# Graph Laplacian in $\ell^2 - \ell^q$ regularization for image reconstruction

Alessandro Buccini
*Department of Mathematics and Computer Science*
*University of Cagliari*
Cagliari, Italy
alessandro.buccini@unica.it

Marco Donatelli
*Department of Science and High Technology*
*University of Insubria*
Como, Italy
marco.donatelli@uninsubria.it

***Abstract*—The use of the Laplacian of a properly constructed graph for denoising images has attracted a lot of attention in the last years. Recently, a way to use this instrument for image deblurring has been proposed. Even though the previously proposed method was able to provide extremely accurate reconstructions, it had several limitations, namely it was only applicable when periodic boundary conditions were employed, the regularization parameter had to be hand-tuned, and only convex regularization terms were allowed. In this paper, we propose two automatic methods that do not need the tuning of any parameter and that can be used for different imaging problems. Moreover, thanks to the projection into properly constructed subspaces of fairly small dimension, the proposed algorithms can be used for solving large scale problems.**

***Index Terms*—Image deblurring, $\ell^2 - \ell^q$ regularization, graph Laplacian.**

## I. Introduction

In this paper we consider image reconstruction problems of the form

$$A\mathbf{x} + \boldsymbol{\eta} = \mathbf{b}^\delta, \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$ is a severely ill-conditioned matrix, i.e., a matrix whose singular values decay to zero rapidly and with no significant gap, $\boldsymbol{\eta} \in \mathbb{R}^m$ collects all the unknown measurement and discretization errors and is often referred to as *noise* vector, $\mathbf{b}^\delta \in \mathbb{R}^m$ is the measured data, and $\mathbf{x} \in \mathbb{R}^n$ is the discretization of the unknown image we wish to recover. Among the possible applications we mention image deblurring and computer tomography; see, e.g., [8], [24].

Since $A$ is ill conditioned and the data $\mathbf{b}^\delta$ are corrupted by noise, the naive solution of (1) $A^\dagger\mathbf{b}^\delta$, where $A^\dagger$ denotes the Moore-Penrose pseudo-inverse of $A$, does not provide a meaningful approximation of $\mathbf{x}^\dagger = A^\dagger\mathbf{b}$, with $\mathbf{b} = \mathbf{b}^\delta - \boldsymbol{\eta}$, i.e., the unavailable noise-free data. To compute an accurate approximation of $\mathbf{x}^\dagger$ one needs to resort to regularization methods. A variational regolarization method that has attracted a lot of attention in the recent years is $\ell^p - \ell^q$ minimization

$$\arg\min_{\mathbf{x}} \frac{1}{p}\left\|A\mathbf{x} - \mathbf{b}^\delta\right\|_p^p + \frac{\mu}{q}\left\|L\mathbf{x}\right\|_q^q, \tag{2}$$

The authors are members of the GNCS-INdAM group. A.B. research is partially supported by the Regione Autonoma della Sardegna research project "Algorithms and Models for Imaging Science [AMIS]" (RASSR57257, intervento finanziato con risorse FSC 2014-2020 - Patto per lo Sviluppo della Regione Sardegna).

where $0 < p, q \leq 2$, $L \in \mathbb{R}^{s\times n}$, $\mu > 0$, and for $\mathbf{z} \in \mathbb{R}^n$

$$\|\mathbf{z}\|_p^p = \sum_{i=1}^n |z_i|^p.$$

A Bayesian justification of this model was derived in [3] and was further analyzed in [2], [4], [6], [7], [12], [16], [25], [29].

We briefly discuss the role of $p$ and $q$. The value of $p$ should be informed by the statistical properties of the noise that affects the data, for Gaussian noise the choice $p = 2$ usually provides accurate results, while for impulsive noise $p < 1$ is more appropriate. In this paper, we will assume that the noise that corrupts the data is Gaussian, therefore, in the following we will fix $p = 2$. On the other hand $q$ is determined by the a priori knowledge available on the solution $\mathbf{x}^\dagger$. In particular, the choice $q \leq 1$ induces sparsity on $L\mathbf{x}$, however, the obtained problem may be non-convex and it is non-smooth. Nevertheless, when $L\mathbf{x}$ is known to be sparse, imposing $q$ close to zero provides very accurate results and the approximation error usually decreases with $q$; see, e.g., [4] for a discussion. Possible choices of $L$ are discretizations of derivative operators and framelets.

The parameter $\mu > 0$, often referred to as *regularization parameter*, determines the sensitivity of the model to the presence of noise in $\mathbf{b}^\delta$. The choice of its value largely affects the quality of the computed reconstruction: a too large value may provide to oversmoothed solution, while a too small value may lead to severe propagation of noise. In the case of Gaussian noise and $p = 2$, one can use the discrepancy principle (DP) to determine the value of $\mu$; see, e.g., [4]. The DP prescribes that the parameter $\mu$ is chosen so that the approximate solution $\mathbf{x}^*$ satisfies

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

where $\tau > 1$ is a user-specified parameter and $\delta$ is such that $\|\boldsymbol{\eta}\|_2 \leq \delta$. The DP allows us to show theoretical properties of the proposed model, however, it requires additional knowledge on the problem. If it is not possible to estimate the norm of the noise or the noise is not Gaussian, the value of $\mu$ can be estimated using heuristic methods like Cross Validation (CV) or Generalized Cross Validation (GCV); see [6], [7].

In this paper, we wish to show how the use of the Laplacian of a properly constructed graph as a regularization operator can improve the accuracy of the computed solution in the $\ell^2-\ell^q$ regularization with respect to the classic choices for $L$. The graph Laplacian has attracted a lot of attention in recent years for image denoising; see, e.g., [27], [28], [30], [31], [33], [35], [37]. In [1] the authors showed that this choice produces extremely satisfactory results in $\ell^2-\ell^1$ regularization for image deblurring. However, in [1] the authors provided an algorithm only for periodic boundary conditions (bc's) and did not provide an automatic way to determine $\mu$. The main novelty of this paper with respect to [1] is that we allow an arbitrary choice of $0 < q \leq 2$, we can deal with more general imaging problems, and we provide automatic rules for choosing $\mu$ whether an estimate of $\|\boldsymbol{\eta}\|_2$ is available or not.

This paper is structured as follows: Section II presents an algorithm for the solution of (2) with $p = 2$, in Section III we describe how the Laplacian of a given graph is constructed. In Section IV we detail our algorithmic proposal and Section V collects some numerical examples. Finally, we draw some conclusions in Section VI.

## II. $\ell^2 - \ell^q$ MINIMIZATION

We briefly describe the approaches proposed in [25] to solve (2). In this paper we set $p = 2$, however, the algorithms proposed in [25] allow for a general $0 < p \leq 2$. We then describe the approaches in [5], [7] to determine the regularization parameter $\mu$.

A possible approach to solve (2) is to use a Majorization-Minimization (MM) method. However, for $q \leq 1$ the minimized functional in (2) is non-smooth. Therefore, we smooth the functional. Let $\varepsilon > 0$ be a fixed parameter and denote by

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

If $\varepsilon$ is small we can approximate $\|\mathbf{x}\|_q^q$ by

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

Note that the function on the right-hand side is everywhere differentiable, while the one on the left is not differentiable if one of the components of $\mathbf{x}$ vanishes. We substitute problem (2) by

$$\min_{\mathbf{x}} \frac{1}{2}\left\|A\mathbf{x}-\mathbf{b}^\delta\right\|_2^2 + \frac{\mu}{q}\sum_{i=1}^{s}\Phi_{q,\varepsilon}((L\mathbf{x})_i). \tag{3}$$

The MM algorithm constructs a sequence $\{\mathbf{x}^{(k)}\}_k$ that converges to a stationary point of the minimized functional in (3).

Denote by $\mathcal{J}_\varepsilon$ the functional minimized in (3) and let $\mathbf{x}^{(k)}$ be the current approximation of the solution of (3), the MM method first determines a quadratic functional $\mathcal{Q}(\mathbf{x},\mathbf{x}^{(k)})$ that majorizes $\mathcal{J}_\varepsilon$ everywhere and that it is tangent to it in $\mathbf{x}^{(k)}$. Then, the new iterate $\mathbf{x}^{(k+1)}$ is the minimizer of $\mathcal{Q}(\mathbf{x},\mathbf{x}^{(k)})$.

**Definition 1.** *Consider an everywhere differentiable function $\mathcal{J}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$. We say that $\mathbf{x} \mapsto \mathcal{Q}(\mathbf{x},\mathbf{y}) \in \mathbb{R}$ is a quadratic tangent majorant of $\mathcal{J}$ in $\mathbf{y}$ if*

1) *$\mathbf{x} \mapsto \mathcal{Q}(\mathbf{x},\mathbf{y})$ is quadratic;*
2) *$\mathcal{Q}(\mathbf{x},\mathbf{y}) \geq \mathcal{J}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^n$;*
3) *$\mathcal{Q}(\mathbf{y},\mathbf{y}) = \mathcal{J}(\mathbf{y})$;*
4) *$\nabla\mathcal{Q}(\mathbf{y},\mathbf{y}) = \nabla\mathcal{J}(\mathbf{y})$.*

In [25] two possible choices for $\mathcal{Q}(\mathbf{x},\mathbf{x}^{(k)})$ where proposed. One was denoted by $\mathcal{Q}^F(\mathbf{x},\mathbf{x}^{(k)})$ and it has a fixed aperture, while the other one has the widest aperture possible and was denoted by $\mathcal{Q}^A(\mathbf{x},\mathbf{x}^{(k)})$. We report here both approaches as we will use either of them depending on the choice rule that we wish to use to determine $\mu$. The term "fixed" aperture is due to the fact that, if $n = m = 1$, then the quadratic majorant is a parabola and its leading coefficient is fixed and does not depend on $\mathbf{x}^{(k)}$.

We first describe the fixed approach. Let

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

and

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

where all operations are meant element-wise. In [25], the authors showed that the quadratic functional

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

where $c$ is a constant that does not depend on $\mathbf{x}$, is a quadratic tangent majorant of $\mathcal{J}_\varepsilon$ in $\mathbf{x}^{(k)}$. The next iterate $\mathbf{x}^{(k+1)}$ can be obtained by

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}}\left\|\begin{bmatrix}A\\ \nu^{1/2}L\end{bmatrix}\mathbf{x}-\begin{bmatrix}\mathbf{b}^\delta\\ \nu^{1/2}\boldsymbol{\omega}_F^{(k)},\end{bmatrix}\right\|_2^2 \tag{4}$$

where $\nu = \mu\varepsilon^{q-2}$. An approximate solution of (4) can be searched 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 and $\hat{k} \ll n$ and assume that the columns of $V_k$ span the search subspace. Therefore, we are looking for a solution of the form

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

Substituting this expression of $\mathbf{x}^{(k+1)}$ in (4) we obtain

$$\mathbf{y}^{(k+1)} = \arg\min_{\mathbf{y}}\left\|\begin{bmatrix}AV_k\\ \nu^{1/2}LV_k\end{bmatrix}\mathbf{y}-\begin{bmatrix}\mathbf{b}^\delta\\ \nu^{1/2}\boldsymbol{\omega}_F^{(k)}\end{bmatrix}\right\|_2^2. \tag{5}$$

Introduce the compact QR factorizations

$$\begin{aligned}AV_k &= Q_AR_A \quad \text{with} \quad Q_A \in \mathbb{R}^{m\times\hat{k}} \quad R_A \in \mathbb{R}^{\hat{k}\times\hat{k}},\\ LV_k &= Q_LR_L \quad \text{with} \quad Q_L \in \mathbb{R}^{s\times\hat{k}} \quad R_L \in \mathbb{R}^{\hat{k}\times\hat{k}},\end{aligned} \tag{6}$$

where $Q_A$ and $Q_L$ have orthonormal columns and $R_A$ and $R_L$ are upper triangular. Note that we use $\hat{k}$ and not $k$ as, in

general, $\hat{k} \geq k$; see below. Plugging the decompositions (6) in (5) we obtain

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

Computing the residual of the normal equation associated with (4) we obtain

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

Following [36] we expand the search subspace by adding the normalized residual to $V_k$, i.e., let $\mathbf{v}_{\text{new}} = \mathbf{r}^{(k+1)} / \left\| \mathbf{r}^{(k+1)} \right\|_2$, then $V_{k+1} = [V_k, \mathbf{v}_{\text{new}}]$. Note that $\mathbf{v}_{\text{new}}$ is orthogonal to the space spanned by the columns of $V_k$.

We store the matrices

$$AV_{k+1} = [AV_k, A\mathbf{v}_{\text{new}}], \quad LV_{k+1} = [LV_k, L\mathbf{v}_{\text{new}}].$$

Following [13], the QR factorizations of these matrices are computed by updating the QR factorizations (6) according to

$$AV_{k+1} = [AV_k, A\mathbf{v}_{\text{new}}] = [Q_A, \tilde{\mathbf{q}}_A] \begin{bmatrix} R_A & \mathbf{r}_A \\ \mathbf{0}^T & \tau_A \end{bmatrix},$$

$$LV_{k+1} = [LV_k, L\mathbf{v}_{\text{new}}] = [Q_L, \tilde{\mathbf{q}}_L] \begin{bmatrix} R_L & \mathbf{r}_L \\ \mathbf{0}^T & \tau_L \end{bmatrix},$$

where

$$\begin{array}{rclrcl} \mathbf{r}_A &=& Q_A^T (A\mathbf{v}_{\text{new}}), & \mathbf{q}_A &=& A\mathbf{v}_{\text{new}} - Q_A \mathbf{r}_A, \\ \tau_A &=& \|\mathbf{q}_A\|_2, & \tilde{\mathbf{q}}_A &=& \mathbf{q}_A / \tau_A, \\ \mathbf{r}_L &=& Q_L^T (L\mathbf{v}_{\text{new}}), & \mathbf{q}_L &=& L\mathbf{v}_{\text{new}} - Q_L \mathbf{r}_L, \\ \tau_L &=& \|\mathbf{q}_L\|_2, & \tilde{\mathbf{q}}_L &=& \mathbf{q}_L / \tau_L. \end{array}$$

We now describe the adaptive approach. Let

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

and construct the diagonal matrix

$$W^{(k)} = \operatorname{diag}\left( \boldsymbol{\omega}_A^{(k)} \right).$$

The adaptive quadratic tangent majorant $\mathcal{Q}_A\left(\mathbf{x}, \mathbf{x}^{(k)}\right)$ is defined by

$$\mathcal{Q}_A(\mathbf{x}, \mathbf{x}^{(k)}) = \frac{1}{2} \left\| A\mathbf{x} - \mathbf{b}^\delta \right\|_2^2 + \frac{\mu}{2} \left\| \left( W^{(k)} \right)^{1/2} L\mathbf{x} \right\|_2^2 + c.$$

Similarly to the fixed aperture case, we determine

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} \mathcal{Q}_A \left( \mathbf{x}, \mathbf{x}^{(k)} \right).$$

Imposing that

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

and introducing the compact QR factorizations

$$AV_k = Q_A R_A \text{ with } Q_A \in \mathbb{R}^{m \times \hat{k}} \; R_A \in \mathbb{R}^{\hat{k} \times \hat{k}},$$

$$\left( W^{(k)} \right)^{1/2} LV_k = Q_L R_L \text{ with } Q_L \in \mathbb{R}^{s \times \hat{k}} \; R_L \in \mathbb{R}^{\hat{k} \times \hat{k}},$$

we obtain

$$\mathbf{y}^{(k+1)} = \arg\min_{\mathbf{y}} \left\| \begin{bmatrix} R_A \\ \mu^{1/2} R_L \end{bmatrix} \mathbf{y} - \begin{bmatrix} Q_A^T \mathbf{b}^\delta \\ \mathbf{0} \end{bmatrix} \right\|_2^2.$$

Like in the fixed aperture case, we can enlarge the space by adding the normalized residual of the normal equations

$$\mathbf{r}^{(k+1)} = A^T (A V_k \mathbf{y}^{(k+1)} - \mathbf{b}^\delta) + \mu L^T W^{(k)} L V_k \mathbf{y}^{(k+1)}.$$

We store the matrix $AV_k$ and update its QR factorization as described above. For computational purposes we store $LV_k$ as well, however, we cannot update the QR factorization of $W^{(k)} LV_k$, therefore, we need to recompute it from scratch at each iteration.

### *A. Determination of the regularization parameter*

We now describe the approaches proposed in [5], [7]. They are based on the DP and the GCV. Both methods are nonstationary, as the parameter $\mu$ is updated at each iteration to satisfy either the DP or the GCV. We briefly describe both methods.

The DP method proposed in [4] uses the fixed aperture approach. We modify the MM method so that at iteration $k$ the algorithm solves the least squares problem

$$\mathbf{y}^{(k+1)} = \arg\min_{\mathbf{y}} \left\| R_A \mathbf{y} - Q_A^T \mathbf{b}^\delta \right\|_2^2 + \nu_k \left\| R_L \mathbf{y} - Q_L^T \boldsymbol{\omega}_F^{(k)} \right\|_2^2,$$

where $\nu_k$ is such that

$$\left\| R_A \mathbf{y}^{(k+1)} - Q_A^T \mathbf{b}^\delta \right\|_2^2 = \tau \delta,$$

with $\delta \geq \|\boldsymbol{\eta}\|_2$ and $\tau > 1$ is a fixed parameter. Note that the computation of $\nu_k$ can be cheaply performed using Newton method and the Generalized Singular Value Decomposition (GSVD) of the matrix pair $\{R_A, R_L\}$; see, e.g., [15], [22].

In [7], the authors proposed to update $\mu_k$ using the GCV method. We recall here the definition of the GCV; see [21]. We consider the adaptive approach, as at each iteration the least squares problem we solve has the same formulation of Tikhonov method in general form, namely

$$\mathbf{y}^{(k+1)} = \arg\min_{\mathbf{y}} \left\| R_A \mathbf{y} - Q_A^T \mathbf{b}^\delta \right\|_2^2 + \mu \left\| R_L \mathbf{y} \right\|_2^2. \tag{7}$$

Denote by $A_\mu^\dagger$ the linear operator such that

$$\mathbf{y}^{(k+1)} = A_\mu^\dagger \widehat{\mathbf{b}}^\delta,$$

where $\widehat{\mathbf{b}}^\delta = Q_A^T \mathbf{b}^\delta$. Note that the operator $A_\mu^\dagger$ can be written in function of the GSVD of the matrix pair $\{R_A, R_L\}$.

The GCV paramter $\mu_k$ can be obtained as

$$\mu_k = \arg\min_{\mu} G(\mu),$$

where $G(\mu)$ is defined by

$$G(\mu) = \frac{\left\| \widehat{\mathbf{b}}^\delta - R_A A_\mu^\dagger \mathbf{b}^\delta \right\|_2^2}{\operatorname{trace}(I - R_A A_\mu^\dagger)^2}. \tag{8}$$

Once the GSVD of $\{R_A, R_L\}$ has been computed, the evaluation of $G$ can be performed extremely cheaply. We refer to the following sections and to Fenu et al. [18], [19] for a discussion on the GCV in Krylov subspaces.

In the following we will refer to $\ell^2 - \ell^q$–DP for the first algorithm, i.e., the one where the parameter is computed using the DP, and to $\ell^2 - \ell^q$–GCV for the second algorithm, i.e., the one where the parameter is computed by the GCV.

## III. Graph Laplacian

We now recall the construction of the Laplacian of a given graph.

An unweighted graph is a pair $\Gamma = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set containing all the vertexes of the graph and $\mathcal{E}$ is a set containing all the edges. The set $\mathcal{E}$ can be seen as a subset of $\mathcal{V} \times \mathcal{V}$. We say that a graph is undirected if $(i,j) \in \mathcal{E}$ implies that $(j,i) \in \mathcal{E}$, otherwise we say that the graph is directed. On the other hand, a weighted graph can still be represented by a pair $\Gamma = (\mathcal{V}, \mathcal{E})$, but this time to each edge in $\mathcal{E}$ is associated a unique positive number, denoted by $\omega_{(i,j)}$, that is often-referred to as the weight of the edge. In this case we can write $\mathcal{E} \subset \mathcal{V} \times \mathcal{V} \times \mathbb{R}$. A the weighted graph can be directed or undirected. In all cases, it is possible to represent the graph by means of the adjacency matrix. Let $n = |\mathcal{V}|$, where $|\mathcal{A}|$ is the cardinality of the set $\mathcal{A}$, i.e., $n$ is the number of nodes in $\Gamma$. Denote by $G \in \mathbb{R}^{n \times n}$ the adjacency matrix of the graph $\Gamma$ defined as follows

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

where we have set $\omega_{(i,j)} \equiv 1$ for unweighted graphs. Note that a graph is undirected if and only if $G$ is symmetric; see, e.g., [17] for a review on graphs and complex networks.

We now describe how to construct the graph Laplacian associated to a given graph $\Gamma$; see, e.g., [**?**], [32] for a discussion on graph Laplacian. There are many definitions for the graph Laplacian. For our purposes we define this operator as follows. Let $G \in \mathbb{R}^{n \times n}$ be the adjacency matrix of $\Gamma$ and denote by $D$ the diagonal matrix

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

We define the graph Laplacian $L_\Gamma$ as

$$L_\Gamma = \frac{D - G}{\|G\|_F},$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Note that, if $G$ is symmetric, $L_\Gamma$ is symmetric as well. Moreover,

$$\mathcal{N}(L_\Gamma) \supseteq \operatorname{span}\{\mathbf{1}\},$$

where $\mathcal{N}(M)$ denotes the null space of the matrix $M$ and $\mathbf{1}$ is the constant vector.

We now describe how to construct a proper graph Laplacian for solving (1).

Let $\widehat{\mathbf{x}}$ be a fairly accurate approximation of $\mathbf{x}^\dagger$. Since $\widehat{\mathbf{x}} \in \mathbb{R}^n$ is the discretization of an image we can reshape it into matrix form $\widehat{X} \in \mathbb{R}^{d_1 \times d_2}$ with $n = d_1 d_2$. To simplify the notation we assume that the image is square and, therefore, $d^2 = n$. Let $\Gamma$ denote the weighted and undirected graph whose nodes are the pixels of $\widehat{X}$. Let $R \in \mathbb{N}$ and $\mathbb{R} \ni \sigma > 0$ be two fixed numbers. We connect two pixels $X_{i_1,j_1}$ and $X_{i_2,j_2}$ if

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

where $\|\mathbf{x}\|_\infty = \max\{|x_i|,\, i = 1, \ldots, n\}$ for $\mathbf{x} \in \mathbb{R}^n$. The weight of the edge is computed by

$$\omega_{(i_1,j_1),(i_2,j_2)} = \omega_{(i_2,j_2),(i_1,j_1)} = \mathrm{e}^{-(X_{i_1,j_1} - X_{i_2,j_2})^2/\sigma}.$$

Summarizing the adjacency matrix $G$ is defined by

$$G_{\mathbf{i}_1,\mathbf{i}_2} = \begin{cases} \mathrm{e}^{-(X_{i_1,j_1} - X_{i_2,j_2})^2/\sigma} & \text{if } 0 < \|\mathbf{i}_1 - \mathbf{i}_2\| \leq R, \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathbf{i}_1 = \begin{bmatrix} i_1, & j_1 \end{bmatrix}^T$ and $\mathbf{i}_2 = \begin{bmatrix} i_2, & j_2 \end{bmatrix}^T$.

It was shown in [1] that the Laplacian related to this graph, if $\widehat{\mathbf{x}}$ is a fairly accurate approximation of $\mathbf{x}^\dagger$ is a good regularization operator for $\ell^2 - \ell^1$ image deblurring.

## IV. Our Proposal

We now detail our algorithmic proposals. Our methods are structured as follows

1) Construct an initial approximation $\widehat{\mathbf{x}}$ of $\mathbf{x}^\dagger$;
2) Construct the graph Laplacian related to $\widehat{\mathbf{x}}$ as described in the previous section;
3) Obtain a regularized solution of (1) using either $\ell^2 - \ell^q$–DP or $\ell^2 - \ell^q$–GCV.

We now discuss how we construct $\widehat{\mathbf{x}}$. We wish to use Tikhonov in general form, i.e.,

$$\mathbf{x}_\alpha = \arg\min_{\mathbf{x}} \left\| A\mathbf{x} - \mathbf{b}^\delta \right\|_2^2 + \alpha \left\| L\mathbf{x} \right\|_2^2, \tag{9}$$

where $\alpha > 0$ is the regularization parameter and $L$ is a differential operator. Since we do not assume any structure for $A$ and $L$ and $n$ is usually very large, we need to resort to Krylov methods to solve (9). We use Golub-Kahan bidiagonalization, $\ell \ll n$ steps of this method with initial vector $\mathbf{b}^\delta$ provide the following factorizations

$$AV_\ell = U_{\ell+1} B_{\ell+1,\ell} \quad \text{and} \quad A^T U_\ell = V_\ell B_{\ell,\ell}^T,$$

where $V_\ell \in \mathbb{R}^{n \times \ell}$ and $U_{\ell+1} \in \mathbb{R}^{m \times (\ell+1)}$ have orthonormal columns, $B_{\ell+1,\ell} \in \mathbb{R}^{(\ell+1) \times \ell}$ is lower bidiagonal. The matrix $U_\ell \in \mathbb{R}^{m \times \ell}$ is composed by the first $\ell$ columns of $U_{\ell+1}$, $B_{\ell,\ell} \in \mathbb{R}^{\ell \times \ell}$ is the $\ell \times \ell$ leading block of $B_{\ell+1,\ell}$, and the first column of $U_{\ell+1}$ is $\mathbf{b}^\delta / \left\| \mathbf{b}^\delta \right\|_2$; see [22]. The columns of $V_\ell$ span the Krylov subspace $\mathcal{K}_\ell(A^T A, A^T \mathbf{b}^\delta)$ defined by

$$\mathcal{K}_\ell(A^T A, A^T \mathbf{b}^\delta) = \operatorname{span}\{A^T \mathbf{b}^\delta, (A^T A) A^T \mathbf{b}^\delta, \\ \ldots, (A^T A)^{\ell-1} A^T \mathbf{b}^\delta\}.$$

We look for an approximate solution of (9) in this space, i.e.,

$$\mathbf{x}_\alpha = V_\ell \mathbf{y}_\alpha.$$

Since $\ell \ll n$ we can explicitly compute $LV_\ell$ and its compact QR factorization

$$LV_\ell = QR \text{ with } Q \in \mathbb{R}^{s \times \ell} \text{ and } R \in \mathbb{R}^{\ell \times \ell},$$

where $Q$ has orthogonal columns and $R$ is upper triangular. We can reformulate (9) in $\mathcal{K}_\ell(A^TA, A^T\mathbf{b}^\delta)$ as

$$\begin{aligned}\mathbf{y}_\alpha &= \arg\min_{\mathbf{y}} \left\|AV_\ell\mathbf{y} - \mathbf{b}^\delta\right\|_2^2 + \alpha\left\|LV_\ell\mathbf{y}\right\|_2^2 \\ &= \arg\min_{\mathbf{y}} \left\|U_{\ell+1}B_{\ell+1,\ell}\mathbf{y} - \mathbf{b}^\delta\right\|_2^2 + \alpha\left\|QR\mathbf{y}\right\|_2^2 \\ &= \arg\min_{\mathbf{y}} \left\|B_{\ell+1,\ell}\mathbf{y} - \left\|\mathbf{b}^\delta\right\|_2\mathbf{e}_1\right\|_2^2 + \alpha\left\|R\mathbf{y}\right\|_2^2 \\ &= \arg\min_{\mathbf{y}} \left\|\begin{bmatrix}B_{\ell+1,\ell}\\ \sqrt{\alpha}R\end{bmatrix}\mathbf{y} - \begin{bmatrix}\left\|\mathbf{b}^\delta\right\|_2\mathbf{e}_1\\ \mathbf{0}\end{bmatrix}\right\|_2^2,\end{aligned} \tag{10}$$

where $\mathbf{e}_1$ is the first column of the $(\ell+1)\times(\ell+1)$ identity matrix. This problem can be solved efficiently thanks to the structure of $\begin{bmatrix}B_{\ell+1,\ell}\\ \sqrt{\alpha}R\end{bmatrix}$; see [14]. In our experiments we use a discretization of the gradient as $L$, i.e.,

$$L = \begin{bmatrix}L_1\otimes I\\ I\otimes L_1,\end{bmatrix} \in \mathbb{R}^{2n\times n} \tag{11}$$

where $\otimes$ denotes the Kronecker product, $I\in\mathbb{R}^{d\times d}$ is the identity matrix, and

$$L_1 = \begin{bmatrix}-1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1\\ 1 & & & -1\end{bmatrix} \in \mathbb{R}^{d\times d}.$$

To determine $\alpha$ we use the GCV. We briefly detail the procedure.

Consider the GSVD of the pair $\{B_{\ell+1,\ell}, R\}$

$$\begin{cases} B_{\ell+1,\ell} = \widehat{U}\Sigma_B X^T,\\ R = \widehat{V}\Sigma_R X^T,\end{cases} \tag{12}$$

where $\widehat{U}\in\mathbb{R}^{(\ell+1)\times(\ell+1)}$ and $\widehat{U}\in\mathbb{R}^{\ell\times\ell}$ have orthonormal columns, $\Sigma_B\in\mathbb{R}^{(\ell+1)\times\ell}$ and $\Sigma_R\in\mathbb{R}^{\ell\times\ell}$ are diagonal matrices, and $X\in\mathbb{R}^{\ell\times\ell}$ is a nonsingular matrix. Plugging the factorizations (12) into (10) we get that $\mathbf{y}_\alpha$ can be computed by

$$\mathbf{y}_\alpha = X^{-T}(\Sigma_B^T\Sigma_B + \alpha\Sigma_R^T\Sigma_R)^{-1}\Sigma_B^T\widehat{U}^T\left\|\mathbf{b}^\delta\right\|_2\mathbf{e}_1.$$

Although this formulation is not more convenient than (10) for the computation of $\mathbf{y}_\alpha$ it useful for computing $G(\alpha)$ in (8). In fact, observe that

$$\begin{aligned}r_\alpha &= \left\|B_{\ell+1,\ell}\mathbf{y}_\alpha - \left\|\mathbf{b}^\delta\right\|\mathbf{e}_1\right\| \\ &= \left\|\left(\Sigma_B(\Sigma_B^T\Sigma_B + \alpha\Sigma_R^T\Sigma_R)^{-1}\Sigma_B^T\widehat{U}^T - I\right)\left\|\mathbf{b}^\delta\right\|\mathbf{e}_1\right\|\end{aligned}$$

and that

$$\begin{aligned}t_\alpha &= \operatorname{trace}(I - B_{\ell+1,\ell}(B_{\ell+1,\ell}^TB_{\ell+1,\ell} + \alpha R^TR)^{-1}B_{\ell+1,\ell}^T) \\ &= \operatorname{trace}(I - \Sigma_B(\Sigma_B^T\Sigma_B + \alpha\Sigma_R^T\Sigma_R)^{-1}\Sigma_B^T),\end{aligned}$$

where we denoted by $r_\alpha^2$ and $t_\alpha^2$ the numerator and the denominator of the expression of $G$ in (8), respectively. Therefore, once the GSVD (12) has been computed, the computation of $G(\alpha) = r_\alpha^2/t_\alpha^2$ is extremely inexpensive. Thus, we can easily compute the minimum of $G$ and obtain $\alpha_{\mathrm{GCV}}$, i.e.,

$$\alpha_{\mathrm{GCV}} = \arg\min_\alpha G(\alpha).$$

We then define $\widehat{\mathbf{x}}$ as

$$\widehat{\mathbf{x}} = V_\ell\mathbf{y}_{\alpha_{\mathrm{GCV}}}.$$

Once we have computed $\widehat{\mathbf{x}}$, we construct the graph Laplacian related to this image as described in Section III and we denote it by $L_\Gamma$. We then use either $\ell^2-\ell^q$–DP or $\ell^2-\ell^q$–GCV to tackle the minimization problem

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

We summarize the computations in Algorithms 1 and 2.

## V. Numerical examples

We now provide some numerical experiments to show the performances of the proposed methods. We consider both image deblurring and computer tomography. Image deblurring is a linear inverse problem of the form

$$\int_{\mathbb{R}^2} k(s,t,u,v)f(u,v)dudv = g(s,t), \tag{13}$$

where $g$ denotes the blurred image, $f$ the unknown sharp image, and $k$ is an integral kernel, possibly smooth. Since $k$ has compact support, problem (13) is ill-posed; see, e.g., [15], [23]. Once we discretize over a finite domain the problem, we obtain a linear system of equations of the form (1). During the discretization process the problem is considered only on a limited domain, the so-called Field Of View (FOV). However, points outside the FOV have an effect inside it, therefore they need to be recovered as well. Therefore, in general, $m\le n$, where $m$ is the number of pixels inside the FOV and $n = m + p$, where $p$ is the number of pixels outside the FOV that influence the blurred image inside the FOV. To avoid an underdetermined system we make assumptions on the boundary conditions (bc's). Using these bc's, i.e., by assuming an algebraic relation between the point inside the FOV and the ones outside, we obtain that $n = m$, see, e.g., [24], [34]. If the discretization is obtained using a collocation method and $\mathbf{x}$ and $\mathbf{b}^\delta$ are ordered in a lexicographic way, the matrix $A$ has the following structure

$$A = T + E + R,$$

where $T$ is a block Toeplitz matrix with Toeplitz blocks, $E$ has small norm, and $R$ is of low rank. The matrices $E$ and $R$ collect the assumptions on the bc's; see [24], [34] for more details on image deblurring and bc's. We note here that, thanks to the structure of $A$, regardless of the bc's it is always possible to perform a matrix-vector product with $A$ cheaply.

In computer tomography, the data are the Radon transform of the attenuation coefficients of some scanned object; see, e.g., [8] for details on computerized tomography. We consider parallel beam tomography, where $J$ parallel X-ray beams are shined through an object at different angles $\theta_k$ with $k =$

**Algorithm 1:** Graph Laplacian $\ell^2 - \ell^q$–DP

Let $A \in \mathbb{R}^{m\times n}$, $\mathbf{b}^\delta \in \mathbb{R}^m$, $\ell > 0$, $R > 0$, $\sigma > 0$;
Construct $L$ as defined in (11);
Run $\ell$ steps of Golub-Kahan bidiagonalization obtaining $V_\ell \in \mathbb{R}^{n\times \ell}$ and $B_{\ell+1,\ell} \in \mathbb{R}^{\ell+1\times \ell}$;
Compute the compact QR factorization $QR = LV_\ell$;
Compute the GSVD of the pair $\{B_{\ell+1,\ell}, R\}$

$$\begin{cases} B_{\ell+1,\ell} = \widehat{U}\Sigma_B X^T, \\ R = \widehat{V}\Sigma_R X^T \end{cases};$$

$\alpha_{\mathrm{GCV}} = \arg\min_\alpha \frac{r_\alpha^2}{t_\alpha^2}$, where $r_\alpha = \left\| (\Sigma_B(\Sigma_B^T\Sigma_B + \alpha\Sigma_R^T\Sigma_R)^{-1}\Sigma_B^T\widehat{U}^T - I)\left\|\mathbf{b}^\delta\right\| \mathbf{e}_1 \right\|$ and $t_\alpha = \mathrm{trace}(I - \Sigma_B(\Sigma_B^T\Sigma_B + \alpha\Sigma_R^T\Sigma_R)^{-1}\Sigma_B^T)$;
$\widehat{\mathbf{y}} = (B_{\ell+1,\ell}^T B_{\ell+1,\ell} + \alpha_{\mathrm{GCV}}\Sigma_R^T\Sigma_R)^{-1} B_{\ell+1,\ell}^T \left\|\mathbf{b}^\delta\right\| \mathbf{e}_1$;
$\widehat{\mathbf{x}} = V_\ell \widehat{\mathbf{y}}$;
Construct $\Omega \in \mathbb{R}^{n\times n}$ as

$$\omega_{(\mathbf{i},\mathbf{j})} = \begin{cases} \mathrm{e}^{-(x_\mathbf{i}^* - x_\mathbf{j}^*)^2/\sigma} & \text{if } 0 < \|\mathbf{i}-\mathbf{j}\|_\infty \le R, \\ 0 & \text{otherwise,} \end{cases}$$

with $\mathbf{i}$ and $\mathbf{j}$ two-dimensional indexes;
$D = \mathrm{diag}\{\sum_{i=1}^n \Omega_{(i,j)}\}$;
$L_\Gamma = \frac{D-\Omega}{\|\Omega\|_F}$;
Generate the initial subspace basis: $V_0 \in \mathbb{R}^{n\times k_0}$ such that $V_0^T V_0 = I$;
Compute and store $AV_0$ and $L_\Gamma V_0$;
Compute the QR factorizations $AV_0 = Q_A R_A$ and $L_\Gamma V_0 = Q_L R_L$;
$\mathbf{y}^{(0)} = V_0^T \mathbf{x}^{(0)}$;
**for** $k = 0, 1, \ldots$ **do**
  $\mathbf{u}^{(k)} = L_\Gamma V_k \mathbf{y}^{(k)}$;
  $\boldsymbol{\omega}_F^{(k)} = \mathbf{u}^{(k)}\left(1 - \left(\frac{(\mathbf{u}^{(k)})^2+\varepsilon^2}{\varepsilon^2}\right)^{q/2-1}\right)$;
  $\nu^{(k)} = \mu^{(k)}\varepsilon^{q-2}$;
  where $\mu^{(k)}$ is such that $\left\|AV_k\mathbf{y}^{(k+1)} - \mathbf{b}^\delta\right\|_2 = \tau\delta$;
  $\mathbf{y}^{(k+1)} = (R_A^T R_A + \eta^{(k)} R_L^T R_L)^{-1}(R_A^T Q_A^T \mathbf{b}^\delta + \eta^{(k)} R_L^T Q_L^T \boldsymbol{\omega}_F^{(k)})$;
  $\mathbf{r} = A^T(AV_k\mathbf{y}^{(k+1)} - \boldsymbol{\omega}_F^{(k)}) + \nu^{(k)} L_\Gamma^T(L_\Gamma V_k \mathbf{y}^{(k+1)} - \boldsymbol{\omega}_F^{(k)})$;
  Reorthogonalize, if needed: $\mathbf{r} = \mathbf{r} - V_k V_k^T \mathbf{r}$;
  $\mathbf{v}_{\mathrm{new}} = \mathbf{r}/\|\mathbf{r}\|_2$; $V_{k+1} = [V_k, \mathbf{v}_{\mathrm{new}}]$;
  Update the QR factorizations: $AV_{k+1} = Q_A R_A$ and $L_\Gamma V_{k+1} = Q_L R_L$;
  $\mathbf{x}^{(k+1)} = V_{k+1}\mathbf{y}^{(k+1)}$;
  **if** $\left\|\mathbf{y}^{(k+1)} - \mathbf{y}^{(k)}\right\|_2 / \left\|\mathbf{y}^{(k+1)}\right\|_2 < tol$ **then**
    exit;

**Algorithm 2:** Graph Laplacian $\ell^2 - \ell^q$–GCV

Let $A \in \mathbb{R}^{m\times n}$, $\mathbf{b}^\delta \in \mathbb{R}^m$, $\ell > 0$, $R > 0$, $\sigma > 0$;
Construct $L$ as defined in (11);
Run $\ell$ steps of Golub-Kahan bidiagonalization obtaining $V_\ell \in \mathbb{R}^{n\times \ell}$ and $B_{\ell+1,\ell} \in \mathbb{R}^{\ell+1\times \ell}$;
Compute the compact QR factorization $QR = LV_\ell$;
Compute the GSVD of the pair $\{B_{\ell+1,\ell}, R\}$

$$\begin{cases} B_{\ell+1,\ell} = \widehat{U}\Sigma_B X^T, \\ R = \widehat{V}\Sigma_R X^T \end{cases};$$

$\alpha_{\mathrm{GCV}} = \arg\min_\alpha \frac{r_\alpha^2}{t_\alpha^2}$, where $r_\alpha = \left\| (\Sigma_B(\Sigma_B^T\Sigma_B + \alpha\Sigma_R^T\Sigma_R)^{-1}\Sigma_B^T\widehat{U}^T - I)\left\|\mathbf{b}^\delta\right\| \mathbf{e}_1 \right\|$ and $t_\alpha = \mathrm{trace}(I - \Sigma_B(\Sigma_B^T\Sigma_B + \alpha\Sigma_R^T\Sigma_R)^{-1}\Sigma_B^T)$;
$\widehat{\mathbf{y}} = (B_{\ell+1,\ell}^T B_{\ell+1,\ell} + \alpha_{\mathrm{GCV}}\Sigma_R^T\Sigma_R)^{-1} B_{\ell+1,\ell}^T \left\|\mathbf{b}^\delta\right\| \mathbf{e}_1$;
$\widehat{\mathbf{x}} = V_\ell \widehat{\mathbf{y}}$;
Construct $\Omega \in \mathbb{R}^{n\times n}$ as

$$\omega_{(\mathbf{i},\mathbf{j})} = \begin{cases} \mathrm{e}^{-(x_\mathbf{i}^* - x_\mathbf{j}^*)^2/\sigma} & \text{if } 0 < \|\mathbf{i}-\mathbf{j}\|_\infty \le R, \\ 0 & \text{otherwise,} \end{cases}$$

with $\mathbf{i}$ and $\mathbf{j}$ two-dimensional indexes;
$D = \mathrm{diag}\{\sum_{i=1}^n \Omega_{(i,j)}\}$;
$L_\Gamma = \frac{D-\Omega}{\|\Omega\|_F}$;
Generate the initial subspace basis: $V_0 \in \mathbb{R}^{n\times k_0}$ such that $V_0^T V_0 = I$;
Compute and store $AV_0$ and $L_\Gamma V_0$;
Compute the QR factorization $Q_A R_A = AV_0$;
$\mathbf{y}^{(0)} = V_0^T \mathbf{x}^{(0)}$;
**for** $k = 0, 1, \ldots$ **do**
  $\mathbf{u}^{(k)} = L_\Gamma V_k \mathbf{y}^{(k)}$;
  $\boldsymbol{\omega}_A^{(k)} := ((\mathbf{u}^{(k)})^2 + \varepsilon^2)^{q/2-1}$;
  $W^{(k)} = \mathrm{diag}(\boldsymbol{\omega}_A^{(k)})$;
  Compute the QR factorization $Q_L R_L = (W_{\mathrm{ref}}^{(k)})^{1/2} L V_k$;
  Compute the GSVD of the pair $\{R_A, R_L\}$

$$\begin{cases} R_A = \widehat{U}\Sigma_A X^T, \\ R_L = \widehat{V}\Sigma_L X^T \end{cases};$$

  $\mu^{(k)} = \arg\min_\mu \frac{r_\mu^2}{t_\mu^2}$, where $r_\mu = \left\| (\Sigma_A(\Sigma_A^T\Sigma_B + \mu\Sigma_L^T\Sigma_L)^{-1}\Sigma_A^T\widehat{U}^T - I)Q_A^T\mathbf{b}^\delta \right\|$ and $t_\mu = \mathrm{trace}(I - \Sigma_A(\Sigma_A^T\Sigma_A + \mu\Sigma_L^T\Sigma_L)^{-1}\Sigma_A^T)$;
  $\mathbf{y}^{(k+1)} = (R_A^T R_A + \mu^{(k)} R_L^T R_L)^{-1} R_A^T Q_A^T \mathbf{b}^\delta$;
  $\mathbf{r} = A^T(AV_k\mathbf{y}^{(k+1)} - \mathbf{b}^\delta) + \mu^{(k)} L_\Gamma^T W^{(k)} L_\Gamma V_k \mathbf{y}^{(k+1)}$;
  $\mathbf{v}_{\mathrm{new}} = \mathbf{r}/\|\mathbf{r}\|$;
  $V_{k+1} = [V_k, \mathbf{v}_{\mathrm{new}}]$;
  Compute and store $AV_k$ and $L_\Gamma V_k$;
  $\mathbf{x}^{(k+1)} = V_k\mathbf{y}^{(k+1)}$;
  Update the QR factorization $Q_A R_A = AV_k$;
  **if** $\left\|\mathbf{y}^{(k+1)} - \mathbf{y}^{(k)}\right\|_2 / \left\|\mathbf{y}^{(k+1)}\right\|_2 < tol$ **then**
    exit;

$1, 2, \ldots, K$. The datum $(b_{j,k})_{j=1,\ldots,J,\ k=1,\ldots,K}$, the so-called sinogram, is the line integral of the attenuation coefficient of the object scanned along the $j$-th beam at angle $\theta_k$.

We compare different choices for the regularization operator $L$. In particular we compare our proposal with the Total Variation (TV) operator, i.e., $L$ in (11), and framelet analysis operator. Following [9], [10], we define it as follows

**Definition 2.** *Let $W \in \mathbb{R}^{r\times n}$ with $1 \leq n \leq r$. The set of the rows of $W$ is a framelet system for $\mathbb{R}^n$ if $\forall \mathbf{x} \in \mathbb{R}^n$ it holds*

$$\|\mathbf{x}\|_2^2 = \sum_{j=1}^{r} (\mathbf{w}_j^T \mathbf{x})^2, \tag{14}$$

*where $\mathbf{w}_j \in \mathbb{R}^n$ denotes the $j$th row of the matrix $W$ (written as a column vector), i.e., $W = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_r]^T$. The matrix $W$ is referred to as an analysis operator and $W^T$ as a synthesis operator.*

Equation (14) is equivalent to the perfect reconstruction formula

$$\mathbf{x} = W^T W \mathbf{x}.$$

Thus, the matrix $W$ defines a tight frame if and only if $W^T W = I$. We remark that in general $WW^T \neq I$, unless $r = n$ and the framelets are orthonormal. We use the same tight frames as in [4], [9]–[11], [26]; they are determined by linear B-splines. Specifically, for problems in one space-dimension, they are formed by a low-pass filter $W_0 \in \mathbb{R}^{n\times n}$ and two high-pass filters $W_1 \in \mathbb{R}^{n\times n}$ and $W_2 \in \mathbb{R}^{n\times n}$. The corresponding masks are given by

$$u^{(0)} = \frac{1}{4}[1, 2, 1], \quad u^{(1)} = \frac{\sqrt{2}}{4}[1, 0, -1], \quad u^{(2)} = \frac{1}{4}[-1, 2, -1].$$

The analysis operator $W$ is determined by these masks and by imposing reflexive boundary conditions, which ensure that $W^T W = I$. Define the matrices

$$W_0 = \frac{1}{4}\begin{bmatrix} 3 & 1 & 0 & \ldots & 0 \\ 1 & 2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 2 & 1 \\ 0 & \ldots & 0 & 1 & 3 \end{bmatrix},$$

$$W_1 = \frac{\sqrt{2}}{4}\begin{bmatrix} -1 & 1 & 0 & \ldots & 0 \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ 0 & \ldots & 0 & -1 & 1 \end{bmatrix},$$

and

$$W_2 = \frac{1}{4}\begin{bmatrix} 1 & -1 & 0 & \ldots & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ 0 & \ldots & 0 & -1 & 1 \end{bmatrix}.$$

Then the operator $W$ is defined as

$$W = \begin{bmatrix} W_0 \\ W_1 \\ W_2 \end{bmatrix}.$$

We are concerned with image restoration problems in two space-dimensions. Therefore, we construct the two-dimensional framelet analysis operator by means of the tensor products

$$W_{ij} = W_i \otimes W_j, \quad i, j = 0, 1, 2.$$

The matrix $W_{00}$ is a low-pass filter; all the other matrices $W_{ij}$ contain at least one high-pass filter. The analysis operator is given by

$$W = \begin{bmatrix} W_{00} \\ W_{01} \\ \vdots \\ W_{22} \end{bmatrix}.$$

We set for all examples $q = 0.1$ and $\varepsilon = 1$. Eventhough the value for $\varepsilon$ may seem large, it is small if compared to the maximum value achievable by 8-bit gray-scale images, i.e., 255. We set the maximum number of iterations to 100 and the tollerance for the stopping criterion to $10^{-5}$. The size of the Krylov subspace needed to generate $\widehat{\mathbf{x}}$ is 50. Finally, we set $R = 10$ and $\sigma = 0.01$.

We compare the considered approaches in term of Relative Restoration Error (RRE) defined by

$$\text{RRE}(\mathbf{x}) = \frac{\|\mathbf{x} - \mathbf{x}_{\text{true}}\|_2}{\|\mathbf{x}_{\text{true}}\|_2}.$$

All computations were carried out in MATLAB R2020b with about 15 significant decimal digits running on a laptop computer with core CPU Intel(R) Core(TM)i7-8750H @2.20GHz with 16GB of RAM.

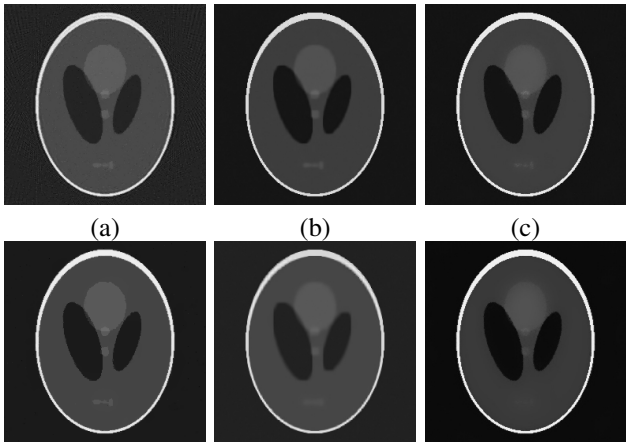
*a) Tomography:* We first consider a tomography example. The exact image is the Shep-Logan Phantom in Fig. 1(a) we shine the phantom with 362 parallel beams at 90 equispaced angles between 0 and $\pi$. To the obtained sinogram we add Gaussian noise such that $\|\boldsymbol{\eta}\|_2 = 0.01\,\|\mathbf{b}\|_2$ obtaining the sinogram in Fig. 1(b). The example was created using the IRtools toolbox [20].

We report the obtained RRE in Table I. We can observe that both the DP and GCV provide satisfactory results. However, the DP provides theoretical assurances, while the GCV does not. On the other hand, the GCV does not require any additional knowledge on the problem. We observe that, regardless of the choice of the parameter, the graph Laplacian is the best choice for regularization operator as it gives extremely accurate reconstruction for both the GCV and DP. These observations are confirmed by the visual inspection of the reconstructions in Fig. 2.

*b) Hubble:* In this example we consider an image deblurring problem. The Hubble image in Fig. 3(a) is blurred by the non-symmetric Point Spread Function (PSF) in Fig. 3(b). We add white Gaussian noise such that $\|\boldsymbol{\eta}\|_2 = 0.01\,\|\mathbf{b}\|_2$ and obtain the blurred and noisy image in Fig. 3(c). We cropped

(a) (b)

Fig. 1. Tomography test case: (a) True image ($256 \times 256$ pixels), (b) Sinogram ($362 \times 90$ pixels) with $1\%$ of white Gaussian noise.



(a) (b) (c)

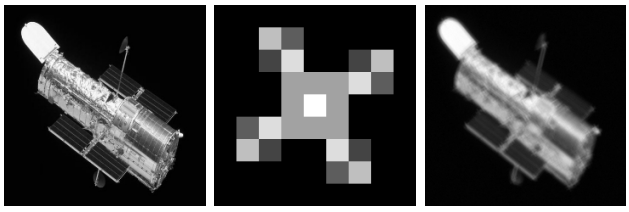
(d) (e) (f)

Fig. 2. Tomography test case reconstructions: (a) $\ell^2 - \ell^q$–DP Total Variation, (b) $\ell^2 - \ell^q$–DP Framelets, (c) $\ell^2 - \ell^q$–DP Graph Laplacian, (d) $\ell^2 - \ell^q$–GCV Total Variation, (e) $\ell^2 - \ell^q$–GCV Framelets, (f) $\ell^2 - \ell^q$–GCV Graph Laplacian.

the boundaries to simulate realistic data. Since the image is an astronomical image we imposed zero boundary conditions.

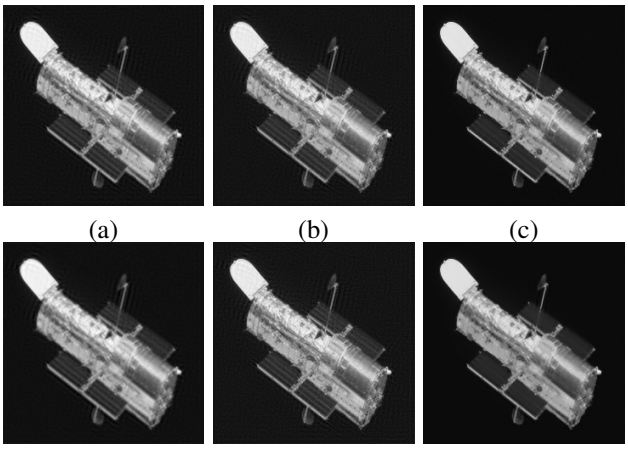
We report the obtained results in Table I. Like in the previous example both DP and GCV provide accurate reconstructions. The best reconstruction is obtained using the DP and the graph Laplacian. In this case for the GCV the graph Laplacian does not provide the best reconstruction among the three operators tested, however, we would like to stress that the GCV is an heuristic method. We report the computed solutions in Fig. 4.

*c) Cameraman:* Our final example is another image deblurring problem. The cameraman image in Fig. 5(a) is blurred by the motion PSF in Fig. 5(b). After adding white Gaussian noise such that $\|\boldsymbol{\eta}\|_2 = 0.02\,\|\mathbf{b}\|_2$ we obtain the blurred and



(a) (b) (b)

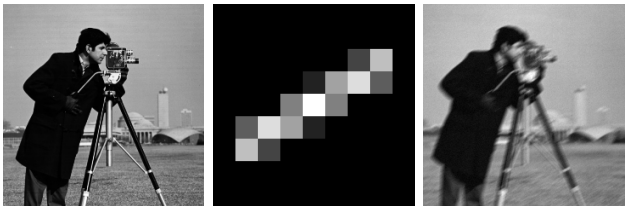
Fig. 3. Hubble test case: (a) True image ($238 \times 238$ pixels), (b) PSF ($9 \times 9$ pixels), (c) Blurred and noisy image ($1\%$ of white Gaussian noise).



(a) (b) (c)

(d) (e) (f)

Fig. 4. Hubble test case reconstructions: (a) $\ell^2 - \ell^q$–DP Total Variation, (b) $\ell^2 - \ell^q$–DP Framelets, (c) $\ell^2 - \ell^q$–DP Graph Laplacian, (d) $\ell^2 - \ell^q$–GCV Total Variation, (e) $\ell^2 - \ell^q$–GCV Framelets, (f) $\ell^2 - \ell^q$–GCV Graph Laplacian.



(a) (b) (b)

Fig. 5. Cameraman test case: (a) True image ($238 \times 238$ pixels), (b) PSF ($9 \times 9$ pixels), (c) Blurred and noisy image ($2\%$ of white Gaussian noise).

noisy image in Fig. 5(c). Like before the boundaries have been cropped to simulate realistic data. Since the image is generic we impose antireflexive boundary conditions; see [34].

From the RRE values reported in Table I we can observe that the graph Laplacian provides the most accurate reconstructions. Moreover, we can see that the difference between the other two regularization operators is extremely evident when the GCV is used to determine the regularization parameter. From the visual inspection of the computed solutions in Fig. 6 we can observe that the reconstructed images obtained with the graph Laplacian are less affected by the propagation of the noise and do not suffer from heavy boundary effects like some of the others.

## VI. Conclusions

In this paper we have shown that the Laplacian of a properly constructed graph can be used as regularization operator in the $\ell^2 - \ell^q$ method for imaging problems. We have provided two completely automatic procedures that provide very accurate reconstructions and can be applied to large scale problems thanks to the projection in Krylov subspaces of fairly small dimension. Matters of future research include the extension of these approaches to the case where the noise is non-Gaussian, e.g., salt-and-pepper or impulsive.

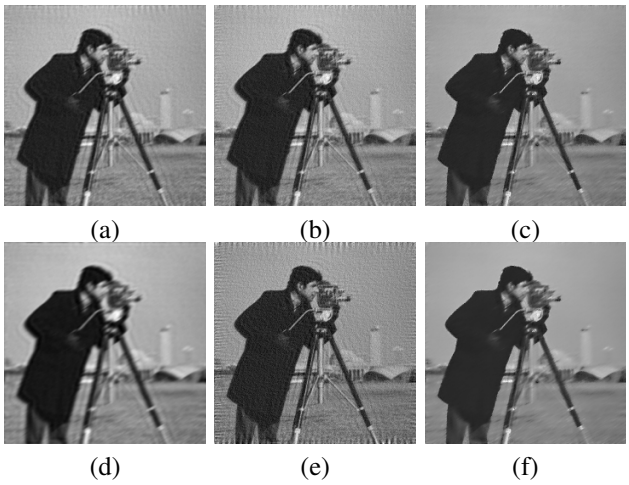


Fig. 6. Cameraman test case reconstructions: (a) $\ell^2 - \ell^q$–DP Total Variation, (b) $\ell^2 - \ell^q$–DP Framelets, (c) $\ell^2 - \ell^q$–DP Graph Laplacian, (d) $\ell^2 - \ell^q$–GCV Total Variation, (e) $\ell^2 - \ell^q$–GCV Framelets, (f) $\ell^2 - \ell^q$–GCV Graph Laplacian.

TABLE I
RRE FOR ALL THE CONSIDERED EXAMPLES AND CHOICES OF $L$ AND PARAMETER.

| Example | Parameter | Reg. Operator | RRE |
|---|---|---|---|
| Tomography | DP | TV | 0.17940 |
| | | Framelets | 0.075467 |
| | | Graph Laplacian | 0.058227 |
| | GCV | TV | 0.11106 |
| | | Framelets | 0.16390 |
| | | Graph Laplacian | 0.074012 |
| Hubble | DP | TV | 0.10594 |
| | | Framelets | 0.10354 |
| | | Graph Laplacian | 0.086858 |
| | GCV | TV | 0.12681 |
| | | Framelets | 0.10010 |
| | | Graph Laplacian | 0.10498 |
| Cameraman | DP | TV | 0.10791 |
| | | Framelets | 0.10904 |
| | | Graph Laplacian | 0.097266 |
| | GCV | TV | 0.13224 |
| | | Framelets | 0.13389 |
| | | Graph Laplacian | 0.10994 |

## REFERENCES


[1] D. Bianchi, A. Buccini, M. Donatelli, and E. Randazzo, "Graph laplacian for image deblurring," *arXiv preprint arXiv:2102.10327*, 2021.

[2] A. Buccini, M. Pasha, and L. Reichel, "Modulus-based iterative methods for constrained $\ell^p - \ell^q$ minimization," *Inverse Problems*, vol. 36, no. 8, p. 084001, 2020.

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

[4] A. Buccini and L. Reichel, "An $\ell^2 - \ell^q$ regularization method for large discrete ill-posed problems," *Journal of Scientific Computing*, vol. 78, no. 3, pp. 1526–1549, 2019.

[5] ——, "An $\ell^2$-$\ell^q$ regularization method for large discrete ill-posed problems," *Journal of Scientific Computing*, vol. 78, no. 3, pp. 1526–1549, 2019.

[6] ——, "An $\ell^p$-$\ell^q$ minimization method with cross-validation for the restoration of impulse noise contaminated images," *Journal of Computational and Applied Mathematics*, vol. 375, p. 112824, 2020.

[7] ——, "Generalized cross validation for $\ell^p$-$\ell^q$ minimization," *Numerical Algorithms*, In press.

[8] T. M. Buzug, "Computed tomography," in *Springer Handbook of Medical Technology*, R. Kramme, K.-P. Hoffmann, and R. S. Pozos, Eds. Springer, Berlin, 2011, pp. 311–342.

[9] J.-F. Cai, S. Osher, and Z. Shen, "Linearized Bregman iterations for frame-based image deblurring," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 226–252, 2009.

[10] ——, "Split Bregman methods and frame based image restoration," *Multiscale Modeling & Simulation*, vol. 8, no. 2, pp. 337–369, 2009.

[11] Y. Cai, M. Donatelli, D. Bianchi, and T.-Z. Huang, "Regularization preconditioners for frame-based image deblurring with reduced boundary artifacts," *SIAM Journal on Scientific Computing*, vol. 38, no. 1, pp. B164–B189, 2016.

[12] R. H. Chan and H.-X. Liang, "Half-quadratic algorithm for $\ell_p$-$\ell_q$ problems with applications to TV-$\ell_1$ image restoration and compressive sensing," in *Efficient Algorithms for Global Optimization Methods in Computer Vision*. Springer, 2014, pp. 78–103.

[13] J. W. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart, "Reorthogonalization and stable algorithms for updating the gram-schmidt qr factorization," *Mathematics of Computation*, vol. 30, no. 136, pp. 772–795, 1976.

[14] L. Eldén, "Algorithms for the regularization of ill-conditioned least squares problems," *BIT Numerical Mathematics*, vol. 17, no. 2, pp. 134–145, 1977.

[15] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of inverse problems*. Springer Science & Business Media, 1996, vol. 375.

[16] C. Estatico, S. Gratton, F. Lenti, and D. Titley-Peloquin, "A conjugate gradient like method for p-norm minimization in functional spaces," *Numerische Mathematik*, vol. 137, no. 4, pp. 895–922, 2017.

[17] E. Estrada, *The structure of complex networks: theory and applications*. Oxford University Press, 2012.

[18] C. Fenu, L. Reichel, and G. Rodriguez, "GCV for Tikhonov regularization via global Golub–Kahan decomposition," *Numerical Linear Algebra with Applications*, vol. 23, no. 3, pp. 467–484, 2016.

[19] C. Fenu, L. Reichel, G. Rodriguez, and H. Sadok, "GCV for Tikhonov regularization by partial SVD," *BIT Numerical Mathematics*, vol. 57, no. 4, pp. 1019–1039, 2017.

[20] S. Gazzola, P. C. Hansen, and J. G. Nagy, "IR Tools: a MATLAB package of iterative regularization methods and large-scale test problems," *Numerical Algorithms*, vol. 81, no. 3, pp. 773–811, 2019.

[21] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, pp. 215–223, 1979.

[22] G. H. Golub and C. F. Van Loan, *Matrix Computations, 4th. edition*. Johns Hopkins University Press, Baltimore, 2013.

[23] P. C. Hansen, *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*. SIAM, 1998.

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

[25] G. Huang, A. Lanza, S. Morigi, L. Reichel, and F. Sgallari, "Majorization-minimization generalized Krylov subspace methods for $\ell_p$-$\ell_q$ optimization applied to image restoration," *BIT Numerical Mathematics*, pp. 1–28, 2017.

[26] J. Huang, M. Donatelli, and R. H. Chan, "Nonstationary iterated thresholding algorithms for images deblurring," *Inverse Problems & Imaging*, vol. 7, pp. 717–736, 2013.

[27] A. Kheradmand and P. Milanfar, "Motion deblurring with graph laplacian regularization," in *Digital Photography XI*, vol. 9404. International Society for Optics and Photonics, 2015, p. 94040C.

[28] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-laplacian priors," in *Advances in neural information processing systems*, 2009, pp. 1033–1041.

[29] A. Lanza, S. Morigi, L. Reichel, and F. Sgallari, "A generalized Krylov subspace method for $\ell_p$-$\ell_q$ minimization," *SIAM Journal on Scientific Computing*, vol. 37, no. 5, pp. S30–S50, 2015.

[30] F. Li and M. K. Ng, "Image colorization by using graph bi-laplacian," *Advances in Computational Mathematics*, vol. 45, no. 3, pp. 1521–1549, 2019.

[31] F. G. Meyer and X. Shen, "Perturbation of the eigenvectors of the graph laplacian: Application to image denoising," *Applied and Computational Harmonic Analysis*, vol. 36, no. 2, pp. 326–334, 2014.

[32] M. Newman, *Networks*. Oxford university press, 2018.

[33] J. Pang and G. Cheung, "Graph laplacian regularization for image denoising: Analysis in the continuous domain," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1770–1785, 2017.

[34] S. Serra-Capizzano, "A note on antireflective boundary conditions and fast deblurring models," *SIAM Journal on Scientific Computing*, vol. 25, no. 4, pp. 1307–1325, 2004.

[35] A. Susnjara, N. Perraudin, D. Kressner, and P. Vandergheynst, "Accelerated filtering on graphs using Lanczos method," *arXiv preprint arXiv:1509.04537*, 2015.

[36] H. Voss, "An arnoldi method for nonlinear eigenvalue problems," *BIT numerical mathematics*, vol. 44, no. 2, pp. 387–401, 2004.

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