required conditions are known, such as those provided by Cauchy-Kowalevski theorem, for PDEs with analytic coefficients. Hence, we do not dwell on this issue - instead, we assume that the unique solution exists (albeit, it can be very unstable - PDEs represent archetypal ill-posed problems [38]).

Looking at (2), one may ask whether there exist an inverse operator $\mathfrak{L}^{-1}$, such that we can compute the solution as $\boldsymbol{x} = \mathfrak{L}^{-1}\boldsymbol{c}$. Indeed, in this setting such operator exists, and is the gist of the method of *Green's functions* for solving PDEs. The operator is (as expected) of integral form, and its kernel is given by the Green's functions $g(\omega, s)$, defined as follows

$$\begin{aligned} \mathfrak{D}\boldsymbol{g}(\omega, s) &= \delta(\omega - s), \quad s \in \Theta, \\ \mathfrak{B}\boldsymbol{g}(\omega, s) &= 0, \quad s \in \partial\Theta, \end{aligned} \tag{3}$$

where $\delta(\cdot)$ represents Dirac's delta distribution. In signal processing language, the Green's function can be seen as the response of the system (2) to the impulse centered at $s \in \Theta$.

If we assume that $\boldsymbol{b} = 0$ on $\partial\Theta$, it is easy to show that the solution of a linear PDE is obtained by integrating the right-hand side of (2). Namely, since

$$\mathfrak{L}\boldsymbol{x}(\omega) = c(\omega) = \int_{\Theta} \delta(s - \omega)\boldsymbol{c}(s)\boldsymbol{d}s = \mathfrak{L}\int_{\Theta} \boldsymbol{g}(s, \omega)\boldsymbol{c}(s)\boldsymbol{d}s, \tag{4}$$

we can identify $\boldsymbol{x}(\omega)$ with the integral. Note that $\boldsymbol{g}(s, \omega) = \boldsymbol{g}(\omega, s)$ for a self-adjoint operator $\mathfrak{L}$, and the latter can be put in front of integration since it acts on ω. When the boundary conditions are inhomogenous ($\boldsymbol{b} \neq 0$), the same approach can be taken except that one needs two types of Green's functions: one defined as in (3), and another with $\delta(\omega - s)$ placed at the boundary (but with $\mathfrak{D}\boldsymbol{g} = 0$, otherwise). Then, one obtains $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ from (4), and the solution $\boldsymbol{x}$ is superposition: $\boldsymbol{x} = \boldsymbol{x}_1 + \boldsymbol{x}_2$.

Since integration is again a linear operation, we can compactly represent (4) in the form of a linear operator:

$$\boldsymbol{x} = \mathfrak{G}\boldsymbol{c}, \tag{5}$$

and, by comparing it with (2), we can deduce that $\mathfrak{G} = \mathfrak{L}^{-1}$.

Green's functions are available in analytic form only for a restricted set of combinations of domain geometries/initial/boundary conditions. In such a case, evaluating $\boldsymbol{g}(\cdot, \cdot)$ is direct, but the integral (4) may still be difficult to evaluate to obtain the integral operator $\mathfrak{G}$. Functional approximations, such as using a *parametrix* [30] can be helpful, however in the most general case, one will have to resort to numerical approximations, as will be discussed in Section 4.

2.3 Linear inverse problem

The inverse problem we are interested in is the *estimation of the field* $\boldsymbol{x}$ (or, equivalently, of the source component $\boldsymbol{c}$) from a limited set of (noisy) field measurements acquired at a sensor array. In the case of a spatial field, such as a heat map, the coordinate parameter is spatial $\omega = r$ and each measurement is typically a scalar estimate of the field at m given locations, $\mathrm{y_j} \approx \boldsymbol{x}(r_\mathrm{j})$, perhaps after some spatial smoothing. In the case of a time series, the coordinate parameter is time $\omega = t$, and the measurements are obtained by analog-to-digital sampling (usually at a fixed sampling rate) at t time instants, corresponding to $\mathrm{y}_\ell \approx (\boldsymbol{h} \star \boldsymbol{x})(\mathrm{t}_\ell)$, where $\boldsymbol{h}(\mathrm{t})$ is a temporal filter, optionally applied for temporal smoothing. In the case of a spatiotemporal field, such as an acoustic pressure field, $\omega = (r, \mathrm{t})$ and the acquired measurement consist of multiple *time series* obtained by analog-to-digital sampling (at a fixed sampling rate) at a number of spatial locations, corresponding to $\mathrm{y}_{\mathrm{j},\ell} \approx (\boldsymbol{h} \star_t \boldsymbol{x})(r_\mathrm{j}, \mathrm{t}_\ell)$ with $\star_t$ a convolution along the temporal dimension. Except when the nature of ω is essential for discussion, we use below the generic notation $\mathrm{y_j} \approx \boldsymbol{x}(\omega_\mathrm{j})$.

Now, we consider a vector $y \in \mathbb{R}^{\mathsf{m}}$ (resp. $\in \mathbb{R}^{\mathsf{t}}$ or $\in \mathbb{R}^{\mathsf{m}\times\mathsf{t}}$) of measurements as described above. Without additional assumptions, recovering $\boldsymbol{c}$ or $\boldsymbol{x}$ given the *measurement vector* y only is impossible. Understanding that (2) and (5) are dual representations of the same physical phenomenon, we term such problems *physics-driven inverse problems*.

Algebraic methods. Rigorous algebraic methods for *particular* instances of inverse source problems have been thoroughly investigated in the past, and are still a very active area of research. The interested reader may consult, *e.g.* [38], and the references therein. A more generic technique (which doesn't prescribe a specific PDE), closely related to the numerical framework we will discuss in next sections, is proposed in [58]. However, its assumptions are strong (although not completely realistic), including in particular the availability of analytic expressions for the Green's functions. In addition, they must respect the approximate Strang-Fix conditions [25] and the source signal must follow the *Finite Rate of Innovation (FRI)* [81] model, *i.e.* it should be a weighted sum of Dirac impulses. In practice, it also requires a high *Signal-to-Noise Ratio (SNR)*, or sophisticated numerical methods in order to combat sensitivity to additive noise, and possibly a large number of sensors.

Sparse regularization. Fortunately, in reality, the support of the source contribution $\boldsymbol{c}$ in (1) is often confined to a much smaller subset $\Theta_0 \subset \Theta$ (*i.e.* $\boldsymbol{c}(\omega) = 0, \forall \omega \in \Theta_0^c$), representing *sources*, *sinks* or other singularities of a physical field. Two practical examples will be given soon in Section 3. This crucial fact is exploited in many regularization approaches for inverse source problems, including the one described in this chapter. Sparsity-promoting regularizers often perform very well in practice (at least empirically) [64, 65, 67, 50]. Even though there exist pathological inverse source problems so severely ill-posed that the source sparsity assumption alone is not sufficient [22, 19], such cases seem to rarely occur in practice. Hence, sparse regularization can be considered as an effective heuristics for *estimating* solutions of various inverse source problems, although not as an all-purpose rigorous methodology for all physics-driven inverse problems.

3 Running examples

As an illustration, in the rest of this chapter we consider two physics-driven inverse problems: acoustic source localization (driven by the acoustic wave equation) and brain source localization (driven by Poisson's equation).

3.1 Acoustic source localization from microphone measurements

The problem we are concerned with is determining the position of one or more sources of sound based solely on microphone recordings. The problem arises in different fields, such as speech and sound enhancement [32], speech recognition [3], acoustic tomography [56], robotics [79] and aeroacoustics [43]. Traditional approaches based on Time Difference of Arrival (TDOA) dominate the field [7], but these usually only provide the direction of arrival of the sound sources, and are generally sensitive to reverberation effects. We take a different approach, and use the physics of acoustic wave propagation to solve the localization problem.

The wave equation. Sound is the manifestation of acoustic pressure, which is a function of position r and time t. Acoustic pressure $\boldsymbol{x} := \boldsymbol{x}(r,\mathrm{t})$, in the presence of a sound source, respects the inhomogeneous acoustic wave equation:

$$\Delta \boldsymbol{x} - \frac{1}{\boldsymbol{v}^2}\frac{\partial^2}{\partial \mathrm{t}^2}\boldsymbol{x} = \boldsymbol{c}, \;\; (r,\mathrm{t}) \in \boldsymbol{\Theta} := \Gamma \times (0,\tau), \tag{6}$$

where Γ denotes the spatial domain, Δ is the Laplacian operator with respect to the spatial variable r and $\boldsymbol{v}$ is the speed of sound (around 334m.s^{-1} at room temperature, it may depend on space and/or time but is often approximated as constant). The right hand side $\boldsymbol{c} := \boldsymbol{c}(r,\mathrm{t})$ represents the pressure emitted by a sound source at position r and time t, if any (if a source is not emitting at some time instant t, then $\boldsymbol{c}(r,\mathrm{t})$ is zero at this source position; as an important consequence, $\boldsymbol{c} = 0$ everywhere, but at the source positions.)

Initial and boundary conditions. To ensure self-adjointness, we impose homogeneous initial and boundary conditions,

$$\forall r \in \Gamma,\; \boldsymbol{x}(r,0) = 0,\; \frac{\partial}{\partial \mathrm{t}}\boldsymbol{x}(r,0) = 0, \tag{7}$$

i.e. the acoustic field is initially at rest.

In addition, we may impose *Dirichlet* ($\boldsymbol{x}\mid_{\partial\Gamma} = 0$) or *Neumann* ($\nabla\boldsymbol{x}\cdot\boldsymbol{n}\mid_{\partial\Gamma} = 0$) boundary conditions, where $\nabla\boldsymbol{x}$ is the spatial gradient (with respect to r) and $\boldsymbol{n}$ is the outward normal vector to the boundary $\partial\Gamma$. A generalisation is the so-called *Robin* boundary condition which models reflective boundaries, or *Mur's* boundary condition [57]:

$$\forall r \in \partial\Gamma, \forall \mathrm{t}\colon\; \frac{\partial \boldsymbol{x}}{\partial \mathrm{t}} + \boldsymbol{v}\xi\nabla\boldsymbol{x}\cdot\boldsymbol{n} = 0, \tag{8}$$

where $\xi \geq 0$ is the specific acoustic impedance (again, possibly dependent on space and time but reasonably considered as fixed over time). For $\xi \approx 1$, Mur's condition *approximates* an absorbant boundary.

Inverse problem. An array consisting of m omnidirectional microphones, with a known geometry, outputs the measurements assembled into the vector $y \in \mathbb{R}^{\mathsf{mt}}$, where t is the number of time samples. Thus, we assume that the microphones output discrete signals, with an antialiasing filter and sampler applied beforehand. The goal is, ideally, to recover the fields $\boldsymbol{x}$ or $\boldsymbol{c}$ from the data y, using prior information that the sound sources are sparse in the spatial domain Γ.

Related work. Sound source localization through wavefield extrapolation and low-complexity regularization was first introduced by Malioutov et al. in [52]. They assumed a free-field propagation model, which allowed them to analytically compute the associated Green's functions. The narrowband sound sources were estimated by applying sparse synthesis or low-rank regularizations. A wideband extension was proposed in [51], which is, however, a two-stage approach that implicitly depends on solving the narrowband problem. The free space assumption was first abandoned by Dokmanić and Vetterli [24, 23], for source localization in the frequency domain. They used the Green's functions dictionary numerically computed by solving the Helmholtz equation with Neumann boundary conditions, by the *Finite Element Method (FEM)*. The wideband scenario was tackled as a jointly sparse problem, to which, in order to reduce computational cost, a modification of the OMP algorithm was applied. However, as argued in [16], this approach is critically dependent on the choice of frequencies, and can fail if modal frequencies are used. Le Roux et al. [48] proposed to use the CoSaMP algorithm for solving the sparse synthesis problem in the same spirit.

3.2 Brain source localization from EEG measurements

Electrical potentials produced by neuronal activity can be measured at the surface of the head using ElectroEncephaloGraphy (EEG). The localization of sources of this neuronal activity (during either cognitive or pathological processes) requires a so-called *head model*, aiming at representing geometrical and electrical properties of the different tissues composing the volume conductor, as well as a *source model*.

Poisson's equation. It is commonly admitted that the electrical potential $\boldsymbol{x} := \boldsymbol{x}(r)$ at location r within the human head mostly reflects the activity of pyramidal cells located in the gray matter and oriented perpendicularly to the cortical surface. This activity is generally modeled by current dipoles. Given the geometry and the scalar field $\{\boldsymbol{\sigma}(r)\}$ of electrical conductivities at location r within the head, Poisson's equation [9, 53] relates the electrical potential $\boldsymbol{x}$ and the current density j:

$$-\nabla.(\boldsymbol{\sigma}\,\nabla \boldsymbol{x}) = \nabla.j, \qquad r \in \Theta \tag{9}$$

where Θ is the spatial domain (interior of the human head) and $\nabla.j$ is the volume current. The operators $\nabla.$ and ∇ respectively denote the divergence and gradient with respect to the spatial variable r.

Boundary condition. We assume the Neumann boundary condition,

$$\boldsymbol{\sigma}\nabla\boldsymbol{x}.n = 0, \qquad r \in \partial\Theta, \tag{10}$$

which reflects the absence of current outside the human head.

Inverse problem. An array consisting of m electrodes located on the scalp (see Figure 2) captures the EEG signal $y = [\boldsymbol{x}(r_1), \ldots, \boldsymbol{x}(r_{\mathsf{m}})]^{\mathsf{T}} \in \mathbb{R}^{\mathsf{m}}$. The brain source localization problem consists in recovering from y the electrical field inside the head (with respect to some reference electrical potential), $\boldsymbol{x}$, or the current density, j, under a sparsity assumption on the latter.

Related work. Numerous methods for brain source localization were developed to localize equivalent current dipoles from EEG recordings. Among them, beamforming techniques [63], subspace approaches [71, 55, 1] and sparse methods [78] are the most popular. Regarding dictionary-based sparse techniques, the most famous is MCE (Minimum Current Estimate) [78], which computes minimum ℓ_1-norm estimates using a so-called *leadfield matrix*, corresponding to discretized Green's functions sampled at the electrode locations.

4 Discretization

A key preliminary step in the deployment of numerical methods to address inverse problems lies in the discretization of the quantities at hand, which amounts to convert the continuous PDE model (2) into a finite-dimensional linear model. A priori, any discretization method could be used within the regularization framework we propose; here, we limit ourselves to two families of them among the most common.

4.1 Finite Difference Methods (FDM)

The simplest way to discretize the original continous PDE is to replace the derivatives by *finite differences* obtained from their Taylor's expansion at a certain order, after discretization of the variable domain Θ itself on a (generally regular) grid. Consider a grid of discretization nodes $\{\omega^\ell\}_{\ell\in I}$ for the domain Θ and its boundary $\partial\Theta$. For each (multi-dimensional) index ℓ corresponding to the interior of the domain, the partial derivative $D^d\boldsymbol{x}(\omega^\ell)$ is approximated by finite linear combination of values of the vector $x = [\boldsymbol{x}(\omega^{\ell'})]_{\ell'\in I}$ associated to indices ℓ' such that $\omega^{\ell'}$ is in the neighborhood of ω^ℓ. The *stencil* defining these positions, as well as the order of the approximation, characterize a particular FDM. A similar approach defines approximations to partial derivatives at the boundary and/or initial points.

Example: the standard Leapfrog Method (LFM). As an example, we describe here the standard *Leapfrog Method (LFM)* applied to the discretization of a 2D, isotropic acoustic wave equation (6). Here, the domain Θ consists in 3 variables

(two spatial coordinates r_x, r_y and time t) and the corresponding PDE to be discretized only involves second-order derivatives of $\boldsymbol{x}$ with respect to these variables. By denoting $\mathrm{x}^{\tau}_{\mathrm{i,j}} := \boldsymbol{x}(\boldsymbol{\omega}^{\tau}_{\mathrm{i,j}})$ the field value at grid position $\boldsymbol{\omega}^{\tau}_{\mathrm{i,j}} := (r_x(\mathrm{i}), r_y(\mathrm{j}), \tau)$ the LFM approximation is (for $\tau > 2$, and excluding the boundaries):

$$\mathfrak{D}\boldsymbol{x}(\boldsymbol{\omega}^{\tau}_{\mathrm{i,j}}) = \left(\frac{\partial^2}{\partial r_x^2} + \frac{\partial^2}{\partial r_y^2} - \frac{1}{\mathrm{v}^2}\frac{\partial^2}{\partial t^2}\right)\boldsymbol{x}(\boldsymbol{\omega}^{\tau}_{\mathrm{i,j}}) \approx \frac{x^{\tau}_{\mathrm{i-1,j}} - 2x^{\tau}_{\mathrm{i,j}} + x^{\tau}_{\mathrm{i+1,j}}}{\mathrm{d}_{\mathrm{x}}^2} + \frac{x^{\tau}_{\mathrm{i,j-1}} - 2x^{\tau}_{\mathrm{i,j}} + x^{\tau}_{\mathrm{i,j+1}}}{\mathrm{d}_{\mathrm{y}}^2} - \frac{1}{\mathrm{v}^2}\frac{x^{\tau+1}_{\mathrm{i,j}} - 2x^{\mathrm{t}}_{\mathrm{i,j}} + x^{\tau-1}_{\mathrm{i,j}}}{\mathrm{d}_{\tau}^2} \quad (11)$$

where $\mathrm{d}_{\mathrm{x}}, \mathrm{d}_{\mathrm{y}}$ and d_{τ} denote the discretized spatial and temporal step sizes, respectively. This FDM scheme can be summarized as the use of a 7-point stencil centered at $x^{\tau}_{\mathrm{i,j}}$ in this case. It is associated to a finite-dimensional linear operator D such that Dx approximate the discretized version of $\mathfrak{D}\boldsymbol{x}(\boldsymbol{\omega})$ in the interior of the domain. The approximation error is of the order of $O(\max(\mathrm{d}_{\mathrm{x}}, \mathrm{d}_{\mathrm{y}}, \mathrm{d}_{\mathrm{t}})^2)$.

Similar formulas for boundary nodes are obtained by substituting a non-existent spatial point in the scheme (11) by the expressions obtained from discretized boundary conditions. For instance, for the frequency-independent acoustic absorbing boundary condition (8), proposed in [46], *e.g.* the missing point $x^{\tau}_{\mathrm{i+1,j}}$ behind the right "wall" is evaluated as:

$$x^{\tau}_{\mathrm{i+1,j}} = x^{\tau}_{\mathrm{i-1,j}} + \frac{\mathrm{d}_{\mathrm{x}}}{\mathrm{v}\mathrm{d}_{\tau}\xi_{\mathrm{i,j}}}\left(x^{\tau-1}_{\mathrm{i,j}} - x^{\tau+1}_{\mathrm{i,j}}\right), \quad (12)$$

When corners (and edges in 3D) are considered, the condition (8) is applied to all directions where the stencil points are missing. Combining (12) and (11) yields a linear operator B such that $Bx = b$ approximate the discretized version of $\mathfrak{B}\boldsymbol{x}(\boldsymbol{\omega}) = \boldsymbol{b}(\boldsymbol{\omega})$ on the initial/boundary points of the domain.

Concatenating D and B yields a square matrix $\boldsymbol{\Omega}$ (of size $\mathsf{n} = \mathsf{st}$ where s is the size of the spatial grid and t the number of time samples)

Using LFM to solve the discretized forward problem. While we are interested to use the above discretization to solve *inverse* problems, let us recall how it serves to address the *forward* problem, i.e. to estimate $\boldsymbol{x}$ when $\boldsymbol{c}$ is given. Under the assumption $Dx = c$, the leapfrog relation (11) allows to compute $\mathrm{x}^{\tau+1}_{\mathrm{i,j}}$ using $c^{\tau}_{\mathrm{i,j}}$ and values of x at two previous discrete time instants ($x^{\tau}_{(\cdot,\cdot)}$ and $x^{\tau-1}_{(\cdot,\cdot)}$). Similarly, homogeneous boundary conditions ($b = 0$) translate into relations between neighboring values of x on the boundaries and over time. For example, the above described discretization of Mur's boundary condition yields an explicit expression of $x^{\tau+1}_{\mathrm{i,j}}$ at the boundary (see Eq. (49) for details.) Neglecting approximation errors, LFM discretization thus yields a convenient *explicit* scheme [49] to solve $\boldsymbol{\Omega}x = c$. An example of a 2D acoustic field discretized by this method is presented in Figure 1. Numerical stability of explicit FDM schemes, such as LFM, can only be ensured if the step sizes respect some constraining condition, such as the Courant-Friedrich-Lewy condition for hyperbolic PDEs [49]. In the abovementioned example, for instance, this condition translates into $\mathrm{v}\mathrm{d}_{\tau}/\min(\mathrm{d}_{\mathrm{x}}, \mathrm{d}_{\mathrm{y}}) \leq 1/\sqrt{2}$. This limits the resolution (for instance in space and time) achievable by these methods.

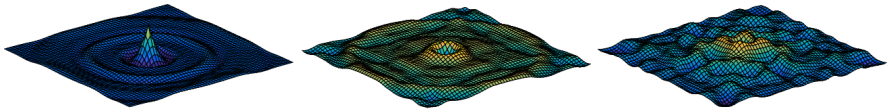

Fig. 1 Example of a discretized 2D acoustic pressure field at different time instants.

4.2 Finite Element Methods (FEM)

The Finite Element Method (FEM) is a numerical approximation used to solve boundary value problems when the solution is intractable analytically due to geometric complexities and inhomogeneities. Among several variants, the Galerkin FEM is famous as it is both well-rooted in theory and simple in derivation [33]. In the Galerkin FEM, a solution is computed in three main steps: 1) the formulation of the problem in its *weak/variational* form, 2) the discretization of the formulated problem, and 3) the choice of the approximating subspace. As an illustrative example, let's consider the well known problem of Poisson's equation (9) with Neumann boundary condition (10).

Weak/variational formulation. The first step aims at expressing the aforementioned PDE in an algebraic form. For a given test function $\boldsymbol{w}(r)$ in some (to be specified) Hilbert space of regular functions $\boldsymbol{H}$ we have, denoting $\boldsymbol{c} = \nabla.j$ the volume current which serves as a source term,

$$\begin{aligned}\int_{\Theta} \boldsymbol{c}(r)\boldsymbol{w}(r)dr &= -\int_{\Theta} \nabla.(\boldsymbol{\sigma}(r)\nabla\boldsymbol{x}(r))\boldsymbol{w}(r)dr \\ &= -\int_{\partial\Theta} n.(\boldsymbol{\sigma}(r)\nabla\boldsymbol{x}(r))\boldsymbol{w}(r)dr + \int_{\Theta} \nabla\boldsymbol{w}(r).(\boldsymbol{\sigma}(r)\nabla\boldsymbol{x}(r))dr \\ &= \int_{\Theta} \nabla\boldsymbol{w}(r).(\boldsymbol{\sigma}(r)\nabla\boldsymbol{x}(r))dr. \end{aligned} \tag{13}$$

The second line in (13) is derived using Green's identity which is the multidimensional analogue of integration by parts [33], whereas, the last line is deduced from the Neumann boundary condition (10). Notice that the resulting equality in (13) can be written as

$$\mathfrak{a}(\boldsymbol{x},\boldsymbol{w}) = \mathfrak{b}(\boldsymbol{w}) \qquad \forall \boldsymbol{w} \in \boldsymbol{H} \tag{14}$$

where $\mathfrak{a}(.,.)$ is a symmetric bilinear form on $\boldsymbol{H}$ and $\mathfrak{b}(.)$ is a linear functional on $\boldsymbol{H}$. The equality in (14) is referred to as the *weak* or the *variational* formulation of the PDE in (9)-(10), a name that stems from the less stringent requirements put on the functions $\boldsymbol{x}$ and $\boldsymbol{c}$. In fact, the former should be differentiable only once (vs. twice in the *strong* formulation), whereas the latter needs to be integrable (vs. continuous over $\bar{\Theta}$ in the *strong* formulation). These relaxed constraints in the *weak* form makes it relevant to a broader collection of problems.

Discretization with the Galerkin method. In the second step, the Galerkin method is applied to the *weak* form. This step aims at discretizing the continuous

problem in (14) by projecting it from the infinite-dimensional space $\boldsymbol{H}$ onto a finite-dimensional subspace $\boldsymbol{H}_h \subset \boldsymbol{H}$ (h refers to the precision of the approximation).

Denoting $\{\phi_\ell\}_{\ell \in I}$ a basis of $\boldsymbol{H}_h$, any function $\boldsymbol{x}$ in the finite-dimensional subspace $\boldsymbol{H}_h$ can be written in a unique way as a linear combination $\boldsymbol{x} = \sum_{\ell \in I} x_\ell \phi_\ell$. Therefore, given $\boldsymbol{x}$ a solution to the problem and if we take as a test function $\boldsymbol{w}$ a basis function ϕ_{i}, the discrete form of (14) is then expressed as

$$\sum_{\ell \in I} \mathfrak{a}_h(\phi_{\mathsf{i}}, \phi_\ell) x_\ell = \mathfrak{b}_h(\phi_{\mathsf{i}}) \qquad \forall \mathsf{i} \in I \tag{15}$$

where

$$\mathfrak{a}_h(\phi_{\mathsf{i}}, \phi_\ell) := \int_{\Theta_h} \sigma(r) \nabla \phi_{\mathsf{i}}(r) . \nabla \phi_\ell(r) dr \tag{16}$$

$$\mathfrak{b}_h(\phi_{\mathsf{i}}) := \int_{\Theta_h} \boldsymbol{c}(r) \phi_{\mathsf{i}}(r) dr \tag{17}$$

with Θ_h a discretized solution domain (see next). The discretization process thus results in a linear system of $\mathsf{n} := card(I)$ equations with n unknowns $\{v_\ell\}_{\ell \in I}$, which can be rewritten in matrix form $\Omega x = c$. The so-called *global stiffness matrix* Ω is a symmetric matrix of size $\mathsf{n} \times \mathsf{n}$ with elements $\Omega_{ij} = \mathfrak{a}_h(\phi_{\mathsf{i}}, \phi_{\mathsf{j}})$, and c is the *load vector* of length n and elements $c_{\mathsf{i}} = \mathfrak{b}_h(\phi_{\mathsf{i}})$. Notice that, in the case where $\sigma(r)$ is a positive function (as in EEG problem for instance), the matrix Ω is also positive semidefinite. This property can be easily deduced from the bilinear form $\mathfrak{a}(.,.)$ where $\mathfrak{a}(\boldsymbol{x}, \boldsymbol{x}) = \int_\Theta \sigma(r)(\nabla \boldsymbol{x}(r))^2 dr \geq 0$ for any function $\boldsymbol{x} \in \boldsymbol{H}$, and from the relationship $\mathfrak{a}(\boldsymbol{x}, \boldsymbol{x}) = \boldsymbol{x}^T \Omega \boldsymbol{x}$.

For the considered Poisson equation the stiffness matrix Ω is also rank deficient by one, corresponding to the fact that, since only the gradient of $\boldsymbol{x}$ appears in Poisson's equation with Neumann's boundary condition, $\boldsymbol{x}$ can only be determined up to an additive constant corresponding to an arbitrary choice of reference for the electrical potential it represents.

Choice of the approximating subspace and discretization basis. The construction of the discretized solution domain Θ_h and the choice of basis functions $\{\phi_\ell\}_{\ell \in I}$ are two pivotal points in FEM since they deeply affect the accuracy of the approximate solution obtained by solving $\Omega x = c$. They also impact the sparsity and conditioning of Ω, hence the computational properties of numerical schemes to solve this linear system.

In FEM, the domain is divided uniformly or non-uniformly into discrete elements composing a mesh, either of triangular shape (tetrahedral in 3D), or rectangular shape (hexahedral in 3D). The triangular (tetrahedral) mesh is often more adapted when dealing with complex geometries (see example in Figure 2 for a mesh of a human head to be used in the context of the EEG inverse problem.

Given the mesh, basis functions are typically chosen as piecewise polynomials, where each basis function is nonzero only on a small part of the domain around a given basic element of the mesh, and satisfy some interpolation condition.

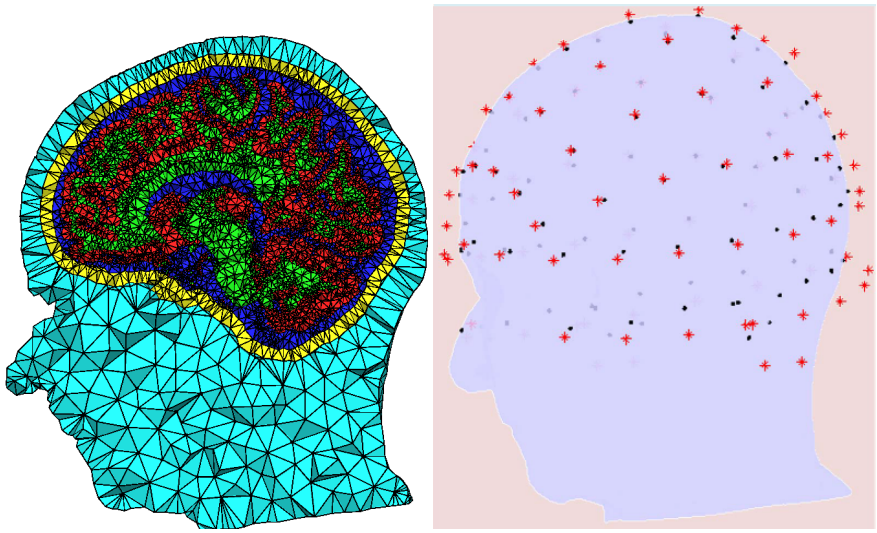

Fig. 2 Left: a sagittal cross-section of tetrahedral mesh generated used iso2mesh software [66] for a segmented head composed of five tissues: gray matter (red), white matter (green), cerebrospinal fluid (blue), skull (yellow) and scalp (cyan). Right: profile view of a human head wearing an electrode helmet (m = 91 electrodes). Red stars indicate the real positions of the electrodes on the helmet while black dots refer to their projection onto the scalp $\{r_{\mathsf{j}}\}_{\mathsf{j}=1:\mathsf{m}}$.

Typical choices lead to families of basis functions whose spatial support overlap little: the support of ϕ_{i} and ϕ_ℓ only intersect if they correspond to close mesh elements. As a result $\mathfrak{a}_h(\phi_{\mathsf{i}}, \phi_\ell)$ is zero for the vast majority of pairs i, ℓ, and the *stiffness* matrix Ω is sparse with $\|\Omega\|_0 = O(\mathsf{n})$.

Using FEM to solve the forward EEG problem. Once again, while our ultimate goal is to exploit FEM for *inverse* problems, its use for *forward* problems is illustrative. In bioelectric field problems, a well-known example of problem modeled by (9)-(10) and solved by FEM is the forward EEG problem that aims at computing the electric field within the brain and on the surface of the scalp using a known current source within the brain and the discretized medium composed of the brain and the surrounding layers (skull, scalp, etc.) [41, 36].

4.3 Numerical approximations of Green's functions

Discretization methods such as FDM or FEM somehow "directly" discretize the PDE at hand, leading to a matrix Ω which is a discrete equivalent of the operator $\mathfrak{L}$. While (2) implicitly defined $\boldsymbol{x}$ given $\boldsymbol{c}$, in the discretized world the matrix Ω allows to implicitly define x given c as

$$\Omega x = c, \tag{18}$$

with $x \in \mathbb{R}^{\mathsf{n}}$ the discretized representation of $\boldsymbol{x}$, and similarly for c and $\boldsymbol{c}$.

We now turn to the discretization of the (potentially more explicit) integral representation of $\boldsymbol{x}$ using Green's functions (5), associated to the integral operator $\mathfrak{G}$, which, as noted in Section 4.3, is often a necessity. One has to discretize the domain $\boldsymbol{\Theta}$, the PDE $\mathfrak{L}$, the field $\boldsymbol{x}$ and the source $\boldsymbol{c}$, and to numerically solve (3) and (4).

Assuming that the equation $\mathfrak{L}\boldsymbol{x} = \boldsymbol{c}$ has a unique solution, we expect the discretized version of $\mathfrak{L}$, the matrix $\Omega \in \mathbb{R}^{\mathsf{n}\times\mathsf{n}}$, to be full rank. Under this assumption, we can write

$$x = \Psi c, \text{ with } \Psi = \Omega^{-1}. \tag{19}$$

In compressive sensing terminology, Ψ is a *dictionary* of discrete Green's functions.

Not surprisingly, the discretized version of the integral operator $\mathfrak{G}$ is the matrix inverse of the discretized version of the differential operator $\mathfrak{L}$. Hence, c and x can be seen as dual representations of the same discrete signal, with linear transformations from one signal space to another. Yet, as we will see, there may be significant differences in sparsity between the matrices Ψ and Ω: while Ψ is typically a dense matrix (Green's function are often delocalized, *e.g.* in the context of propagation phenomena), with $\|\Psi\|_0$ of the order of n^2, the analysis operator is typically very sparse, with $\|\Omega\|_0 = O(\mathsf{n})$. In the context of linear inverse problems where one only observes $y \approx Ax$, algorithms may thus have significantly different computational properties whether they are designed with one representation in mind or the other.

4.4 Discretized inverse problem

We now have all elements in place to consider the discretized version of the inverse problems expressed in Section 2.3. The signals and measurement vectors are respectively denoted by $x \in \mathbb{R}^{\mathsf{s}}$, $c \in \mathbb{R}^{\mathsf{s}}$ and $y \in \mathbb{R}^{\mathsf{m}}$, in the case of a spatial field, or $x \in \mathbb{R}^{\mathsf{st}}$, $c \in \mathbb{R}^{\mathsf{st}}$, and $y \in \mathbb{R}^{\mathsf{mt}}$, in the case of the spatio-temporal field. We denote n the dimension of x and c, which is $\mathsf{n} = \mathsf{s}$ in the former case and $\mathsf{n} = \mathsf{st}$ in the latter.

The vector of measurements y can be seen as a subsampled version of x, possibly contaminated by some additive noise e. In the case of a spatial field, $y = Ax + e$, where A is an $\mathsf{m} \times \mathsf{s}$ spatial subsampling matrix (row-reduced identity), and e is a discrete representation of additive noise $\boldsymbol{e}$. In the case of a spatio-temporal field, the same holds where A is an $(\mathsf{mt}) \times (\mathsf{st})$ block-diagonal concatenation of identical (row-reduced identity) spatial subsampling matrices. Overall, we have to solve

$$y = Ax + e, \tag{20}$$

where A is a given subsampling matrix. Given Ω (and, therefore, $\Psi = \Omega^{-1}$), equivalently to (20) we can write

$$y = A\Psi c + e, \tag{21}$$

where x and c satisfy (18)-(19).

Sparsity or cosparsity assumptions. Since $A\Psi \in \mathbb{R}^{\mathsf{m}\times\mathsf{s}}$ (resp $\in \mathbb{R}^{\mathsf{mt}\times\mathsf{st}}$), and $\mathsf{m} < \mathsf{s}$, it is obvious that one cannot recover every possible source signal c from the measurements y, hence the need for a low-dimensional model on x or on c. As discussed in 2.3, a typical assumption is the sparsity of the source field $\boldsymbol{c}$, which in the discretized setting translates into c being a very sparse vector, with $\|c\|_0 \ll \mathsf{n}$ (or well approximated by a sparse vector), possibly with an additional structure. This gives rise to *sparse synthesis* regularization, usually tackled by convex relaxations or greedy methods that exploit the mode $x = \Psi c$ with sparse c. Alternatively, this is expressed as a *sparse analysis* or *cosparse* model on x asserting that Ωx is sparse.

5 Sparse and cosparse regularization

Previously discussed techniques for solving inverse source problems suffer from two serious practical limitations, *i.e.* algebraic methods (Section 2.3) impose strong, often unrealistic assumptions, whereas sparse synthesis approaches based on numerical Green's functions approaches (Section 3) do not scale gracefully for non-trivial geometries. Despite the fact that physical fields are not perfectly sparse in any finite basis, as demonstrated in one of the chapters of the previous version of this monograph [62], it becomes obvious that we can escape discretization only for very restricted problem setups. Therefore, we focus on the second issue using the *analysis* version of sparse regularization.

5.1 Optimization problems

Following traditional variational approaches [69], estimating the unknown parameters x and c corresponds to an abstract optimization problem, which we uniformly term *physics-driven (co)sparse regularization*:

$$\min_{x,c} f_d(Ax-y) + f_r(c) \qquad (22)$$
$$\text{subject to } \Omega x = c,\ Cx = h.$$

Here, f_d is the data fidelity term (enforcing consistency with the measured data), whereas f_r is the regularizer (promoting (structured) sparse solutions c). The matrix C and vector h represent possible additional constraints, such as source support restriction or specific boundary/initial conditions, as we will see in Section 7.

We restrict both f_d and f_r to be convex, lower semicontinuous functionals, *e.g.* f_d can be the standard sum of squares semimetric, and f_r can be the ℓ_1 norm. In some cases the constraints can be omitted. Obviously, we can solve (22) for either c or x, and recover another using (18) or (19). The former gives rise to sparse synthesis

$$\min_{c} f_d(A\Psi c - y) + f_r(c) \ \text{ subject to } \ C\Psi c = h. \qquad (23)$$

or *sparse analysis (*aka *cosparse)* optimization problem

$$\min_x f_d(Ax-y)+f_r(\Omega x) \quad \text{subject to} \quad Cx=h. \tag{24}$$

The discretized PDE encoded in Ω is the analysis operator. As mentioned, the two problems are equivalent in this context [27], but as we will see in Section 6, their computational properties are very different. Additionaly, note that the operator Ω is obtained by explicit discretization of (2), while the dictionary Ψ of Green's functions is discretized implicitly (in general, since analytic solutions of (3) are rarely available), *i.e.* by inverting Ω, which amounts to computing numerical approximations to Green's functions (see Section 4.3).

5.2 Optimization algorithm

Discretization can produce optimization problems of huge scale (see Sections 6-7 for examples), some of which can be even intractable. Since (23) or (24) are nominally equivalent, the question is whether there is a computational benefit in solving one or another. Answering this question is one of the goals of the present chapter.

Usually, problems of such scale are tackled by first order optimization algorithms, that require only the objective and the (sub)gradient oracle at a given point [60]. The fact that we allow both f_d and f_r to be non-smooth, forbids using certain popular approaches, such as *Fast Iterative Soft Thresholding Algorithm* [4]. Instead we focus on the *Alternating Direction Method of Multipliers (ADMM)* algorithm [31, 26], which has become a popular scheme due to its scalability and simplicity. Later in this subsection, we discuss two variants of the ADMM algorithm, convenient for tackling different issues related to the physics-driven framework.

Alternating Direction Method of Multipliers (ADMM). For now, consider a convex optimization problem of the form[2]

$$\min_z f(z)+g(Kz-b), \tag{25}$$

where the functions f and g are convex, proper and lower semicontinuous [60]. Either of these can account for hard constraints, if given as a *characteristic function* $\chi_S(z)$ of a convex set S:

$$\chi_S(z) := \begin{cases} 0 & z\in S, \\ +\infty & \text{otherwise.} \end{cases} \tag{26}$$

An equivalent formulation of the problem (25) is

[2] The change of notation, in particular from x/c to z for the unknown, is meant to cover both cases in a generic framework.

$$\min_{z_1,z_2} f(z_1)+g(z_2) \quad \text{subject to} \quad Kz_1-b=z_2, \tag{27}$$

for which the (scaled) *augmented Lagrangian* [10] writes:

$$L_\rho(z_1,z_2,u)=f(z_1)+g(z_2)+\frac{\rho}{2}\|Kz_1-z_2-b+u\|_2^2-\frac{\rho}{2}\|u\|_2^2 \tag{28}$$

with ρ a positive constant. Note that the augmented Lagrangian is equal to the standard (unaugmented) Lagrangian plus the quadratic penalty on the constraint residual Kz_1-b-z_2.

The ADMM algorithm consists in iteratively minimizing the augmented Lagrangian with respect to z_1 and z_2, and maximizing it with respect to u. If the standard Lagrangian has a saddle point [10, 14], iterating the following expressions yields a global solution of the problem[3]:

$$z_1^{(\mathrm{j}+1)}=\underset{z_1}{\operatorname{argmin}}\, f(z_1)+\frac{\rho}{2}\|Kz_1-b-z_2^{(\mathrm{j})}+u^{(\mathrm{j})}\|_2^2 \tag{29}$$

$$z_2^{(\mathrm{j}+1)}=\operatorname{prox}_{\frac{1}{\rho}g}\left(Kz_1^{(\mathrm{j}+1)}-b+u^{(\mathrm{j})}\right) \tag{30}$$

$$u^{(\mathrm{j}+1)}=u^{(\mathrm{j})}+Kz_1^{(\mathrm{j}+1)}-b-z_2^{(\mathrm{j}+1)}. \tag{31}$$

The iterates $z_1^{(\mathrm{j})}$ and $z_2^{(\mathrm{j})}$ update the primal variables, and $u^{(\mathrm{j})}$ updates the dual variable of the convex problem (27). The expression $\operatorname{prox}_{f/\rho}(v)$ denotes the well-known *proximal operator* [54] of the function f/ρ applied to some vector v:

$$\operatorname{prox}_{\frac{1}{\rho}f}(v)=\underset{w}{\operatorname{argmin}}\, f(w)+\frac{\rho}{2}\|w-v\|_2^2. \tag{32}$$

Proximal operators of many functionals of our interest are computationally efficient to evaluate (linear or linearithmic in the number of multiplications, often admiting closed form expressions). Such functionals are usually termed "simple" in the optimization literature [13].

Weighted Simultaneous Direction Method of Multipliers (SDMM). The first ADMM variant we consider is *weighted SDMM (Simultaneous Direction Method of Multipliers)* [17]. It refers to an application of ADMM to the case where more than two functionals are present in the objective:

$$\min_{z,z_1\ldots z_{\mathrm{f}}} \sum_{\mathrm{i}=1}^{\mathrm{f}} f_{\mathrm{i}}(z_{\mathrm{i}}) \quad \text{subject to} \quad K_{\mathrm{i}}z-b_{\mathrm{i}}=z_{\mathrm{i}}. \tag{33}$$

Such an extension can be written [17] as a special case of the problem (27), for which the iterates are given as follows:

[3] j denotes an iteration index.

$$z^{(\mathrm{j}+1)} = \underset{z}{\operatorname{argmin}} \sum_{\mathrm{i}=1}^{\mathrm{f}} \frac{\rho_\mathrm{i}}{2} \|K_\mathrm{i} z - b_\mathrm{i} + u_\mathrm{i}^{(\mathrm{j})} - z_\mathrm{i}^{(\mathrm{j})}\|_2^2, \tag{34}$$

$$z_\mathrm{i}^{(\mathrm{j}+1)} = \operatorname{prox}_{\frac{1}{\rho_\mathrm{i}} f_\mathrm{i}} \left(K_\mathrm{i} z^{(\mathrm{j}+1)} - b_\mathrm{i} + u_\mathrm{i}^{(\mathrm{j})} \right), \tag{35}$$

$$u_\mathrm{i}^{(\mathrm{j}+1)} = u_\mathrm{i}^{(\mathrm{j})} + K_\mathrm{i} z^{(\mathrm{j}+1)} - b_\mathrm{i} - z_\mathrm{i}^{(\mathrm{j}+1)}. \tag{36}$$

We can now instantiate (33), with I denoting the identity matrix:

- for the sparse synthesis problem: $K_1 = I$, $K_2 = A\Psi$ and $K_3 = C\Psi$;
- for the sparse analysis problem, by $K_1 = \Omega$, $K_2 = A$ and $K_3 = C$.

In both cases $b_1 = 0$, $b_2 = y$ and $b_3 = h$, and the functionals f_i are f_r, f_d and 0.

Choice of the multipliers. The multipliers ρ_i only need to be strictly positive, but a suitable choice can be helpful for the overall convergence speed of the algorithm. In our experiments, we found that assigning larger values for ρ_i's corresponding to hard constraints (*e.g.* $\|Az - y\|_2 \leq \varepsilon$ or $C\Psi c = h$), and proportionally smaller values for other objectives was beneficial.

The weighted SDMM is convenient for comparison, since it can be easily shown that it yields iteration-wise numerically identical solutions for both the synthesis and analysis problems, if the intermediate evaluations are exact. However, solving a large system of normal equations per iteration of the algorithm seems wasteful in practice. For an improved efficiency, another ADMM variant is more convenient, known as the *preconditioned ADMM* or the *Chambolle-Pock (CP)* algorithm [13].

Chambolle-Pock (CP). For simplicity, we demonstrate the idea on the setting involving only two objectives, as in (27). The potentially expensive step is the ADMM iteration (29), due to the presence of a matrix K in the square term. Instead, as proposed in [13] and analysed in [73], an additional term is added to the subproblem

$$z_1^{(\mathrm{j}+1)} = \underset{z_1}{\operatorname{argmin}}\, f(z_1) + \frac{\rho}{2} \|Kz_1 - b - z_2^{(\mathrm{j})} + u^{(\mathrm{j})}\|_2^2 + \frac{\rho}{2} \|z_1 - z_1^{(\mathrm{j})}\|_P^2, \tag{37}$$

where $\|v\|_P = v^\mathsf{T} P v$. A clever choice is $P = \frac{1}{\tau\sigma} I - K^\mathsf{T} K$, which, after some manipulation, yields:

$$z_1^{(\mathrm{j}+1)} = \operatorname{prox}_{\sigma f} \left(z_1^{(\mathrm{j})} + \sigma K^\mathsf{T} u^{(\mathrm{j})} - \sigma\tau K^\mathsf{T} \left(K z_1^{(\mathrm{j})} - b - z_2^{(\mathrm{j})} \right) \right). \tag{38}$$

Thus, P acts as a preconditioner and simplifies the subproblem.

In the original formulation of the algorithm [13], the z_2 and u updates are merged together. The expression for $u^{(\mathrm{j}+1)}$, along with a straightforward application of Moreau's identity [14], leads to

$$\begin{aligned} z^{(\mathrm{j}+1)} &= \operatorname{prox}_{\sigma f} \left(z^{(\mathrm{j})} + \sigma K^\mathsf{T} \left(2u^{(\mathrm{j})} - u^{(\mathrm{j}-1)} \right) \right) \\ u^{(\mathrm{j}+1)} &= \operatorname{prox}_{\tau g^*} \left(u^{(\mathrm{j})} - \tau \left(K z^{(\mathrm{j}+1)} - b \right) \right), \end{aligned} \tag{39}$$

where the primal variable index has been dropped (z instead of z_1), since the auxiliary variable z_2 does not appear explicitly in the iterations any longer. The functional g^* represents the *convex conjugate* [4] of g, and the evaluation of its associated proximal operator $\text{prox}_{g^*}(\cdot)$ is of the same computational complexity as of $\text{prox}_g(\cdot)$, again thanks to Moreau's identity.

As mentioned, different flavours of the CP algorithm [13, 18] can easily lend to settings where a sum of multiple objectives is given, but, for the purpose of demonstration, in the present chapter we use this simple formulation involving only two objectives (the boundary conditions are presumably absorbed by the regularizers in (23) and (24)). We instantiate (39):

- in the synthesis case, with $K = A\Psi$, $z = c$, $f = f_r$ and $g = f_d$.
- in the analysis case, we exploit the fact that A has a simple structure, and set $K = \Omega$, $z = x$, $f(\cdot) = f_d(A\cdot)$ and $g(\cdot) = f_r(\cdot)$. Since A is a row-reduced identity matrix, and thus a tight frame, evaluation of the proximal operators of the type $\text{prox}_{f_d}(A\cdot)$ is usually as efficient as evaluating $\text{prox}_{f_d}(\cdot)$, *i.e.* without composition with the measurement operator [10]. Moreover, if $\text{prox}_{f_d}(\cdot)$ is separable component-wise (*i.e.* can be evaluated for each component independently), so is the composed operator $\text{prox}_{f_d}(A\cdot)$.

Accelerated variants. If the objective has additional regularity, such as strong convexity, accelerated variants of ADMM algorithms are available [34, 21]. Thus, since the evaluation of proximal operators is assumed to be computationally "cheap", the main computational burden comes from matrix-vector multiplications (both in CP and SDMM) and from solving the linear least squares subproblem (in SDMM only). For the latter, in the large-scale setting, one needs to resort to iterative algorithms to approximate the solution (ADMM is robust to inexact computations of intermediate steps, as long as the accumulated error is finite [26]). These iterative algorithms can often be initialized (*warm-started*) using a previous iterations' estimate, which may greatly help their convergence. We can also control the accuracy, thus ensuring that there is no large drift between the sparse and cosparse versions.

5.3 Computational complexity

The overall computational complexity of the considered algorithms results from a combination of their iteration cost and their convergence rate.

Iteration cost. It appears that the iteration cost of ADMM is driven by that of the multiplication of vectors with matrices and their transposes. In practice, most discretization schemes, such as Finite Difference (FD) or Finite Element Method (FEM) are *locally supported* [49, 75]. By this we mean that the number of non-zero coefficients required to approximate $\mathfrak{L}$, *i.e.* $\texttt{nnz}(\Omega)$, is linear with respect to $\texttt{n}$, the dimension of the discretized space. In turn, applying Ω and its transpose, is

[4] $g^*(\lambda) := \sup_z g(z) - z^\mathsf{T}\lambda$

in the order of $O(\mathsf{n})$ operations, thanks to the sparsity of the analysis operator. This is in stark contrast with synthesis minimization, whose cost is dominated by much heavier $O(\mathsf{mn})$ multiplications with the dense matrix $A\Psi$ and its transpose. The density of the dictionary Ψ is not surprising - it stems from the fact that the physical quantity modeled by $\boldsymbol{x}$ is spreading in the domain of interest (otherwise, we would not be able to obtain remote measurements). As a result, and as will be confirmed experimentally the following sections, *the analysis minimization is computationally much more efficient*.

Convergence rate. In [13], the authors took a different route to develop the CP algorithm, where they considered a saddle point formulation of the original problem (25) directly. The asymptotic convergence rate of the algorithm was discussed for various regimes. In the most general setting considered, it can be shown that, for $\tau\sigma\|K\| \leq 1$ and any pair (z,u), the *weak primal-dual gap* is bounded, and that, when $\tau\sigma\|K\| < 1$, the iterates $z^{(\mathsf{j})}, u^{(\mathsf{j})}$ converge *("ergodic convergence")* to saddle points of the problem (25). Thus, it can be shown that the algorithm converges with a rate of $O(\frac{1}{\mathsf{j}})$. In terms of the order of iteration count, this convergence rate cannot be improved in the given setting, as shown by Nesterov [60]. However, considering the bounds derived from [13], we can conclude that the rate of convergence is proportional to:

- the operator norm $\|K\|$ (due to the constraint on the product $\tau\sigma$);
- the distance of the initial points $(z^{(0)}, u^{(0)})$ from the optimum (z^*, u^*).

In both cases, a lower value is preferred. Concerning the former, the unfortunate fact is that the ill-posedness of PDE-related problems is reflected in the conditioning of Ω and Ψ. Generally, the rule of thumb is that the finer the discretization, the larger the condition number, since either (or both) $\|\Omega\|$ and $\|\Psi\|$ can grow unbounded [29].

Multiscale acceleration. A potential means for addressing the increasing condition numbers of Ω and Ψ is to apply multiscale schemes, in the spirit of widely used *multigrid methods* for solutions of PDE-generated linear systems. The multigrid methods are originally exploiting smoothing capabilities of Jacobi and Gauss-Seidel iterations [68], and are based on hierarchical discretizations of increasing finesses. Intuitively, the (approximate) solution at a lower level is interpolated, and forwarded as the initial point for solving a next-in-hierarchy higher resolution problem, until the target (very) high resolution problem (in practice, more sophisticated schemes are often used). In the same spirit, one could design a hierarchy of discretizations for problems (23) or (24), and exploit the fact that $\|\Omega\|$ and $\|A\Psi\|$ are reducing proportionally to lowering discretization finesse. At the same time, matrix-vector multiplications become much cheaper to evaluate.

Initialization strategy. Finally, for the synthesis optimization problem (23), we often expect the solution vector c^* to be sparse, *i.e.* to mostly contain zero components. Therefore, a natural initialization point would be $c^{(0)} = 0$, and we expect $\|c^* - c^{(0)}\|$ to be relatively small. However, as mentioned, the synthesis version is not generally preferable, due to high per-iteration cost and memory requirements. On the other hand, we do not have such a simple intuition for initializing $z^{(0)}$ for

the cosparse problem (24). Fortunately, we can leverage the multiscale scheme described in the previous paragraph: we would solve the analysis version of the regularized problem at all levels in hierarchy, except at the coarsest one, where the synthesis version with $c^{(0)} = 0$ would be solved instead. The second problem in hierarchy would be initialized by the interpolated version of $z^* = \Psi c^*$, with c^* being the solution at the coarsest level. Ideally, such a scheme would inherit good properties of both the analysis- and synthesis-based physics-driven regularization. In subsection 6.2, we empirically investigate this approach, to confirm the predicted performance gains.

6 Scalability

In this section we empirically investigate differences in computational complexity of the synthesis and analysis physics-driven regularization, through simulations based on the weighted SDMM (34), and the multiscale version of the Chambolle-Pock algorithm (39). First, we explore the scalability of the analysis compared to the synthesis physics-driven regularization, applied to the acoustic source localization problem (results and discussions are adopted from [44]). Next, we demonstrate the effectiveness of the mixed synthesis-analysis multiscale approach on a problem governed by Poisson's equation.

6.1 Analysis vs *synthesis*

Let us recall that, for acoustic source localization, we use the *Finite Difference Time Domain (FDTD) Standard Leap Frog (SLF)* method [49, 75] for discretization of the acoustic wave equation (6) with imposed initial/boundary conditions. This yields a discretized spatio-temporal pressure field $x \in \mathbb{R}^{\mathsf{nt}}$ and a discretized spatio-temporal source component $c \in \mathbb{R}^{\mathsf{nt}}$, built by vectorization and sequential concatenation of t corresponding n-dimensional scalar fields. The matrix operator Ω is a banded, lower triangular, sparse matrix with a very limited number of non-zeros per row (*e.g.* maximum seven in the 2D case). Note that the Green's functions dictionary $\Psi = \Omega^{-1}$ cannot be sparse, since it represents the truncated impulse responses of an infinite impulse response ("reverberation") filter. Finally, the measurements are presumably discrete, and can be represented as $y \approx Ax$, where $A \in \mathbb{R}^{\mathsf{mt} \times \mathsf{nt}}$ is a block diagonal matrix, where each block is an identical spatial subsampling matrix.

Optimization problems. To obtain $\hat{x}$ and $\hat{c}$, we first need to solve *one* of the two optimization problems:

$$\hat{x} = \underset{x}{\operatorname{argmin}} f_r(\Omega_\Theta x) + f_d(Ax - y) \text{ subject to } \Omega_{\partial\Theta} x = 0 \tag{40}$$

$$\hat{c} = \underset{c}{\operatorname{argmin}} f_r(c_\Theta) + f_d(A\Psi c - y) \text{ subject to } c_{\partial\Theta} = 0, \tag{41}$$

where the matrix $\Omega_{\partial\Theta}$ is formed by extracting rows of Ω corresponding to initial conditions (7), and boundary conditions (8), while Ω_{Θ} is its complement corresponding to (the interior of) the domain itself. Analogously, the vector $c_{\partial\Theta}$ corresponds to components of c such that $\Omega_{\partial\Theta}x = \Omega_{\partial\Theta}\Psi c = c_{\partial\Theta}$ (due to $\Omega_{\partial\Theta} \perp \Psi$), while c_{Θ} is the vector built from complementary entries of c. The data fidelity term is the ℓ_2 norm constraint on the residual, *i.e.* $f_d = \chi_{\|\cdot\|\leq\varepsilon}(A\cdot - y)$.

Source model and choice of the penalty function for source localization. Assuming a small number of sources that remain at fixed locations, the true source vector is group sparse: denoting by $\{c_{\mathsf{j}} \in \mathbb{R}^{\mathsf{t}}\}_{\mathsf{j}=1\ldots\mathsf{s}}$ the subvectors of c corresponding to the s discrete spatial locations in Γ, we assume that only few of these vectors are nonzero. As a consequence the regularizer f_r is chosen as the joint $\ell_{2,1}$ group norm [40] with non-overlapping groups associated to this partition of c.

Detection of source locations. Given the estimated $\hat{x}$, or equivalently $\hat{c}$, the localization task becomes straightforward. Denoting $\{\hat{c}_{\mathsf{j}} \in \mathbb{R}^{\mathsf{t}}\}_{\mathsf{j}=1\ldots\mathsf{s}}$ the subvectors of $\hat{c}$ corresponding to discrete locations in Γ, estimated source positions can be retrieved by setting an energy threshold on each $\hat{c}_{\mathsf{j}}$. Conversely, if the number of sound sources k is known beforehand (for simplicity this is our assumption in the rest of the text), one can consider the k spatial locations with highest magnitude $\|\hat{c}_{\mathsf{j}}\|_2$ to be the sound source positions.

Results. An example localization result of this approach is presented in Figure 3. The simulated environment is a reverberant 3D acoustic chamber, with boundaries modeled by the Neumann (hard wall) condition, corresponding to highly reverberant conditions that are difficult for traditional TDOA methods cf. [7]. The problem dimension is $\mathsf{n} = \mathsf{st} \approx 3 \times 10^6$.

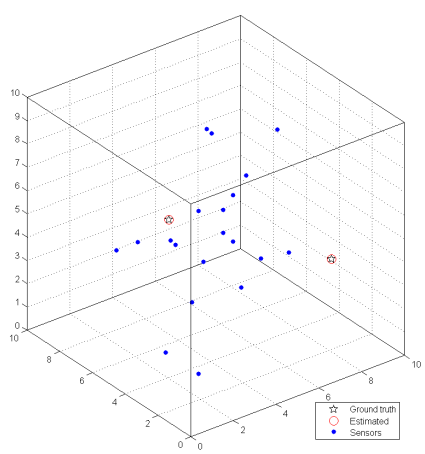

Fig. 3 Localization of 2 simulated sources (stars) by a 20-microphone random array (dots) in 3D.

Empirical computational complexities. To see how the two regularizations scale in the general setting, we explicitly compute the matrix $A\Psi$, and use it in computations. The SDMM algorithm (34) requires solving a system of normal equa-

tions, with a coefficient matrix of size $\mathsf{n} \times \mathsf{n}$ with $\mathsf{n} = \mathsf{st}$. Its explicit inversion is infeasible in practice, and we use the *Least Squares Minimum Residual (LSMR)* [28] iterative method instead. This method only evaluates matrix-vector products, thus its per iteration cost is driven by the (non-)sparsity of the applied coefficient matrix, whose number of non-zero entries is $O(\mathsf{st})$, in the analysis, and $O(\mathsf{smt}^2)$, in the synthesis case. In order to ensure there is no bias towards any of the two approaches, an oracle stopping criterion is used: SDMM iterations stop when the objective function $f_r(c^{(\mathsf{j})})$ falls below a predefined threshold, close to the ground truth value. Given this criterion, and by setting the accuracy of LSMR sufficiently high, the number of SDMM iterations remains equal for both the analysis and synthesis regularizations.

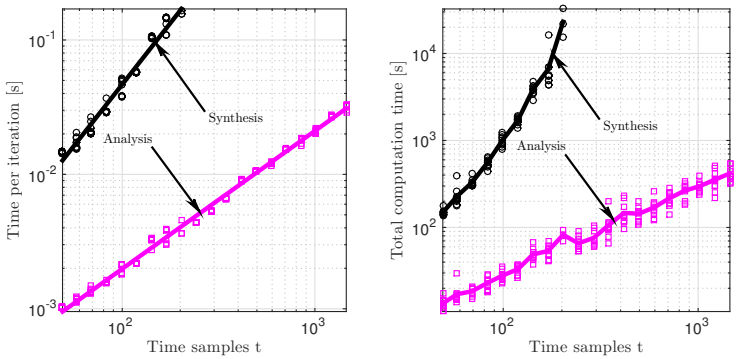

Fig. 4a Computation time *vs* problem size: (left) per inner iteration, (right) total. Solid line: average, dots: individual realizations.

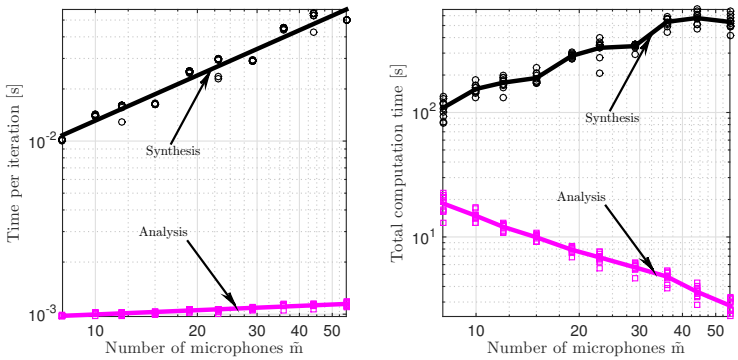

Fig. 4b Computation time *vs* number of measurements: (left) per inner iteration, (right) total. Solid line: average, dots: individual realizations.

In Figure 4a, the results with varying number of time samples t are presented[5], verifying that the two approaches scale differently with respect to problem size. Indeed, the per-iteration cost of the LSMR solver grows linearly with t, in the analysis case, while being nearly quadratic for the synthesis counterpart. The difference between the two approaches becomes striking when the total computation time is considered, since the synthesis-based problem exhibits cubic growth (in fact, above a certain size, it becomes infeasible to scale the synthesis problem due to high memory requirements and computation time).

Keeping the problem size $\mathsf{n} = \mathsf{st}$ fixed, we now vary the number of microphones m (corresponding to a number of measurements mt). We expect the per-iteration complexity of the analysis regularization to be almost independent of m, while the cost of the synthesis version should grow linearly. The results in the left part of figure 4b confirm this behavior. However, we noticed that the number of SDMM iterations *decreases* with m for both models, at the same pace. The consequence is that the total computation time increases in the synthesis case, but this *computation time decreases when the number of microphones increases* in the analysis case, as shown in the right graph. While perhaps a surprise, this is in line with recent theoretical studies [15, 72] 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.

6.2 Multiscale acceleration

The back-to-back comparison of the analysis and synthesis regularizations reveals that the former is a preferred option for large scale problems, when a *numerically identical* SDMM algorithm (34) is used. We are now interested to understand how the two approaches behave when more suitable, but non-identical versions of the CP algorithm (39) are used instead. To investigate this question, let us consider a very simple one-dimensional differential equation:

$$-\frac{d^2\boldsymbol{x}(r)}{dr^2} = \boldsymbol{c}(r), \tag{42}$$

with $\boldsymbol{x}(0) = \boldsymbol{x}(\phi) = 0$ (*e.g.* modeling a potential distribution of a grounded thin rod, with sparse "charges" $\boldsymbol{c}(r)$).

Optimization problems. Given the discretized analysis operator Ω and the dictionary Ψ, and assuming, for simplicity, noiseless measurements $y = Ax^*$, physics-driven regularization boils down to solving either of the following two problems:

[5] The spatial dimensions remain fixed to ensure solvability of the inverse problem.

$$\hat{x} = \underset{x}{\operatorname{argmin}} \|\Omega x\|_1 \text{ subject to } Ax = y \tag{43}$$

$$\hat{c} = \underset{c}{\operatorname{argmin}} \|c\|_1 \text{ subject to } A\Psi c = y \tag{44}$$

As noted in Section 5.3, the operator norms $\|\Omega\|$ and $\|\Psi\|$ are key quantities for convergence analysis. To obtain Ω and Ψ, we apply finite (central) differences, here at n points with the dicretization step $\delta r = 1/\mathsf{n}$. We end up with the well-known symmetric tridiagonal Toeplitz[6] matrix Ω, *i.e.* the 1D discrete Laplacian operator, with a "stencil" defined as $\delta r^2[-1,\ 2,\ -1]$. Its singular values admit simple analytical formula [75]:

$$\sigma_{\mathsf{i}} = 4\mathsf{n}^2 \sin^2\left(\frac{\pi \mathsf{i}}{2\mathsf{n}}\right), \quad \mathsf{i} = 1 \ldots \mathsf{n}. \tag{45}$$

We can immediatelly deduce $\|\Omega\| \approx 4\mathsf{n}^2$ and $\|\Psi\| \approx 1/\pi^2$, which is very unfavorable for the analysis approach, but appreciated in the synthesis case[7]. The situation is opposite if a discretization with unit stepsize is applied. Note that we can safely scale each term in the objective (24) by a constant value, without affecting the optimal solution x^*. Provided that f_r can be factored – for the ℓ_1 norm in (43) we have $\|\Omega x\|_1 = |\mathrm{w}| \|\frac{1}{\mathrm{w}}\Omega x\|_1$, $\mathrm{w} \neq 0$ – we can normalize the problem by multiplying with $1/\delta r^2$, which yields $\|\frac{1}{\delta r^2}\Omega\| \approx 4$, irrespective of the problem dimension n.

Numerical experiments. Considering the multiscale scheme described in subsection 5.3, we would preferably solve the non-normalized synthesis problem at the coarsest scale, and consequently solve the normalized analysis problems from the second level in hierarchy onward. However, to see the benefits of the multiscale approaches more clearly, here we turn a blind eye on this fact, and use the non-normalized finite difference discretization for both approaches (thereby crippling the analysis approach from the start). To investigate the influence of different aspects discussed in subsection 5.3, we set the target problem dimension to $\mathsf{n} = 10^4$, and build a multiscale pyramid with 5 levels of discretization, the coarsest using only 500 points to approximate (42).

Optimization algorithms. Six variants of the CP algorithm (39) are considered:

- ***Analysis***: the matrices Ω and A are built for the target (high resolution) problem, and the algorithm is initialised by an all-zero vector ($x^{(0)} = 0$).
- ***Analysis multiscale***: A set of analysis operators and measurement matrices is built for each of the five scales in hierarchy. At the coarsest scale, the algorithm is initialised by an all-zero vector; at subsequent scales, we use a (linearly) interpolated estimate $\hat{x}$ from the lower hierarchical level as a starting point.

[6] Note that, in this simplistic setting, a fast computation of $\Omega^{-1}c$ using the Thomas algorithm [76] could be exploited. The reader is reminded that this is not a generally available commodity, which is the main incentive for considering the analysis counterpart.

[7] The value of $\|A\Psi\|$ is actually somewhat lower than $\|\Psi\|$ - it depends on the number of microphones m and their random placement.

- ***Synthesis (zero init)***: Analogous to the first, single scale analysis approach, the target resolution problem is solved by the synthesis version of CP initialised by an all-zero vector $z^{(0)} = 0$.
- ***Synthesis (random init)***: Same as above, but initialised by a vector whose components are sampled from a high-variance univariate normal distribution.
- ***Synthesis multiscale***: Analogous to analysis multiscale approach, a set of reduced dictionary matrices $A\Psi$ is built for each scale in hierarchy. The algorithm at the coarsest scale is initialised by an all-zero vector, and the estimation-interpolation scheme is continued until the target resolution.
- ***Mixed multiscale***: We use the solution of the synthesis multiscale approach *at the coarsest scale* to initialise the second level in hierarchy of the analysis version. Then, the analysis multiscale proceeds as before.

Performance metrics. Even with an oracle stopping criterion, the number of iterations between different versions of the CP algorithm may vary. To have comparable results, we fix the number of iterations to 10^4, meaning that the full-resolution (single-scale) approaches are given an unfair advantage, due to their higher per-iteration cost. Therefore, we output two performance metrics: i) a relative error: $\varepsilon = \|\hat{x} - x^*\| / \|x^*\|$, $\hat{x}$ and x^* being respectively the estimated and the ground truth (propagated) signal[8]; and ii) processing time for the given number of iterations. The experiments are conducted for different values of m, the number of measurements.

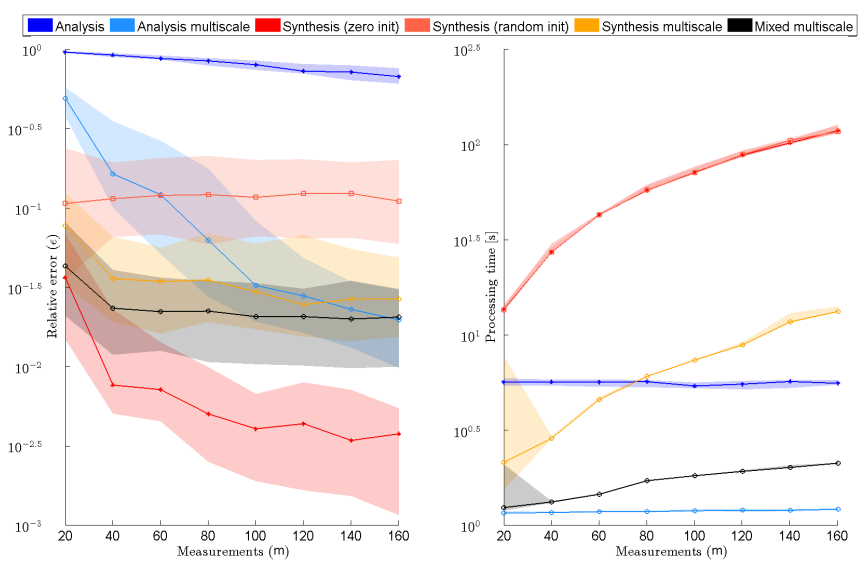

Fig. 5 Median performance metrics (logarithmic scale) wrt number of measurements (shaded regions correspond to 25% - 75% percentiles). Left: relative error ε; right: processing time.

[8] This metric is more reliable than the corresponding one with respect to the source signal, since small defects in support estimation should not disproportionately affect performance.

Results. The results presented in Figure 5 (left) confirm our predictions: the synthesis approach initialized with all-zeros, as well as the proposed mixed synthesis-analysis approach, perform better than the rest in terms of the relative error metric. It is clear that improper initialization significantly degrades performance - notably, for the synthesis algorithm initialized randomly, and the two analysis approaches. The single-scale analysis version is the slowest to converge, due to its large Lipschitz constant $\|\Omega\|^2$ at $\delta r = 1/\mathsf{n}$, and trivial initialization. However, processing time results on the right graph of Figure 5 reveal that synthesis based approaches imply much higher computational cost than the analysis ones, even if the multiscale scheme is applied. In addition their computational performance suffers when the number of measurements increases – which is, naturally, beneficial with regards to the relative error – due to the increased cost of matrix-vector products with $G = A\Psi$ (where G is precomputed once and for all before iterating the algorithm). Fortunately, the mixed approach is mildly affected, since only the computational cost at the coarsest scale increases with m.

7 Versatility

In this section we demonstrate the versatility of physics-driven cosparse regularization. First, we discuss two notions of "blind" acoustic source localization enabled by the physics-driven approach. All developments and experiments in this part refer to 2D spatial domains, however, the principles are straightforwardly extendable to three spatial dimensions. In the second subsection, we apply the regularization to another problem, using a different discretization method: source localization in electroencephalography with FEM. There we consider a three dimensional problem, with physically-relevant domain geometry (real human head).

7.1 Blind acoustic source localization

The attentive reader may have noticed that so far no explicit assumption has been made on the shape of the spatial domain under investigation. In fact, it has been shown in [45] that the proposed regularization facilitates acoustic source localization in spatial domains of exotic shape, even if there is no line of sight between the sources and microphones. This is an intriguing capability, as such a scenario prevents the use of more traditional methods based on TDOA. One example is presented in Figure 6 (left), termed "hearing behind walls". Here the line of sight between the sources and microphones is interrupted by a soundproof obstacle, hence the acoustic waves can propagate from the sources to the microphones only by reverberation. Yet, as shown with the empirical probability (of exactly localizing all sources) results in Figure 6 (right), the physics-driven localization is still possible.

However, an issue with applying physics-driven regularization is that it comes with the strong assumption of knowing the parametrized physical model almost per-

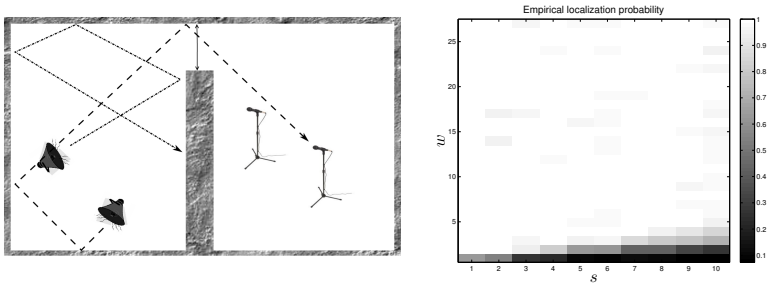

Fig. 6 Left: hearing behind walls scenario; right: empirical probability of accurate localization given the gap width w and number of sources k (results from [45]).

fectly. In reality, such knowledge is not readily available, hence there is an interest in inferring certain unknown physical parameters directly from the data. Ideally, such estimation would be done simultaneously with solving the original inverse problem, which is the second notion of "blind" localization in this subsection. For the acoustic wave equation (6) and boundary conditions (8), various parameters could be unknown. In this section, we consider two of them: sound speed $\boldsymbol{v}$, and the specific acoustic impedance ξ. Note that imposing a parameter model is necessary in this case; otherwise, these blind estimation problems would be ill-posed.

Blind Localization and Estimation of Sound Speed (BLESS). The speed of sound $\boldsymbol{v}$ is usually a slowly varying function of position and time, *e.g.* due to a temperature gradient of space caused by an air-conditioner or radiator. Provided that the temperature is in steady state and available, one could approximate $\boldsymbol{v}$ as constant. However, if such approximation is very inaccurate, the physical model embedded in the analysis operator will be wrong. The effects of such model inaccuracies have been exhaustively investigated [37, 12], and are known to significantly alter regularization performance. Therefore, our goal here is to simultaneously recover the pressure signal (in order to localize sound sources), and estimate the sound speed functional $\boldsymbol{v}$. For demonstrational purpose, we regard $\boldsymbol{v} := \boldsymbol{v}(r)$, *i.e.* a function that varies only in space.

To formalize the problem, consider the FDM Leapfrog discretization scheme presented in (11). Instead of a scalar sound speed parameter v, we now have a vector unknown corresponding to the sampled functional $\mathrm{v}_{\mathrm{ij}} = \boldsymbol{v}(r_x(\mathrm{i}), r_y(\mathrm{j})) > 0$. Denoting $q \in \mathbb{R}^{\mathrm{n}}$ the vector with stacked entries $\mathrm{q}_{\mathrm{i,j}} = \mathrm{v}_{\mathrm{i,j}}^{-2}$, we can represent the analysis operator Ω as follows:

$$\Omega = \Omega_1 + \mathtt{diag}\,(q)\,\Omega_2, \tag{46}$$

where the singular matrices Ω_1 and Ω_2 are obtained by factorizing wrt v in (11).

Assume the entries of q are in some admissible range $[\mathrm{v}_{\mathrm{max}}^{-2}, \mathrm{v}_{\mathrm{min}}^{-2}]$, e.g. given by the considered temperature range in a given environment. Moreover, assume that v and q are slowly varying functions. We model this smoothness by a vector space of polynomials of degree $\mathsf{r}-1$ in the space variables (constant over the time

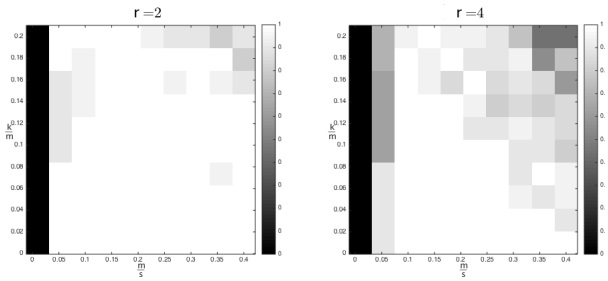

Fig. 7a Empirical localization probability with estimated sound speed.

dimension), which leads to the model $q = Fa$, where F is a dictionary of sampled polynomials and a is a weight vector [6].

Adding a as an unknown in (40) (instantiated, *e.g.*, with f_d a simple quadratic penalty), and introducing the auxiliary sparse variable c, yields the optimization problem:

$$\begin{aligned} &\min_{x,c,a} \; f_r(c_\Theta) + \lambda \|Ax - y\|_2^2 \\ &\text{subject to} \quad \Omega = \Omega_1 + \mathtt{diag}(Fa)\,\Omega_2, \quad \mathrm{v}_{\max}^{-2} \preceq Fa \preceq \mathrm{v}_{\min}^{-2} \\ &\qquad\qquad\quad \Omega x = c, \quad c_{\partial\Theta} = 0. \end{aligned} \tag{47}$$

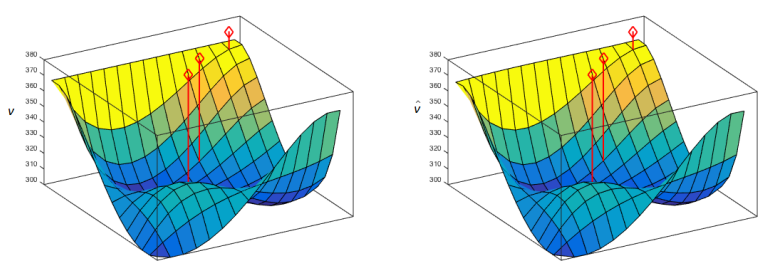

Fig. 7b The original sound speed (left) v and the estimate (right) $\hat{v}$ (the diamond markers indicate the spatial position of the sources).

Unfortunately, due to the presence of the bilinear term $\mathtt{diag}(Fa)\,\Omega_2 x$ relating optimization variables x and a, (47) is not a convex problem. However, it is biconvex - fixing either of these two, makes the modified problem convex again, thus its global solution is attainable. This enables us to design an ADMM-based heuristic, by developing an augmented Lagrangian (28) comprising the three variables:

$$L_{\rho_1,\rho_2}(c,x,a,u_1,u_2) = f_r(c_\Theta) + \chi_{\cdot=0}(c_{\partial\Theta}) + \frac{\rho_1}{2}\|(\Omega_1 + \mathtt{diag}(Fa)\,\Omega_2)x - c + u_1\|_2^2 + \frac{\rho_2\lambda}{2}\|Ax - y + u_2\|_2^2 + \chi_{\mathrm{v}_{\max}^{-2}\preceq\cdot\preceq\mathrm{v}_{\min}^{-2}}(Fa) - \frac{\rho_1}{2}\|u_1\|_2^2 - \frac{\rho_2}{2}\|u_2\|_2^2. \quad (48)$$

From here, the ADMM iterates are straightforwardly derived, similar to (29)-(31). We skip their explicit formulation to reduce the notational load of the chapter.

In order to demonstrate the joint estimation performance of the proposed approach, we vary the number of sources k and microphones m. First, a vector $\tilde{a}$ is randomly generated from centered Gaussian distribution of unit variance. Then, a is computed as the Euclidean projection of $\tilde{a}$ to a set $\left\{a \mid \mathrm{u}_{\max}^{-2} \preceq F_{\mathtt{null}}^{[\mathsf{r}]} a \preceq \mathrm{u}_{\min}^{-2}\right\}$. We let $\mathrm{u}_{\min} = 300$m/s and $\mathrm{u}_{\max} = 370$m/s, and use Neumann boundary conditions. The performance is depicted as an empirical localization probability graph in Figure 7a, for two values of the degree r of the polynomials used to model the smoothness of q. One can notice that the performance deteriorates with q less smooth (*i.e.* with larger r), since the dimension of the model polynomial space increases. When localization is successful, q is often perfectly recovered, as exemplified in Figure 7b.

Cosparse Acoustic Localization, Acoustic Impedance Estimation and Signal recovery (CALAIS). A perhaps even more critical acoustic parameter is the specific boundary impedance ξ in (8). While we may have an approximate guess of the sound speed, the impedance varies more abruptly, as it depends on the type of material composing a boundary of the considered enclosed space [47]. The approach recently proposed in [2] relies on the training phase using a known sound source, allowing one to calibrate the acoustic model for later use with unknown sources in the same environment. Here we present a method [5] to avoid the calibration phase, and, as for the sound speed, to simultaneously infer the unknown parameter ξ and the acoustic pressure x.

Now we consider discretization of the spatial domains' boundary. Let $\Omega_{\partial\Gamma}$ represent the subset of rows of the analysis operator Ω corresponding to the boundary conditions only, and let Ω_0 denote the matrix corresponding to initial conditions only (we have $\Omega_{\partial\Theta} = [\Omega_0^\mathsf{T}\ \Omega_{\partial\Gamma}^\mathsf{T}]^\mathsf{T}$, up to a row permutation). To account for Mur's boundary conditions, FDM discretization can be explicitly written as:

$$x_{\mathsf{i},\mathsf{j}}^{\tau+1}(1+\tfrac{\lambda}{\xi_{\mathsf{i},\mathsf{j}}}) - \left[2(1-2\lambda^2)x_{\mathsf{i},\mathsf{j}}^\tau + \lambda^2(x_{\mathsf{i},\mathsf{j}+1}^\tau + x_{\mathsf{i},\mathsf{j}-1}^\tau) + 2\lambda^2 x_{\mathsf{i}-1,\mathsf{j}}^\tau - \left(1 - \tfrac{\lambda}{\xi_{\mathsf{i},\mathsf{j}}}\right)x_{\mathsf{i},\mathsf{j}}^{\tau-1}\right] = 0, \quad (49)$$

where $\lambda = \mathrm{v}\mathrm{d}_\tau/\mathrm{d}_\mathsf{x} = \mathrm{v}\mathrm{d}_\tau/\mathrm{d}_\mathsf{y}$. Denote $\eta = \left[\xi_{1,1}\ \xi_{2,1}\ \ldots\ \xi_{\mathsf{i},\mathsf{j}}\ \ldots\right]^{-\mathsf{T}}$ the vector of inverse acoustic impedances, *i.e.* of *specific acoustic admittances*, which we assume does not change in time. We introduce the matrix S which distributes the admittances stored in η at appropriate positions in discretized space, and repeats these across all time instances $[1,\mathsf{t}]$. Factorizing (49) with respect to η, we can represent $\Omega_{\partial\Gamma}$ (up to an adequate row permutation) as:

$$\Omega_{\partial\Gamma} = \underbrace{\begin{bmatrix} \boxed{\underline{\Omega}_{\partial\Gamma_1}} & & \\ & \boxed{\underline{\Omega}_{\partial\Gamma_1}} & \\ & & \ddots \\ & & & \boxed{\underline{\Omega}_{\partial\Gamma_1}} \end{bmatrix}}_{\Omega_{\partial\Gamma_1}} + \ \mathtt{diag}(S\eta) \underbrace{\begin{bmatrix} \boxed{\underline{\Omega}_{\partial\Gamma_2}} & & \\ & \boxed{\underline{\Omega}_{\partial\Gamma_2}} & \\ & & \ddots \\ & & & \boxed{\underline{\Omega}_{\partial\Gamma_2}} \end{bmatrix}}_{\Omega_{\partial\Gamma_2}}. \tag{50}$$

where the rows of each block $\underline{\Omega}_{\partial\Gamma_1}$ (resp.) $\underline{\Omega}_{\partial\Gamma_2}$ are indexed by the space coordinate, while the blocks themselves are indexed by time.

Note that, for standard rooms, the boundaries are composed of walls, floor, ceiling, windows etc. At least on macroscopic scale, each of these structures is approximately homogeneous. Hence, we suppose that η admits a *piecewise constant* model, provided we take care of the ordering of elements within η. This weak assumption usually holds in practice, unless the discretization is very crude. To promote such a signal model, the discrete total variation norm $\|\eta\|_{\mathrm{TV}} = \|\nabla\eta\|_1$ is commonly used.

This model, along with the assumption that the initial/boundary conditions are homogeneous, inspires the following optimization problem:

$$\begin{aligned} &\min_{x,\eta} f_r(\Omega_\Theta x) + \|\eta\|_{\mathrm{TV}} + \lambda \|\Omega_{\partial\Gamma} x\|_2^2, \\ &\text{subject to } \Omega_0 x = 0,\ Ax = y,\ \eta \succeq 0 \\ &\qquad \Omega_{\partial\Gamma} = \Omega_{\partial\Gamma_1} + \mathtt{diag}(S\eta)\,\Omega_{\partial\Gamma_2}, \end{aligned} \tag{51}$$

where λ is a user-defined positive constant. Therefore, we end up with another biconvex problem, and propose to address it again by an ADMM heuristics. As in the previous case, one proceeds by defining the augmented Lagrangian which determines the algorithm. We do not develop these steps here, due to spatial constraints.

To illustrate the performance and robustness of the approach, we propose a somewhat challenging experimental setup. Within a simulated 2D spatial domain, modeled by high-resolution LFM discretization, we randomly distribute k sources and m microphones. The ground truth admittance is generated such that it approximately satisfies the assumed model - a white Gaussian noise is added to a piecewise constant vector η to account for model inaccuracy. The measurements y are also corrupted by noise, such that *per-sample* SNR is 20dB. Finally, the matrices Ω_Θ, $\Omega_{\partial\Gamma_1}$, $\Omega_{\partial\Gamma_2}$ and S, used in the optimization problem (51), are obtained by discretizing the physical model by a low-resolution LFM. This embeds some inaccuracy at the PDE modeling level, making simulations more physically relevant.

The results in Figure 8 are with respect to average Euclidean distance between ground truth source locations (empirical *Root Mean Square Error (RMSE)*), and the estimated ones. The dashed line denotes the spatial step size of the coarse grid, thus the errors below this threshold are tolerated. The median RMSE values indicate that localization is possible provided that the number of microphones is sufficiently high. The error increases with the number of sources to localize, but usually remains lower than the double of spatial stepsize, suggesting that the sources are localized in their true, immediate neighborhoods.

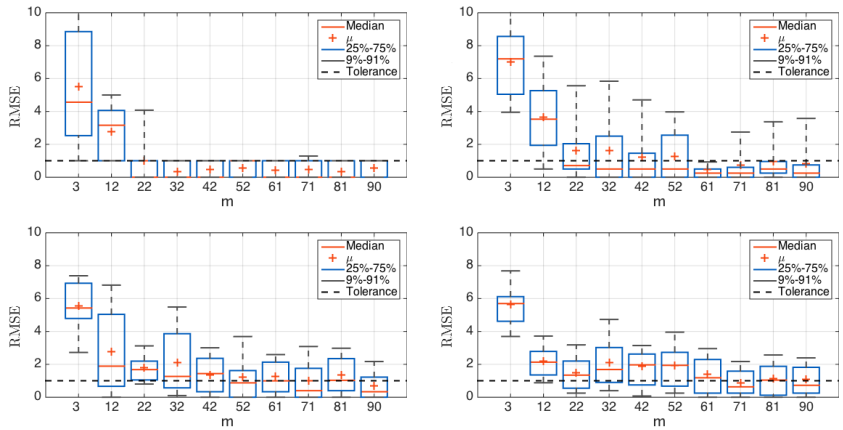

Fig. 8 RMSE for $\mathsf{k} = 1$, $\mathsf{k} = 2$ (top); and $\mathsf{k} = 3$, $\mathsf{k} = 4$ (bottom) simulated sources.

7.2 Cosparse brain source localization

Most source localization algorithms use one of the two following source models: the point source model, which explains the data with a small number of equivalent current dipoles and the distributed source model, which uses thousands of dipoles. Whereas the latter allows for an estimation of the spatial extent of the source, it requires to make assumptions about the spatial source distribution, which may lead to blurred (or even distorted) solutions [35]. On the other hand, the former often gives helpful first approximations and superior performance in environments where there are few sources which are clustered [35]. Regarding head models, they aim at representing geometrical and electrical properties of the different tissues composing the volume conductor. Various models were proposed going from concentric homogeneous spheres with isotropic conductivities for which analytic computations of Green's functions are possible, to realistically shaped models with refined tissue conductivity values [80].

FEM discretization and head model. As seen in Section 4.2, FEM can be used to discretize Poisson's equation with Neumann boundary condition (9)-(10) and derive an equation of the form $\Omega x = c$ where the so-called linear analysis operator [59] Ω is the *stiffness matrix*, and vectors x and c respectively contain the potential and total current flow values at the different nodes of the mesh. A realistic head model obtained from anatomical imaging modalities, such as Computed Tomography (CT) and structural Magnetic Resonance Imaging (sMRI), is segmented and a linear tetrahedral mesh of n nodes is then generated to divide it into small elements where a unique conductivity value is assigned to each one. In Figure 9, we illustrate the different steps performed to come to the sought algebraic system.

As previously explained in section 4.2, the $(\mathsf{n} \times \mathsf{n})$ matrix Ω is symmetric, positive semidefinite, rank deficient by one and sparse with only few components in each row [68]. Generally, instead of considering the singular linear system $\Omega x = c$,

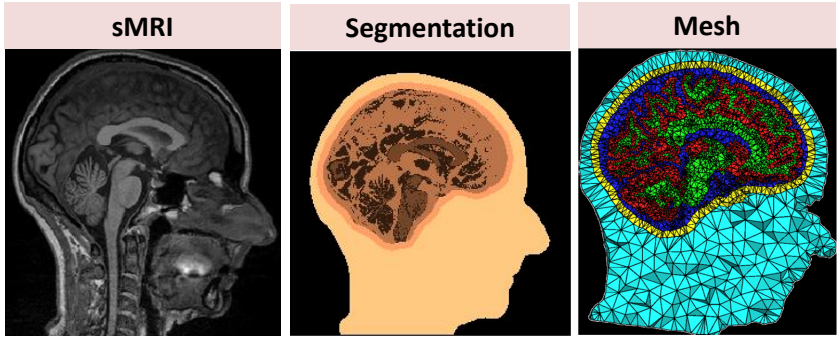

Fig. 9 The different steps of preprocessing in discretizing realistic head volume using FEM

another possibility is to transform it into a regular one and solve this instead. The regular system is chosen such that its unique solution belongs to the set of solutions of the original singular system. As described in [9], the easiest approach is to fix the value of the potential to zero in one node. The special structure of the matrix Ω then allows us to cancel the corresponding row and column in Ω and also the respective entry in the right-hand side vector c. This leads to a system for which the $(\mathsf{n}-1\times\mathsf{n}-1)$ resulting matrix is symmetric, sparse and positive definite, as it can be derived from a bilinear form satisfying the sames properties as $\mathfrak{a}(.,.)$ in section 4.2. By abuse of notation we still denote it Ω. The solution of this system solves the initial system with a zero potential value in the reference node.

Source model. We consider the following assumptions:

A1. There are g possible current sources that cover the volume of the gray matter $\mathscr{G}$.
A2. Each current source is modeled as a "pure" dipole, consisting in a point dipole characterized by its position ρ_{q} and moment $p_{\mathsf{q}} = \|p_{\mathsf{q}}\|_2\, n_{\mathsf{q}}$ where $\|p_{\mathsf{q}}\|_2$ and n_{q} correspond to the activity level and the orientation of the dipole, respectively. In this model, the current density is expressed as $j = \sum_{\mathsf{q}=1}^{\mathsf{g}} p_{\mathsf{q}}\, \delta(\rho_{\mathsf{q}})$ [70].
A3. Each current dipole is oriented orthogonally to the cortical surface (p_{q} is the normal to the cortical surface at position ρ_{q}).
A4. At most k ($\mathsf{k} < \mathsf{m}$) current dipoles have non-negligible activity.

In addition to the "pure" dipole, there are other numerical methods used to model the current dipole, such as the subtraction potential method and the direct potential approach using Saint Venant's principle [82, 70]. Though the "pure" dipole is a mathematical idealization of the real "physical" dipole that has finite current and separation between its monopoles, it is frequently used for its simple derivation.

According to the variational form introduced in (15), the component c_{i} in the *load vector* is

$$c_{\mathsf{i}} = \int_{\Theta_h} \nabla.\, j\, \phi_{\mathsf{i}}(\boldsymbol{r})\, d\boldsymbol{r} \tag{52}$$

By applying Green's identity to (52), using the fact that no current sources are present in the scalp $\partial\Theta$, considering the expression of the domain Θ_h as a union of d tetrahedra Θ_h^{e} present in the mesh, and eventually assuming that each dipole position ρ_{q} coincides with a node position in $\mathscr{G}$, appropriate calculations lead to rewrite the entry in the *load vector* as:

$$c_{\mathsf{i}} = \begin{cases} -p_{\mathsf{q}} \cdot \sum_{\mathsf{e}\in\mathsf{d}_{\mathsf{i}}} \nabla^{\mathsf{e}}\phi_{\mathsf{i}}(\rho_{\mathsf{q}}) & \text{if} \quad \boldsymbol{r}_{\mathsf{i}} = \rho_{\mathsf{q}} \\ 0 & \text{otherwise} \end{cases} \tag{53}$$

where $\nabla^{\mathsf{e}}\phi_{\mathsf{i}}$ is the gradient of function ϕ_{i} over element Θ_h^{e}. By injecting the expression of dipole moment in (53) and in the case of non-zero c_{i}, the latter can be expressed as the product $c_{\mathsf{i}} = \|p_{\mathsf{q}}\|_2 \; n_{\mathsf{q}} . \sum_{\mathsf{e}\in\mathsf{d}_{\mathsf{i}}} \nabla^{\mathsf{e}}\phi_{\mathsf{i}}(\rho_{\mathsf{q}})$, which allows us to factorize vector c as $c = Bz$ where B is a $(\mathsf{n}-1 \times \mathsf{n}-1)$ diagonal matrix defined by

$$B_{\mathsf{i},\mathsf{i}} = \begin{cases} -n_{\mathsf{q}} . \sum_{\mathsf{e}\in\mathsf{d}_{\mathsf{i}}} \nabla^{\mathsf{e}}\phi_{\mathsf{i}}(\rho_{\mathsf{q}}) & \text{if} \quad \boldsymbol{r}_{\mathsf{i}} = \rho_{\mathsf{q}} \\ 1 & \text{otherwise} \end{cases} \tag{54}$$

In addition, z is an $(\mathsf{n}-1)$-dimensional sparse vector with $\mathsf{n}-1-\mathsf{g}$ known zero elements defined as

$$z_{\mathsf{i}} = \begin{cases} \|p_{\mathsf{q}}\|_2 & \text{if} \quad \boldsymbol{r}_{\mathsf{i}} = \rho_{\mathsf{q}} \\ 0 & \text{otherwise} \end{cases} \tag{55}$$

Consequently, matrix B conveys our knowledge about the orientation of the g dipoles of the grey matter $\mathscr{G}$, while the non-zero elements in vector z represent the activity of dipoles restricted to the cortical volume. It is noteworthy that, even when dipoles positions do not coincide with node positions, the factorization of vector c is still possible. However, in that case, the matrix B is no longer a square matrix but rather a tall matrix (not left invertible), which makes the computation of Ω more complicated. In addition, the assumption on the dipoles position is still realistic and affordable by using a dense mesh in $\mathscr{G}$.

Overall model. Combining the properties $\Omega x = c$, the source model $c = Bz$ and the observation model $y = Ax$, the brain source localization defined above can finally be reformulated as a cosparse analysis model fitting problem given by:

$$\begin{cases} \tilde{\Omega}\, x = z \\ y = Ax \end{cases} \tag{56}$$

where the analysis operator $\tilde{\Omega}$ is given by $\tilde{\Omega} = B^{-1}\,\Omega$ and the sensing matrix A is an $\mathsf{m} \times \mathsf{n}-1$ row-reduced identity matrix. As B is diagonal, $\tilde{\Omega}$ is still sparse.

Optimization problem. To address the cosparse analysis model fitting problem (56), we express the following convex optimization problem:

$$\min_{v} \|\tilde{\Omega}_1 x\|_1 + \lambda \|\tilde{\Omega}_2 x\|_2^2 \tag{57}$$
$$\text{subject to } Ax = y.$$

where $\tilde{\Omega}_1$ is the $(\mathsf{g} \times \mathsf{n} - 1)$ submatrix of $\tilde{\Omega}$ obtained by extracting the rows of $\tilde{\Omega}$ corresponding to the support set of the grey matter $\mathscr{G}$, whereas $\tilde{\Omega}_2$ corresponds to the rows indicated by the complementary set $\bar{\mathscr{G}}$. By choosing the appropriate weight λ, the cosparse solution of the optimization problem (57) will fulfill the assumptions (**A1**) to (**A4**). Namely, $\|\tilde{\Omega}_1 x\|_1$ will promote sparsity at the surface of the cortex, while $\lambda \|\tilde{\Omega}_2 x\|_2^2$ will attenuate the signal in the other regions. The linear constraints $Ax = y$ ensure that the model fits the electrode measurements. Depending on the resolution of the cubic grid tuned by n, the problem can reach considerably large scale. Therefore, we use the Chambolle-Pock method as described in section 5.2.

Experiments and performance criterion. One scenario was considered for a comparison of performance between the analysis and synthesis approaches. It aims at studying the influence of the SNR.

More particularly, $\mathsf{k} = 2$ synchronous epileptic dipoles were placed in $\mathscr{G}$ at $\rho_1 = [-71, 31, 92]^\mathsf{T}$ and $\rho_2 = [-70, 27, 92]^\mathsf{T}$, respectively, (locations are given in centimeters). Note that the origin (O) of the head model was defined as the intersection of the O-Cz axis (z-axis), the O-T4 axis (x-axis) and the O-Fpz axis (y-axis). A physiologically-relevant model [39] was used to generate the time series corresponding to epileptic activity. It is noteworthy that this activity was the same for both epileptic dipoles, leading to synchronous epileptic sources. On the other hand, the background activity, i.e. the activity of non-epileptic dipoles of $\mathscr{G}$, was generated as Gaussian and as temporally and spatially white. Its power was controlled by a multiplicative coefficient in order to get different SNR values.

As far as the head model is concerned, we used a realistic head model obtained from sMRI. Ninety-one electrodes ($\mathsf{m} = 91$) were placed on the scalp using the 10-5 system [61]. In addition, in order to apply the FEM and compute the analysis operator Ω, we created a linear tetrahedral mesh of $\mathsf{n} = 180585$ nodes. Consequently, the size of $\tilde{\Omega}$ and the number of dipoles of $\mathscr{G}$ were $(\mathsf{n}-1) \times (\mathsf{n}-1) = 180584 \times 180584$ and $\mathsf{g} = 3110$, respectively.

The quality of the source localization was quantified for each method by means of the average Root Mean Square Error (RMSE), which is defined by:

$$RMSE = \frac{1}{\mathsf{k}\,\mathsf{mc}} \sum_{\mathsf{q}=1}^{\mathsf{k}} \sum_{\mathsf{i}=1}^{\mathsf{mc}} \min_{1 \le \mathsf{j} \le \mathsf{k}} \|\rho_\mathsf{q} - \widehat{\rho}_\mathsf{j}\| \tag{58}$$

where mc is the number of realizations fixed to 71, where ρ_q is the ground truth position of the q-th epileptic dipole and where $\widehat{\rho}_\mathsf{j}$ is the j-th dipole location estimated during the i-th Monte Carlo trial. It is noteworthy that from one realization to another, the temporal dynamics of the g dipoles of $\mathscr{G}$ were changed while the location of the three epileptic dipoles stayed unchanged.

Figure 10 shows the RMSE criterion at the output of both algorithms as a function of the SNR. It appears that the analysis method is more robust with respect to the presence of noise than the synthesis one. Indeed, it succeeds in localizing perfectly both epileptic dipoles beyond 12 dB while the synthesis-based method does not manage to do it perfectly.

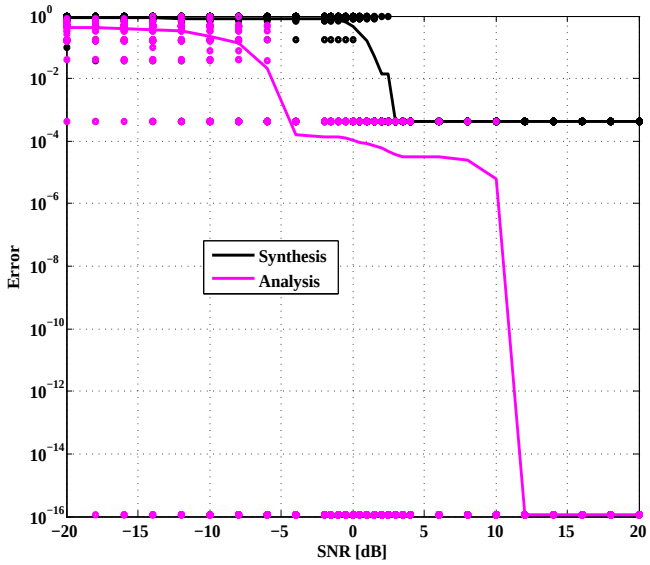

Fig. 10 Behavior of the analysis and synthesis approaches as a function of the SNR for $\mathsf{k} = 2$ epileptic dipoles and $\mathsf{m} = 91$ electrodes (solid line: average, dots: individual realizations).

8 Summary and conclusion

In many physics-driven inverse problems, one can leverage both a sparsity hypothesis on some source term and the properties of the underlying PDE. A classical approach to combine these ingredients is to build a dictionary of Green's function of the PDE and to exploit traditional techniques (ℓ_1 regularization) for sparse synthesis reconstruction to estimate the source field. Yet, for high spatial and/or temporal resolutions, pre-computing Green's functions can be challenging, and the synthesis version of the ℓ_1 optimization problem may become numerically intractable due to polynomial complexities of too high degree in the overall size of the discretization.

An alternative is to discretize the PDE itself, *e.g.* through a Finite Difference Scheme or the Finite Element Methods, which naturally leads to very sparse *analysis operators* rather than dictionaries. While the two approaches (synthesis and analysis) are formally equivalent, a primary advantage of the *cosparse analysis* regularization is a much smaller iteration cost. Although demonstration of the full potential of the existing cosparse approaches on *real* acoustic or EEG data remains to be done, results shown on simulated data allow to support our claims on their interest. Overall, as illustrated in this chapter, a promising approach to achieve precision and scalability is to combine the synthesis approach and the analysis one in a multiscale optimization strategy. Besides scalability, the cosparse analysis approach opens interesting perspectives regarding the ability to solve extended inverse problems where some physical parameters such as impedance or speed of sound may be unknown. Using FEM, it allows to handle complex geometries, and as demon-

strated on some brain source localization problems, it offert competitive robustness to noise.

Beyond model-based methods, an intensive research in machine learning has recently inspired several training-based approaches, *e.g.* [42, 20, 2], which, however, either focus on a specific aspect of the problem at hand, or even neglect its explicit physical nature. Instead, we feel that a promising research avenue are pre-trained physics-driven cosparse models, potentially leading to "fully-learnable" methods, unrestricted by parameterization or the geometry of the environment.

References

1. L. Albera, A. Ferreol, D. Cosandier-Rimele, I. Merlet, and F. Wendling. Brain source localization using a fourth-order deflation scheme. *IEEE Transactions Biomedical Engineering*, 55(2):490–501, Feb. 2008.
2. N. Antonello, T. van Waterschoot, M. Moonen, and P. A. Naylor. Identification of surface acoustic impedances in a reverberant room using the FDTD method. In *Acoustic Signal Enhancement (IWAENC), 2014 14th International Workshop on*, pages 114–118. IEEE, 2014.
3. J. Barker, R. Marxer, E. Vincent, and S. Watanabe. The third chime speech separation and recognition challenge: Dataset, task and baselines. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 504–511, Dec 2015.
4. A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.
5. N. Bertin, S. Kitić, and R. Gribonval. Joint estimation of sound source location and boundary impedance with physics-driven cosparse regularization. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 6340–6344. IEEE, 2016.
6. C. Bilen, S. Kitić, N. Bertin, and R. Gribonval. Sparse acoustic source localization with blind calibration for unknown medium characteristics. In *iTwist-2nd international-Traveling Workshop on Interactions between Sparse models and Technology*, 2014.
7. C. Blandin, A. Ozerov, and E. Vincent. Multi-source TDOA estimation in reverberant audio using angular spectra and clustering. *Signal Processing*, 92(8):1950–1960, 2012.
8. T. Blumensath and M. E. Davies. Sampling theorems for signals from the union of finite-dimensional linear subspaces. *IEEE Transactions on Information Theory*, 55(4):1872–1882, 2009.
9. P. Bochev and R. B. Lehoucq. On the finite element solution of the pure Neumann problem. *SIAM Review*, 47(1):50–66, 2005.
10. 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*, 3(1):1–122, 2011.
11. J. Campbell. *Introduction to Remote Sensing*. Guilford Press, 1996.
12. E. J. Candès, J. K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223, 2006.
13. A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.
14. A. Chambolle and T. Pock. An introduction to continuous optimization for imaging. *Acta Numerica*, 25:161–319, 2016.
15. V. Chandrasekaran and M. I. Jordan. Computational and statistical tradeoffs via convex relaxation. *Proceedings of the National Academy of Sciences*, 110(13):E1181–E1190, 2013.
16. G. Chardon, T. Nowakowski, J. De Rosny, and L. Daudet. A blind dereverberation method for narrowband source localization. *Journal of Selected Topics in Signal Processing*, 2015.

17. P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.
18. L. Condat. A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *Journal of Optimization Theory and Applications*, 158(2):460–479, 2013.
19. G. Dassios and A. Fokas. The definite non-uniqueness results for deterministic eeg and meg data. *Inverse Problems*, 29(6):065012, 2013.
20. A. Deleforge, F. Forbes, and R. Horaud. Acoustic space learning for sound-source separation and localization on binaural manifolds. *International journal of neural systems*, 25(01):1440003, 2015.
21. W. Deng and W. Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. *Journal of Scientific Computing*, 66(3):889–916, 2016.
22. A. Devaney and G. Sherman. Nonuniqueness in inverse source and scattering problems. *IEEE Transactions on Antennas and Propagation*, 30(5):1034–1037, 1982.
23. I. Dokmanić. *Listening to Distances and Hearing Shapes*. PhD thesis, EPFL, 2015.
24. I. Dokmanić and M. Vetterli. Room helps: Acoustic localization with finite elements. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012*, pages 2617–2620. Ieee, 2012.
25. P. L. Dragotti, M. Vetterli, and T. Blu. Sampling moments and reconstructing signals of finite rate of innovation: Shannon meets strang–fix. *IEEE Transactions on Signal Processing*, 55(5):1741–1757, 2007.
26. J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1):293–318, 1992.
27. M. Elad, P. Milanfar, and R. Rubinstein. Analysis versus synthesis in signal priors. *Inverse problems*, 23(3):947, 2007.
28. D. C.-L. Fong and M. Saunders. LSMR: An iterative algorithm for sparse least-squares problems. *SIAM Journal on Scientific Computing*, 33(5):2950–2971, 2011.
29. I. Fried. Bounds on the spectral and maximum norms of the finite element stiffness, flexibility and mass matrices. *International Journal of Solids and Structures*, 9(9):1013–1034, 1973.
30. A. Friedman. *Partial differential equations of parabolic type*. Courier Dover Publications, 2008.
31. D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.
32. S. Gannot, D. Burshtein, and E. Weinstein. Signal enhancement using beamforming and nonstationarity with applications to speech. *IEEE Transactions on Signal Processing*, 49(8):1614–1626, 2001.
33. M. S. Gockenbach. *Understanding and implementing the finite element method*. SIAM, 2006.
34. T. Goldstein, B. O'Donoghue, S. Setzer, and R. Baraniuk. Fast alternating direction optimization methods. *SIAM Journal on Imaging Sciences*, 7(3):1588–1623, 2014.
35. R. Grech, T. Cassar, J. Muscat, K. P. Camilleri, S. G. Fabri, M. Zervakis, P. Xanthopoulos, V. Sakkalis, and B. Vanrumste. Review on solving the inverse problem in EEG source analysis. *Journal of NeuroEngineering and Rehabilitation*, 7:5–25, November 2008.
36. H. Hallez, B. Vanrumste, R. Grech, J. Muscat, W. De Clercq, A. Vergult, Y. D 'asseler, K. P. Camilleri, S. G. Fabri, S. V. Huffel, and I. Lemahieu. Review on solving the forward problem in EEG source analysis. *J. Neuroeng. Rehabil.*, 4(4), 2007.
37. M. Herman, T. Strohmer, et al. General deviants: An analysis of perturbations in compressed sensing. *IEEE Journal of Selected topics in signal processing*, 4(2):342–349, 2010.
38. V. Isakov. *Inverse problems for partial differential equations*, volume 127. Springer Science & Business Media, 2006.
39. B. H. Jansen and V. G. Rit. Electroencephalogram and visual evoked potential generation in a mathematical model of coupled cortical columns? *Biological Cybernetics*, 73(4):357–366, Sept. 1995.

40. R. Jenatton, J.-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. *The Journal of Machine Learning Research*, 12:2777–2824, 2011.
41. C. R. Johnson. Computational and numerical methods for bioelectric field problems. *Crit. Rev. Biomed. Eng.*, 25(1):1–81, 1997.
42. U. S. Kamilov and H. Mansour. Learning optimal nonlinearities for iterative thresholding algorithms. *IEEE Signal Processing Letters*, 23(5):747–751, 2016.
43. Y. Kim and P. Nelson. Optimal regularisation for acoustic source reconstruction by inverse methods. *Journal of Sound and Vibration*, 275(35):463 – 487, 2004.
44. S. Kitić, L. Albera, N. Bertin, and R. Gribonval. Physics-driven inverse problems made tractable with cosparse regularization. *IEEE Transactions on Signal Processing*, 64(2):335–348, 2016.
45. S. Kitić, N. Bertin, and R. Gribonval. Hearing behind walls: localizing sources in the room next door with cosparsity. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 3087–3091. IEEE, 2014.
46. K. Kowalczyk and M. v. Walstijn. Modeling frequency-dependent boundaries as digital impedance filters in FDTD and K-DWM room acoustics simulations. *Journal of the Audio Engineering Society*, 56(7/8):569–583, 2008.
47. H. Kuttruff. *Room acoustics*. Crc Press, 2016.
48. J. Le Roux, P. T. Boufounos, K. Kang, and J. R. Hershey. Source localization in reverberant environments using sparse optimization. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013*, pages 4310–4314. IEEE, 2013.
49. R. J. LeVeque. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. SIAM, 2007.
50. T. T. Lin and F. J. Herrmann. Compressed wavefield extrapolation. *Geophysics*, 72(5):SM77–SM93, 2007.
51. D. Malioutov. A sparse signal reconstruction perspective for source localization with sensor arrays, master's thesis, 2003.
52. D. Malioutov, M. Çetin, and A. S. Willsky. A sparse signal reconstruction perspective for source localization with sensor arrays. *IEEE Transactions on Signal Processing*, 53(8):3010–3022, 2005.
53. 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*, 41:75–84, 2003.
54. J.-J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965.
55. J. C. Mosher and R. M. Leahy. Source localization using Recursively Applied and Projected (RAP) music. *IEEE Transactions On Signal Processing*, 47(2):332–340, Feb. 1999.
56. W. Munk, P. Worcester, and C. Wunsch. *Ocean acoustic tomography*. Cambridge University Press, 2009.
57. G. Mur. Absorbing boundary conditions for the finite-difference approximation of the time-domain electromagnetic-field equations. *IEEE transactions on Electromagnetic Compatibility*, EMC-23(4):377–382, 1981.
58. J. Murray-Bruce and P. L. Dragotti. Solving physics-driven inverse problems via structured least squares. In *Signal Processing Conference (EUSIPCO), 2016 24th European*, pages 331–335. IEEE, 2016.
59. S. Nam, M. E. Davies, M. Elad, and R. Gribonval. The cosparse analysis model and algorithms. *Applied and Computational Harmonic Analysis*, 34(1):30–56, January 2013.
60. Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.
61. R. Oostenvelda and P. Praamstra. The five percent electrode system for high-resolution EEG and ERP measurements. *Clinical Neurophysiology, Elsevier*, 112(4):713–719, April 2001.
62. A. Pezeshki, Y. Chi, L. L. Scharf, and E. K. Chong. Compressed sensing, sparse inversion, and model mismatch. In *Compressed Sensing and its Applications*, pages 75–95. Springer, 2015.

63. T. Piotrowski, D. Gutierrez, I. Yamada, and J. Zygierewicz. Reduced-rank neural activity index for EEG/MEG multi-source localization. In *ICASSP'14, IEEE International Conference on Acoustics Speech and Signal Processing*, Florence, Italy, May 4-9 2014.
64. L. C. Potter, E. Ertin, J. T. Parker, and M. Cetin. Sparsity and compressed sensing in radar imaging. *Proceedings of the IEEE*, 98(6):1006–1020, 2010.
65. J. Provost and F. Lesage. The application of compressed sensing for photo-acoustic tomography. *IEEE transactions on medical imaging*, 28(4):585–594, 2009.
66. Qianqian Fang and D. A. Boas. Tetrahedral mesh generation from volumetric binary and grayscale images. In *2009 IEEE Int. Symp. Biomed. Imaging From Nano to Macro*, pages 1142–1145. IEEE, jun 2009.
67. A. Rosenthal, D. Razansky, and V. Ntziachristos. Fast semi-analytical model-based acoustic inversion for quantitative optoacoustic tomography. *IEEE transactions on medical imaging*, 29(6):1275–1285, 2010.
68. Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.
69. O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier, and F. Lenzen. *Variational methods in imaging*, volume 320. Springer, 2009.
70. P. Schimpf, C. Ramon, and J. Haueisen. Dipole models for the EEG and MEG. *IEEE Trans. Biomed. Eng.*, 49(5):409–418, may 2002.
71. R. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions On Antennas Propagation*, 34(3):276–280, Mar. 1986. Reprint of the original 1979 paper from the RADC Spectrum Estimation Workshop.
72. S. Shalev-Shwartz and N. Srebro. SVM optimization: inverse dependence on training set size. In *Proceedings of the 25th international conference on Machine learning*, pages 928–935. ACM, 2008.
73. R. Shefi and M. Teboulle. Rate of convergence analysis of decomposition methods based on the proximal method of multipliers for convex minimization. *SIAM Journal on Optimization*, 24(1):269–297, 2014.
74. J.-L. Starck and J. M. Fadili. An overview of inverse problem regularization using sparsity. In *IEEE ICIP*, pages 1453 – 1456, Cairo, Egypt, Nov. 2009.
75. G. Strang. *Computational science and engineering*, volume 791. Wellesley-Cambridge Press Wellesley, 2007.
76. L. Thomas. Using a computer to solve problems in physics. *Applications of digital computers*, pages 44–45, 1963.
77. J. A. Tropp and S. J. Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, June 2010.
78. K. Uutela, M. Hamalainen, and E. Somersalo. Visualization of magnetoencephalographic data using minimum current estimates. *Elsevier NeuroImage*, 10(2):173–180, August 1999.
79. J.-M. Valin, F. Michaud, J. Rouat, and D. Létourneau. Robust sound source localization using a microphone array on a mobile robot. In *Proceedings. 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003.(IROS 2003).*, volume 2, pages 1228–1233. IEEE, 2003.
80. F. Vatta, F. Meneghini, F. Esposito, S. Mininel, and F. Di Salle. Realistic and spherical head modeling for EEG forward problem solution: A comparative cortex-based analysis. *Computational Intelligence and Neuroscience*, 2010.
81. M. Vetterli, P. Marziliano, and T. Blu. Sampling signals with finite rate of innovation. *IEEE transactions on Signal Processing*, 50(6):1417–1428, 2002.
82. C. Wolters, H. Köstler, C. Möller, J. Härdtlein, and A. Anwander. Numerical approaches for dipole modeling in finite element method based source analysis. *International Congress Series*, 1300:189 – 192, 2007.