A Fast Homotopy Algorithm for Gridless Sparse Recovery

Jean-Baptiste Courbot, Bruno Colicchio

A FAST HOMOTOPY ALGORITHM FOR GRIDLESS SPARSE RECOVERY*

JEAN-BAPTISTE COURBOT AND BRUNO COLICCHIO †

Abstract. In this paper, we study the solving of the gridless sparse optimization problem and its application to 3D image deconvolution. Based on the recent works of [14] introducing the Sliding Frank-Wolfe algorithm to solve the Beurling LASSO problem, we introduce an accelerated algorithm, denoted BSFW, that preserves its convergence properties, while removing most of the costly local descents. Besides, as the solving of BLASSO still relies on a regularization parameter, we introduce an homotopy algorithm to solve the constrained BLASSO that allows to use a more practical parameter based on the image residual, *e.g.* its standard deviation. Both algorithms benefit from a finite termination property, *i.e.* they are guaranteed to find the solution in a finite number of step under mild conditions. These methods are then applied on the problem of 3D tomographic diffractive microscopy images, with the purpose of explaining the image by a small number of atoms in convolved images. Numerical results on synthetic and real images illustrates the improvement provided by the BSFW method, the homotopy method and their combination.

Key words. Beurling LASSO, gridless sparse optimization, homotopy algorithm, 3D deconvolution, tomographic diffractive microscopy

1. Introduction.

1.1. Observation model and work hypothesis. In this paper, we consider the problem of the deconvolution of some image $\mathbf{y}$ containing a small number of component. Our target application is the 3D deconvolution of Tomographic Diffractive Microscopy (TDM), so we assume without loss of generality that $\mathbf{y}$ is a 3D image.

The components of the image are represented by a Radon measure $\mu_{\boldsymbol{w},\boldsymbol{\theta}}$ observed through an imaging operator $\boldsymbol{\Phi}$ under an additive noise:

$$\mathbf{y} = \boldsymbol{\Phi}\mu_{\boldsymbol{w},\boldsymbol{\theta}} + \boldsymbol{\epsilon} \tag{1.1}$$

Here and in the following, we consider for the measures $\mu_{\boldsymbol{w},\boldsymbol{\theta}}$ a weighted Dirac mass sum of the form $\sum_{n=1}^{N} w_n \delta_{\boldsymbol{\theta}_n}$, with the weight vector $\boldsymbol{w} = \{w_1, \dots, w_N\} \in \mathbb{R}_*^N$. $\boldsymbol{\theta}_n$ locates, for each atom, its parameters within the bounded domain $\mathcal{D}$ of dimension D. The Radon space corresponding to $\mathcal{D}$ is denoted $\mathcal{M}(\mathcal{D})$ so that $\mu_{\boldsymbol{w},\boldsymbol{\theta}} \in \mathcal{M}(\mathcal{D})$. We assume that the representation of each Dirac mass in the image space is given by some $\mathcal{G} : \mathcal{D} \times \mathbb{R} \to \mathbb{R}^3$. Furthermore, $\boldsymbol{\Phi}$ embeds a point spread function (PSF) $\mathbf{H}$ that blurs the observations of atoms.

Summing up, we can rephrase (1.1) as:

$$\mathbf{y} = \mathbf{H} * \sum_{n=1}^{N} \mathcal{G}(\boldsymbol{\theta}_n, w_n) + \boldsymbol{\epsilon}. \tag{1.2}$$

Generally, we assume N is reasonably small, so that μ embeds a compact representation of the large $\mathbf{y}$. The problem handled here consists in estimating N, and $\{\boldsymbol{\theta}_n, \boldsymbol{w}_n\}_{n=1}^{n=N}$ while knowing only $\mathbf{y}$ and $\mathbf{H}$.

1.2. Gridless sparse recovery. Inverting (1.2) is typically made using sparsity-enforcing methods. To do so, we search for the *sparser* solution being close enough to

*Submitted to the editors on August 25, 2020.

†IRIMAS UR 7499, 61 rue Albert Camus, 68100 Mulhouse, France (jean-baptiste.courbot@uha.fr).

the observed $\mathbf{y}$, solving :

$$\min_{\mu \in \mathcal{M}(\mathcal{D})} \mathcal{R}(\mu) \text{ subject to } \|\mathbf{y} - \mathbf{\Phi}\mu\|_2 \leq e. \tag{1.3}$$

where $e > 0$ can be, *e.g.*, an estimation of the noise level in the data. The regularizer $\mathcal{R}$ can take many forms. It is well documented in the literature that the ℓ_0 "norm" of μ directly enforces sparsity [35], at the cost of making the problem non-convex, non-differentiable and thus difficult to solve. On the other hand, the relaxation based on the ℓ_1 norm provides an interesting framework, since the resulting problem remains differentiable in almost all points, while a ℓ_2 norm may induce data overfitting.

Besides, we are interested in inverting (1.2) in a continuous fashion: the atoms to search for do not lie on a pre-established grid or dictionary, which form the conventional framework for sparse recovery. Indeed, assuming the prior existence of such a grid leads to practical difficulties, since the choice of the dictionary critically influence the outcome of many methods.

In [18], the Continuous Basis Pursuit interpolates between parameters defined on a grid so as to reach subgrid accuracy. Other approach, seen in [8, 10, 13], rephrase the problem in a fully continuous formalism. By doing so, the sparsity-promoting ℓ_1 norm is replaced by its continuous counterpart, the total variation of measures. The problem is then referred as Beurling LASSO or BLASSO [2].

In this work, we focus on solving the constrained BLASSO problem:

$$\min_{\mu_{\boldsymbol{w},\boldsymbol{\theta}} \in \mathcal{M}(\mathcal{D})} |\mu_{\boldsymbol{w},\boldsymbol{\theta}}| \text{ subject to } \|\mathbf{y} - \mathbf{\Phi}\mu_{\boldsymbol{w},\boldsymbol{\theta}}\|_2 \leq e, \tag{$\mathcal{P}_1$}$$

where $|\mu_{\boldsymbol{w},\boldsymbol{\theta}}|$ denotes the total mass of the measure $\mu_{\boldsymbol{w},\boldsymbol{\theta}} \in \mathcal{M}(\mathcal{D})$. In the case of a sum of Dirac masses ($\mu_{\boldsymbol{w},\boldsymbol{\theta}} = \sum_{n=1}^{N} w_n \delta_{\boldsymbol{\theta}_n}$), it is analog to a ℓ_1 norm with $|\mu_{\boldsymbol{w},\boldsymbol{\theta}}| = \sum_{n=1}^{N} |w_n|$. In other words, we search for the best compromise between the sparsity prior and data fit.

Introducing $\lambda > 0$, this trade-off is made apparent in the following equivalent minimization problem:

$$\min_{\mu_{\boldsymbol{w},\boldsymbol{\theta}} \in \mathcal{M}(\mathcal{D})} C(\mathbf{y}, \lambda, \mu_{\boldsymbol{w},\boldsymbol{\theta}}) \overset{\text{def.}}{=} \min_{\mu_{\boldsymbol{w},\boldsymbol{\theta}} \in \mathcal{M}(\mathcal{D})} \frac{1}{2} \|\mathbf{y} - \mathbf{\Phi}\mu_{\boldsymbol{w},\boldsymbol{\theta}}\|_2^2 + \lambda |\mu_{\boldsymbol{w},\boldsymbol{\theta}}|. \tag{$\mathcal{P}_\lambda$}$$

This unconstrained formulation is generally preferred because it is more closely related to convex quadratic programming.

To numerically solve the BLASSO problem, several approaches have been proposed. In [10, 30], the problem is rephrased as a semi-definite program, while the ADCG solver proposed in [8, 7] relies on an alternating gradient based method which iteratively adds Dirac masses to the solution. Recently, a variant of the ADCG called Sliding Frank-Wolfe (SFW) appeared in [14], which is guaranteed to converge in a finite number of steps under mild assumptions.

1.3. Homotopy. Despite providing a quite appealing way to solve ($\mathcal{P}_\lambda$), solving the BLASSO in practice require to set λ at an adequate value. So the choice of λ is often left to the practitioner, making it a tuning parameter: a large λ yields few atoms, while a low λ provides a better data fit with a denser solution. However there is no straight relation between the value of λ and the properties of the solution.

So, in this paper, we are interested in methods to search for the best λ according to some criterion which can be relevant in practice.This is the purpose of the *homotopy*

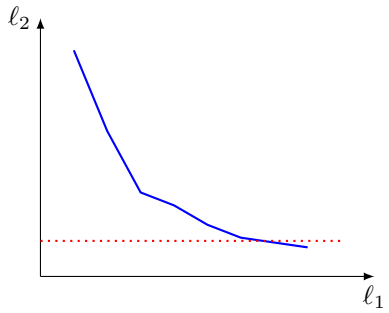


Fig. 1: Schematic depiction of a Pareto frontier. Points on the line are solution to the problem at hand, which forms different compromises between the ℓ_2 and ℓ_1 elements. Homotopy algorithms follow (a part of) this piecewise linear frontier, starting at a high ℓ_2 norm and evolving towards a high ℓ_1 norm. A limit value can be set (red dots) to stop the homotopy when attained.

algorithms, introduced in [24, 25]. These methods solve ($\mathcal{P}_1$) for some noise level value e, by solving ($\mathcal{P}_\lambda$) for a sequence of λ.

Such algorithms explore the so-called *Pareto frontier*, materializing the solutions of ($\mathcal{P}_1$) for a set of $\lambda > 0$ (see Fig. 1). Indeed, decreasing λ while keeping the same minimizing value of $\mathcal{P}_\lambda$ moves the ℓ_2 objective to the ℓ_1 objective. This is why these algorithms are referred to as an *homotopy* from ℓ_2 to ℓ_1. For ($\mathcal{P}_1$), the Pareto frontier forms a polygonal path with a discrete number of vertices [15]: changes only occur at critical values of λ. This important property helps probing this frontier.

Most homotopy algorithms in the literature are designed for ℓ_1-based problems [31, 3, 33, 1, 34], with the exception of [29]. All of them are designed in a discretized framework based on a dictionary. To our knowledge, there is no work in the literature on homotopy-based algorithms on the space of measures.

1.4. This paper. In this paper, we aim at bringing together the BLASSO solvers and the homotopy algorithms. To do so, we first study (in section 2) the unconstrained BLASSO solving of ($\mathcal{P}_\lambda$) by SFW [14]. We show that removing some steps of SFW yield a significantly faster solver, that we name *Boosted* SFW or BSFW. BSFW also preserves the convergence properties of SFW. Then, we introduce in section 3 an homotopy embedding of the unconstrained BLASSO solvers to solve the constrained BLASSO ($\mathcal{P}_1$). We also show that this embedding stops in a finite number of iteration. Finally, we investigate the numerical behavior of the proposed methods on synthetic and real images in section 4.

This paper follows the preliminary work published in [11], in which the BSFW algorithm was briefly introduced, without studying in depth its properties.

Along this paper, we will search for Dirac masses within an image. Without distinction, these masses will also be referred as *atoms*, *spikes*, or *components*. In addition, we refer to the measure of interest with $\mu_{\boldsymbol{w},\boldsymbol{\theta}}$ when $\boldsymbol{w}$ and $\boldsymbol{\theta}$ are useful for the comprehension, and with μ otherwise. Superscript in bracket, as in $\mu^{[k]}$, will refer to the $k-$th iteration of a solver, while bracket indexes, as in $\mu_{[t]}$, will indicate the $t-$th iteration of an homotopy algorithm. Finally, N will generically refer to the number of spikes within μ.

2. Boosting the SFW BLASSO solver. After recalling the main elements of the SFW algorithm, we show how it can be accelerated while preserving its finite termination property.

2.1. The SFW solver. SFW [14] is a greedy solver for the unconstrained BLASSO problem ($\mathcal{P}_\lambda$). At each iterations, it adds a Dirac mass to an estimated measure μ, then fit $\boldsymbol{w}$ in order to quickly approach a minimizer of C, and then both $\boldsymbol{w}$ and $\boldsymbol{\theta}$ are optimized within a local descent in order to finely minimize C.

For a given measure $\mu \in \mathcal{M}(\mathcal{D})$, we can define a *certificate* that helps ensuring the solution is attainable. It is defined as:

$$\eta_\lambda(\mu) \stackrel{\text{def.}}{=} \frac{1}{\lambda}\boldsymbol{\Phi}^T(\mathbf{y} - \boldsymbol{\Phi}\mu). \tag{2.1}$$

Notably, the certificate is said to be *non-degenerate* when

$$\left| \begin{array}{l} \forall \mu \in \mathcal{M}(\mathcal{D}) \setminus \bigcup_{n=1}^{N} \{\delta_{\boldsymbol{\theta}_n}\},\ |\eta_\lambda(\mu_{\boldsymbol{w},\boldsymbol{\theta}})| < 1, \\ \forall n \in \{1, \dots, N\}, {\eta_\lambda}''(w_n \delta_{\boldsymbol{\theta}_n}) \neq 0. \end{array} \right. \tag{2.2}$$

This property ensures the stable recovery of the solution μ^* in low noise regime [17]. We refer the reader to [14] for the complete introduction of certificates relative to the SFW algorithm.

The certificate η_λ is unknown when handling real-world problems. Nevertheless, within a step of SFW, it can be approached based on the current estimation of μ, denoted $\mu^{[k]}$ for step k:

$$\eta^{[k]} \stackrel{\text{def.}}{=} \frac{1}{\lambda}\boldsymbol{\Phi}^\top \left(\mathbf{y} - \boldsymbol{\Phi}\mu^{[k-1]}\right) \tag{2.3}$$

Informally, it can be seen as the result of the convolution between the residual and the volume of an atom located at one point in $\mathcal{D}$. This forms the first step of SFW. Besides, the sequence $(\eta^{[k]})_{k\in\mathbb{N}}$ produced by SFW converges (in infinite norm) towards η_λ. Hence, testing if $\|\eta^{[k]}\|_\infty < 1$ forms a stopping condition of SFW[1]. When this is not the case, $\eta^{[k]}$ has low amplitudes where $\mu^{[k-1]}$ explains well the observed $\mathbf{y}$, and high amplitudes elsewhere. Thus, high absolute values locate where mass has to be appended in order to better explain $\mathbf{y}$, yielding the second step of SFW.

SFW is described in Algorithm 2.1, and Figure 2a depicts the steps encountered in a 2D toy example. More generally, details on this algorithm can be found in [14]. Notably, SFW is proven to find the solution of ($\mathcal{P}_\lambda$) in a finite number of step when η_λ is non-degenerate.

2.2. Boosting SFW. In addition to its interesting theoretical properties, in most cases SFW is efficient and retrieves, on synthetic data, the N atoms provided in exactly N steps. This has already been noted in [14] but has not been proven.

Investigating more closely the course of SFW, we observe that:

- most computation time is spent in the local descents of step 4, which handle all $k \times D$ parameters. Then, all parameters are again updated, so that for all iterations except the last one, the finely-optimized parameters are modified afterwards.

[1] This condition is in fact equivalent to $\nabla C(\mathbf{y}, \lambda, \mu^{[k]}) \leq 0$ on $\mathcal{M}(\mathcal{D})$, see *e.g.* [14, Remark 7] or [12, Appendix A].

Algorithm 2.1 Sliding Frank-Wolfe algorithm solving $(\mathcal{P}_\lambda)$[14]

Input: $\mathbf{y}$, PSF $\mathbf{H}$, λ
Output: $\mu_{\boldsymbol{w},\boldsymbol{\theta}}$, solution of $(\mathcal{P}_\lambda)$
Initialization: $\mu_{\boldsymbol{w}^{[0]},\boldsymbol{\theta}^{[0]}} = 0$ or an initial guess if available.
repeat(iteration k):
1. Compute $\max_{\boldsymbol{\theta}\in\mathcal{D}} \eta^{[k]}$ by local ascent using *e.g.* L-BFGS-B [9], starting from a maximum attained on a grid.
if $\max_{\boldsymbol{\theta}\in\mathcal{D}} \eta^{[k]} > 1$:
2. Expand the support: $\boldsymbol{\theta}^{[k]} = \boldsymbol{\theta}^{[k-1]} \cup \arg\max_{\boldsymbol{\theta}\in\mathcal{D}} \eta^{[k]}$, and let N_k be the current number of Dirac masses.
3. Adjust weights only (LASSO):
$$\tilde{\boldsymbol{w}}^{[k]} = \underset{\boldsymbol{w}\in\mathbb{R}_*^{N_k}}{\arg\min}\ C(\mathbf{y}, \lambda, \mu_{\boldsymbol{w},\boldsymbol{\theta}^{[k]}})$$
4. Local descent on all parameters using *e.g.* L-BFGS-B, starting at $\mu_{\tilde{\boldsymbol{w}}^{[k]},\boldsymbol{\theta}^{[k]}}$:
$$\boldsymbol{w}^{[k]}, \boldsymbol{\theta}^{[k]} = \underset{\boldsymbol{w}\in\mathbb{R}_*^{N_k}, \boldsymbol{\theta}\in\mathcal{D}^{N_k}}{\text{local descent of}}\ C(\mathbf{y}, \lambda, \mu_{\boldsymbol{w},\boldsymbol{\theta}})$$
5. Remove zero-weighted masses and update the measure:
$$\mu^{[k]} = \textstyle\sum_{n=1}^{N_k} w_n^{[k]} \delta_{\boldsymbol{\theta}_n^{[k]}}$$
else:
$\hat{\mu}^{[k-1]}$ is a solution. **End** of SFW

- this step often marginally decreases the objective criterion C from $(\mathcal{P}_\lambda)$ (this is the case, *e.g.* in Fig. 2a). We noted that this is partly due to the result of step 1: the local ascent maximizing $\eta^{[k]}$ yields an already relevant result, so the local descent of step 4 does not decrease much C.

Based on these observations, we propose a boosted version of the SFW algorithm, denoted BSFW, which removes most of the local descents. As in SFW, the certificate $\eta^{[k]}$ locates new atoms and indicates when the algorithm should stop. Then, at iteration k, either:

- $\max_{\boldsymbol{\theta}\in\mathcal{D}} \eta^{[k]} > 1$, and a new Dirac mass is added, then only $\boldsymbol{w}^{[k]}$ is adjusted,
- $\max_{\boldsymbol{\theta}\in\mathcal{D}} \eta^{[k]} < 1$, and a local descent is made to adjust $(\boldsymbol{w}^{[k]}, \boldsymbol{\theta}^{[k]})$. Afterwards, if we still have $\max_{\boldsymbol{\theta}\in\mathcal{D}} \eta^{[k]} < 1$ then BSFW stops, otherwise BSFW continues.

To summarize, BSFW performs local descent when needed, and not systematically, thanks to the insights given by $\max_{\boldsymbol{\theta}\in\mathcal{D}} \eta^{[k]}$. The procedure is described in Algorithm 2.2. and Figure 2b depicts the BSFW steps on a toy case.

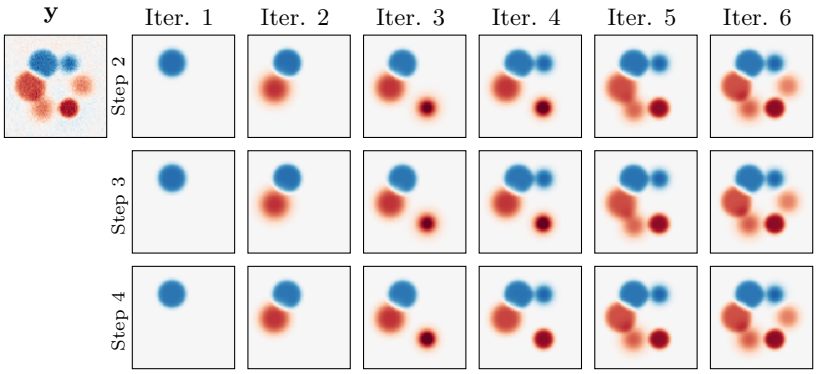
2.3. BSFW convergence and time complexity analysis.

Convergence analysis. The following proposition expands the finite termination property of SFW to the BSFW algorithm introduced in the previous section.

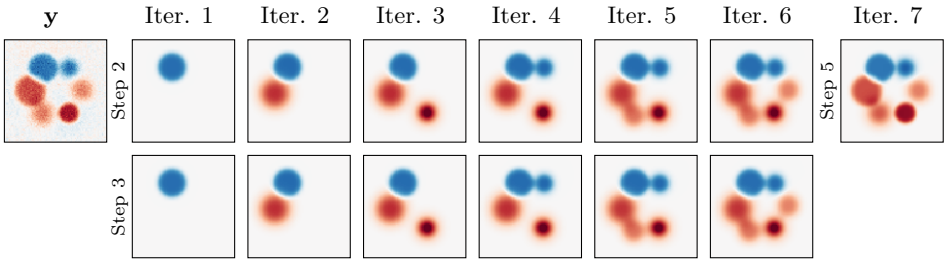
PROPOSITION 2.1 (Finite termination of BSFW). *Let μ^* be the unique solution of $(\mathcal{P}_\lambda)$. Assuming that η_λ is non-degenerate (2.2), then the BSFW algorithms recovers μ^* after a finite number of steps.*

The proof of this proposition is given in A, and is mostly based on the proof established for the SFW solver in [14, Sec. 4.2]. The main difference lies in the fact that the local descent of step 5 is not performed at all iterations, but at least once during BSFW.

Time complexity.. In order to study the time complexity of SFW and BSFW, let us denote K the number of atoms in a given image $\mathbf{y}$, $L = D + 1$ the number of parameter per atom, and S the size of $\mathbf{y}$. Let us assume that we are at an iteration



(a) SFW. In the first line, of the central block, a new atom is appended. In the second line, the weights $\boldsymbol{w}$ are adjusted, and in the third line, both $\boldsymbol{\theta}$ and $\boldsymbol{w}$ are adjusted.



(b) BSFW. In the first line of the central block, a new atom is appended. In the second line, the weights $\boldsymbol{w}$ are adjusted. The only local descent over $\boldsymbol{w}$ and $\boldsymbol{\theta}$ occurs In the last column.

Fig. 2: Depiction of SFW and BSFW on a 2D toy example, where each column represent an iteration of the algorithms. The positive and negative intensities are coded in red and blue respectively. In this example, BSFW obtains a result similar to SFW while avoiding all but one local descents.

containing, at its end, k atoms. All local descents are performed using L-BFGS-B [9], because the parameter to search for lie in the bounded domain $\mathcal{D}$. Table 1 summarizes the time complexities of the different steps involved an iteration of SFW and BSFW.

From this table, we can see that an iteration containing k atoms has the following complexity:

- $\mathcal{O}\left(k^2S(L+1) + 2kLS\log S\right)$ for SFW,
- $\mathcal{O}\left(k^2S + 2LS\log S\right)$ for an iteration of BSFW if $\max_{\boldsymbol{\theta}\in\mathcal{D}} \eta^{(k)} > 1$,
- $\mathcal{O}\left(k^2LS + 2kLS\log S + 2LS\log S\right)$ otherwise.

Given an image $\mathbf{y}$ containing K atoms, let us assume that K iterations are needed to find the solution, *i.e.* no Dirac mass was removed in the process. Then, summing from $k = 1$ to K yields, for SFW, the following best-case time complexity:

$$\mathcal{O}\left(\frac{K^3}{3}LS + K^2LS\log S\right). \tag{2.4}$$

For BSFW, depending on the values reached by $\|\eta^{[k]}\|_\infty$, two scenario can be considered. In the worst case, there is an alternance of steps 2–4 and 5–6 until the solution contains K atoms. Here, the time complexity of BSFW is the same as SFW. In the best case, BSFW go through steps 2–4 K times and through steps 5–6 only

Algorithm 2.2 Boosted Sliding Frank-Wolfe solving ($\mathcal{P}_\lambda$)

Input: $\mathbf{y}$, PSF $\mathbf{H}$, λ
Output: Estimated minimizer $\hat{\mu}_{\boldsymbol{w},\boldsymbol{\theta}}$ of ($\mathcal{P}_\lambda$)

repeat (iteration k):

1. Compute $\max_{\boldsymbol{\theta}\in\mathcal{D}} \eta^{[k]}$ by local ascent using *e.g.* L-BFGS-B, starting from a maximum attained on a grid.

if $\max_{\boldsymbol{\theta}\in\mathcal{D}} \eta^{[k]} > 1$:

2. Expand the support: $\boldsymbol{\theta}^{[k]} = \boldsymbol{\theta}^{[k-1]} \cup \arg\max_{\boldsymbol{\theta}\in\mathcal{D}} \eta^{[k]}$
3. Adjust weights only (LASSO):
$$\tilde{\boldsymbol{w}}^{[k]} = \underset{\boldsymbol{w}\in\mathbb{R}_*^k}{\arg\min}\ C(\mathbf{y}, \lambda, \mu_{\boldsymbol{w},\boldsymbol{\theta}^{[k]}})$$
4. Remove zero-weighted masses, update the measure:
$$\mu^{[k]} = \sum_{n=1}^{k} \tilde{w}_n^{[k]} \delta_{\boldsymbol{\theta}_n^{[k]}}$$

else:

5. Local descent on all parameters using *e.g.* L-BFGS-B, starting at $\mu_{\tilde{\boldsymbol{w}}^{[k]},\boldsymbol{\theta}^{[k]}}$:
$$\boldsymbol{w}^{[k]}, \boldsymbol{\theta}^{[k]} = \underset{\boldsymbol{w}\in\mathbb{R}_+^k, \boldsymbol{\theta}\in\mathcal{D}^k}{\text{local descent of}}\ C(\mathbf{y}, \lambda, \mu_{\boldsymbol{w},\boldsymbol{\theta}})$$
6. Remove zero-weighted masses, update the measure:
$$\mu^{[k]} = \sum_{n=1}^{k} w_n^{[k]} \delta_{\boldsymbol{\theta}_n^{[k]}}$$
7. Compute $\max_{\boldsymbol{\theta}\in\mathcal{D}} \eta^{[k+1]}$.

if $\max_{\boldsymbol{\theta}\in\mathcal{D}} \eta^{[k+1]} > 1$:

continue

else:

$\mu_{\boldsymbol{w}^{[k]},\boldsymbol{\theta}^{(k]}}$ is a solution. **End** of BSFW

Table 1: Time complexity of the operations involved in SFW and BSFW over k atoms.

Optimization algorithm	**Complexity of a function evaluation**	**Algorithm time complexity**
LASSO over k parameters.	$\mathcal{O}(S)$	$\mathcal{O}(k^2 S)$
L-BFGS-B over v variables.	$\mathcal{O}(S)$	$\mathcal{O}(vS)$
L-BFGS-B for estimating $\max_{\boldsymbol{\theta}\in\mathcal{D}} \eta^{[k]}$ over $L-1$ parameters.	Convolution (FFT and IFFT) and 2 elementary operations: $\mathcal{O}(2S\log S)$	$\mathcal{O}(2(L-1)S\log S)$
L-BFGS-B for local descent over $k \times L$ parameters.	Convolution (FFT and IFFT) and $k+1$ elementary operations: $\mathcal{O}(kS + 2S\log S)$	$\mathcal{O}(k^2 LS + \mathbf{2}kLS\log S)$

once, and no Dirac mass is removed in the process. Then the best-case time complexity for BSFW is:

$$\mathcal{O}\left(\frac{K^3}{3}S + 3KLS\log S\right). \tag{2.5}$$

Fig. 3 depicts the time complexity of BSFW and SFW. While the complexity order is the same (cubic in K for instance), BSFW offers room for reducing the time complexity of SFW without dropping its finite termination property.

3. Homotopy embedding of BLASSO solvers.

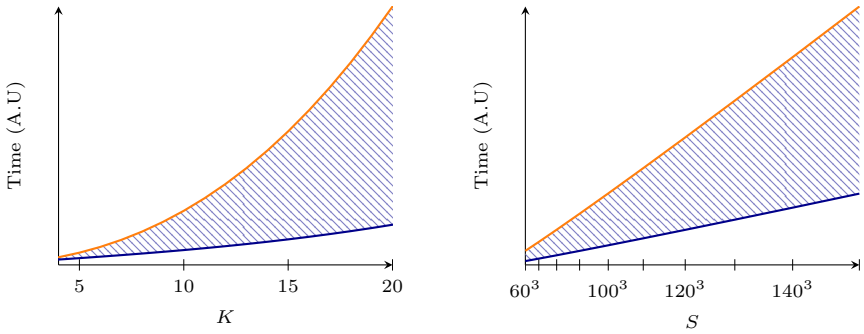


Fig. 3: Time complexity for a fixed $L = 6$, $S = 100^3$ and a varying K (left) and for a fixed $L = 6$, $K = 10$ (right). SFW is depicted in orange and BSFW in blue. The hatched region cover the interval between the worst and best case for BSFW.

3.1. Motivation. We have seen that SFW and BSFW provide a solution for the BLASSO problem stated in ($\mathcal{P}_\lambda$). However, this require to choose the regularization parameter λ before solving the problem. In practice, this choice is difficult to make, because there is no direct link between λ and properties of the solution. On the other hand, the solution of ($\mathcal{P}_1$) can be related to the residual of the image, which is easily interpretable.

In this section, we introduce an homotopy algorithm solving ($\mathcal{P}_1$) by embedding the BLASSO solvers seen in the previous section. The purpose is not to explore the full regularization path, but only its first segment (higher values of λ) until the ℓ_2 condition of ($\mathcal{P}_1$) is satisfied (see Fig. 1). This condition is equivalent to bounding the standard deviation of the residual, so we will work from this point of view which is more useful for practitioners.

The homotopy algorithms existing in the literature rely on the fact that the regularization path is piecewise linear (see *e.g.* [22]). These algorithms are stated over a discretized space, but this property holds for the space of measure. Indeed, let us consider λ_1 and λ_2, and μ_1, μ_2 solutions of ($\mathcal{P}_{\lambda_1}$), ($\mathcal{P}_{\lambda_2}$) respectively, such that $\eta_{\lambda_1}(\mu_1) = \eta_{\lambda_2}(\mu_2)$. Then we can see that $\forall a \in [0, 1]$, $a\mu_1 + (1 - a)\mu_2$ is a solution of $(\mathcal{P}_{a\lambda_1+(1-a)\lambda_2})$. So when two solutions μ_1 and μ_2 have the same certificate value for $\lambda_1 \neq \lambda_2$, the regularization path between λ_1 and λ_2 is a linear segment.

3.2. Homotopy for BLASSO solvers. The purpose of our approach is to solve ($\mathcal{P}_\lambda$) for a decreasing sequence of λ until the solution also solves ($\mathcal{P}_1$). Let $\{\lambda_0, \lambda_1, \ldots, \lambda_T\}$ be the decreasing sequence of regularization parameters, and $\{\hat{\mu}_{[0]}, \hat{\mu}_{[1]}, \ldots, \hat{\mu}_{[T]}\}$ the sequence of corresponding solutions provided by the BLASSO solvers (SFW or BSFW), such that $\hat{\mu}_{[T]}$ is also solution to ($\mathcal{P}_1$). When designing an homotopy algorithm, one must state what is the starting value of λ, how to account for the past knowledge, and also how λ will evolve.

Starting point.. λ_0 must be chosen carefully in order to start at a reasonably low value. By construction of SFW and BSFW, if for the initial value of $\|\eta^{[0]}\|_\infty < 1$, the algorithms stop. Because $\eta^{[0]} = \frac{1}{\lambda}\mathbf{\Phi}^T\mathbf{y}$, SFW and BSFW find at least one spike in the solution for $\lambda_0 < \|\mathbf{\Phi}^T\mathbf{y}\|_\infty$. So a trivial first solution is to set $\lambda_0 = \|\mathbf{\Phi}^T\mathbf{y}\|_\infty$ to have $\hat{\mu}_{[0]} = 0$. Note that a similar starting point is used in the discretized case [25, Remark 2.3][22].

Algorithm 3.1 Homotopy algorithm for the BLASSO

Input: $\mathbf{y}$, PSF $\mathbf{H}$, σ_{target}, $c > 0$
Output: Estimation $\hat{\mu}_{\boldsymbol{w},\boldsymbol{\theta}}$, solution of $(\mathcal{P}_1)$
Initialization: $\lambda_0 = \|\mathbf{\Phi}^T\mathbf{y}\|_\infty$ and $\mu_{[0]} = 0$.
repeat (iteration t)
1. Starting from $\hat{\mu}_{[t-1]}$, solve the BLASSO problem $(\mathcal{P}_{\lambda_t})$ to obtain $\hat{\mu}_{[t]}$, using either SFW (Alg. 2.1) or BSFW (Alg. 2.2).
2. Compute σ_t from the residual $\mathbf{y} - \mathbf{\Phi}\hat{\mu}_{[t]}$
if $\sigma_t < \sigma_{\text{target}}$: $\hat{\mu}_{[t]}$ is a solution. **End** of the algorithm.
else:
3. Compute $\max_{\boldsymbol{\theta}\in\mathcal{D}} \eta_{[t]}$ by local ascent.
4. Update $\lambda_{t+1} = \frac{\lambda_t \max \eta_{[t]}}{1+c}$

Continuation.. By construction, the solution to $(\mathcal{P}_{\lambda_{t+1}})$ contains at least as much atoms as the solution of $(\mathcal{P}_{\lambda_t})$. Hence, instead of solving $\mathcal{P}_{\lambda_{t+1}}$ starting from an empty $\mu^{[0]}_{[t+1]} = 0$, we use the previous estimation as a warm start. So the initialization of SFW and BSFW within the homotopy algorithm is,$\forall t > 0$

$$\mu^{[0]}_{[t+1]} = \hat{\mu}_{[t]} \tag{3.1}$$

This approach is referred as the *continuation* between steps [31, 19], and significantly speed up the numerical computations.

Jumping to the next step.. Besides, both BSFW and SFW are guided and stopped by successive computations of $\eta^{[k]}$ (2.3). Notably, when $\|\eta^{[k]}\|_\infty < 1$, the algorithms stop. If the resulting solution of $(\mathcal{P}_\lambda)$ does not solve $(\mathcal{P}_1)$, a lower value of λ has to be found. In order to ensure that at least one step of SFW or BSFW is made in a new homotopy iteration, it is necessary to have $\|\eta^{[k]}\|_\infty$ greater than one at the next step. Since $\mu^{(0)}_{t+1} = \hat{\mu}_{\lambda_t}$, it is sufficient to set $\forall t > 0$

$$\lambda_{t+1} = \frac{\lambda_t \|\eta^{[k]}_{[t]}\|_\infty}{1+c} \tag{3.2}$$

with c any positive real value. Using this approach guarantees that each homotopy step happens on a different linear segment of the regularization path.

To summarize, the homotopy algorithm for BLASSO requires the definition of some $c > 0$ and of the target standard deviation σ_{target}. The latter is easily available on practical problems, while the former rules the speed of the algorithm but not its output. In the following, we will set $c = 1$. In the remaining of the paper, we will refer as Homotopy-SFW (H-SFW) and Homotopy-BSFW (H-BSFW) for the respecting embedding of SFW and BSFW into Alg. 3.1.

3.3. Convergence and time complexity analysis.

PROPOSITION 3.1. *The homotopy method (Alg. 3.1) for the BLASSO problem has the finite termination property,* i.e. *there exists some $T \in \mathbb{N}$ satisfying $\sigma_T < \sigma_{\text{target}}$, such that $\hat{\mu}_T$ is a solution to $(\mathcal{P}_1)$.*

This proposition is proved in B, and relies on the fact that $\forall t$, $\|\mathbf{y} - \mathbf{\Phi}\mu_{[t]}\|_2^2 > \|\mathbf{y} - \mathbf{\Phi}\mu_{[t+1]}\|_2^2$.

Time complexity.. We are interested in the first T segments of the regularization path, not in its full exploration. Hence, the complexity is no longer exponential as could be expected from [22] in the discretized case. Let us consider an image $\mathbf{y}$ containing K atoms. We study two situations :

- worst case: each homotopy iteration adds exactly one atom to the solution, so the stopping condition is attained in at least $T = K$ homotopy iterations. Then each homotopy iteration corresponds to one SFW iteration.
- best case : the first homotopy iteration is sufficient to attain the stopping criterion, so $T = 1$. Hence SFW is run once but has K steps.

So in both case, H-SFW has the same complexity as SFW.

However, H-BSFW may be faster:

- in the worst case: one homotopy step corresponds to two BSFW step, hence the complexity is the same as in the SFW case
- in the best case: BSFW is run once over $K + 1$ steps. So the complexity of H-BSFW is the complexity of BSFW.

Note that as in the previous section, we did not account for the time complexity caused by the possible removal of zero-weighted Dirac masses. Indeed, it is impossible to know in advance if this step will happen, or not. In practice, it occurs seldomly, but helps reducing the dimension of the solution when necessary.

4. Numerical results. This section presents the experimental results obtained in the case of 3D deconvolution by SFW, BSFW and their homotopy embedding.

4.1. Experimental setting. The observation model is set so as to reflect a tomographic diffractive microscopy setup [28] under the Born approximation, which linearize the problem linking the optical index to the electric field. In this framework, several holograms are acquired under varying illumination conditions. Their combination in the Fourier space forms a synthetic aperture, which acts similarly to an Optical Transfer Function (OTF) [27]. The OTF, and the corresponding PSF, are depicted in Figure 4. We consider only transparent objects, *i.e.* there is no absorption and the real-valued voxel intensity is a measure of the refractive index (RI).

In order to capture smooth and sharp objects as well, we choose to use generalized isotropic Gaussians as atoms, yielding $\forall 1 \leq n \leq N$ and $\forall \boldsymbol{s} \in \mathbb{R}^3$:

$$\mathcal{G}(\boldsymbol{\theta}_n, w_n; \boldsymbol{s}) = w_n \exp\left(-\frac{1}{2\sigma_n^{d_n}} \|\boldsymbol{m}_n - \boldsymbol{s}\|_2^{d_n}\right) \tag{4.1}$$

so that $\boldsymbol{\theta}_n = \{\boldsymbol{m}_n, \sigma_n, d_n\} \in \mathcal{D} \subset \mathbb{R}^3 \times \mathbb{R}_{+*}^2$.

Indeed, the d_n parameter allows to fit object with different intensity gradient : a greater d yields a sharper object than an usual Gaussian.

Implementation. Several implementation points have been leveraged to improve the speed of both SFW and BSFW:

- convolutions are made in the Fourier domain,
- ∇C is computed analytically, so as to avoid a costly numerical approximation. For completeness, its analytical form is reported in C.
- the computations of C and ∇C are parallelized across atoms,
- a lookup table for convolution in $\mathcal{D}$ is computed once in order to accelerate the grid initialization for the local ascent of $\eta^{[k]}$ (2.3).

Alternative. In the context of image reconstruction, a popular criterion is based on regularization by the total variation (TV) between pixels [5]. Its unconstrained

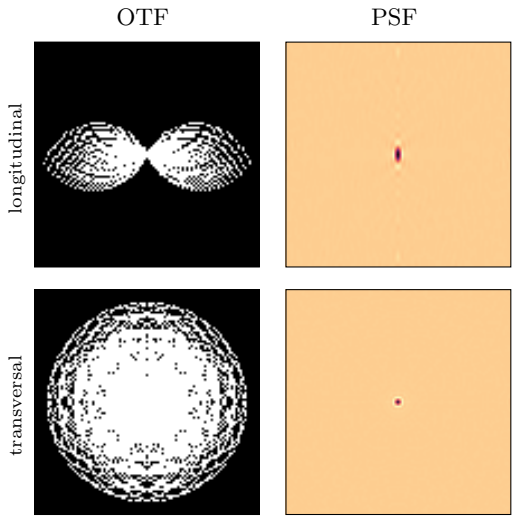


Fig. 4: Slices of the OTF (left) and PSF (right) of the considered observation model. The first line depicts slices along the optical axis (vertical axis) and the second line depicts slices in the perpendicular plane. The PSF and OTF are axisymmetric, yielding a "missing cone" in the Fourier space along the said axis.

formulation is:

$$(\mathcal{P}_{\mathsf{TV}}) \qquad \min_{\mathbf{x}\in\mathbb{R}^S} \mathsf{TV}(\mathbf{x}) \text{ subject to } \|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2 \leq e,$$

and it constrained counterpart is

$$(\mathcal{P}_{\lambda\mathsf{TV}}) \qquad \min_{\mathbf{x}\in\mathbb{R}^S} \frac{1}{2}\|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2 + \lambda\mathsf{TV}(\mathbf{x}),$$

where $\mathbf{x}$ is the noiseless image prior to convolution. We choose to use order-1 TV regularization, so that $\mathsf{TV}(\mathbf{x}) = \sum_{i\in\mathcal{S}} \left(\sum_{j\in\mathcal{N}(i)} |x_i - x_j|\right)$ where for any voxel i, $\mathcal{N}(i)$ represents its neighboring voxels. To solve ($\mathcal{P}_{\lambda\mathsf{TV}}$), we use a solver based on the accelerated proximal gradient (APG), which is described in D. Using this approach as a counterpart of SFW and BSFW, we can compare its output $\hat{\mathbf{x}}$ with the estimations $\mathbf{\Phi}\hat{\mu}_{\mathrm{SFW}}$ and $\mathbf{\Phi}\hat{\mu}_{\mathrm{BSFW}}$. Besides, the TV solver can be also used within a modified homotopy algorithm denoted H-TV, which is reported in D.

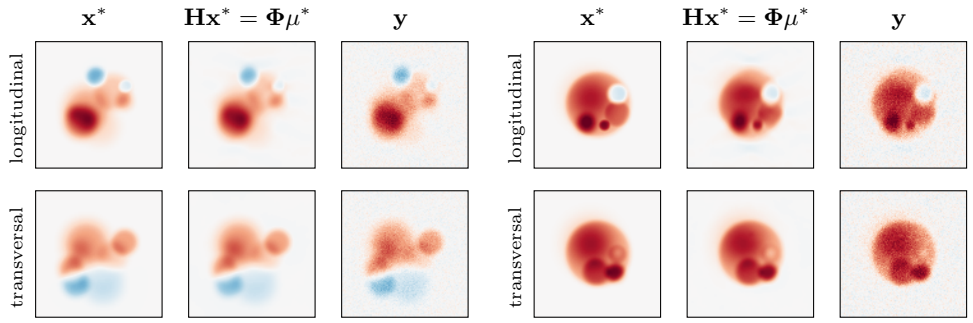
In the following, we study two synthetic cases:

A. A 3D Gaussian mixtures with randomly-chosen parameters (see Fig. 5a). In this case, we will vary the number of atoms K and the number of voxels S.

B. A 3D phantom cell, with a large spherical component (analog to the cytoplasm) containing several smaller components (see Fig. 5b). On a real image, these small components could be the cell core, mitochondria, or vacuoles (for lower index). We keep 10 small components and vary S in the experiments. This phantom cell model is inspired by [20, 21].

Note that, as in real applications, there is no prior knowledge on the real η and in particular, if it satisfies the non-degeneracy condition (2.2) or not.

In this experimental section, we are interested in two quantities :



(a) Case A: mixture of generalized Gaussians.

(b) Case B: mixture of 10 generalized Gaussians over a central atom whose parameters are fixed for all simulations.

Fig. 5: Examples of the two synthetic cases considered for the numerical study, seen from their central transversal or longitudinal (along the optical axis) slices. Red (resp. blue) indicates positive (resp. negative) values. In both cases, the values of $\boldsymbol{\theta}$ are randomly chosen.

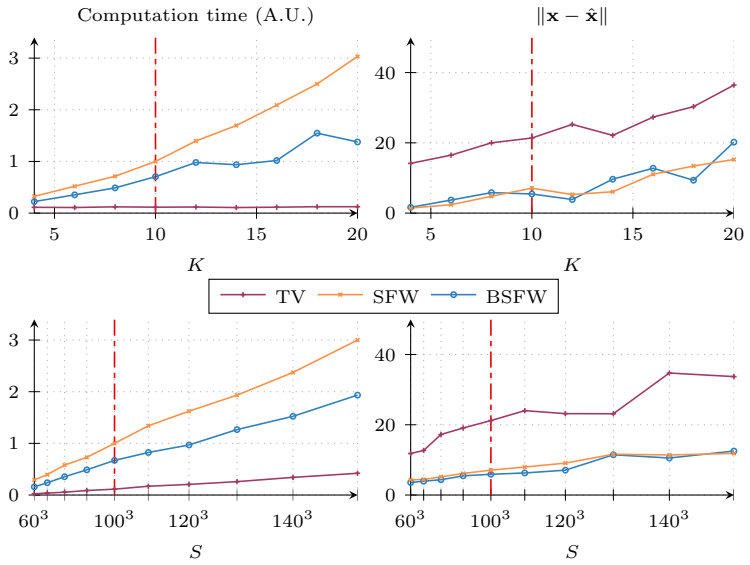
- How far is the solution to the original image $\mathbf{x}^*$. It is evaluated trough the imaging operator $\mathbf{\Phi}$, so that the quantity of interest is $\|\mathbf{\Phi}\hat{\mu} - \mathbf{x}^*\|_2$ when applying a BLASSO solver and $\|\hat{\mathbf{x}}^{\mathsf{TV}} - \mathbf{x}^*\|_2$ when applying the TV solver.
- The computation time. Since it is implementation dependent, we will use the computation time of SFW (or H-SFW) with $S = 100^3$ voxels and $K = 10$ as a reference.

4.2. Solving the unconstrained BLASSO problem. In this section, we investigate the numerical behavior of the three studied solvers, apart from their homotopy embedding. This allows to distinguish what is due to each solver from what is due to their interaction with the homotopy method. So in this section, we focus again on solving ($\mathcal{P}_\lambda$) and ($\mathcal{P}_{\lambda\mathsf{TV}}$) respectively. Again, choosing λ has a critical influence on the result. Hence, in order to avoid tuning λ, for each experiment its value was taken from the output of the homotopy method, for which results are reported in the next section.

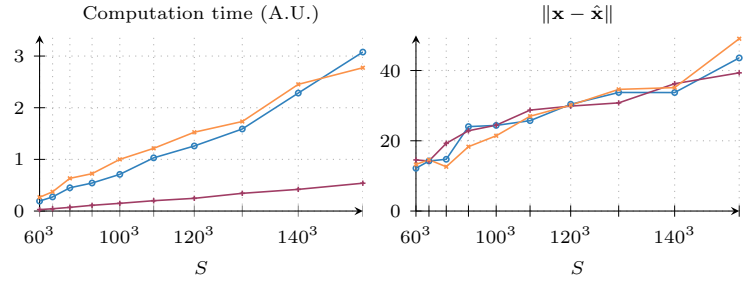
The numerical results of the TV, SFW and BSFW solvers are reported in Fig. 6:

- The main observation from case A is that BSFW yields similar results than SFW, while providing notably reduced computation time: on average BSFW is 30% to 38% faster than SFW.
- The computation time are mostly linear as a function of K and S over the investigated ranges. This observation only indicates that over these range, the linear terms overcome the higher-degree terms seen in (2.4) and (2.5). This is expected since the linear term originates from the L-BFGS-B algorithm, which is the computational bottleneck of the methods.
- The results for case B are quite different from case A: SFW and BSFW results and speed are similar. This result is somewhat counter-intuitive, and the next section will provide additional insights on this point.

As a partial conclusion, we see that depending on the complexity of the observation, the use of SFW or BSFW at a fixed value of λ may not be fully satisfying.



(a) Numerical results for case A (Fig. 5a). The red vertical line ($S = 100^3$, $K = 10$) is shared between the first and the second line.



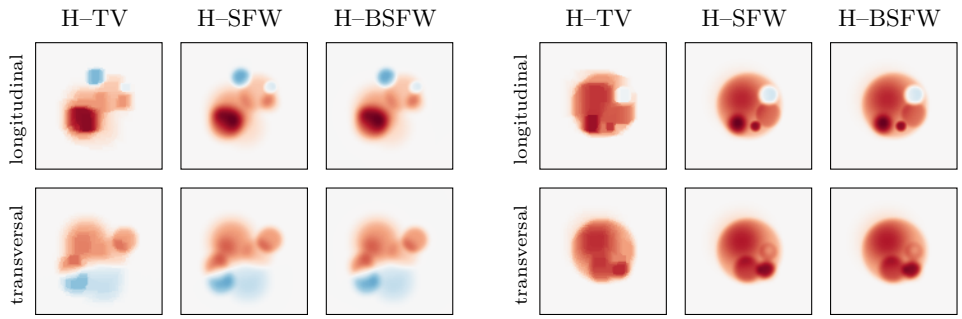
(b) Numerical results for case B (Fig. 5b), with the same legend as case A.

Fig. 6: Error rate for the three solver evaluated in this study. Each point is averaged over 10 random values of $\boldsymbol{\theta}$.

4.3. Homotopy solving of the constrained BLASSO. In this section, we investigate the constrained reconstruction problem as stated in ($\mathcal{P}_1$) and ($\mathcal{P}_{\mathsf{TV}}$) by the homotopy method in Alg. 3.1. We especially focus on the interaction between homotopy and the TV, SFW and BSFW solvers whose results are reported in the previous section. Here, we set $\sigma_{\text{target}} = 1.05\sigma$ where σ is the known noise standard deviation.

An instance of results obtained by the homotopy algorithm is given in Fig. 7, and the complete results are reported in Fig. 8:

- The H-TV method is always the fastest, but provide poor reconstruction results in all cases.
- Unlike H-TV, the H-SFW solver provides the best results but is also the most time consuming.
- The H-BSFW solver yields, on average, results at least as good as its SFW



(a) Case A. (b) Case B.

Fig. 7: Example results obtained on the images from Fig. 5, using the same colormap. We observe that the TV regularization enforces some intensity and morphological flattening, so the solver hardly cope with smooth intensity and spherical shapes. Note also that the flattening of TV follows the voxel grid because of the structure of the considered local neighborhood.

counterpart, but the computation time is notably reduced.

- The homotopy embedding makes noticeable the difference between the SFW and BSFW approaches, in the two considered cases. Let us recall that the results from figure 6 are obtained with the value of λ found by homotopy. It follows that SFW and BSFW are sensible to their initialization, and that the homotopy embedding succeeds in providing the best one.
- Besides, by comparing figure 6 and 8, we also observe that the homotopy algorithms are a bit slower than their embedded solvers, but the computation times remain similar.

Thus, the homotopy algorithm allows a finer exploration of the solution path, while having moderate additional computational cost with respect to the original solver.

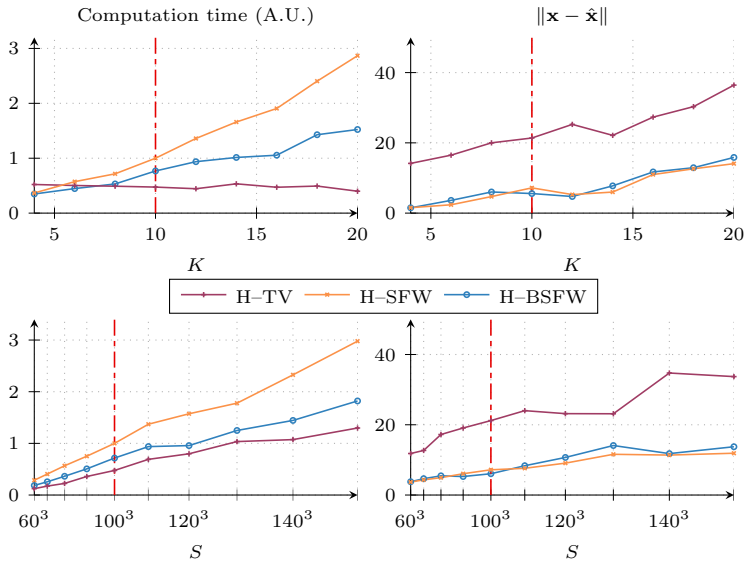
In other words, we provided a way to solve the constrained BLASSO problem ($\mathcal{P}_1$) at a similar speed than its unconstrained counterpart ($\mathcal{P}_\lambda$), while providing a better solution and setting a constraint on the result.

4.4. Application on real images. In this section, we present the results obtained on images from a real TDM setup [28]. The observation system is more intricated than the observation model (1.2): the image is formed in presence of a non-zero background, the holograms contain some aberrations, and the Born hypothesis remains approximate for thick samples (a few μm). We study two cases:

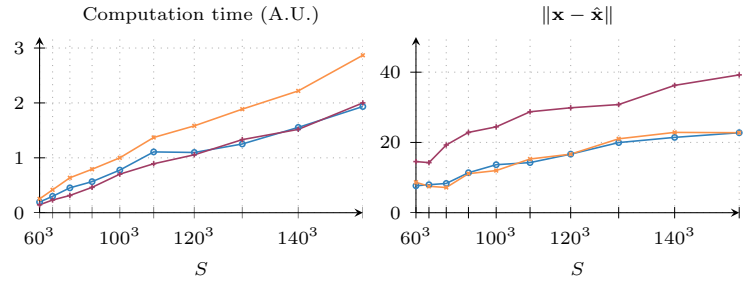
C. the observation of a polymer bead whose size and refractive index are known. Such observation is useful to verify the estimations of reconstruction algorithms. Noteworthy, the reconstruction suffers from some imperfections that make the apparent refractive index vary along the optical axis, which should not be the case of the physical bead.

D. the image of a snowdrop pollen (*Galanthus nivalis*), whose physical properties are unknown. The pollen has an ovoid shape and is slightly flattened long the z axis, because of the transparent slides used to prepare the sample.

The estimated standard deviation, $\hat{\sigma}$, is computed over an empty region of the volume. Then, we set $\sigma_{\text{target}} = 1.5\hat{\sigma}$. The volumes, as well as the corresponding results, are



(a) Numerical results for case A , with the same legend as Fig. 6a.



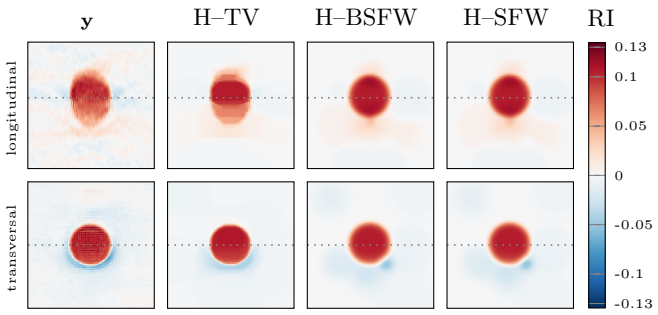
(b) Numerical results for case B.

Fig. 8: Numerical results for the homotopy embedding of the three studied solvers. The legend is the same as in Fig. 6.

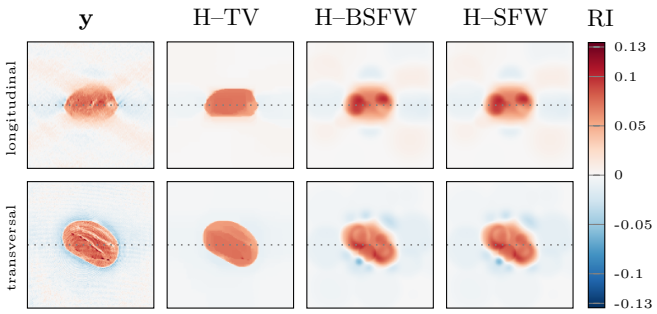
reported in Fig. 9:

- For case C, the reconstruction using TV is efficient in removing the noise, but is limited by the reconstruction artifacts, making the shape blurry along the optical axis. Instead, the reconstructions with H-SFW and H-BSFW do separate the central bead from its neighborhood, which is depicted by a few lower-intensity atoms.
- Regarding case C, the H-TV algorithm is able to retrieve the outer morphology of the pollen, but erases all its internal gradient. Instead, regarding H-SFW and H-BSFW, the sharp edges are less captures, but the internal refractive index variations are fitted. In particular, the pollen has an outer shell that is visible in $\mathbf{y}$ from texture changes. This shell seems well-captured by H-SFW and H-BSFW.

5. Discussion. In this paper, we studied the problem of gridless sparse recovery. We found that the recent SFW solver could be notably accelerated, while preserving



(a) Case C: polymer bead.



(b) Case D: snowdrop pollen.

Fig. 9: Images from the reconstruction under the Born hypothesis (left columns) and the corresponding results using the homotopy algorithms. The grey dotted lines locate the different slices, *i.e.* they are common between the transversal and longitudinal slices. Both volumes contain $S = 150^3$ voxels, with a voxel edge length of 165nm.

its finite termination property. Furthermore, we showed that these solvers could be embedded within an homotopy algorithm, allowing thus to solve the constrained BLASSO problem. Numerically, the methods were applied in the case of 3D deconvolution, showing that the BSFW does accelerate SFW, and that the homotopy embedding is also effecient to accurately parse the solution path.

Future works will focus on the use of other atomic shapes, such as splines, and on the generalization of the measure model beyond the sum of Dirac masses.

Acknowledgment. The author are very grateful to Nicolas Verrier and Matthieu Debailleul (IRIMAS, UR 7499) for providing the TDM observations and the reconstruction pipeline.

This work was partially funded by the French National Research Agency with the grants HORUS ANR-18-CE45-0010 and THTTM ANR-19-CE42-0004.

REFERENCES

[1] M. S. Asif and J. Romberg, *Sparse recovery of streaming signals using l1-homotopy*, IEEE Transactions on Signal Processing, 62 (2014), pp. 4209–4223.

[2] J.-M. Azais, Y. De Castro, and F. Gamboa, *Spike detection from inaccurate samplings*, Applied and Computational Harmonic Analysis, 38 (2015), pp. 177–195.

[3] H. P. Babcock, J. R. Moffitt, Y. Cao, and X. Zhuang, *Fast compressed sensing analysis for*

super-resolution imaging using l1-homotopy, Optics express, 21 (2013), pp. 28583–28596.
[4] A. Barbero and S. Sra, *Modular proximal optimization for multidimensional total-variation regularization*, The Journal of Machine Learning Research, 19 (2018), pp. 2232–2313.
[5] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM journal on imaging sciences, 2 (2009), pp. 183–202.
[6] A. Beck and M. Teboulle, *Gradient-based algorithms with applications to signal recovery*, Convex optimization in signal processing and communications, (2009), pp. 42–88.
[7] N. Boyd, G. Schiebinger, and B. Recht, *The alternating descent conditional gradient method for sparse inverse problems*, SIAM Journal on Optimization, 27 (2017), pp. 616–639.
[8] K. Bredies and H. K. Pikkarainen, *Inverse problems in spaces of measures*, ESAIM: Control, Optimisation and Calculus of Variations, 19 (2013), pp. 190–218.
[9] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, *A limited memory algorithm for bound constrained optimization*, SIAM Journal on Scientific Computing, 16 (1995), pp. 1190–1208.
[10] E. J. Candès and C. Fernandez-Granda, *Towards a mathematical theory of super-resolution*, Communications on Pure and Applied Mathematics, 67 (2014), pp. 906–956.
[11] J.-B. Courbot and B. Colicchio, *Boosting the Sliding Frank-Wolfe solver for 3D deconvolution*, in iTWIST 2020: international-Traveling Workshop on Interactions between Sparse models and Technology, 2020.
[12] J.-B. Courbot, V. Duval, and B. Legras, *Sparse analysis for mesoscale convective systems tracking*, Signal Processing: Image Communication, (2020), p. 115854.
[13] Y. De Castro and F. Gamboa, *Exact reconstruction using Beurling minimal extrapolation*, Journal of Mathematical Analysis and applications, 395 (2012), pp. 336–354.
[14] Q. Denoyelle, V. Duval, G. Peyré, and E. Soubies, *The sliding Frank–Wolfe algorithm and its application to super-resolution microscopy*, Inverse Problems, 36 (2019), p. 014001.
[15] D. L. Donoho and Y. Tsaig, *Fast solution of l1-norm minimization problems when the solution may be sparse*, IEEE Transactions on Information Theory, 54 (2008), pp. 4789–4812.
[16] C. Dünner, S. Forte, M. Takác, and M. Jaggi, *Primal-dual rates and certificates*, in Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, 2016, pp. 783–792.
[17] V. Duval and G. Peyré, *Exact support recovery for sparse spikes deconvolution*, Foundations of Computational Mathematics, 15 (2015), pp. 1315–1355.
[18] C. Ekanadham, D. Tranchina, and E. P. Simoncelli, *Recovery of sparse translation-invariant signals with continuous basis pursuit*, IEEE transactions on signal processing, 59 (2011), pp. 4735–4744.
[19] E. T. Hale, W. Yin, and Y. Zhang, *A fixed-point continuation method for l1-regularized minimization with applications to compressed sensing*, CAAM TR07-07, Rice University, 43 (2007), p. 44.
[20] W. Krauze, P. Makowski, M. Kujawińska, and A. Kuś, *Generalized total variation iterative constraint strategy in limited angle optical diffraction tomography*, Optics Express, 24 (2016), pp. 4924–4936.
[21] J. Lim, A. B. Ayoub, E. E. Antoine, and D. Psaltis, *High-fidelity optical diffraction tomography of multiple scattering samples*, Light: Science & Applications, 8 (2019), pp. 1–12.
[22] J. Mairal and B. Yu, *Complexity analysis of the lasso regularization path*, in Proceedings of the 29th International Conference on Machine Learning (ICML 2012), 2012.
[23] J. Nocedal and S. Wright, *Numerical optimization*, Springer Science & Business Media, 2006.
[24] M. R. Osborne, B. Presnell, and B. A. Turlach, *A new approach to variable selection in least squares problems*, IMA journal of numerical analysis, 20 (2000), pp. 389–403.
[25] M. R. Osborne, B. Presnell, and B. A. Turlach, *On the lasso and its dual*, Journal of Computational and Graphical statistics, 9 (2000), pp. 319–337.
[26] N. Parikh and S. Boyd, *Proximal algorithms*, Foundations and Trends® in Optimization, 1 (2014), pp. 127–239.
[27] C. Park, S. Shin, and Y. Park, *Generalized quantification of three-dimensional resolution in optical diffraction tomography using the projection of maximal spatial bandwidths*, JOSA A, 35 (2018), pp. 1891–1898.
[28] B. Simon and O. Haeberlé, *Tomographic diffractive microscopy: Principles, implementations, and applications in biology*, in Label-Free Super-Resolution Microscopy, Springer, 2019, pp. 85–112.
[29] C. Soussen, J. Idier, J. Duan, and D. Brie, *Homotopy based algorithms for l0-regularized least-squares*, IEEE Transactions on Signal Processing, 63 (2015), pp. 3301–3316.
[30] G. Tang, B. N. Bhaskar, P. Shah, and B. Recht, *Compressed sensing off the grid*, IEEE transactions on information theory, 59 (2013), pp. 7465–7490.
[31] E. Van Den Berg and M. P. Friedlander, *Probing the Pareto frontier for basis pursuit*

solutions, SIAM Journal on Scientific Computing, 31 (2009), pp. 890–912.
[32] L. Vandenberghe, *Fast proximal gradient methods*, EE236C course notes, Online, http://www. seas. ucla. edu/vandenbe C, 236 (2010).
[33] L. Xiao and T. Zhang, *A proximal-gradient homotopy method for the sparse least-squares problem*, SIAM Journal on Optimization, 23 (2013), pp. 1062–1091.
[34] L. Zhang, T. Yang, R. Jin, and Z.-H. Zhou, *A simple homotopy algorithm for compressive sensing*, in Artificial Intelligence and Statistics, 2015, pp. 1116–1124.
[35] Y.-B. Zhao, *Sparse optimization theory and methods*, CRC Press, 2018.

Appendix A. Finite termination of BSFW.

In this appendix, we prove the finite termination property of BSFW. The proof is reported here for completeness, but follow mostly the one given in [14].

Let $(\mu^{[k]})_{k\in\mathbb{N}}$ be the sequence of estimations made by BSFW, μ^* be the unique solution of $(\mathcal{P}_\lambda)$ and N be the number of atoms in μ^*.

Because steps 3 (weight fit) and 5 (local descent on all parameters) both decrease C_λ, we have $\forall k > 0$:

$$C(\mathbf{y}, \lambda, \mu^{[k]}_{\boldsymbol{w},\boldsymbol{\theta}}) \leq C(\mathbf{y}, \lambda, \tilde{\mu}^{[k]}) \leq C(\mathbf{y}, \lambda, \mu^{[k]}) \leq C(\mathbf{y}, \lambda, \mu^{[k-1]}) \tag{A.1}$$

where $\tilde{\mu}^{[k]}$ is the result of step 2 (support expansion) at iteration k. As noted in [14], $\tilde{\mu}^{(k)}$ is the result of a standard Frank-Wolfe algorithm, so we benefit from the same convergence [23]: there exists $C_1 > 0$ such that:

$$\forall k > 0, C_\lambda(\mu^{[k]}) - C_\lambda(\mu^*) \leq \frac{C_1}{k} \tag{A.2}$$

Hence, $(\mu^{[k]})_{k\in\mathbb{N}}$ is a bounded minimizing sequence. One can extract from it a subsequence that converges towards some $\mu \in \mathcal{M}(\mathcal{D})$ for the weak-* topology. Since C_λ is convex and lower semi-continuous (l.s.c.), it is also weak-* l.s.c. so that $C_\lambda(\mu) = C_\lambda(\mu^*)$, making μ a solution of $(\mathcal{P}_\lambda)$. Hence, if $\mu^* \in \mathcal{M}(\mathcal{D})$ is the unique solution of $(\mathcal{P}_\lambda)$, then $(\mu^{[k]})_{k\in\mathbb{N}}$ weak-* converges towards μ^*.

Such convergence can also be established for the certificate η and its derivatives. From [14], we obtain that because $\boldsymbol{\Phi}$ is weak-* to weak continuous and $\boldsymbol{\Phi}^T$ is a compact operator, $\forall j \in \{0, 1, 2\}$:

$$(\eta^{[k]})^{(j)} \underset{k\to+\infty}{\overset{\|\cdot\|_\infty, \mathcal{M}(\mathcal{D})}{\longrightarrow}} \eta^{(j)}_\lambda \tag{A.3}$$

Let us denote $\mu_i = \boldsymbol{w}_i \delta_{\boldsymbol{\theta}_i}$ for the $i-$th spike in μ^*. As in [14], the proof of finite termination is made in three steps.

First step. Because η_λ is non-degenerate, and because of (A.3), there exists $e > 0$ and $k_1 \in \mathbb{N}$ such that:

$$\forall k \geq k_1, \forall i \in \{1, \dots, N\}, \forall \mu \in I_{\mu_i, e}, \eta^{(k)\prime\prime}(\mu) \neq 0 \tag{A.4}$$

where $I_{\mu_i,e} =]\mu_i - e, \mu_i + e[$.

Second step. Since $\mu^{[k]}$ converges towards μ^* in the weak-* topology, and $|\mu^*|$ does not charge the boundary of $I_{\mu_i,e}$ we have:

$$\forall i \in \{1, \dots, N\}, \mu^{[k]}(I_{\mu_i,e}) \underset{k\to+\infty}{\to} \mu^*(I_{\mu_i,e}) = w_i \neq 0 \tag{A.5}$$

Hence, there exists $k_2 \in \mathbb{N}$ such that $\forall k \geq k_2$, $\mu^{[k]}$ contains at least one spike in each $I_{\mu_i,e}$. Notably, $\mu^{[k]}$ contains at least N spikes.

Third step. Because of (A.3), there exists some k_3 such that $\|\eta^{[k_3]}\|_\infty < 1$, so step 5 (local descent on all parameters) happens. In addition, from (A.3), there exists k_4 such that

$$\mathrm{Sat}^{\pm}(\eta^{[k]}) \subset \mathrm{Sat}^{\pm}(\eta_\lambda) \oplus (]-e, e[\times\{0\}) \tag{A.6}$$

With the set of saturation point of a given $\eta \in \mathcal{C}_0(\mathcal{M}(\mathcal{D}), \mathbb{R})$ defined as $\mathrm{Sat}^{\pm}(\eta) = \{(x, v) \in \mathcal{M}(\mathcal{D}) \times \{-1, 1\}; \eta(x) = v\}$. In addition:

$$\forall \mu \in \mathcal{M}(\mathcal{D}) \setminus \bigcup_{i=1}^{N} I_{\mu_i, e},\ |\eta^{[k]}(\mu)| < 1 \tag{A.7}$$

In particular, for $k \geq k_3$ and $k \geq k_4$, $\mu^{[k]}$ has no spikes in $\mathcal{M}(\mathcal{D}) \setminus \bigcup_{i=1}^{N} I_{\mu_i,e}$ because it would contradict the optimality of step 5.
Gathering the three steps, let us assume that $k \geq \max(k_1, k_2, k_3, k_4)$. Then:

- $\eta^{[k]\prime\prime}(x) \neq 0$ inside $\bigcup_{i=1}^{N} I_{\mu_i,e}$
- $\mu^{[k]}$ has at least one spike in each $I_{\mu_i,e}$.
- step 2b happens, so $\mu^{[k]}$ has no spikes outside of $\bigcup_{i=1}^{N} I_{\mu_i,e}$

Let $\mu_j^{[k]}$ be the position of the $j-$th spike in $\mu^{[k]}$. From step 5 (local descent on all parameters) we have that $\eta^{(k)\prime}(\mu_j^{[k]}) = 0$. Because $\eta^{(k)\prime\prime} \neq 0$ in $I_{\mu_i,e}$, this implies that $|\eta^{(k)}| < 1$ except in $\mu_j^{[k]}$. So $\mu^{[k]}$ has exactly one spike in $I_{\mu_i,e}$, and:

$$\forall \mu \in \mathcal{M}(\mathcal{D}) \setminus \bigcup_{i=1}^{N} \mu_j^{[k]},\ |\eta^{(k)}(\mu)| < 1 \tag{A.8}$$

Hence $\mu^{[k]}$, containing N spikes, is a solution of $(\mathcal{P}_\lambda)$. Since μ^* is assumed to be the unique solution of $(\mathcal{P}_\lambda)$, $\mu^{[k]} = \mu^*$: the BSFW algorithms recovers μ^* in a finite number of steps.

Appendix B. Finite termination of the BLASSO homotopy algorithm.

In this second appendix, we show that the homotopy algorithm (Alg. 3.1) stops in a finite number of steps. To do so, we show that $\left\{\|y - \mathbf{\Phi}\mu_{[t]}\|\right\}_{t\in\mathbb{N}}$ is a strictly decreasing sequence, ensuring that there exists t_1 such that $\forall t \geq t_1$, $\|y - \mathbf{\Phi}\mu_{[t]}\| < e$ (or equivalently that $\sqrt{\|y - \mathbf{\Phi}\mu_{[t]}\|/S} \leq \sigma_{\text{target}}$) so that Alg. 3.1 stops at t_1.

Because $\lambda_t > \lambda_{t+1}$, we have $\forall \mu \in \mathcal{M}(\mathcal{D})$:

$$C(\mu, \lambda_t) = \frac{1}{2}\|y - \mathbf{\Phi}\mu\|^2 + \lambda_t|\mu| > \frac{1}{2}\|y - \mathbf{\Phi}\mu\|^2 + \lambda_{t+1}|\mu| = C(\mu, \lambda_{t+1}) \tag{B.1}$$

In particular for $\mu = \mu_{[t]}$:

$$C(\mu_{[t]}, \lambda_t) > C(\mu_{[t]}, \lambda_{t+1}) > C(\mu_{[t+1]}, \lambda_{t+1}) \tag{B.2}$$

because $\mu_{[t+1]}$ is a minimizer of $C(\cdot, \lambda_{t+1})$.

From (B.2) we have:

$$\begin{aligned} \frac{1}{2}\|y - \mathbf{\Phi}\mu_{[t]}\|^2 + \lambda_t|\mu_{[t]}| &> \frac{1}{2}\|y - \mathbf{\Phi}\mu_{[t+1]}\|^2 + \lambda_{t+1}|\mu_{[t+1]}| \\ \|y - \mathbf{\Phi}\mu_{[t]}\|^2 - \|y - \mathbf{\Phi}\mu_{[t+1]}\|^2 &> 2\lambda_{t+1}|\mu_{[t+1]}| - 2\lambda_t|\mu_{[t]}| \end{aligned} \tag{B.3}$$

Let us denote:

$$\mu^*(\lambda) = \arg\min \frac{1}{2}\|y - \mathbf{\Phi}\mu\|^2 + \lambda|\mu| \tag{B.4}$$

Then, we have:

$$\left.\frac{\partial C(\mu,\lambda)}{\partial \mu}\right|_{\mu=\mu^*(\lambda)} = \mathbf{\Phi}^T(\mathbf{\Phi}\mu^* - \mathbf{y}) + \lambda = 0 \tag{B.5}$$

Deriving with respect to λ yields:

$$\begin{aligned} \frac{\partial}{\partial\lambda}\left(\left.\frac{\partial C(\mu,\lambda)}{\partial\mu}\right|_{\mu=\mu^*(\lambda)}\right) + \frac{\partial\mu^*(\lambda)}{\partial\lambda}\,\frac{\partial}{\partial\lambda}\left.\frac{\partial^2 C(\mu,\lambda)}{\partial^2\mu}\right|_{\mu=\mu^*(\lambda)} &= 0 \\ 1 + \mathbf{\Phi}^T\mathbf{\Phi}\frac{\partial\mu^*(\lambda)}{\partial\lambda} &= 0 \end{aligned} \tag{B.6}$$

Because $\mathbf{\Phi}^T\mathbf{\Phi} \neq 0$, we have that $\frac{\partial\mu^*(\lambda)}{\partial\lambda} < 0$. Hence, $\lambda_t > \lambda_{t+1}$ implies that $|\mu_{[t]}| < |\mu_{[t+1]}|$.

Going back to (B.3), we showed that $\|y - \mathbf{\Phi}\mu_{[t]}\|^2 - \|y - \mathbf{\Phi}\mu_{[t+1]}\|^2 > 0$, hence $\sigma_{t+1} < \sigma_t\ \forall t$. Hence, the $\{\sigma_t\}_{t\in\mathbb{N}}$ form a strictly decreasing sequence, and so there exists some $T \in \mathbb{N}$ such that $\sigma_T < \sigma_{\text{target}}$: the homotopy algorithm stops in a finite number of iterations.

Appendix C. Gradient of C.

In this appendix, we report the analytical form of the gradient of C from ($\mathcal{P}_\lambda$), given that atoms are generalized 3D Gaussians (4.1). Let us recall that C is parametrized by $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n, \ldots, \boldsymbol{\theta}_K\}$ and that each $\boldsymbol{\theta}_n$ rules a different atom. Let us denote $m_{n,i}$ one of the three scalars forming the position vector $\boldsymbol{m}_n \in \mathbb{R}^3$ and s_i one of the three scalars of $\boldsymbol{s}$, which locates a voxel in $\mathbf{y}$. For each $\boldsymbol{\theta}_n \in \boldsymbol{\theta}$ and for each parameter $\theta_{n,i} \in \boldsymbol{\theta}_n = \{m_{n,1}, m_{n,2}, m_{n,3}, \sigma_n, d_n\}$:

$$\frac{\partial C(\mathbf{y},\lambda,\mu_{\boldsymbol{w},\boldsymbol{\theta}})}{\partial\theta_{n,i}} = \mathbf{H} * \sum_{s\in\mathcal{S}} \frac{\partial\mathcal{G}(\boldsymbol{\theta}_n, w_n, s)}{\partial\theta_{n,i}} \times 2\left(\mathbf{\Phi}\mu_{\boldsymbol{w},\boldsymbol{\theta}} - \mathbf{y}\right) \tag{C.1}$$

We now write the derivatives of $\mathcal{G}$ along each component of one atom $\boldsymbol{\theta}_n$. For each location parameter, we have:

$$\frac{\partial\mathcal{G}(\boldsymbol{\theta}_n, w_n, \boldsymbol{s})}{\partial m_{n,i}} = \frac{d_n(s_i - m_{n,i})}{2\sigma_n^{d_n}}\|\boldsymbol{m}_n - \boldsymbol{s}\|_2^{d_n-2}\mathcal{G}(\boldsymbol{\theta}_n, w_n, \boldsymbol{s}) \tag{C.2}$$

Deriving $\mathcal{G}$ along the $n-$th standard deviation yields:

$$\frac{\partial\mathcal{G}(\boldsymbol{\theta}_n, w_n, \boldsymbol{s})}{\partial\sigma_n} = \frac{d_n}{2\sigma_n^{d_n+1}}\|\boldsymbol{m}_n - \boldsymbol{s}\|_2^{d_n}\mathcal{G}(\boldsymbol{\theta}_n, w_n, \boldsymbol{s}), \tag{C.3}$$

and the derivative with respect to the $n-$th degree is:

$$\frac{\partial\mathcal{G}(\boldsymbol{\theta}_n, w_n; \boldsymbol{s})}{\partial d_n} = -\frac{1}{2\sigma_n^{d_n}}\|\boldsymbol{m}_n - \boldsymbol{s}\|_2^{d_n}\log\left(\frac{1}{\sigma_n}\|\boldsymbol{m}_n - \boldsymbol{s}\|_2\right)\mathcal{G}(\boldsymbol{\theta}_n, w_n, \boldsymbol{s}). \tag{C.4}$$

Finally,

$$\frac{\partial C(\mathbf{y},\lambda,\mu_{\boldsymbol{w},\boldsymbol{\theta}})}{\partial w_n} = \lambda\,\mathrm{sgn}(w_n) + \mathbf{H} * \sum_{s\in\mathcal{S}} \frac{2\mathcal{G}(\boldsymbol{\theta}_n, w_n, \boldsymbol{s})}{w_n}\left(\mathbf{\Phi}\mu_{\boldsymbol{w},\boldsymbol{\theta}} - \mathbf{y}\right) \tag{C.5}$$

Algorithm D.1 TV - Homotopy algorithm

Input: $\mathbf{y}$, PSF $\mathbf{H}$, σ_{target}, $c > 0$
Output: Solution of ($\mathcal{P}_{\mathsf{TV}}$)
Initialization: $\lambda_0 = \|\mathbf{\Phi}^T \mathbf{y}\|_\infty$ and $\mathbf{x}_{[0]} = \mathbf{y}$.
repeat (iteration t)
1. Starting from $\hat{\mathbf{x}}_{[t-1]}$, solve the BLASSO problem ($\mathcal{P}_{\lambda_t \mathsf{TV}}$) using λ_t to obtain $\hat{\mathbf{x}}_{[t]}$ using the APG algorithm.
2. Update $\lambda_{t+1} = \frac{\lambda_t}{2}$
3. Estimate the standard deviation σ_t of the residual $\mathbf{y} - \hat{\mathbf{x}}_{[t]}$

until $\sigma_t < \sigma_{\text{target}}$

when $w_n \neq 0$. When $w_n = 0$ we adopt the convention $\frac{\partial C(\mathbf{y}, \lambda, \mu_{\boldsymbol{w},\boldsymbol{\theta}})}{\partial w_n} = 0$.

Appendix D. 3D image reconstruction with TV regularization. In this last appendix, we describe how TV regularization is used as an optimization constraint for 3D TDM images reconstruction. The goal is to reconstruct the volume $\mathbf{x}$ over the voxel grid $\mathcal{S}$, based on the observation $\mathbf{y}$.

Proximal gradient algorithms [26] are popular to solve problems like ($\mathcal{P}_{\lambda \mathsf{TV}}$). We choose to use the accelerated proximal gradient (APG) method [32]. We start at $\mathbf{x}^0 = \mathbf{y}$ and repeat the following steps $\forall k > 0$:

$$\begin{aligned} \overline{\mathbf{x}}^k &:= \mathbf{x}^k + \omega^k (\mathbf{x}^k - \mathbf{x}^{k-1}) \\ \mathbf{x}^{k+1} &:= \overline{\mathbf{x}}^k - \gamma^k G_{\gamma^k}(\overline{\mathbf{x}}^k) \end{aligned} \tag{D.1}$$

where

$$G_\gamma(\mathbf{x}) = \frac{1}{\gamma} \left(\mathbf{x} - \mathsf{prox}_{\gamma, \lambda \mathsf{TV}}(\mathbf{x} - \gamma \mathbf{H}^\top (\mathbf{H}\mathbf{x} - \mathbf{y})) \right); \tag{D.2}$$

with the proximal operator defined as:

$$\mathsf{prox}_{\gamma, \lambda \mathsf{TV}}(\mathbf{x}) = \underset{\mathbf{u} \in \mathbb{R}^S}{\arg\min} \left(\lambda \mathsf{TV}(\mathbf{u}) + \frac{1}{2\gamma} \|\mathbf{u} - \mathbf{x}\|_2^2 \right) \tag{D.3}$$

The latter is implemented using [4]. Besides, at each step k the value of $\gamma^k > 0$ is found by line search [6].

Finally, the APG algorithm is stopped based on a computation of the dual gap, *i.e.* the difference between the solution to ($\mathcal{P}_{\lambda \mathsf{TV}}$) and the solution to its dual. The dual gap for this problem is [16]:

$$G(\mathbf{x}^k) = \|\mathbf{H}\mathbf{x}^k - \mathbf{y}\|_2^2 + \lambda \mathsf{TV}(\mathbf{x}^k) + \langle \mathbf{H}\mathbf{x}^k - \mathbf{y}, \mathbf{y} \rangle + \lambda \|\mathbf{x} \nabla \cdot (-\mathbf{H}^\top \mathbf{x})\|_\infty \tag{D.4}$$

The closest G is to 0, the better is the solution. However 0 is never attained because the problem is defined in the presence of noise. Instead, we choose to stop the APG algorithm when the dual gap decrease stops. In practice, APG is stopped when the relative gap between $G(\mathbf{x}^k)$ and the average of the 10 previous values is lower than 10^{-3}.

Finally, in the same fashion as SFW and BSFW, the APG method can be embedded within an homotopy method, which is reported in Alg. D.1.