Fractional graph Laplacian for image reconstruction

Stefano Aleotti[a], Alessandro Buccini[b], Marco Donatelli[a]

[a] *Department of Science and High Technology, University of Insubria, Como, Italy*
[b] *Department of Mathematics and Computer Science, University of Cagliari, Cagliari, Italy*

Abstract

Image reconstruction problems, like image deblurring and computer tomography, are usually ill-posed and require regularization. A popular approach to regularization is to substitute the original problem with an optimization problem that minimizes the sum of two terms, an ℓ^2 term and an ℓ^q term with $0 < q \leq 1$. The first penalizes the distance between the measured data and the reconstructed one, the latter imposes sparsity on some features of the computed solution.

In this work, we propose to use the fractional Laplacian of a properly constructed graph in the ℓ^q term to compute extremely accurate reconstructions of the desired images. A simple model with a fully automatic method, i.e., that does not require the tuning of any parameter, is used to construct the graph and enhanced diffusion on the graph is achieved with the use of a fractional exponent in the Laplacian operator. Since the fractional Laplacian is a global operator, i.e., its matrix representation is completely full, it cannot be formed and stored. We propose to replace it with an approximation in an appropriate Krylov subspace. We show that the algorithm is a regularization method under some reasonable assumptions. Some selected numerical examples in image deblurring and computer tomography show the performance of our proposal.

Keywords: Fractional graph Laplacian, image reconstruction, $\ell^2 - \ell^q$ minimization.
2010 MSC: 65F22, 65K10, 65R30

1. Introduction

Image reconstruction algorithms are of fundamental importance in several fields of science and engineering, such as medicine, astronomy, and geophysics. In most cases, the problem is ill-posed, i.e., extremely sensitive to perturbations in the collected data. Regularization methods aim at reducing this sensitivity [4, 16, 20].

Email addresses: saleotti@uninsubria.it (Stefano Aleotti), alessandro.buccini@unica.it (Alessandro Buccini), marco.donatelli@uninsubria.it (Marco Donatelli)

In this paper we are concerned with the solution of problems of the form

$$\arg\min_{\mathbf{x}\in\mathbb{R}^n} \left\|A\mathbf{x}-\mathbf{b}^\delta\right\|_2^2, \tag{1}$$

where $\|\cdot\|_2$ denotes the Euclidean norm, $A\in\mathbb{R}^{m\times n}$ is the discretization of an integral operator, e.g., a blurring matrix, $\mathbf{b}^\delta\in\mathbb{R}^m$ collects some measurements that we assume are corrupted by errors, and $\mathbf{x}\in\mathbb{R}^n$ represents an unknown two-dimensional image with n pixels, whose entries have been reordered in lexicographical order. We assume that the data $\mathbf{b}^\delta$ can be written as

$$\mathbf{b}^\delta=\mathbf{b}+\boldsymbol{\eta}, \tag{2}$$

where $\boldsymbol{\eta}$ collects all the perturbations and is often referred to as noise and $\mathbf{b}$ is the unavailable noise-free data. Moreover, we assume that each entry of $\boldsymbol{\eta}$ is the realization of a Gaussian random variable with zero mean and fixed variance and that all the entries are independent. We denote by $\delta>0$ an upper-bound of the norm of the noise, i.e.,

$$\left\|\mathbf{b}^\delta-\mathbf{b}\right\|_2\leq\delta.$$

Due to the presence of noise in the observed data $\mathbf{b}^\delta$ and the severely ill-conditioning of A, the naive solution of (1) is a poor approximation of the desired solution $\mathbf{x}^\dagger=A^\dagger\mathbf{b}$, where $A^\dagger$ denotes the Moore-Penrose pseudo-inverse of A. Regularization methods substitute the original ill-posed problem (1) with a nearby well-posed one whose solution well approximates $\mathbf{x}^\dagger$. Among the various regularization methods in the literature, we consider here the $\ell^2-\ell^q$ regularization (see, e.g., [13, 17, 23, 21])

$$\arg\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\left\|A\mathbf{x}-\mathbf{b}^\delta\right\|_2^2+\frac{\mu}{q}\left\|L\mathbf{x}\right\|_q^q, \tag{3}$$

where $L\in\mathbb{R}^{s\times n}$ is called regularization operator, $\mu>0$ is referred to as regularization parameter, and $0<q\leq 2$. We define $\|\mathbf{x}\|_q^q=\sum_{i=1}^n|x_i|^q$ and we refer to this quantity as ℓ^q-norm, even though, if $q<1$, this is not a norm since it does not satisfy the triangular inequality. A Bayesian justification of (3) was given in [7], while its application to statistics was explored in [8]. For a review on how the parameter μ can be selected, we refer the interested reader to [10].

If $q\leq 1$, then the ℓ^q-norm approximates the so-called ℓ^0-norm that counts the non-vanishing entries of a vector; see, e.g., [23] for a discussion. Therefore, in this case, it is beneficial to select L such that $L\mathbf{x}^\dagger$ is as sparse as possible. It was shown in [11] that, if $L\mathbf{x}^\dagger$ is sparse, the quality of the computed solutions increases as q approaches 0. Popular choices are framelet operators and differential operators. Fractional differential operators have also been investigated to enhance diffusion, in particular with denoising problems [1, 33]. Moreover, several authors have recently considered selecting L as the graph Laplacian of a properly constructed graph obtained from a given approximation of $\mathbf{x}^\dagger$; see, e.g., [5, 9, 22, 25, 26, 27, 29, 32]. On the other hand, to the best of our knowledge,

the fractional graph Laplacian was yet to be used as a regularization operator for image restoration problems.

Fractional graph Laplacian has recently attracted the attention of the community working on complex networks [3, 6]. It allows to explore non-local dynamics that can spread the information in the graph. The drawback of this strategy is that the fractional graph Laplacian is a full matrix even if the graph Laplacian is sparse. Therefore, approximation tools need to be explored to perform computation with the fractional graph Laplacian. In this direction, the spectral approximation of the graph Laplacian proposed in [29] is very useful and will be employed in our method. In detail, the authors explore the use of the Lanczos method for filtering signals on graphs and observe that only few Lanczos iterations are sufficient to obtain a good approximation of a filtering function of the graph Laplacian.

In this paper, we expand the algorithmic proposal in [9]. In the latter, the authors first constructed an approximation of the solution of (1) with Tikhonov regularization, i.e., by setting $q = 2$ in (3). Starting from this approximation they constructed a graph Laplacian to use as a regularization operator in (3) with $q < 1$. Here we improve this method as follows. We employ an improved algorithm for the minimization of (3) recently proposed in [12]. Moreover, instead of considering L, we consider the fractional graph Laplacian L^α with $\alpha > 0$, where the graph is computed by the approximation of $\mathbf{x}^\dagger$ obtained by the graph Laplacian. In practice, we add a further step to the algorithm proposed in [9] updating the graph and forcing enhanced diffusion by a fractional exponent. Finally, we prove that the proposed method is a regularization method, i.e., that the computed solutions converge to the exact one as $\delta \to 0$ under some suitable assumptions. Though we have two parameters μ and α to estimate, the proposed approach is completely automatic. This is achieved by combining the Discrepancy Principle (DP), which prescribes that the computed solution $\bar{\mathbf{x}}$ satisfies

$$\left\| A\bar{\mathbf{x}} - \mathbf{b}^\delta \right\|_2 = \tau\delta,$$

where $\tau > 1$ is a user-defined constant, and the whiteness residual principle (see [24]) which requires that the residual $A\bar{\mathbf{x}} - \mathbf{b}^\delta$ is as white as possible; see below for more details.

This paper is structured as follows. In Section 2 we report the minimization algorithm in [12] tailored to the $\ell^2 - \ell^q$ problem in (3). Section 3 presents how to construct the graph Laplacian starting from a given image. In Section 4 we describe how to apply the MM method to the $\ell^2 - \ell^q$ case with the graph Laplacian as a regularizer. We provide some pseudocode describing the implementation of these methods. We present our algorithmic proposal in Section 5 and report some numerical examples in Section 6. Finally, we draw our conclusions in Section 7.

2. Majorization-Minimization for $\ell^2 - \ell^q$

We briefly describe the approaches proposed in [21] to solve (3). In this paper we set $p = 2$, however, the algorithms proposed in [21] allow for a general

$0 < p \leq 2$.

A possible approach to solve (3) is to use a Majorization-Minimization (MM) method. Note that, for $q \leq 1$, the minimized functional in (3) is non-smooth, therefore, we substitute it with a smooth approximation. Let $\varepsilon > 0$ be a fixed parameter and denote by

$$\Phi_{q,\varepsilon}(t) = \left(\sqrt{t^2 + \varepsilon^2}\right)^q.$$

Assuming that ε is small enough, we can approximate $\|\mathbf{x}\|_q^q$ by

$$\|\mathbf{x}\|_q^q \approx \sum_{i=1}^{n} \Phi_{q,\varepsilon}(x_i), \qquad \mathbf{x} \in \mathbb{R}^n.$$

Note that the function on the right-hand side is everywhere differentiable, while the one on the left is not differentiable if at least one of the components of $\mathbf{x}$ vanishes. Therefore, we substitute problem (3) by

$$\min_{\mathbf{x}\in\mathbb{R}^n} \mathcal{J}_\varepsilon(\mathbf{x}), \tag{4}$$

where

$$\mathcal{J}_\varepsilon(\mathbf{x}) = \frac{1}{2}\left\|A\mathbf{x} - \mathbf{b}^\delta\right\|_2^2 + \frac{\mu}{q}\sum_{i=1}^{s} \Phi_{q,\varepsilon}((L\mathbf{x})_i).$$

The MM algorithm constructs a sequence $\left\{\mathbf{x}^{(k)}\right\}_k$ that converges to a stationary point of $\mathcal{J}_\varepsilon$. Let $\mathbf{x}^{(k)}$ be the current approximation of the solution of (4), the MM method first determines a quadratic functional $\mathcal{Q}\left(\mathbf{x}, \mathbf{x}^{(k)}\right)$ that majorizes $\mathcal{J}_\varepsilon$ everywhere and that is tangent to it in $\mathbf{x}^{(k)}$. Then, the new iterate $\mathbf{x}^{(k+1)}$ is the minimizer of $\mathcal{Q}\left(\mathbf{x}, \mathbf{x}^{(k)}\right)$.

Given $\mathcal{J}_\varepsilon$ and $\mathbf{x}^{(k)}$ one can construct infinitely many quadratic tangent majorants $\mathcal{Q}\left(\mathbf{x}, \mathbf{x}^{(k)}\right)$. In [21] the authors proposed two choices. We describe here the so-called fixed majorant. The name derives from the fact that, in the one-dimensional case, it coincides with a parabola whose leading coefficient does not depend on $\mathbf{x}^{(k)}$. Fix $\varepsilon > 0$, let $\mathbf{u}^{(k)} = L\mathbf{x}^{(k)}$ and

$$\boldsymbol{\omega}^{(k)} = \mathbf{u}^{(k)}\left(1 - \left(\frac{(\mathbf{u}^{(k)})^2 + \varepsilon^2}{\varepsilon^2}\right)^{q/2-1}\right),$$

where all operations are meant element-wise, then,

$$\mathcal{Q}(\mathbf{x}, \mathbf{x}^{(k)}) = \frac{1}{2}\left\|A\mathbf{x} - \mathbf{b}^\delta\right\|_2^2 + \frac{\mu\varepsilon^{q-2}}{2}\left(\|L\mathbf{x}\|_2^2 - 2\left\langle \boldsymbol{\omega}^{(k)}, L\mathbf{x}\right\rangle\right) + c,$$

where c is a constant that does not depend on $\mathbf{x}$. Note that $\mathcal{Q}\left(\mathbf{x}, \mathbf{x}^{(k)}\right)$ is a quadratic tangent majorant of $\mathcal{J}_\varepsilon$ in $\mathbf{x}^{(k)}$; see [21]. The approximation $\mathbf{x}^{(k+1)}$ is obtained by minimizing $\mathcal{Q}$ with respect to $\mathbf{x}$, i.e.,

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}\in\mathbb{R}^n} \left\| \begin{bmatrix} A \\ \nu^{1/2}L \end{bmatrix}\mathbf{x} - \begin{bmatrix} \mathbf{b}^\delta \\ \nu^{1/2}\boldsymbol{\omega}^{(k)} \end{bmatrix}\right\|_2^2, \tag{5}$$

where $\nu = \mu\varepsilon^{q-2}$. Therefore, we solve the least squares problem (5) at each iteration.

An approximate solution of (5) can be computed in a subspace of $\mathbb{R}^n$ of fairly small dimension. Let $V_k \in \mathbb{R}^{n\times\hat{k}}$ be a matrix with orthonormal columns. Assuming that the columns of V_k span the search subspace, we look for a solution of the form

$$\mathbf{x}^{(k+1)} = V_k \mathbf{y}^{(k+1)}, \tag{6}$$

where $\mathbf{y}^{(k+1)}$ is obtained solving

$$\mathbf{y}^{(k+1)} = \arg\min_{\mathbf{y}\in\mathbb{R}^{\hat{k}}} \left\| \begin{bmatrix} AV_k \\ \nu^{1/2} LV_k \end{bmatrix} \mathbf{y} - \begin{bmatrix} \mathbf{b}^\delta \\ \nu^{1/2}\boldsymbol{\omega}^{(k)} \end{bmatrix} \right\|_2^2. \tag{7}$$

Note that (7) is obtained by plugging $\mathbf{x} = V_k\mathbf{y}$ in (5). Since the matrices AV_k and LV_k have more rows than columns, we can compute just the first $\hat{k}$ rows of R and the first $\hat{k}$ columns of Q in their QR factorizations. This is called *economic* (or economy-size) QR factorization. These factorizations read

$$\begin{aligned} AV_k &= Q_A R_A \quad \text{with} \quad Q_A \in \mathbb{R}^{m\times\hat{k}}, \quad R_A \in \mathbb{R}^{\hat{k}\times\hat{k}}, \\ LV_k &= Q_L R_L \quad \text{with} \quad Q_L \in \mathbb{R}^{s\times\hat{k}}, \quad R_L \in \mathbb{R}^{\hat{k}\times\hat{k}}, \end{aligned} \tag{8}$$

where Q_A and Q_L have orthonormal columns and R_A and R_L are upper triangular. Plugging the decompositions (8) in (7), we obtain

$$\mathbf{y}^{(k+1)} = \arg\min_{\mathbf{y}\in\mathbb{R}^{\hat{k}}} \left\| \begin{bmatrix} R_A \\ \nu^{1/2} R_L \end{bmatrix} \mathbf{y} - \begin{bmatrix} Q_A^T\mathbf{b}^\delta \\ \nu^{1/2} Q_L^T\boldsymbol{\omega}^{(k)} \end{bmatrix} \right\|_2^2,$$

which can be solved with direct methods since $\hat{k} \ll n$; see, e.g., [15].

Computing the residual of the normal equation associated with (5) and recalling that $\mathbf{x}^{(k+1)} = V_k\mathbf{y}^{(k+1)}$, we obtain

$$\mathbf{r}^{(k+1)} = A^T(AV_k\mathbf{y}^{(k+1)} - \mathbf{b}^\delta) + \nu L^T(LV_k\mathbf{y}^{(k+1)} - \boldsymbol{\omega}^{(k)}).$$

Following [30], at each iteration, we expand the search subspace by adding to the basis V_k the normalized residual, i.e.,

$$V_{k+1} = [V_k, \mathbf{v}^{(k+1)}], \qquad \mathbf{v}^{(k+1)} = \mathbf{r}^{(k+1)} / \left\|\mathbf{r}^{(k+1)}\right\|_2.$$

Note that, in exact arithmetic, $\mathbf{v}^{(k+1)}$ is orthogonal to the space spanned by the columns of V_k.

Since the following computations are exactly the same for both matrices A and L, we can describe them using a generic matrix $C \in \{A, L\}$.

Following [14], the QR factorizations of CV_{k+1} is computed by updating the QR factorization $CV_k = Q_C R_C$, cf. (8), according to

$$CV_{k+1} = [CV_k, C\mathbf{v}^{(k+1)}] = [Q_C, \mathbf{q}_C] \begin{bmatrix} R_C & \mathbf{r}_C \\ \mathbf{0}^T & \tau_C \end{bmatrix},$$

where

$$\begin{array}{rclcrclcrcl} \tilde{\mathbf{v}}^{(k+1)} & = & C\mathbf{v}^{(k+1)}, & & \mathbf{r}_C & = & Q_C^T\tilde{\mathbf{v}}^{(k+1)}, & & & & \\ \tilde{\mathbf{q}}_C & = & \tilde{\mathbf{v}}^{(k+1)} - Q_C\mathbf{r}_C, & & \tau_C & = & \|\tilde{\mathbf{q}}_C\|_2\,, & & \mathbf{q}_C & = & \tilde{\mathbf{q}}_C/\tau_C, \end{array}$$

We now briefly discuss the strategy proposed in [12] to reduce the computational cost of the MM algorithm. The authors observed that in real applications only a few vectors of the Krylov subspace are actually used and that most of the coefficients of $\mathbf{y}^{(k)}$ almost vanish. Therefore, they propose to restart the space every r iterations. More in details, if $k \equiv 0 \bmod r$ we set

$$V_k = \mathbf{x}^{(k)} / \left\|\mathbf{x}^{(k)}\right\|_2 \in \mathbb{R}^n.$$

We compute CV_k and its economic QR factorization is easily obtained as

$$CV_k = Q_C R_C, \quad \text{with} \quad Q_C = CV_k / \|CV_k\|_2 \quad \text{and} \quad R_C = \|CV_k\|_2\,,$$

We then proceed with the iterations as in the MM method. In [12] the authors proved that the obtained algorithm is a descent method, i.e., it holds

$$\mathcal{J}_\varepsilon\left(\mathbf{x}^{(k+1)}\right) \leq \mathcal{J}_\varepsilon\left(\mathbf{x}^{(k)}\right).$$

Moreover, there exists a converging subsequence $\mathbf{x}^{(k_j)}$. Extensive numerical experience, however, suggests that the whole sequence converges and there is no need to consider subsequences.

3. Graph Laplacian for image reconstruction

In this section, we describe how, given an approximation of $\mathbf{x}^\dagger$, we construct the graph Laplacian. We first recall the definition of a graph and how to construct its Laplacian. We then describe how to build a graph that we can use for our purposes starting from an image $\widehat{\mathbf{x}}$.

An *unweighted graph* is a pair $G = (V, E)$, where V is the vertex set and $E \subseteq V \times V$ is the set containing all the edges. A graph G is said to be undirected if $(i, j) \in E$ implies that $(j, i) \in E$, otherwise we say that the graph is directed. Sometimes, a graph G is associated with a measure $\omega : E \to \mathbb{R}^+$ which associates each edge of the graph with a unique positive value that is often referred to as the weight of the edge. In this case, the graph G is called a *weighted graph*. It is possible to represent it by means of the adjacency matrix. Let $n = |V|$, where $|\mathcal{A}|$ is the cardinality of the set $\mathcal{A}$, i.e., n is the number of nodes in G. Then, we denoted the adjacency matrix $\Omega \in \mathbb{R}^{n\times n}$ of the graph G as

$$\Omega_{i,j} = \begin{cases} \omega(i,j) & \text{if } (i,j) \in E, \\ 0 & \text{otherwise,} \end{cases}$$

where we set $\omega(i, j) \equiv 1$ for all $(i, j) \in E$ for unweighted graphs.

Now, we describe how to construct the graph Laplacian associated with a given graph G. There are many definitions for the graph Laplacian. For our purposes, we define this operator as follows. Given the adjacency matrix $\Omega \in \mathbb{R}^{n\times n}$ of G we define the degree matrix D as the diagonal matrix

$$D_{j,j} = \deg(j) = \sum_{i=1}^{n} \Omega_{i,j}.$$

Then, the graph Laplacian L_ω is defined by

$$L_\omega = \frac{D - \Omega}{\|\Omega\|},$$

where $\|\cdot\|$ denotes the Frobenius norm. Note that, if ω is symmetric, that is $\omega(i,j) = \omega(j,i)$, $\forall (x_i, x_j) \in E$, then the adjacency matrix Ω and L_ω are symmetric. Moreover, the graph Laplacian is a positive semi-definite matrix, i.e. $\forall \mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}^T L_\omega \mathbf{x} \geq 0$, and it is also stochastic by rows. Thus, we have that

$$\ker(L_\omega) \supseteq \operatorname{span}\{\mathbf{1}\},$$

where $\mathbf{1}$ is the constant vector.

We now describe how to construct a proper graph Laplacian for solving (3). Given a fairly accurate approximation $\widehat{\mathbf{x}}$ of the minimal norm solution $\mathbf{x}^\dagger$, we reshape it into matrix form $\widehat{X} \in \mathbb{R}^{d_1\times d_2}$ and we set $n = d_1 d_2$. To simplify the notation we assume that $d_1 = d_2$ so that the image is square. We define the undirected graph G related to the image $\widehat{X}$ identifying each pixel of $\widehat{X}$ with a vertex of G. Let $R \in \mathbb{N}$ and $\sigma \in \mathbb{R}^+$ be two fixed numbers, we connect two pixels $\widehat{X}_{i_1,i_2}$ and $\widehat{X}_{j_1,j_2}$ if

$$0 < \|\mathbf{i} - \mathbf{j}\|_\infty = \left\| \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} - \begin{bmatrix} j_1 \\ j_2 \end{bmatrix} \right\|_\infty \leq R,$$

where $\|\mathbf{x}\|_\infty = \max\{|x_k|,\, k = 1,\dots,n\}$ for $\mathbf{x} \in \mathbb{R}^n$. Then, we define the weight measure ω by

$$\omega(\widehat{X}_{\mathbf{i}}, \widehat{X}_{\mathbf{j}}) = \mathrm{e}^{-(\widehat{X}_{i_1,i_2} - \widehat{X}_{j_1,j_2})^2/\sigma}.$$

In this way, denoting by i and j the lexicographically ordered indexes of $\mathbf{i}$ and $\mathbf{j}$ respectively, the adjacency matrix Ω is defined by

$$\Omega_{i,j} = \begin{cases} \mathrm{e}^{-(\widehat{X}_{i_1,i_2} - \widehat{X}_{j_1,j_2})^2/\sigma} & \text{if } 0 < \|\mathbf{i} - \mathbf{j}\|_\infty \leq R, \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathbf{i} = \begin{bmatrix} i_1, & i_2 \end{bmatrix}^T$ and $\mathbf{j} = \begin{bmatrix} j_1, & j_2 \end{bmatrix}^T$. Intuitively, if two pixels (nodes of the graph G) are close to each other and have similar intensity then we connect them with a strong weight. On the other hand, if two pixels have different values or are too far away from each other we connect them with a low weight or we do not connect them at all. In this way, the graph Laplacian encodes the structure of the image $\widehat{X}$ with strong connections between pixels belonging

to the same area of the image and low connections between nodes belonging to different parts of the picture.

It was shown in [5, 9] that the Laplacian related to this graph is a good regularization operator for $\ell^2 - \ell^q$ image deblurring whenever $\widehat{\mathbf{x}}$ is a fairly accurate approximation of $\mathbf{x}^\dagger$.

4. Graph Laplacian for $\ell^2 - \ell^q$

In this section, we apply the MM algorithm described in Section 2 to the $\ell^2 - \ell^q$ problem in (3) with L_ω as regularization operator. In practice, we provide a new implementation of the algorithm proposed in [9] based on the restarting strategy proposed in [12].

We first discuss how we select the initial approximation of $\mathbf{x}^\dagger$ that we use to construct L_ω. We wish to solve (3) with L defined as the finite difference approximation of the 2D gradient, which is

$$L = \begin{bmatrix} L_1 \otimes I \\ I \otimes L_1 \end{bmatrix} \qquad \text{with} \qquad L_1 = \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ 1 & & & -1 \end{bmatrix}, \tag{9}$$

where $\otimes$ is the Kronecker product and I is the identity matrix. The size of the two square matrices L_1 and I depends on the size of the image $\mathbf{x}^\dagger$ to restore. For simplicity, if we assume that $\mathbf{x}^\dagger \in \mathbb{R}^{n \times n}$, then the matrices L_1 and I have the same size of $\mathbf{x}^\dagger$. Therefore, the matrix $L \in \mathbb{R}^{2n^2 \times n^2}$ is extremely sparse.

To determine the parameter μ we use the DP as described in [11, 10]. We consider in the MM iterations (5) a non-stationary μ_k (and in turn ν_k) such that at each iteration it holds

$$\left\| A\mathbf{x}^{(k+1)} - \mathbf{b}^\delta \right\|_2 = \tau\delta,$$

with $\tau > 1$. In [11] the authors proved that such μ_k exists and the iterates converge (up to subsequences) to a certain $\hat{\mathbf{x}}$ that satisfies the DP as well. We use this $\hat{\mathbf{x}}$ to construct our graph Laplacian.

For completeness, we report the computations in Algorithm 1.

After computing an initial approximation $\widehat{\mathbf{x}}$ of $\mathbf{x}^\dagger$, we use $\widehat{\mathbf{x}}$ to construct the graph Laplacian L_ω. Note that by construction L_ω is symmetric and hence in Algorithm 1 the only operation involving L_ω is the matrix-vector product. Moreover, since L_ω is a sparse matrix, the matrix-vector product can be computed efficiently with a linear cost in n. Therefore, the same Algorithm 1 can be used to solve the $\ell^2 - \ell^q$ problem in (3) with L_ω as regularization operator. We report the computations in Algorithm 2.

5. Fractional graph Laplacian for $\ell^2 - \ell^q$

To improve the quality of the reconstruction $\mathbf{x}^*$ obtained by Algorithm 2, we construct a new graph based on $\mathbf{x}^*$ and we take the α-th power of the new graph

Algorithm 1: Nonstationary $\ell^2 - \ell^q$ with the Discrepancy Principle

Input : A, $\mathbf{b}^\delta$, δ, q, L, $\mathbf{x}^0$, ε, τ, K, r, γ

Construct $V_0 \in \mathbb{R}^{n \times \hat{k}}$ such that $V_0^T V_0 = I$;

Compute and store AV_0 and LV_0, and their economic QR factorizations $AV_0 = Q_A R_A$; $LV_0 = Q_L R_L$;

for $k = 0, 1, \ldots, K$ **do**

  if *($k \equiv 0 \bmod r$) and ($k \neq 0$)* **then**

    $V_k = \mathbf{x}^{(k)} / \left\|\mathbf{x}^{(k)}\right\|_2$;

    Compute and store AV_k;

    $R_A = \|AV_k\|_2$;

    $Q_A = AV_k / R_A$;

    Compute and store LV_k;

    $R_L = \|LV_k\|_2$;

    $Q_L = LV_k / R_L$;

  $\mathbf{u}^{(k)} = L\mathbf{x}^{(k)}$;

  $\boldsymbol{\omega}^{(k)} = \mathbf{u}^{(k)} \left(1 - \left(\frac{(\mathbf{u}^{(k)})^2 + \varepsilon^2}{\varepsilon^2}\right)^{q/2-1}\right)$;

  $\mathbf{y}^{(k+1)} = \arg\min_{\mathbf{y}} \left\|R_A\mathbf{y} - Q_A^T\mathbf{b}^\delta\right\|_2^2 + \nu^{(k)} \left\|R_L\mathbf{y} - Q_L^T\boldsymbol{\omega}^{(k)}\right\|_2^2$, where ν^k is such that $\left\|R_A\mathbf{y}^{(k+1)} - Q_A^T\mathbf{b}^\delta\right\|_2 = \tau\delta$;

  if $\left\|\mathbf{y}^{(k+1)} - \mathbf{y}^{(k)}\right\|_2 \leq \gamma \left\|\mathbf{y}^{(k)}\right\|_2$ **then**

    exit;

  $\mathbf{r}^{(k+1)} = A^T(AV_k\mathbf{y}^{(k+1)} - \mathbf{b}^\delta) + \nu L^T(LV_k\mathbf{y}^{(k+1)} - \boldsymbol{\omega}^{(k)})$;

  $\mathbf{v}^{(k+1)} = \mathbf{r}^{(k+1)} / \left\|\mathbf{r}^{(k+1)}\right\|_2$;

  $V_{k+1} = [V_k,\ \mathbf{v}^{(k+1)}]$;

  $AV_{k+1} = [AV_k,\ A\mathbf{v}^{(k+1)}]$;

  $LV_{k+1} = [LV_k,\ L\mathbf{v}^{(k+1)}]$;

  Update the QR factorizations of AV_{k+1} and LV_{k+1};

$\mathbf{x}^* = V_k\mathbf{y}^{(k+1)}$;

Output: $\mathbf{x}^*$

Algorithm 2: Graph Laplacian $\ell^2 - \ell^q$

Input : A, $\mathbf{b}^\delta$, δ, q, L, $\mathbf{x}^0$, ε, τ, K, r, γ, σ, R, $\alpha_{\min}$, $\alpha_{\max}$, J, d

1 Compute, using Algorithm 1 with inputs A, $\mathbf{b}^\delta$, δ, q, L, $\mathbf{x}^0$, ε, τ, K, r, the approximation $\hat{\mathbf{x}}$;

2 Construct the adjacency matrix

$$\Omega_{i,j} = \begin{cases} \mathrm{e}^{-(\widehat{X}_{i_1,i_2} - \widehat{X}_{j_1,j_2})^2/\sigma} & \text{if } 0 < \|\mathbf{i} - \mathbf{j}\|_\infty \leq R, \\ 0 & \text{otherwise,} \end{cases}$$

where $\hat{\mathbf{x}} = \mathrm{vec}\left(\widehat{X}\right)$, i and j are the lexicographic indexes of $\mathbf{i} = \begin{bmatrix} i_1, & i_2 \end{bmatrix}^T$ and $\mathbf{j} = \begin{bmatrix} j_1, & j_2 \end{bmatrix}^T$, respectively;

3 Construct the diagonal matrix $D_{j,j} = \sum_{i=1}^{n} \Omega_{i,j}$;

4 $L_\omega = \frac{D - \Omega}{\|\Omega\|_F}$;

5 Compute, using Algorithm 1 with inputs A, $\mathbf{b}^\delta$, δ, q, L_ω, $\mathbf{x}^0$, ε, τ, K, r, the approximation $\mathbf{x}^*$, where every product with L_ω is performed using sparse matrices;

Output: $\mathbf{x}^*$

Laplacian L_ω in order to better diffuse the information along the graph. Finally, we solve the $\ell^2 - \ell^q$ problem in (3), for fixed $\alpha > 0$, with L_ω^α as regularization operator.

We are now faced with the following issues, L_ω^α is a full matrix and hence it cannot be explicitly formed, therefore, it has to be approximated. Moreover, we have to provide an automatic rule for the computation of a suitable value for α. In the next subsections, we describe how to perform matrix-vector products with L_ω^α, we propose an algorithm for an automatic estimation of α, and we study the convergence of the proposed algorithm.

5.1. Approximation of the fractional graph Laplacian

Let $\alpha > 0$, we wish to solve

$$\arg\min_{\mathbf{x}} \frac{1}{2} \left\| A\mathbf{x} - \mathbf{b}^\delta \right\|_2^2 + \frac{\mu}{q} \left\| L_\omega^\alpha \mathbf{x} \right\|_q^q . \tag{10}$$

Note that, by construction, L_ω is symmetric and positive semidefinite, hence it is possible to determine an orthonormal basis of eigenvectors of L_ω. Let $\lambda_j \geq 0$ be the eigenvalues of L_ω for $j = 1, \ldots, n$, and let Q be the matrix formed by the eigenvectors of L_ω, then

$$L_\omega^\alpha = Q \Lambda^\alpha Q^T, \tag{11}$$

where $\Lambda^\alpha = \mathrm{diag}\left(\lambda_1^\alpha, \ldots, \lambda_n^\alpha\right)$, is symmetric and positive semidefinite. Unfortunately, it is computationally too expensive to compute the spectral decomposition of L_ω.

We now discuss how we implement the MM algorithm when applied to the minimization (10). The only difference with Algorithm 1 is in the computation of the matrix-vector products with L_ω^α and $\left(L_\omega^\alpha\right)^T$. Since L_ω^α is symmetric and positive semidefinite as L_ω, it is sufficient to discuss how to implement the matrix-vector products with L_ω^α without explicitly constructing the matrix L_ω^α.

We distinguish the case $\alpha \in \mathbb{N}$ and $\alpha \in \mathbb{R}^+ \setminus \mathbb{N}$. In the first case, if $\alpha = 1$, there is nothing to discuss, therefore, we assume $\alpha > 1$. Since L_ω is sparse, it is computationally attractive to perform the matrix-vector product as

$$L_\omega^\alpha \mathbf{x} = \underbrace{L_\omega(L_\omega(\ldots(L_\omega \mathbf{x})))}_{\alpha \text{ times}},$$

instead of to explicitly form the matrix L_ω^α.

If $\alpha \in \mathbb{R}^+ \setminus \mathbb{N}$, since $n \gg 1$, we cannot either explicitly form the matrix L_ω^α or compute its spectral decomposition (11). Therefore, following the proposal in [29], to compute $L_\omega^\alpha \mathbf{x}$ for a given $\mathbf{x}$, we project the problem in the Krylov subspace

$$\mathcal{K}_d(L_\omega, \mathbf{x}) = \operatorname{span}\left\{\mathbf{x}, L_\omega \mathbf{x}, \ldots, L_\omega^{d-1}\mathbf{x}\right\},$$

where we assume $d \ll n$ small enough so that the dimension of $\mathcal{K}_d(L_\omega, \mathbf{x})$ is d. Using d steps of the Lanczos algorithm (see, e.g., [19]), with starting vector $\mathbf{x}$, we obtain the following relation

$$L_\omega V_d = V_{d+1} T_{d+1,d},$$

where $V_{d+1} \in \mathbb{R}^{n \times (d+1)}$ has orthonormal columns that spans the space $\mathcal{K}_{d+1}(L_\omega, \mathbf{x})$ and $T_{d+1,d} \in \mathbb{R}^{\times (d+1) \times d}$ is of the form

$$T_{d+1,d} = \begin{bmatrix} \alpha_1 & \beta_2 & & & & & \\ \beta_2 & \alpha_2 & \beta_3 & & & & \\ & \beta_3 & \ddots & \ddots & & & \\ & & \ddots & \ddots & \beta_{d-1} & & \\ & & & \beta_{d-1} & \alpha_{d-1} & \beta_d & \\ & & & & \beta_d & \alpha_d & \\ & & & & & \beta_{d+1} \end{bmatrix}.$$

We use the matrix $T_{d,d}$, i.e., the leading $d \times d$ block of $T_{d+1,d}$, to approximate the matrix-vector product $L_\omega^\alpha \mathbf{x}$ as

$$\begin{aligned} L_\omega^\alpha \mathbf{x} &\approx \left(V_d T_{d,d} V_d^T\right)^\alpha \mathbf{x} \\ &= \left(V_d U \Sigma U^T V_d^T\right)^\alpha \mathbf{x} \\ &= V_d U \Sigma^\alpha U^T V_d^T \mathbf{x}, \end{aligned}$$

where $T_{d,d} = U \Sigma U^T$ is the eigenvalue decomposition of $T_{d,d}$, that exists since $T_{d,d}$ is symmetric. If $d \ll n$ the computation of U and Σ can be performed with small computational burden.

5.2. Selection of the parameters

According to the analysis and the numerical results in [29], only a few iterations of the Lanczos method are sufficient to obtain a good approximation of the graph Laplacian, in particular, when a filtering function is applied to it. Therefore, the dimension d of the Krylov subspace introduced in the previous subsection is not crucial and even a small d is enough to obtain a good approximation. For instance, we fix $d = 10$ in the numerical results for different applications (deblurring and computer tomography) and images of different sizes.

In (10), we need to determine two parameters, the regularization parameter μ and the fractional parameter α. Moreover, we would like to ensure that our choices ensure that the obtained algorithm is a regularization method; see below. Due to the Bakushinskii veto [2], in order to construct a regularization method, we need to assume that an accurate estimate of the norm of the noise δ is available.

We proceed as follows. Let $0 < \alpha_{\min} < \alpha_{\max}$ be two fixed values and let $J \in \mathbb{N}$ be given. We define

$$\alpha_j = \alpha_{\min} + j\frac{\alpha_{\max} - \alpha_{\min}}{J}, \quad j = 0, 1, \ldots, J.$$

For each j we consider the minimization problem

$$\mathbf{x}_\mu = \arg\min_{\mathbf{x}} \frac{1}{2} \left\| A\mathbf{x} - \mathbf{b}^\delta \right\|_2^2 + \frac{\mu}{q} \left\| L_\omega^{\alpha_j} \mathbf{x} \right\|_q^q .$$

We would like to select μ so that the discrepancy principle is satisfied. This can be done a posteriori, as described in [10], by trying several values of μ and selecting the largest one such that $\left\| A\mathbf{x}_\mu - \mathbf{b}^\delta \right\|_2 \leq \tau\delta$. However, this may become computationally expensive if J is large or if many values of μ are considered. Therefore, we follow the strategy proposed in [11] and described in Algorithm 1, where the only difference is that every multiplication with $L_\omega^{\alpha_j}$ is performed using the Lanczos algorithm as described above. Therefore, for each α_j, we compute a $\mathbf{x}_j$ such that

$$\left\| A\mathbf{x}_j - \mathbf{b}^\delta \right\|_2 = \tau\delta, \quad j = 0, \ldots, J.$$

We now discuss how we select the solution $\mathbf{x}_j$. Following the idea in [10, 24] we wish to select $\mathbf{x}_j$ using the residual whiteness principle. Ideally, if $\mathbf{x}_j = \mathbf{x}^\dagger$ for a certain j, then $\mathbf{r}_j = \mathbf{b}^\delta - A\mathbf{x}_j = \boldsymbol{\eta}$ and, therefore, $\mathbf{r}_j$ would be white since $\boldsymbol{\eta}$ defined in (2) has this property. We propose to select j such that $\mathbf{r}_j$ is as white as possible. We consider the measure of whiteness, introduced by Lanza et al. [24], defined by

$$\mathcal{W}(\mathbf{r}) = \frac{\|\mathbf{r} \star \mathbf{r}\|_2^2}{\|\mathbf{r}\|_2^4}, \tag{12}$$

where $\star$ denotes the two-dimensional convolution. The computation of $\mathcal{W}(\mathbf{r})$ can be performed cheaply thanks to the convolution theorem. Let F be the

discrete Fourier matrix, then

$$\mathcal{W}(\mathbf{r}) = \frac{\left\| |F\mathbf{r}|^2 \right\|_2^2}{\|F\mathbf{r}\|_2^4},$$

where $|\cdot|$ denotes the modulus of a complex number and the operations are meant element-wise.

Using the function $\mathcal{W}$ we compute

$$\widehat{j} = \arg\min_j \mathcal{W}(\mathbf{r}_j)$$

and select our approximate solution as $\mathbf{x}^* = \mathbf{x}_{\widehat{j}}$. We summarize the computations in Algorithm 3.

Algorithm 3: Fractional Graph Laplacian $\ell^2 - \ell^q$

Input : A, $\mathbf{b}^\delta$, δ, q, L, $\mathbf{x}^0$, ε, τ, K, r, γ, σ, R, $\alpha_{\min}$, $\alpha_{\max}$, J, d

Compute the approximation $\hat{\mathbf{x}}$, using Algorithm 2 with inputs A, $\mathbf{b}^\delta$, δ, q, L, $\mathbf{x}^0$, ε, τ, K, r, ;

Construct the adjacency matrix

$$\Omega_{i,j} = \begin{cases} \mathrm{e}^{-(\widehat{X}_{i_1,i_2} - \widehat{X}_{j_1,j_2})^2/\sigma} & \text{if } 0 < \|\mathbf{i}-\mathbf{j}\|_\infty \le R, \\ 0 & \text{otherwise,} \end{cases}$$

where $\hat{\mathbf{x}} = \mathrm{vec}\left(\hat{X}\right)$, i and j are the lexicographic indexes of $\mathbf{i} = \begin{bmatrix} i_1, & i_2 \end{bmatrix}^T$ and $\mathbf{j} = \begin{bmatrix} j_1, & j_2 \end{bmatrix}^T$, respectively;

Construct the diagonal matrix $D_{j,j} = \sum_{i=1}^n \Omega_{i,j}$;

$L_\omega = \frac{D-\Omega}{\|\Omega\|_F}$;

for *j=1,...,J* **do**

  Compute the approximation $\mathbf{x}_j$, using Algorithm 1 with inputs A, $\mathbf{b}^\delta$, δ, q, $L_\omega^{\alpha_j}$, $\mathbf{x}^0$, ε, τ, K, r, where every product with $L_\omega^{\alpha_j}$ is performed using d steps of Lanczos, as discussed above;

  $\boldsymbol{\theta}_j = F\left(\mathbf{b}^\delta - A\mathbf{x}_j\right)$;

  $w_j = \frac{\left\| |\boldsymbol{\theta}_j|^2 \right\|^2}{\|\boldsymbol{\theta}_j\|^4}$;

$\widehat{j} = \arg\min_j \{w_j\}$;

$\mathbf{x}^* = \mathbf{x}_{\widehat{j}}$;

Output: $\mathbf{x}^*$

5.3. Theoretical results

We now discuss some theoretical properties of our method. In particular, we wish to show that Algorithm 3 is a regularization method, i.e., that, if

$\delta_j \searrow 0$ as $j \to \infty$, denoting by $\mathbf{x}_j^*$ the solution obtained with data $\mathbf{b}^{\delta_j}$, where $\left\|\mathbf{b} - \mathbf{b}^{\delta_j}\right\|_2 \leq \delta_j$, then

$$\limsup_{j\to\infty} \left\|\mathbf{x}_j^* - \mathbf{x}^\dagger\right\|_2 = 0.$$

In order to show this, as it was done in [11], we need to assume that $A \in \mathbb{R}^{m\times n}$ is of full rank, that $m \geq n$, and that $\mathbf{b} \in \mathcal{R}(A)$. This ensures that the least square solution of

$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|_2$$

is unique and coincides with $\mathbf{x}^\dagger$. If A is not of full rank, one may consider the slightly modified problem

$$\min_{\mathbf{x}} \left\|\widetilde{A}\mathbf{x} - \widetilde{\mathbf{b}}\right\|_2,$$

with

$$\widetilde{A} = \begin{bmatrix} A \\ \theta I \end{bmatrix} \quad \text{and} \quad \widetilde{A} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix},$$

where $I \in \mathbb{R}^{n\times n}$ denotes the identity matrix and $\theta \in \mathbb{R}^+$ is a small number. This usually does not change the numerical results, especially if θ is smaller than the machine epsilon, and we do not consider this modification in our computations.

We are now in a position to show our main result

Theorem 1. *Let $A \in \mathbb{R}^{m\times n}$ be of full column rank with $m \geq n$ and let $\mathbf{b} \in \mathcal{R}(A)$. Let $\left\{\mathbf{b}^{\delta_j}\right\}_{j\in\mathbb{N}} \subset \mathbb{R}^m$ be a sequence of vectors such that*

$$\left\|\mathbf{b}^{\delta_j} - \mathbf{b}\right\|_2 \leq \delta_j,$$

with $\delta_j \searrow 0$ as $j \to \infty$. Denote by $\mathbf{x}_j^*$ the output of Algorithm 3 with input data $\mathbf{b}^{\delta_j}$, then there exists a converging subsequence $\left\{\mathbf{x}_{j_k}^*\right\}_{j_k\in\mathbb{N}}$ such that

$$\limsup_{j_k\to\infty} \left\|\mathbf{x}_{j_k}^* - \mathbf{x}^\dagger\right\|_2 = 0,$$

where $\mathbf{x}^\dagger = A^\dagger\mathbf{b}$.

Proof. The proof is similar to the one in [11]. Let us first observe that, by construction, for all $j \in \mathbb{N}$ we have

$$\left\|A\mathbf{x}_j^* - \mathbf{b}^{\delta_j}\right\|_2 = \tau\delta_j.$$

Using the fact that

$$\left\|A\mathbf{x}_j^* - \mathbf{b}^{\delta_j}\right\|_2 \geq \left|\left\|A\mathbf{x}_j^*\right\|_2 - \left\|\mathbf{b}^{\delta_j}\right\|_2\right| \geq \left\|A\mathbf{x}_j^*\right\|_2 - \left\|\mathbf{b}^{\delta_j}\right\|_2,$$

we have

$$\begin{aligned}
\left\|A\mathbf{x}_j^*\right\|_2 &\leq \left\|A\mathbf{x}_j^* - \mathbf{b}^{\delta_j}\right\|_2 + \left\|\mathbf{b}^{\delta_j}\right\|_2 \\
&= \tau\delta_j + \left\|\mathbf{b}^{\delta_j} - \mathbf{b} + \mathbf{b}\right\|_2 \\
&\leq \tau\delta_j + \left\|\mathbf{b}^{\delta_j} - \mathbf{b}\right\|_2 + \|\mathbf{b}\|_2 \\
&\leq (1+\tau)\delta_j + \|\mathbf{b}\|_2
\end{aligned}$$

Let σ_n denote the smallest singular value of A. Since we assumed that A is of full column rank and $m \geq n$ we have that $\sigma_n > 0$, therefore, $\|A\mathbf{x}\|_2 \geq \sigma_n \|\mathbf{x}\|_2$ for all $\mathbf{x} \in \mathbb{R}^n$. Combining these two inequalities and the fact that δ_j is monotonically decreasing we obtain

$$\left\|\mathbf{x}_j^*\right\|_2 \leq \frac{(1+\tau)\delta_1 + \|\mathbf{b}\|_2}{\sigma_n}, \quad \forall j \in \mathbb{N}.$$

i.e., that the sequence $\left\{\mathbf{x}_j^*\right\}_{j\in\mathbb{N}}$ is uniformly bounded. Since $\left\{\mathbf{x}_j^*\right\}_{j\in\mathbb{N}}$ is uniformly bounded, it admits a converging subsequence $\left\{\mathbf{x}_{j_k}^*\right\}_{j_k\in\mathbb{N}}$. We obtain

$$\begin{aligned}
0 &\leq \limsup_{j_k\to\infty} \left\|\mathbf{x}_{j_k}^* - \mathbf{x}^\dagger\right\|_2 \leq \limsup_{j_k\to\infty} \frac{1}{\sigma_n} \left\|A\mathbf{x}_{j_k}^* - A\mathbf{x}^\dagger\right\|_2 \\
&= \limsup_{j_k\to\infty} \frac{1}{\sigma_n} \left\|A\mathbf{x}_{j_k}^* - \mathbf{b}\right\|_2 = \limsup_{j_k\to\infty} \frac{1}{\sigma_n} \left\|A\mathbf{x}_{j_k}^* - \mathbf{b}^{\delta_{j_k}} + \mathbf{b}^{\delta_{j_k}} - \mathbf{b}\right\|_2 \\
&\leq \limsup_{j_k\to\infty} \frac{1}{\sigma_n} \left\{\left\|A\mathbf{x}_{j_k}^* - \mathbf{b}^{\delta_{j_k}}\right\|_2 + \left\|\mathbf{b}^{\delta_{j_k}} - \mathbf{b}\right\|_2\right\} \\
&\leq \limsup_{j_k\to\infty} \frac{1}{\sigma_n}(1+\tau)\delta_{j_k} = 0
\end{aligned}$$

which concludes the proof. □

6. Numerical examples

In this section, we show some numerical examples obtained using the fractional graph Laplacian. We compare our results with the methods proposed in [11] and [9]. The $\ell^2 - \ell^q$ TV algorithm proposed in [11] is Algorithm 1 where the operator L is the gradient defined in (9). The $\ell^2 - \ell^q$ algorithm with graph Laplacian proposed in [9] is Algorithm 2 up to the modification of the restarting strategy described in Section 2, which reduces the computational time without deteriorating the quality of the restored image. The initial approximation $\hat{\mathbf{x}}$ used to construct the graph in Algorithm 2 is computed solving the $\ell^2 - \ell^2$ TV method, i.e., the minimization problem (3) with $q = 2$ and L being the gradient defined in (9). This solution $\hat{\mathbf{x}}$ can be computed by the fast Fourier transform for image deblurring or by a generalized Krylov subspace method [28], where the regularization parameter μ is estimated by the generalized cross validation (GCV).

As it was shown in [11] the quality of the reconstructions increases as q approaches 0. However, a too small value of q may lead to numerical instability. Therefore, we set $q = 0.1$.

Finally, for our fractional graph Laplacian $\ell^2 - \ell^q$ method we use Algorithm 3 according to the following strategy:

1. compute an initial reconstruction $\hat{\mathbf{x}}$ by solving the $\ell^2 - \ell^2$ TV problem;

2. construct the graph associated to $\hat{\mathbf{x}}$ and compute a better reconstruction $\mathbf{x}^*$ by Algorithm 2 ($\ell^2 - \ell^q$ graph Laplacian);
3. construct the graph associated to $\mathbf{x}^*$ and compute a new reconstruction by Algorithm 3 ($\ell^2 - \ell^q$ fractional graph Laplacian).

Vast numerical experience suggests that this combination of the three algorithms reliably products extremely accurate approximate solutions. Moreover, thanks to the projection in the GKS and Krylov subspace as well as the restart technique employed, the computational cost of the procedure is reasonable and the computations can be easily performed on any machine.

We compare the methods above in terms of accuracy using the Relative Restoration Error (RRE) computed as

$$\mathrm{RRE}(\mathbf{x}) = \frac{\left\|\mathbf{x} - \mathbf{x}^\dagger\right\|_2}{\left\|\mathbf{x}^\dagger\right\|_2},$$

and the Peak Signal to Noise Ration (PSNR) defined as

$$\mathrm{PSNR}(\mathbf{x}) = 20 \log_{10}\left(\frac{nM}{\left\|\mathbf{x} - \mathbf{x}^\dagger\right\|_2}\right),$$

where M denotes the maximum value achievable by $\mathbf{x}^\dagger$. Moreover, we consider the Structure SIMilarity index (SSIM), introduced in [31]. The definition of the SSIM is extremely involved, here we simply recall that this statistical index measures how structurally similar two images are, in particular, the higher the SSIM the more similar the images are, and its highest achievable value is 1.

We will consider two different applications: a deblurring problem and a Computer Tomography (CT) reconstruction.

We set the restarting parameter $r = 30$, the window size $R = 5$, the variance in the weight function of the graph $\sigma = 10^{-3}$, and the smoothing parameter $\varepsilon = 10^{-1}$. We would like to stress that the results obtained by the algorithm is not very sensitive to the choice of this parameters.

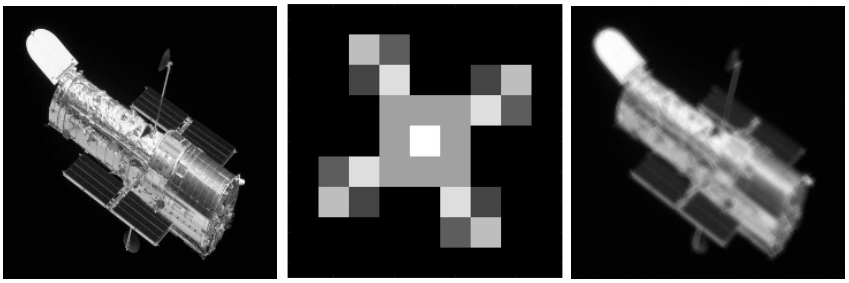
We stop the iterations of all considered algorithms as soon as either

$$\frac{\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\|_2}{\left\|\mathbf{x}^{(k)}\right\|_2} \leq 10^{-4}$$

or the maximum number of iterations, i.e., $K = 500$, is reached.

6.1. Example 1

In our first example we consider a 256×256 pixels image of the Hubble space telescope. We blur it with a PSF of dimension 9×9 pixels and we add white Gaussian noise such that $\|\boldsymbol{\eta}\|_2 = 0.01 \|\mathbf{b}\|_2$. We say that, in the case, the noise level is 1%. We crop the image to simulate realistic data and boundary effects; see, e.g., [20]. Since the image has a black background we impose zero boundary conditions. Figure 1 shows the true image, the PSF, and the observed picture.



(a) (b) (c)

Figure 1: Example 1. (a) true image (228×238 pixels), (b) PSF (9×9 pixels), (c) blurred image corrupted by 1% of white Gaussian Noise (238×238 pixels).

In Figure 2 we report the reconstructions obtained using the considered methods. In Figure 2(a) we show the approximate solution obtained with the $\ell^2 - \ell^2$ model with TV regularization, while, in Figure 2(b), we report the reconstruction obtained with the $\ell^2 - \ell^q$ model, like before, with L being the TV operator (Algorithm 1). In Figures 2(c) and 2(d) we consider the standard graph Laplacian (Algorithm 2), that is $\alpha = 1$, and the fractional graph Laplacian with $\alpha = 1.5$ in the $\ell^2 - \ell^q$ setting (Algorithm 3), respectively. We also report the results obtained with $\alpha = 1.6$ and $\alpha = 2$ in Figures 2(e) and 2(f). Although visual inspection seems to suggest that there are no differences with the case $\alpha = 1.5$, the computed reconstructions achieve a higher value of the PSNR and the SSIM respectively. The considered statistics for the numerical results obtained with the four different methods are reported in Table 1.

In Figure 3 we show a blow-up of the lower-right part of the image. We observe that, if one can properly choose the fractional exponent α, then it is possible to accurately reconstruct the details of the image. To this aim, we compute the value of the fractional exponent α using the residual whiteness principle described in Section 5.2.

Figures 4(a) and 4(b) depict the behavior of PSNR and SSIM, respectively, for different values of the fractional parameter α. These two measures are used to evaluate the accuracy of the computed solutions, we recall that high values of these quantities correspond to more accurate reconstructions. We highlight with a red asterisk the value of the fractional exponent computed using the residual whiteness principle. We observe that this criterion provides a fairly accurate estimate of the optimal value, i.e., the one that maximizes the two functionals. Moreover, we can observe that, in this case, the choice of a fractional exponent different from 1 improves the quality of the results with respect to $\alpha = 1$. This means that the fractional graph Laplacian can be a better regularizer than the standard graph Laplacian, provided that one can estimate the fractional exponent properly. Finally, Figures 4(c) and 4(d) report the RRE and the residual whiteness function for different values of α. As before, the red asterisk denotes the minimizer of the functional (12) and is also close to the value that minimizes the RRE.

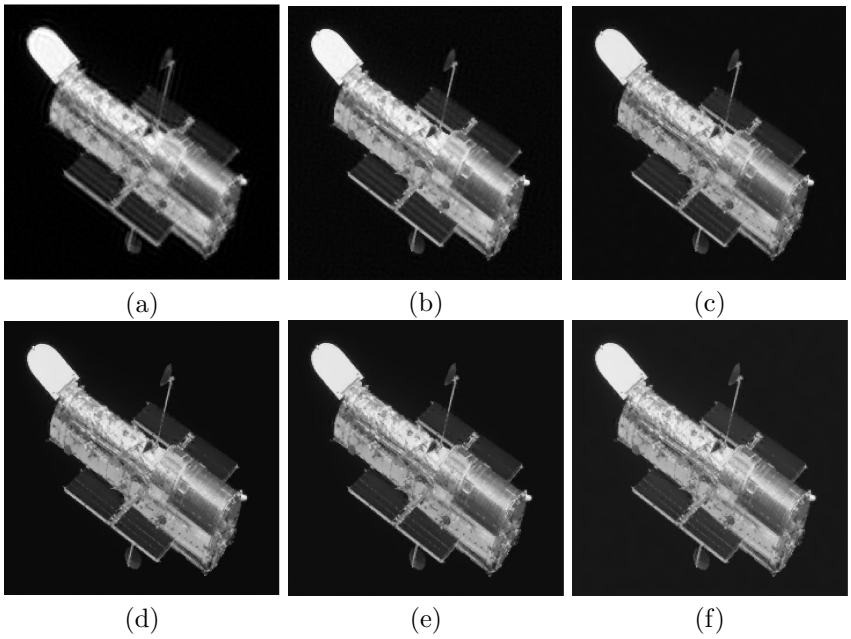


Figure 2: Example 1. Reconstructions obtained with four different methods. (a) $\ell^2 - \ell^2$ with TV, (b) $\ell^2 - \ell^q$ with TV, (c) $\ell^2 - \ell^q$ with the graph Laplacian by Algorithm 1. (d)-(e)-(f) $\ell^2 - \ell^q$ with the fractional graph Laplacian by Algorithm 3 with fractional exponent $\alpha = 1.5,\ 1.6,\ 2$ respectively.

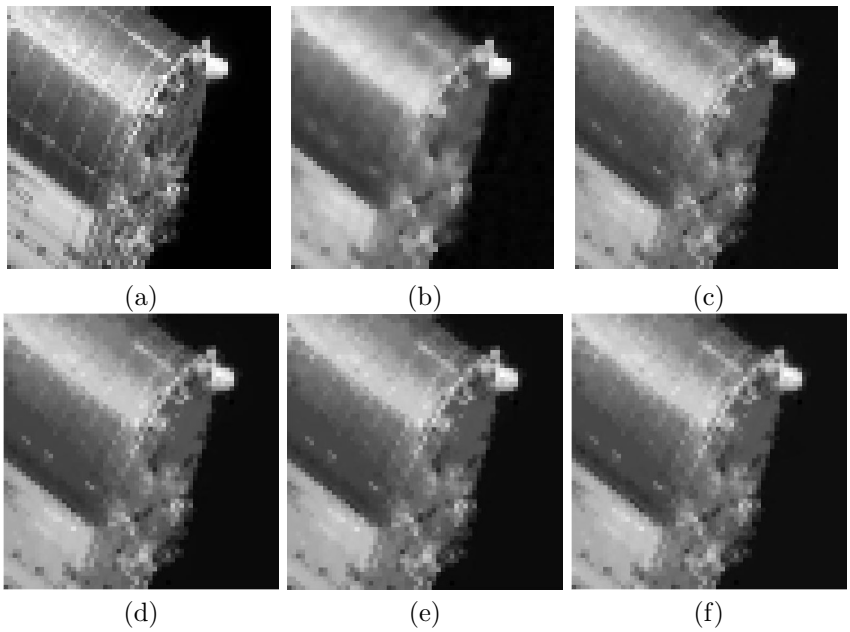


Figure 3: Example 1. Blow-up on the frontal part of the Hubble: (a) original image, (b) $\ell^2 - \ell^q$ TV, (c) $\ell^2 - \ell^q$ graph Laplacian, (d)-(e)-(f) $\ell^2 - \ell^q$ fractional graph Laplacian with $\alpha = 1.5$, 1.6, 2, respectively.

6.2. Example 2

The second example is a CT reconstruction problem. We construct this example using the IRtools toolbox [18]. The original image is the Shepp-Logan Phantom of dimension 128×128 pixels, it is shined with 181 parallel beams at 180 equispaced angles between 0 and π. Moreover, we perturb the sinogram $\mathbf{b} \in \mathbb{R}^{181\times 180}$ with white Gaussian noise η with noise level 2%, i.e., $\|\eta\|_2 = 0.02\|b\|_2$. The real image and the observed sinogram are shown in Figure 5.

In Figure 6 we compare the different reconstructions obtained with the same methods we used for the deblurring example. We follow the same strategy described before for the deblurring problem and we set the parameters of Algorithm 3 as in the previous example. Despite the fact that the initial approximation provided by the $\ell^2 - \ell^2$ method, reported in Figure 6(a), is not very accurate, since its SSIM is only 0.6378, our proposal was able to provide an almost optimal reconstruction, reported in Figure 6(d), that that achieves a SSIM value of 0.9929. All the artifacts present in the first approximation have been completely removed in the final reconstruction. The numerical results obtained for the four different cases can be found in Table 1.

In Figure 7 we reported the PSNR, SSIM, RRE, and values of the whiteness residual $\mathcal{W}$ for different values of α. We observe that the quality of the reconstruction strongly depends on the choice of the fractional exponent. A red asterisk highlights the value of α that minimizes $\mathcal{W}$ in (12). Once again, we

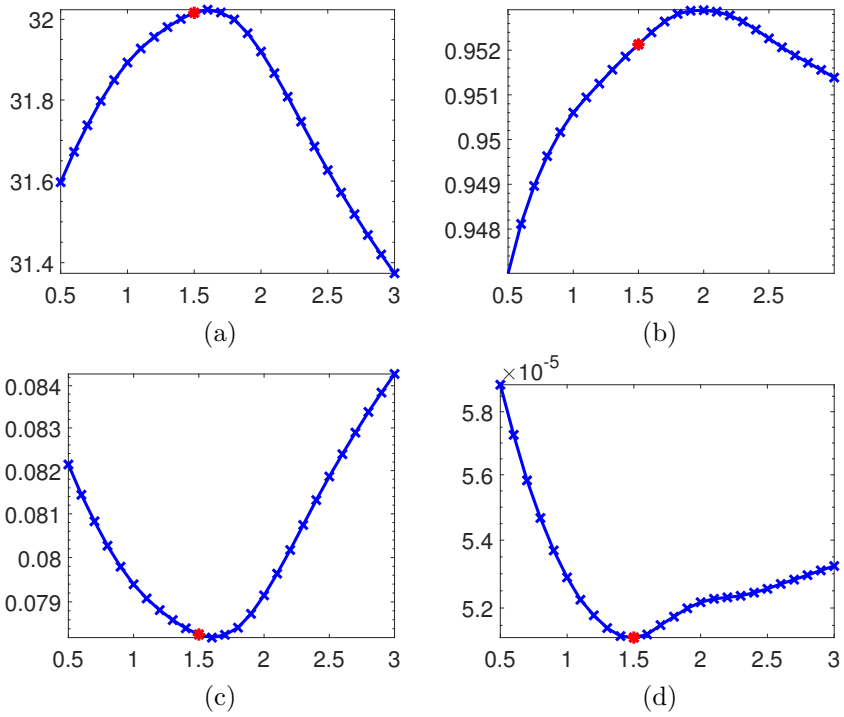


Figure 4: Example 1. Behavior of (a) PSNR, (b) SSIM, (c) RRE, and (d) residual whiteness $\mathcal{W}$, for different values of the fractional exponent α.

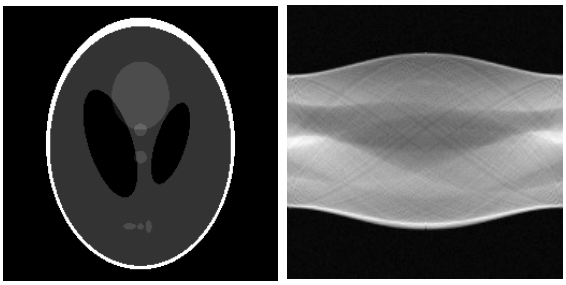


Figure 5: Example 2. (a) true image (128×128 pixels, (b) observed sinogram corrupted with 2% of white Gaussian noise (181×180 pixels).

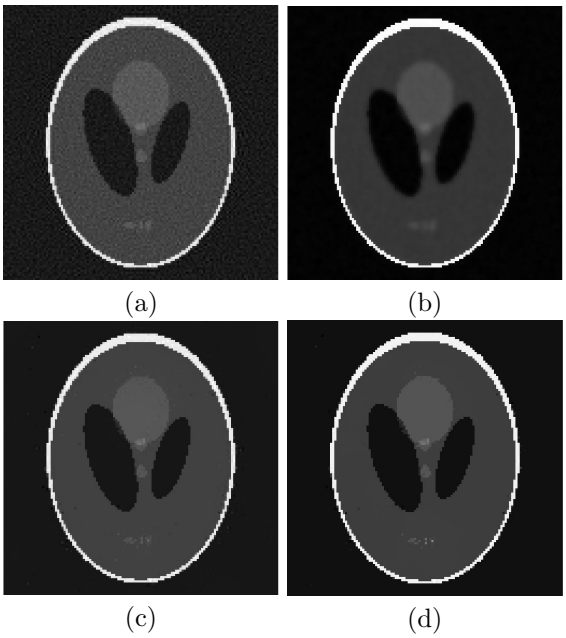


Figure 6: Example 2. Reconstructions obtained with four considered methods. (a) $\ell^2 - \ell^2$ with TV, (b) $\ell^2 - \ell^q$ with TV, (c) $\ell^2 - \ell^q$ with graph Laplacian, and (d) $\ell^2 - \ell^q$ with fractional graph Laplacian with $\alpha = 0.5$.

note that this strategy provides extremely accurate values for α.

7. Conclusions

In this paper, we developed an algorithm for the solution of some ill-posed image reconstruction problems. The main novelty of our proposal is the use of a fractional exponent in the graph Laplacian used in the regularization term to diffuse the information on the graph. This introduction improves the quality of the computed reconstructions. Moreover, the algorithm is completely automatic and, provided that a fairly accurate estimate of the norm of the noise that corrupts the data is available, does not require the tuning of any parameter. Finally, we showed that the proposed method is a regularization method and analyzed its theoretical properties. Selected numerical examples showed that the introduction of this fractional exponent can improve the quality of the computed solutions. The application of this method to non-linear problems and to a different type of noise, e.g., impulse noise or Cauchy noise, will be a matter of future research.

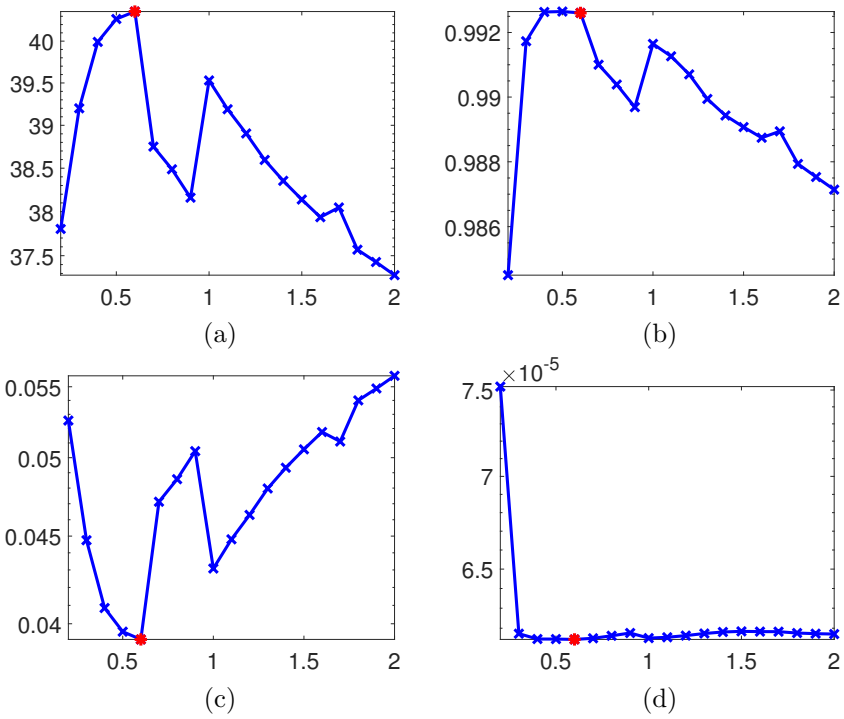


Figure 7: Example 2. Behavior of (a) PSNR, (b) SSIM, (c) RRE, and (d) residual whiteness $\mathcal{W}$, for different values of the fractional exponent α.

Table 1: Quality of the computed reconstructions for the considered methods.

Example	Method	RRE	SSIM	PSNR
Example 1	$\ell^2 - \ell^2$ TV	0.1318	0.8695	27.49
	$\ell^2 - \ell^q$ TV (Alg. 1)	0.0933	0.9114	30.49
	$\ell^2 - \ell^q$ $\alpha = 1$ (Alg. 2)	0.0857	0.9445	31.23
	$\ell^2 - \ell^q$ $\alpha = 1.5$ (Alg. 3)	0.0783	0.9521	32.02
	$\ell^2 - \ell^q$ $\alpha = 1.6$	0.0782	0.9524	32.03
	$\ell^2 - \ell^q$ $\alpha = 2$	0.0791	0.9529	31.92
Example 2	$\ell^2 - \ell^2$ TV	0.1468	0.6378	28.88
	$\ell^2 - \ell^q$ TV (Alg. 1)	0.0539	0.9593	37.58
	$\ell^2 - \ell^q$ $\alpha = 1$ (Alg. 2)	0.0560	0.9878	37.24
	$\ell^2 - \ell^q$ $\alpha = 0.5$ (Alg. 3)	0.0396	0.9926	40.27

Acknowledgments

The authors are members of the GNCS group of INdAM that partially financed their research with the INdAM-GNCS 2022 Project "Metodi e modelli di regolarizzazione per problemi mal posti di grandi dimensioni" (CUP_E55F22000270001) and with INdAM-GNCS 2023 Project "Tecniche numeriche per lo studio dei problemi inversi e l'analisi delle reti complesse" (CUP_E53C22001930001). A. B. is partially supported by Fondazione di Sardegna, Progetto biennale bando 2021, "Computational Methods and Networks in Civil Engineering (COMANCHE)".

References

[1] Antil, H., Bartels, S.: Spectral approximation of fractional pdes in image processing and phase field modeling. Computational Methods in Applied Mathematics **17**(4), 661–678 (2017)

[2] Bakushinskii, A.: Remarks on choosing a regularization parameter using the quasi-optimality and ratio criterion. USSR Computational Mathematics and Mathematical Physics **24**(4), 181–182 (1984)

[3] Benzi, M., Bertaccini, D., Durastante, F., Simunec, I.: Non-local network dynamics via fractional graph laplacians. Journal of Complex Networks **8**(3), cnaa017 (2020)

[4] Bertero, M., Boccacci, P.: Introduction to inverse problems in imaging. CRC press (1998)

[5] Bianchi, D., Buccini, A., Donatelli, M., Randazzo, E.: Graph Laplacian for image deblurring. Electronic Transactions on Numerical Analysis **55**, 169–186 (2022)

[6] Bianchi, D., Donatelli, M., Durastante, F., Mazza, M.: Compatibility, embedding and regularization of non-local random walks on graphs. Journal of Mathematical Analysis and Applications **511**(1), 126,020 (2022)

[7] Buccini, A., De la Cruz Cabrera, O., Donatelli, M., Martinelli, A., Reichel, L.: Large-scale regression with non-convex loss and penalty. Applied Numerical Mathematics **157**, 590–601 (2020)

[8] Buccini, A., De la Cruz Cabrera, O., Koukouvinos, C., Mitrouli, M., Reichel, L.: Variable selection in saturated and supersaturated designs via minimization. Communications in Statistics - Simulation and Computation, in press (2021)

[9] Buccini, A., Donatelli, M.: Graph Laplacian in $\ell^2 - \ell^q$ regularization for image reconstruction. In: 2021 21st International Conference on Computational Science and Its Applications (ICCSA), pp. 29–38. IEEE (2021)

[10] Buccini, A., Pragliola, M., Reichel, L., Sgallari, F.: A comparison of parameter choice rules for $\ell^p - \ell^q$ minimization. ANNALI DELL'UNIVERSITA'DI FERRARA **68**, 441–463 (2022)

[11] Buccini, A., Reichel, L.: An ℓ^2-ℓ^q regularization method for large discrete ill-posed problems. Journal of Scientific Computing **78**, 1526–1549 (2019)

[12] Buccini, A., Reichel, L.: Limited memory restarted $\ell^p - \ell^q$ minimization methods using generalized Krylov subspaces. Advances in Computational Mathematics **49**, 26 (2023)

[13] Chan, R.H., Liang, H.X.: Half-quadratic algorithm for ℓ_p-ℓ_q problems with applications to TV-ℓ_1 image restoration and compressive sensing. In: Efficient Algorithms for Global Optimization Methods in Computer Vision, pp. 78–103. Springer, New York (2014)

[14] Daniel, J.W., Gragg, W.B., Kaufman, L., Stewart, G.W.: Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization. Mathematics of Computation **30**(136), 772–795 (1976)

[15] Eldén, L.: Algorithms for the regularization of ill-conditioned least squares problems. BIT Numerical Mathematics **17**(2), 134–145 (1977)

[16] Engl, H.W., Hanke, M., Neubauer, A.: Regularization of Inverse Problems. Kluwer, Dordrecht (1996)

[17] Estatico, C., Gratton, S., Lenti, F., Titley-Peloquin, D.: A conjugate gradient like method for p-norm minimization in functional spaces. Numerische Mathematik **137**(4), 895–922 (2017)

[18] Gazzola, S., Hansen, P.C., Nagy, J.G.: IR Tools: a MATLAB package of iterative regularization methods and large-scale test problems. Numerical Algorithms **81**(3), 773–811 (2019)

[19] Golub, G.H., Van Loan, C.F.: Matrix Computations, 4th. edition. Johns Hopkins University Press, Baltimore (2013)

[20] Hansen, P.C., Nagy, J.G., O'Leary, D.P.: Deblurring Images: Matrices, Spectra, and Filtering. SIAM, Philadelphia (2006)

[21] Huang, G., Lanza, A., Morigi, S., Reichel, L., Sgallari, F.: Majorization-minimization generalized Krylov subspace methods for ℓ_p-ℓ_q optimization applied to image restoration. BIT Numerical Mathematics **57**, 351–378 (2017)

[22] Krishnan, D., Fergus, R.: Fast image deconvolution using hyper-laplacian priors. In: Advances in neural information processing systems, pp. 1033–1041 (2009)

[23] Lanza, A., Morigi, S., Reichel, L., Sgallari, F.: A generalized Krylov subspace method for ℓ_p-ℓ_q minimization. SIAM Journal on Scientific Computing **37**, S30–S50 (2015)

[24] Lanza, A., Pragliola, M., Sgallari, F.: Residual whiteness principle for parameter-free image restoration. Electronic Transactions on Numerical Analysis **53**, 329–351 (2020)

[25] Li, F., Ng, M.K.: Image colorization by using graph bi-laplacian. Advances in Computational Mathematics **45**(3), 1521–1549 (2019)

[26] Meyer, F.G., Shen, X.: Perturbation of the eigenvectors of the graph laplacian: Application to image denoising. Applied and Computational Harmonic Analysis **36**(2), 326–334 (2014)

[27] Pang, J., Cheung, G.: Graph laplacian regularization for image denoising: Analysis in the continuous domain. IEEE Transactions on Image Processing **26**(4), 1770–1785 (2017)

[28] Reichel, L., Sgallari, F., Ye, Q.: Tikhonov regularization based on generalized Krylov subspace methods. Applied Numerical Mathematics **62**(9), 1215–1228 (2012)

[29] Susnjara, A., Perraudin, N., Kressner, D., Vandergheynst, P.: Accelerated filtering on graphs using Lanczos method. arXiv preprint arXiv:1509.04537 (2015)

[30] Voss, H.: An Arnoldi method for nonlinear eigenvalue problems. BIT numerical mathematics **44**(2), 387–401 (2004)

[31] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004)

[32] Yağan, A.C., Özgen, M.T.: A spectral graph wiener filter in graph fourier domain for improved image denoising. In: 2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp. 450–454. IEEE (2016)

[33] Yang, Q., Chen, D., Zhao, T., Chen, Y.: Fractional calculus in image processing: a review. Fractional Calculus and Applied Analysis **19**(5), 1222–1249 (2016)