

An ℓ^2 - ℓ^q regularization method for large discrete ill-posed problems

Alessandro Buccini · Lothar Reichel

the date of receipt and acceptance should be inserted later

Abstract Ill-posed problems arise in many areas of science and engineering. Their solutions, if they exist, are very sensitive to perturbations in the data. Regularization aims to reduce this sensitivity. Typically, regularization methods replace the original problem by a minimization problem with a fidelity term and a regularization term. Recently, the use of a p -norm to measure the fidelity term, and a q -norm to measure the regularization term, has received considerable attention. The relative importance of these terms is determined by a regularization parameter. This paper discussed how the latter parameter can be determined with the aid of the discrepancy principle. We primarily focus on the situation when $p = 2$ and $0 < q \leq 2$, where we note that when $0 < q < 1$, the minimization problem may be non-convex.

Keywords ℓ^2 - ℓ^q minimization · ill-posed problem · iterative method

Mathematics Subject Classification (2000) 65F10 · 65R32 · 90C26

1 Introduction

We consider the computation of an approximate solution of minimization problems of the form

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|A\mathbf{x} - \mathbf{b}^\delta\|_2, \quad (1)$$

where $A \in \mathbb{R}^{m \times n}$ is a large matrix, whose singular values decrease to zero gradually with no significant gap, and the vector $\mathbf{b}^\delta \in \mathbb{R}^m$ represents measured error-contaminated data. We will sometimes refer to the error in \mathbf{b}^δ as “noise.” The norm $\|\cdot\|_2$ in (1) denotes the Euclidean norm. We also will comment on the use of other norms.

A. Buccini
Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA.
E-mail: abuccini@kent.edu

L. Reichel
Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA.
E-mail: reichel@math.kent.edu

Minimization problems of the kind (1) are commonly referred to as *discrete ill-posed problems*. They typically arise from the discretization of ill-posed problems, such as Fredholm integral equations of the first kind with a smooth kernel; see, e.g., [10, 14, 16] for discussions on ill-posed and discrete ill-posed problems.

Let $\mathbf{b} \in \mathbb{R}^m$ denote the (unknown) error-free vector associated with \mathbf{b}^δ . We will assume that \mathbf{b} is in the range of A and that a fairly sharp bound δ for the error in \mathbf{b}^δ is known, i.e.,

$$\|\mathbf{b} - \mathbf{b}^\delta\|_2 \leq \delta. \quad (2)$$

These assumptions will allow us to determine a regularization parameter with the aid of the discrepancy principle.

Since A is ill-conditioned and \mathbf{b}^δ is contaminated by error, the naïve solution, $A^\dagger \mathbf{b}^\delta$, of (1), where A^\dagger denotes the Moore-Penrose pseudoinverse of A , usually is not a meaningful approximation of the desired vector

$$\widehat{\mathbf{x}} := A^\dagger \mathbf{b} \quad (3)$$

due to severe amplification and propagation of the error in \mathbf{b}^δ . To achieve a more accurate approximation of $\widehat{\mathbf{x}}$, the original discrete ill-posed problem (1) is replaced by a nearby well-posed problem, whose solution is less sensitive to the error in \mathbf{b}^δ . This replacement is known as *regularization*.

A regularization technique that recently has received considerable attention is to replace (1) by an ℓ^p - ℓ^q minimization problem of the form

$$\mathbf{x}^* := \arg \min_{\mathbf{x}} \left\{ \frac{1}{p} \|\mathbf{Ax} - \mathbf{b}^\delta\|_p^p + \frac{\mu}{q} \|L\mathbf{x}\|_q^q \right\} =: \arg \min_{\mathbf{x}} \mathcal{J}(\mathbf{x}), \quad (4)$$

where the regularization matrix $L \in \mathbb{R}^{\ell \times n}$ is such that

$$\mathcal{N}(A) \cap \mathcal{N}(L) = \{\mathbf{0}\}. \quad (5)$$

Here $\mathcal{N}(M)$ denotes the null space of the matrix M and

$$\|\mathbf{z}\|_s := \left(\sum_{j=1}^r |z_j|^s \right)^{1/s}, \quad \mathbf{z} = [z_1, \dots, z_r]^T \in \mathbb{R}^r.$$

We will refer to $\|\cdot\|_s$ as the s -norm of \mathbf{z} also for $0 < s < 1$, even though the mapping $\mathbf{z} \rightarrow \|\mathbf{z}\|_s$ does not satisfy the triangle inequality and therefore is not a norm for these s -values.

The regularization parameter $\mu > 0$ in (4) balances the relative influence of the first term (the fidelity term) and the second term (the regularization term). The use of $0 < p, q \leq 2$ has received considerable attention; see, e.g., [2, 3, 6, 11, 19, 21, 23] and references therein. Note that if either $0 < p < 1$ or $0 < q < 1$, then the functional (4) generally is non-convex. When $p = q = 2$, the problem (4) reduces to Tikhonov regularization in general form [13, 16, 22].

We briefly comment on the choice of q . In many situations it is known that the desired vector (3) is sparse in some basis. To enhance sparsity, we may consider using a regularization term with ℓ^0 -norm. However, the minimization problem so obtained is very difficult to solve. Therefore, it is common to approximate the ℓ^0 -norm by the ℓ^1 -norm. The main advantage of using this approximation is that the ℓ^1 -norm is convex.

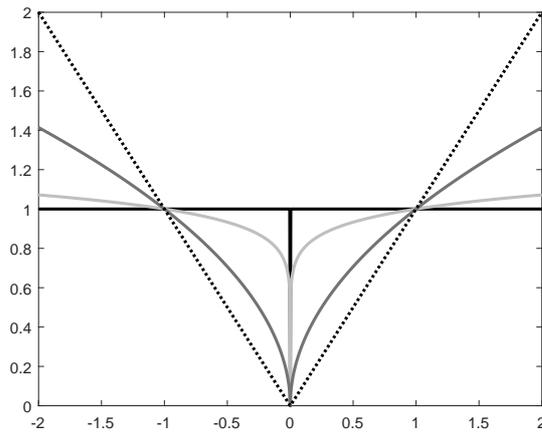


Fig. 1: Comparison of different ℓ^q -norms. The solid black line represents the ℓ^0 -norm, the dotted black line shows the ℓ^1 -norm, the dark gray solid line displays the $\ell^{0.5}$ -norm, and the light gray solid line depicts the $\ell^{0.1}$ -norm.

This makes the computation of a solution easier. However, ℓ^q -norms with $0 < q < 1$ are better approximations to the ℓ^0 -norm. In particular, the smaller q , the better the approximation, i.e., if q_1 is smaller than q_2 then the ℓ^{q_1} -norm is a better approximation of the 0-norm than the ℓ^{q_2} -norm; see Figure 1 for an illustration and [23] for a discussion. The main drawback of using $q < 1$ is that the resulting minimization problem (4) may be non-convex.

We turn to the choice of p . The value of p should depend on the type of noise in the data \mathbf{b}^δ . For white Gaussian noise, the choice $p = 2$ is appropriate. We will primarily consider this choice in the present paper.

A possible approach for solving the minimization problem (4) with $p = 2$ is to approximate the ℓ^q -norm by a weighted ℓ^2 -norm. By iteratively refining this approximation, it is possible to effectively compute a solution of (4). This process is known as the iteratively reweighted norm (IRN) method and has been applied in several different situations with good results [9, 12, 23, 24, 26]. IRN methods proceed by solving a sequence of weighted least-squares problems until a solution of (4) has been determined to desired accuracy. Applications of IRN-type methods to minimization problems (4) with p or q smaller than unity are described in [19, 23]. It is shown in [19] that the solutions of the sequence of weighted least-squares problems converge to a stationary point of the functional (4).

None of the works on solution methods for (4) mentioned discuss the choice of the regularization parameter $\mu > 0$. It is the purpose of the present paper to describe several algorithms, that are based on the discrepancy principle, for choosing a suitable μ -value, and to provide theoretical results that shed some light on their performances.

Our algorithms for determining μ are based on the IRN-type method described in [19] and there denoted by FMM-GKS. We outline the idea behind this method and denote it simply by MM-GKS. Each iteration with this method can be divided into two steps: The first step majorizes the functional to be minimized in (4) by a quadratic

function that is tangent to the functional at the current approximation. Then, in the second step, the unique minimizer of the majorant is computed and used as the new iterate.

In our first algorithm for determining a suitable value of the regularization parameter μ , we supplement the iterations described in [19] with an a priori chosen monotonically decreasing sequence of values $\mu^{(k)}$, $k = 1, 2, \dots$, of the regularization parameter. The sequence is chosen such that $\mu^{(k)} \rightarrow 0$ as $k \rightarrow \infty$. Inspired by results due to Hanke and Groetsch [15], we will use a geometrically decreasing sequence in the numerical examples of Section 4. However, any monotonically decreasing sequence that converges to zero can be used. We will show that, when no stopping criterion is employed, the iterates generated in this manner converge to a solution of the least-squares problem (1). This solution is not useful. Regularization is achieved by early termination of the iterations. The stopping rule is based on the discrepancy principle, which prescribes that the iterations be terminated as soon as an iterate $\mathbf{x}^{(k)}$ has been determined that satisfies

$$\left\| A\mathbf{x}^{(k)} - \mathbf{b}^\delta \right\|_2 \leq \tau\delta, \quad (6)$$

where δ is defined in (2) and $\tau > 1$ is a user-supplied constant independent of δ .

Our second algorithm for determining a suitable value of the regularization parameter differs from the first one in that the sequence of regularization parameters $\mu^{(k)}$, $k = 1, 2, \dots$, is not chosen a priori. Instead, we compute at each iteration a regularization parameter $\mu^{(k)}$ such that the iterate $\mathbf{x}^{(k)}$ satisfies the discrepancy principle, i.e., such that

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

We show convergence of this algorithm (up to a subsequence) and regularization properties of the computed iterates $\mathbf{x}^{(k)}$. A third algorithm considered in Section 4 uses the regularization parameter value obtained in this manner in the algorithm described in [19].

This paper is organized as follows: Section 2 outlines the FMM-GKS method described in [19] for the solution of (4). Our algorithms for determining the regularization parameter μ are described in Section 3, and a few numerical example are presented in Section 4. Finally, Section 5 contains concluding remarks and discusses some extensions.

2 A majorization-minimization method

We briefly describe the FMM-GKS method proposed in [19]. Introduce a smoothed version of the norm $\|\mathbf{x}\|_q^q$ as follows. Consider the function $\Phi_q : \mathbb{R} \rightarrow \mathbb{R}$ defined by

$$\Phi_q(t) = |t|^q.$$

If $0 < q \leq 1$, then the function $t \rightarrow \Phi_q(t)$ is not differentiable at $t = 0$. Introduce the smoothed version of Φ_q as

$$\Phi_{q,\varepsilon}(t) = \left(\sqrt{t^2 + \varepsilon^2} \right)^q, \quad (7)$$

where $\varepsilon > 0$ is a small constant. Clearly, $\Phi_{q,\varepsilon}(t)$ is everywhere differentiable. A smoothed version of $\|\mathbf{x}\|_q^q$ for $\mathbf{x} = [x_1, \dots, x_n]^t \in \mathbb{R}^n$ is given by the right-hand side of

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

Throughout this paper, the superscript t denotes transposition.

Define the smoothed version of the functional with $p = 2$ that is minimized in (4),

$$\mathcal{J}_\varepsilon(\mathbf{x}) := \frac{1}{2} \left\| A\mathbf{x} - \mathbf{b}^\delta \right\|_2^2 + \frac{\mu}{q} \sum_{i=1}^{\ell} \Phi_{q,\varepsilon}((L\mathbf{x})_i). \quad (8)$$

Thus, the smoothed minimization problem associated with (4) reads

$$\mathbf{x}^* := \arg \min_{\mathbf{x}} \mathcal{J}_\varepsilon(\mathbf{x}).$$

The FMM-GKS method described in [19] for computing a stationary point of \mathcal{J}_ε is based on a majorization-minimization method. It constructs a sequence of iterates $\mathbf{x}^{(k)}$ that converge to a stationary point of \mathcal{J}_ε . At each step the functional \mathcal{J}_ε is majorized by a quadratic function $\mathbf{x} \rightarrow \mathcal{Q}(\mathbf{x}, \mathbf{x}^{(k)})$ that is tangent to \mathcal{J}_ε at $\mathbf{x}^{(k)}$. The next iterate $\mathbf{x}^{(k+1)}$ is the unique minimizer of $\mathbf{x} \rightarrow \mathcal{Q}(\mathbf{x}, \mathbf{x}^{(k)})$. We outline this method in the remainder of this section.

Definition 1 Consider the differentiable function $\mathcal{J}(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$. We say that the function $\mathbf{x} \rightarrow \mathcal{Q}(\mathbf{x}, \mathbf{y}) : \mathbb{R}^n \rightarrow \mathbb{R}$ is a *quadratic tangent majorant* for $\mathcal{J}(\mathbf{x})$ at $\mathbf{y} \in \mathbb{R}^n$ if the following conditions hold:

- $\mathcal{Q}(\mathbf{x}, \mathbf{y})$ is quadratic;
- $\mathcal{Q}(\mathbf{x}, \mathbf{y}) \geq \mathcal{J}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^n$;
- $\mathcal{Q}(\mathbf{y}, \mathbf{y}) = \mathcal{J}(\mathbf{y})$ and $\nabla \mathcal{Q}(\mathbf{y}, \mathbf{y}) = \nabla \mathcal{J}(\mathbf{y})$.

2.1 Majorization step

We outline the construction of a quadratic tangent majorant at the point $\mathbf{x}^{(k)}$. Two approaches are described in [19], one yields a majorant with fixed aperture and the other one a majorant with the largest aperture possible. The second kind of majorant approximates the function \mathcal{J}_ε better than the first kind, but its computation is more demanding. In the following we will consider only the majorant with fixed aperture, although all the theoretical results hold true also for the majorant with adaptive aperture.

Let

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

and introduce the vector

$$\boldsymbol{\omega}^{(k)} := \mathbf{u}^{(k)} \left(1 - \left(\frac{(\mathbf{u}^{(k)})^2 + \varepsilon^2}{\varepsilon^2} \right)^{q/2-1} \right), \quad (9)$$

where all the operations, including squaring, are meant element-wise. It is shown in [19] that the function

$$\mathcal{Q}(\mathbf{x}, \mathbf{x}^{(k)}) = \frac{1}{2} \left\| A\mathbf{x} - \mathbf{b}^\delta \right\|_2^2 + \frac{\mu\varepsilon^{q-2}}{2} \left(\|L\mathbf{x}\|_2^2 - 2 \left\langle \boldsymbol{\omega}^{(k)}, L\mathbf{x} \right\rangle \right) + c, \quad (10)$$

with c a suitable constant independent of \mathbf{x} , is a quadratic tangent majorant for \mathcal{J}_ε at $\mathbf{x}^{(k)}$. We refer to [19] for a derivation of $\boldsymbol{\omega}^{(k)}$ and c .

2.2 Minimization step

Given $\mathbf{x}^{(k)}$, the next iterate $\mathbf{x}^{(k+1)}$ is the minimizer of $\mathbf{x} \rightarrow \mathcal{Q}(\mathbf{x}, \mathbf{x}^{(k)})$. Since \mathcal{Q} is quadratic, $\mathbf{x}^{(k+1)}$ can be computed by determining the zero of the gradient, i.e., by solving the linear system of equations

$$(A^t A + \eta L^t L) \mathbf{x}^{(k+1)} = A^t \mathbf{b}^\delta + \eta L^t \boldsymbol{\omega}^{(k)}, \quad \eta := \mu \varepsilon^{q-2}. \quad (11)$$

The matrix on the left-hand side is nonsingular for $\mu > 0$ due to the requirement (5). Therefore $\mathbf{x}^{(k+1)}$ is the unique minimizer of $\mathcal{Q}(\mathbf{x}, \mathbf{x}^{(k)})$.

An approximate solution of (11) can be computed efficiently by seeking a solution in a low-dimensional subspace. Let the columns of $V_k \in \mathbb{R}^{n \times d}$, with $1 \leq d \ll n$, form an orthonormal basis for the subspace in which we determine an approximate solution $\mathbf{x}^{(k+1)}$ of (11). We compute $\mathbf{x}^{(k+1)}$ by solving the minimization problem

$$\mathbf{y}^{(k+1)} := \arg \min_{\mathbf{y}} \left\| \begin{bmatrix} AV_k \\ \eta^{1/2} LV_k \end{bmatrix} \mathbf{y} - \begin{bmatrix} \mathbf{b}^\delta \\ \eta^{1/2} \boldsymbol{\omega}^{(k)} \end{bmatrix} \right\|_2^2 \quad (12)$$

and letting

$$\mathbf{x}^{(k+1)} = V_k \mathbf{y}^{(k+1)}. \quad (13)$$

Introduce the QR factorizations

$$\begin{aligned} AV_k &= Q_A R_A & \text{with } Q_A &\in \mathbb{R}^{m \times d}, & R_A &\in \mathbb{R}^{d \times d}, \\ LV_k &= Q_L R_L & \text{with } Q_L &\in \mathbb{R}^{\ell \times d}, & R_L &\in \mathbb{R}^{d \times d}. \end{aligned} \quad (14)$$

Inserting these factorizations into (12) yields

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

and substituting (13) into (11) gives the residual vector

$$\mathbf{r} := A^t (AV_k \mathbf{y}^{(k+1)} - \mathbf{b}^\delta) + \eta L^t (LV_k \mathbf{y}^{(k+1)} - \boldsymbol{\omega}^{(k)}). \quad (15)$$

We expand the solution subspace by including the scaled residual vector $\mathbf{v}_{\text{new}} = \mathbf{r} / \|\mathbf{r}\|$ in the solution subspace. This vector is orthogonal to the columns of the matrix V_k , and we define the new matrix $V_{k+1} = [V_k, \mathbf{v}_{\text{new}}] \in \mathbb{R}^{n \times (d+1)}$ with an orthonormal basis for the expanded solution subspace. The so determined solution subspaces are referred to as a *Generalized Krylov subspaces*. Another related application of this kind of solution subspaces is described in [22].

We store the matrices

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

The QR factorizations of these matrices are computed by updating the QR factorizations (14) according to

$$\begin{aligned} AV_{k+1} &= [AV_k, A\mathbf{v}_{\text{new}}] = [Q_A, \tilde{\mathbf{q}}_A] \begin{bmatrix} R_A & \mathbf{r}_A \\ \mathbf{0}^t & \tau_A \end{bmatrix}, \\ LV_{k+1} &= [LV_k, L\mathbf{v}_{\text{new}}] = [Q_L, \tilde{\mathbf{q}}_L] \begin{bmatrix} R_L & \mathbf{r}_L \\ \mathbf{0}^t & \tau_L \end{bmatrix}, \end{aligned}$$

where

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

see Daniel et al. [7] for details.

Algorithm 1 summarizes the computations. Following Huang et al. [19], we refer to the scheme described as a majorization-minimization generalized Krylov subspace (MM-GKS) method. To initiate the computations a user chooses a k_0 -dimensional solution subspace of \mathbb{R}^n . The columns of the matrix V_0 form an orthonormal basis for this subspace.

Algorithm 1 (The MM-GKS method) Let $0 < q \leq 2$ and $\mu > 0$. Consider $A \in \mathbb{R}^{m \times n}$ and $L \in \mathbb{R}^{\ell \times n}$ such that (5) holds. Fix $\varepsilon > 0$ and $k_0 > 0$.

$$\eta = \mu\varepsilon^{q-2};$$

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 LV_0 ;

Let $\mathbf{x}_0 \in \mathbb{R}^n$ be an initial approximate solution. Let $\mathbf{y}^{(0)} = V_0^t \mathbf{x}_0^{(0)}$;

Compute the QR factorizations $AV_0 = Q_A R_A$ and $LV_0 = Q_L R_L$;

for $k = 0, 1, \dots$ **do**

$$\mathbf{u}^{(k)} = LV_k \mathbf{y}^{(k)};$$

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

$$\mathbf{y}^{(k+1)} = (R_A^t R_A + \eta R_L^t R_L)^{-1} (R_A^t Q_A^t \mathbf{b}^\delta + \eta R_L^t Q_L^t \boldsymbol{\omega}^{(k)});$$

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

$$\mathbf{v}_{\text{new}} = \mathbf{r} / \|\mathbf{r}\|_2; V_{k+1} = [V_k, \mathbf{v}_{\text{new}}];$$

Update the QR factorizations: $AV_{k+1} = Q_A R_A$ and $LV_{k+1} = Q_L R_L$;

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

end

To evaluate $\mathbf{y}^{(k+1)}$, we solve a least-squares problem instead of inverting the matrix $R_A^t R_A + \eta R_L^t R_L$; see [19] for details.

The main computational effort of the algorithm for large matrices A and L is the evaluation of matrix-vector products with these matrices and with their transposes. Since the matrices AV_k and LV_k are stored, each iteration requires the evaluation of the matrix-vector products AV_{k+1} and LV_{k+1} , which are needed for updating AV_k and LV_k , and of their QR factorizations. The evaluation of a matrix-vector products with each one of the matrices A^t and L^t is required when computing the residual vector (15).

The following result for the approximate solutions $\mathbf{x}^{(k)}$ computed by Algorithm 1 is shown in [19].

Theorem 1 Let (5) hold. Then for any initial approximate solution $\mathbf{x}^{(0)} \in \mathbb{R}^n$, the sequence $\{\mathbf{x}^{(k)}\}_k$ converges to a stationary point of $\mathcal{J}_\varepsilon(\mathbf{x})$. Thus,

$$(i) \lim_{k \rightarrow \infty} \left\| \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right\|_2 = 0,$$

$$(ii) \lim_{k \rightarrow \infty} \nabla \mathcal{J}_\varepsilon(\mathbf{x}^{(k)}) = \mathbf{0}.$$

Remark 1 When $q < 1$, the minimization problems (12) may be non-convex. The stationary point computed by MM-GKS therefore might not be a global minimum; in fact, it is not assured that it is a local minimum. The stationary point computed depends on the initial approximate solution and, hence, may vary if the starting vector is changed. It is outside the scope of this paper to analyze the convexity or local convexity properties of the functionals (4) and (8), and of the minimization problem (12). Thus, we are not able to provide theoretical insight on the closeness of the computed stationary point to a global minimum. However, numerical evidences indicates that changing the initial approximate solution does not significantly alter the quality of the restoration determined by the MM-GKS method.

3 Determining the regularization parameter

This section describes two methods for determining the regularization parameter μ . Our analysis requires the matrix A to be of full column rank. This requirement might not be satisfied for certain matrices that stem from the discretization of ill-posed problems. We may replace A by the full-rank matrix

$$\tilde{A} = \begin{bmatrix} A \\ \alpha I \end{bmatrix}, \quad (16)$$

where $\alpha > 0$ is a fixed tiny constant and I denotes the identity matrix of proper dimension. For sufficiently small $\alpha > 0$, this substitution will not adversely affect the quality of the computed solution. Henceforth, we assume that such a replacement has been carried out and refer to the matrix so obtained by A . Of course, the matrix (16) does not have to be explicitly stored.

3.1 The MM-GKS-MD method

We first describe a method in which the regularization parameter decreases monotonically during the iterations. This method is derived from Algorithm 1 and uses the discrepancy principle to determine when to terminate the computations. Let $\mu^{(k)}$ denote the regularization parameter at step k . We will require that

$$\mu^{(k-1)} > \mu^{(k)} > 0 \quad \forall k, \quad \lim_{k \rightarrow \infty} \mu^{(k)} = 0. \quad (17)$$

The algorithm of this subsection is referred to as a majorization-minimization generalized Krylov subspace method with monotonically decreasing regularization parameter, in brief as the MM-GKS-MD method. We describe an algorithm that defines this method and subsequently discuss some of its properties.

Algorithm 2 (The MM-GKS-MD method) Let $0 < q < 2$ be fixed and let $\mu^{(k)}$, $k = 1, 2, \dots$, be a sequence that satisfies (17). Let the matrices $A \in \mathbb{R}^{m \times n}$ and $L \in \mathbb{R}^{\ell \times n}$ satisfy (5). Let $\mathbf{x}_0 \in \mathbb{R}^n$ be an initial approximate solution, and let $\varepsilon > 0$ and $k_0 > 0$ be constants.

```

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  $LV_0$ ;
Compute QR factorizations  $AV_0 = Q_A R_A$  and  $LV_0 = Q_L R_L$ ;
 $\mathbf{y}^{(0)} = V_0^t \mathbf{x}^{(0)}$ ;
for  $k = 0, 1, \dots$  do
     $\mathbf{u}^{(k)} = LV_k \mathbf{y}^{(k)}$ ;
     $\boldsymbol{\omega}^{(k)} = \mathbf{u}^{(k)} \left( 1 - \left( \frac{(\mathbf{u}^{(k)})^2 + \varepsilon^2}{\varepsilon^2} \right)^{q/2-1} \right)$ ;
     $\eta^{(k)} = \mu^{(k)} \varepsilon^{q-2}$ ;
     $\mathbf{y}^{(k+1)} = (R_A^t R_A + \eta^{(k)} R_L^t R_L)^{-1} (R_A^t Q_A^t \mathbf{b}^\delta + \eta^{(k)} R_L^t Q_L^t \boldsymbol{\omega}^{(k)})$ ;
    if  $\|AV_k \mathbf{y}^{(k+1)} - \mathbf{b}^\delta\|_2 \leq \tau \delta$  then
        | Exit;
    end
     $\mathbf{r} = A^t (AV_k \mathbf{y}^{(k+1)} - \mathbf{b}^\delta) + \eta^{(k)} L^t (LV_k \mathbf{y}^{(k+1)} - \boldsymbol{\omega}^{(k)})$ ;
    Reorthogonalize, if needed:  $\mathbf{r} = \mathbf{r} - V_k V_k^t \mathbf{r}$ ;
     $\mathbf{v}_{\text{new}} = \mathbf{r} / \|\mathbf{r}\|_2$ ;  $V_{k+1} = [V_k, \mathbf{v}_{\text{new}}]$ ;
    Update the QR factorizations:  $AV_{k+1} = Q_A R_A$  and  $LV_{k+1} = Q_L R_L$ ;
     $\mathbf{x}^{(k+1)} = V_{k+1} \mathbf{y}^{(k+1)}$ ;
end

```

The residual vector \mathbf{r} in Algorithm 2 is defined as in (15).

Before showing some properties of Algorithm 2, we introduce some notation that will be used in the following. Define the functional

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

and let $\mathcal{Q}^{(k)}(\mathbf{x}, \mathbf{x}^{(k)})$ denote the quadratic tangent majorant for $\mathcal{J}_\varepsilon^{(k)}$ at $\mathbf{x}^{(k)}$, i.e.,

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

where $c^{(k)}$ is a suitable constant independent of \mathbf{x} and $\boldsymbol{\omega}^{(k)}$ is defined in (9). We use the notation $\mathcal{Q}^{(k)}(\mathbf{x}, \mathbf{x}^{(k)})$ instead of $\mathcal{Q}(\mathbf{x}, \mathbf{x}^{(k)})$ to stress the fact that the parameter $\mu^{(k)}$ in $\mathcal{Q}^{(k)}$ is non-stationary.

Proposition 1 Let $\mathbf{x}^{(k)}$, $k = 1, 2, \dots$, denote the approximate solutions generated by Algorithm 2. Then

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

Proof By the properties of quadratic tangent majorants, we have

$$\begin{aligned} \mathcal{J}_\varepsilon^{(k)}(\mathbf{x}^{(k)}) &= \mathcal{Q}^{(k)}(\mathbf{x}^{(k)}, \mathbf{x}^{(k)}) \stackrel{(a)}{\geq} \mathcal{Q}^{(k)}(\mathbf{x}^{(k+1)}, \mathbf{x}^{(k)}) \stackrel{(b)}{\geq} \mathcal{J}_\varepsilon^{(k)}(\mathbf{x}^{(k+1)}) \\ &= \frac{1}{2} \left\| A\mathbf{x}^{(k+1)} - \mathbf{b}^\delta \right\|_2^2 + \frac{\mu^{(k)}}{q} \sum_{i=1}^{\ell} \Phi_{q,\varepsilon}((L\mathbf{x}^{(k+1)})_i) \\ &\stackrel{(c)}{\geq} \frac{1}{2} \left\| A\mathbf{x}^{(k+1)} - \mathbf{b}^\delta \right\|_2^2 + \frac{\mu^{(k+1)}}{q} \sum_{i=1}^{\ell} \Phi_{q,\varepsilon}((L\mathbf{x}^{(k+1)})_i) = \mathcal{J}_\varepsilon^{(k+1)}(\mathbf{x}^{(k+1)}), \end{aligned}$$

where (a) holds since $\mathbf{x}^{(k+1)}$ is a minimizer of $\mathcal{Q}^{(k)}$, (b) follows from the fact that $\mathcal{Q}^{(k)}(\mathbf{x}, \mathbf{x}^{(k)}) \geq \mathcal{J}_\varepsilon^{(k)}(\mathbf{x})$ for all \mathbf{x} , and (c) holds since $\mu^{(k)} \geq \mu^{(k+1)}$.

The following result is a consequence of Proposition 1.

Corollary 1 *Let the iterates $\mathbf{x}^{(k)}$, $k = 0, 1, \dots$, be determined by Algorithm 2. Then*

$$\mathcal{J}_\varepsilon^{(k)}(\mathbf{x}^{(k)}) \rightarrow J^* \text{ as } k \rightarrow \infty,$$

for some $J^* \geq 0$.

Proof This is obvious since the sequence $\mathcal{J}_\varepsilon^{(k)}(\mathbf{x}^{(k)})$, $k = 1, 2, \dots$, is decreasing and bounded from below by zero.

We are now in position to show that, when A is of full column rank, the iterates generated by Algorithm 2 converge to the solution of the least-squares problem (1) if no stopping criterion is employed.

Theorem 2 *Assume that the matrix A is of full column rank. Let Algorithm 2 determine the iterates $\mathbf{x}^{(k)}$, $k = 1, 2, \dots$. Then*

$$\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = A^\dagger \mathbf{b}^\delta,$$

i.e., the limit is the solution of the least-squares problem (1).

Proof Let $\rho^{(k)}$ denote the modulus of convexity of $\mathcal{Q}^{(k)}$, see, e.g., [18] for a definition, and let

$$\rho := \inf_k \rho^{(k)}.$$

Since A is of full column rank, we have $\rho > 0$. We will show that $\mathbf{x}^{(k)}$, $k = 1, 2, \dots$, is a Cauchy sequence. To this aim consider

$$\begin{aligned} \frac{\rho}{2} \left\| \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right\|_2^2 &\leq \mathcal{Q}^{(k)}(\mathbf{x}^{(k)}, \mathbf{x}^{(k)}) - \mathcal{Q}^{(k)}(\mathbf{x}^{(k+1)}, \mathbf{x}^{(k)}) \\ &= \mathcal{J}_\varepsilon^{(k)}(\mathbf{x}^{(k)}) - \mathcal{Q}^{(k)}(\mathbf{x}^{(k+1)}, \mathbf{x}^{(k)}) \\ &\stackrel{(a)}{\leq} \mathcal{J}_\varepsilon^{(k)}(\mathbf{x}^{(k)}) - \mathcal{J}_\varepsilon^{(k)}(\mathbf{x}^{(k+1)}) \\ &\stackrel{(b)}{\leq} \mathcal{J}_\varepsilon^{(k)}(\mathbf{x}^{(k)}) - \mathcal{J}_\varepsilon^{(k+1)}(\mathbf{x}^{(k+1)}), \end{aligned}$$

where (a) follows from the fact that $\mathcal{Q}^{(k)}$ is a quadratic tangent majorant, and (b) holds because $\mu^{(k+1)} \leq \mu^{(k)}$.

Since the sequence $\mathcal{J}_\varepsilon^{(k)}(\mathbf{x}^{(k)})$, $k = 1, 2, \dots$, is convergent, we have

$$\mathcal{J}_\varepsilon^{(k)}(\mathbf{x}^{(k)}) - \mathcal{J}_\varepsilon^{(k+1)}(\mathbf{x}^{(k+1)}) \rightarrow 0 \text{ as } k \rightarrow \infty.$$

Thus,

$$\left\| \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right\|_2 \rightarrow 0 \text{ as } k \rightarrow \infty,$$

and we have established that the sequence $\mathbf{x}^{(k)}$, $k = 1, 2, \dots$, is a Cauchy sequence. Since \mathbb{R}^n is complete, there is a limit point \mathbf{x}^* .

It remains to show that \mathbf{x}^* is the solution of $A\mathbf{x} = \mathbf{b}^\delta$. Since we assume the matrix A to be of full column rank, it suffices to prove that the residual converges to $\mathbf{0}$ as $k \rightarrow \infty$. Assume that k is large enough so that the columns of V_k span \mathbb{R}^n and consider the residual at step k ,

$$\mathbf{r}^{(k)} = A\mathbf{x}^{(k)} - \mathbf{b}^\delta = A(A^t A + \eta^{(k)} L^t L)^{-1} A^t \mathbf{b}^\delta + \eta^{(k)} L^t \boldsymbol{\omega}^{(k)} - \mathbf{b}^\delta.$$

Since $\mu^{(k)} \rightarrow 0$ as $k \rightarrow \infty$, it holds that $\eta^{(k)} \rightarrow 0$ as $k \rightarrow \infty$. Thus,

$$\mathbf{r}^{(k)} \rightarrow (A^t A)^{-1} A^t A \mathbf{b}^\delta - \mathbf{b}^\delta = \mathbf{0} \text{ as } k \rightarrow \infty.$$

This shows the theorem.

Corollary 2 *Let δ be a bound for the norm of the error in \mathbf{b}^δ , see (2), and let $\tau > 1$ be a parameter independent of δ . The computations with Algorithm 2 equipped with the discrepancy principle (6) as stopping criterion terminate after a finite number, k_δ , of steps. The iterate $\mathbf{x}^{(k_\delta)}$ generated at step k_δ satisfies*

$$\left\| A\mathbf{x}^{(k_\delta)} - \mathbf{b}^\delta \right\|_2 \leq \tau\delta.$$

We are now in a position to show that Algorithm 2 equipped with the discrepancy principle as a stopping rule is an iterative regularization method.

Theorem 3 *Let the conditions of Corollary 2 hold and let $\mathbf{x}^\delta = \mathbf{x}^{(k_\delta)}$ denote the last iterate determined by Algorithm 2 equipped with the discrepancy principle as stopping criterion. Then*

$$\limsup_{\delta \searrow 0} \left\| \mathbf{x}^\delta - \widehat{\mathbf{x}} \right\|_2 = 0,$$

where $\widehat{\mathbf{x}}$ denotes the desired solution vector (3).

Proof All norms are equivalent in \mathbb{R}^n and A is of full column rank. Therefore, there is a constant $\tilde{c} > 0$ such that

$$\left\| \mathbf{x}^\delta - \widehat{\mathbf{x}} \right\|_2 \leq \tilde{c} \left\| \mathbf{x}^\delta - \widehat{\mathbf{x}} \right\|_A$$

for all vectors $\mathbf{x}^\delta - \widehat{\mathbf{x}}$, where $\|\mathbf{y}\|_A = \|A\mathbf{y}\|_2$ is the vector norm induced by A . Then we have

$$\begin{aligned}
\limsup_{\delta \searrow 0} \|\mathbf{x}^\delta - \widehat{\mathbf{x}}\|_2 &\leq \tilde{c} \limsup_{\delta \searrow 0} \|\mathbf{x}^\delta - \widehat{\mathbf{x}}\|_A \\
&= \tilde{c} \limsup_{\delta \searrow 0} \|A\mathbf{x}^\delta - A\widehat{\mathbf{x}}\|_2 \\
&= \tilde{c} \limsup_{\delta \searrow 0} \|A\mathbf{x}^\delta - \mathbf{b}^\delta - (A\widehat{\mathbf{x}} - \mathbf{b}^\delta)\|_2 \\
&= \tilde{c} \limsup_{\delta \searrow 0} \|A\mathbf{x}^\delta - \mathbf{b}^\delta - (\mathbf{b} - \mathbf{b}^\delta)\|_2 \\
&\leq \tilde{c} \limsup_{\delta \searrow 0} \left(\|A\mathbf{x}^\delta - \mathbf{b}^\delta\|_2 + \|\mathbf{b} - \mathbf{b}^\delta\|_2 \right) \\
&\stackrel{(a)}{\leq} \tilde{c} \limsup_{\delta \searrow 0} (\tau\delta + \delta) \\
&\leq \tilde{c}(1 + \tau) \limsup_{\delta \searrow 0} \delta \\
&= 0,
\end{aligned}$$

where (a) follows from Corollary 2 and (2).

3.2 The MM-GKS-DP method

This subsection describes another method derived from Algorithm 1 for choosing a suitable value of the regularization parameter. The method determines a value of the regularization parameter that satisfies the discrepancy principle in each iteration. We refer to this scheme as the MM-GKS-DP method. The theoretical results for this method are less complete than those of the previous subsection. For instance, we will not prove convergence of the method, but only show the existence of a converging subsequence. Nevertheless, numerical experiments reported in Section 4 show the MM-GKS-DP method to achieve very good results. We first describe the algorithm and then discuss its properties.

Algorithm 3 (The MM-GKS-DP method) Let $0 < q < 2$ be fixed and let $\mu^{(k)}$, $k = 1, 2, \dots$, be a sequence that satisfies (17). Let the matrices $A \in \mathbb{R}^{m \times n}$ and $L \in \mathbb{R}^{\ell \times n}$ satisfy (5). Let $\mathbf{x}_0 \in \mathbb{R}^n$ be an initial approximate solution, and let $\varepsilon > 0$ and $k_0 > 0$ be constants.

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 LV_0 ;

Compute the QR factorizations $AV_0 = Q_A R_A$ and $LV_0 = Q_L R_L$;

$\mathbf{y}^{(0)} = V_0^t \mathbf{x}^{(0)}$;

for $k = 0, 1, \dots$ **do**

$\mathbf{u}^{(k)} = LV_k \mathbf{y}^{(k)}$;

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

$\eta^{(k)} = \mu^{(k)} \varepsilon^{q-2}$;

 where $\mu^{(k)}$ is such that $\|AV_k \mathbf{y}^{(k+1)} - \mathbf{b}^\delta\|_2 = \tau\delta$;

$\mathbf{y}^{(k+1)} = (R_A^t R_A + \eta^{(k)} R_L^t R_L)^{-1} (R_A^t Q_A^t \mathbf{b}^\delta + \eta^{(k)} R_L^t Q_L^t \boldsymbol{\omega}^{(k)})$;

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

 Reorthogonalize, if needed: $\mathbf{r} = \mathbf{r} - V_k V_k^t \mathbf{r}$;

$\mathbf{v}_{\text{new}} = \mathbf{r} / \|\mathbf{r}\|_2$; $V_{k+1} = [V_k, \mathbf{v}_{\text{new}}]$;

 Update the QR factorizations: $AV_{k+1} = Q_A R_A$ and $LV_{k+1} = Q_L R_L$;

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

end

Again the residual vector \mathbf{r} is defined as in (15).

We now show that the sequence $\mathbf{x}^{(k)}$, $k = 1, 2, \dots$, computed by Algorithm 3 has a converging subsequence with a limit that satisfies the discrepancy principle.

Proposition 2 Assume that A is of full column rank and let $\mathbf{x}^{(k)}$, $k = 1, 2, \dots$, denote the iterates generated by Algorithm 3. Then there is a subsequence $\mathbf{x}^{(k_j)}$, $j = 1, 2, \dots$, with a limit \mathbf{x}^* , such that

$$\|A\mathbf{x}^* - \mathbf{b}^\delta\|_2 = \tau\delta.$$

Proof We have for all k that $\|A\mathbf{x}^{(k)} - \mathbf{b}^\delta\|_2 = \tau\delta$. Since all norms are equivalent in \mathbb{R}^n , there is a constant $\hat{c} > 0$ independent of A and $\mathbf{x}^{(k)}$ such that

$$\begin{aligned} \tau\delta &= \|A\mathbf{x}^{(k)} - \mathbf{b}^\delta\|_2 \geq \|A\mathbf{x}^{(k)}\|_2 - \|\mathbf{b}^\delta\|_2 \\ &= \|\mathbf{x}^{(k)}\|_A - \|\mathbf{b}^\delta\|_2 \geq \hat{c} \|\mathbf{x}^{(k)}\|_2 - \|\mathbf{b}^\delta\|_2, \end{aligned}$$

where $\|\mathbf{x}^{(k)}\|_A = \|A\mathbf{x}^{(k)}\|_2$. Thus, for all k we have the bound

$$\|\mathbf{x}^{(k)}\|_2 \leq \frac{\tau\delta + \|\mathbf{b}^\delta\|_2}{\hat{c}}.$$

This shows that the sequence $\mathbf{x}^{(k)}$, $k = 1, 2, \dots$, is bounded. Therefore, there is a subsequence $\mathbf{x}^{(k_j)}$, $j = 1, 2, \dots$ that converges to a limit \mathbf{x}^* as j increases. Since

$\|A\mathbf{x}^{(k_j)} - \mathbf{b}^\delta\|_2 = \tau\delta$ for all j , we have by continuity of the norm that

$$\|A\mathbf{x}^* - \mathbf{b}^\delta\|_2 = \tau\delta.$$

We now can show that the MM-GKS-DP method is an iterative regularization method (up to a subsequence).

Theorem 4 *Let $\mathbf{x}^{(k_j)}$, $j = 1, 2, \dots$, denote the subsequence defined in Proposition 2 and let $\mathbf{x}^\delta = \mathbf{x}^{(k_j)}$ denote the approximate solution determined by Algorithm 3 with noise level $\delta > 0$, where we assume that A is of full column rank. Then*

$$\limsup_{\delta \searrow 0} \|\mathbf{x}^\delta - \widehat{\mathbf{x}}\|_2 = 0,$$

where $\widehat{\mathbf{x}}$ denotes the solution (3) of the error-free system.

Proof The result can be shown in the same manner as Theorem 3. We therefore omit the details.

Remark 2 In actual computations we have not experienced the need to take a subsequence of the iterates $\mathbf{x}^{(k)}$, $k = 1, 2, \dots$, determined by Algorithm 3.

Remark 3 This section shows results for the MM-GKS method of [19]. This reference also considers an approach to construct quadratic majorants, whose aperture depends on $\mathbf{x}^{(k)}$. The resulting scheme is referred to as the AMM-GKS in [19]. It requires more work per iteration, but the number of iterations required may be smaller than for the MM-GKS method applied in the present paper. The theory developed in this section carries over to the AMM-GKS method without change.

Remark 4 Recently, Huang et al. [20] used the approach of this subsection to determine the regularization parameter in an iterated Tikhonov regularization method in general form with $p = q = 2$. Then the quadratic majorant Q , defined by (10), is independent of k . Theorem 4 applies to the iterates generated by the method in [20].

Remark 5 This section focused on the use of the Euclidean norm for the fidelity term in (4). The results of this section carry over to fidelity terms with $p \neq 2$, such as $p = 1$. Then Algorithms 1-3 have to be modified to implement the FMM-GKS or AMM-GKS methods of [19].

4 Numerical examples

We present some numerical examples. All computations are carried out in MATLAB 2016a with about 15 significant decimal digits running on a laptop computer with a quad core CPU Intel Core i7-6700HQ @2.60GHz and with 16 GB of RAM. We compare Algorithms 2 and 3 with the MM-GKS method described in [19], which requires a user to explicitly specify a value of the regularization parameter, and with the methods described in [1, 8]. We do not compare our approach with other methods from the literature for solving (4) since the main focus of this paper is to show that, using the discrepancy principle, we are able to determine a regularization parameter that provides accurate restorations, i.e., restorations that are of about the same quality

as restorations obtained with the optimal regularization parameter. For a thorough analysis of the performances of the $\ell^2 - \ell^q$ approach and for a comparison with other methods from the literature we refer to [19,23].

The error in the vector \mathbf{b} is in all examples white Gaussian noise. We fix $\varepsilon = 1$ in (7). This value is small compared to elements of the vector \mathbf{b}^δ , which in all examples represents a contaminated image. Each element is a pixel value in the range $[0, 255]$. Numerical experiments show that the choice of ε does not largely affect the quality of the restorations, but can greatly affect the required number of iterations. This is illustrated below.

We would like to compute a sparse solution and therefore use a two-level framelet analysis operator as regularization operator L . Framelets are extensions of wavelets. They are defined as follows.

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

$$\|\mathbf{x}\|_2^2 = \sum_{j=1}^r \mathbf{y}_j^t \mathbf{x},$$

where $\mathbf{y}_j \in \mathbb{R}^n$ is the j -th row of \mathcal{A} (written as a column vector), i.e., $\mathcal{A} = [\mathbf{y}_1, \dots, \mathbf{y}_r]^t$. The matrix \mathcal{A} is referred to as an *analysis operator* and \mathcal{A}^t as a *synthesis operator*.

Tight frames have been used in many image restoration applications, including inpainting and deblurring [2,4,5]. The essential feature of tight frames is their redundancy. Due to the redundancy, the loss of some information can be tolerated. Usually images have a very sparse representation in the framelet domain.

We will use tight frames determined by linear B-splines. These frames are in one space-dimension made up of a low-pass filter W_0 and two high-pass filters W_1 and W_2 . The corresponding masks are given by

$$\mathbf{w}^{(0)} = \frac{1}{2} (1, 2, 1), \quad \mathbf{w}^{(1)} = \frac{\sqrt{2}}{4} (1, 0, -1), \quad \mathbf{w}^{(2)} = \frac{1}{4} (-1, 2, -1).$$

We now derive the synthesis operator \mathcal{A} from these masks. Imposing reflexive boundary conditions, so that $\mathcal{A}^t \mathcal{A} = I$, we obtain

$$W_0 = \frac{1}{4} \begin{pmatrix} 3 & 1 & 0 & \dots & 0 \\ 1 & 2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 2 & 1 \\ 0 & \dots & 0 & 1 & 3 \end{pmatrix}, \quad W_1 = \frac{\sqrt{2}}{4} \begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & & -1 & 0 & 1 \\ 0 & \dots & 0 & -1 & 1 \end{pmatrix},$$

and

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

We obtain the synthesis operator

$$\mathcal{A} = \begin{pmatrix} W_0 \\ W_1 \\ W_2 \end{pmatrix}.$$

We define operators for two space-dimensions by using tensor products

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

This yields the analysis operator

$$\mathcal{A} = \begin{pmatrix} W_{00} \\ W_{01} \\ \vdots \\ W_{22} \end{pmatrix}.$$

The matrix W_{00} is a low-pass filter. All the other matrices W_{ij} contain at least one high-pass filter in some direction.

While we have discussed the stopping criterion for the MM-GKS-MD method, we have not defined one for the MM-GKS-DP and MM-GKS methods. We will terminate the iterations with the latter two methods as soon as two consecutive iterates are sufficiently close, i.e., as soon as

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

Inspired by the geometric sequence of regularization parameters used for iterated Tikhonov methods described in [15], we use in the MM-GKS-MD method the following sequence of regularization parameters

$$\mu^{(k)} = 0.7^k, \quad k = 1, 2, \dots$$

We also have to address the selection of the regularization parameter for the MM-GKS method. Our first computed example illustrates that the regularization parameter value determined by the MM-GKS-DP method is quite close to the optimal one, i.e., to the value that minimizes the restoration error defined below. We therefore will in our computed examples use the regularization parameter value determined by the MM-GKS-DP method for the MM-GKS method as well.

We measure the quality of the restored images with the relative restoration error

$$\text{RRE}(\mathbf{x}) = \frac{\|\mathbf{x} - \widehat{\mathbf{x}}\|_2}{\|\widehat{\mathbf{x}}\|_2},$$

where $\widehat{\mathbf{x}}$ denotes the desired solution (3).

Example 1 We consider the restoration of an image that has been contaminated by noise and spatially invariant blur; see, e.g., [17] for a discussion of this kind of restoration problems. The blurring phenomenon can be modeled with a Fredholm integral equation of the first kind

$$B(k, f) = \int k(s - x, t - y) f(s, t) ds dt = k * f,$$

which, due to the spatial invariance, reduces to a convolution. Discretization yields a linear system of equations with a severely ill-conditioned matrix, i.e., the singular values of the matrix decrease to zero with increasing index with no significant gap; see,

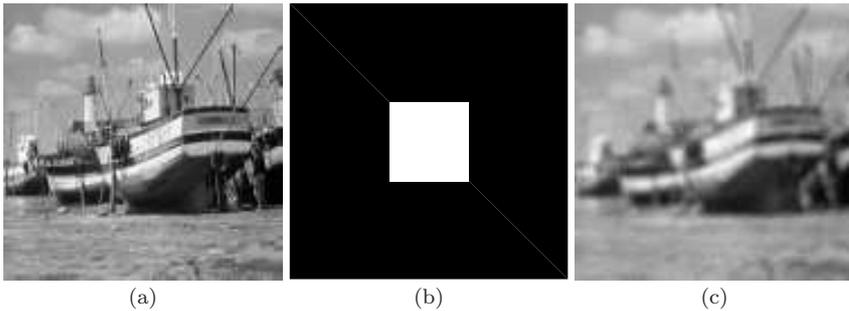


Fig. 2: Boat test image: (a) True image (124×124 pixels), (b) PSF (14×14 pixels), (c) blurred and noisy image ($\delta = 0.02 \|\mathbf{b}\|_2$).

e.g., [14, 16]. We impose anti-reflective boundary conditions to take into account that the available image is finite; see [25] for details.

We first illustrate how the selection of ε and q affect the accuracy and convergence rate of the MM-GKS-DP method. Then we compare the proposed approaches with other methods from the literature. In particular, we compare with the MM-GKS method described in [19] and with methods proposed in [1, 8]. Specifically, we compare with the Approximated Iterated Tikhonov (AIT) method described in [8], the Approximated Iterated Tikhonov with General Penalty term (AIT-GP) method, the Approximated Projected Iterated Tikhonov (APIT) method, and the Approximated Projected Iterated Tikhonov with General Penalty term (APIT-GP) method described in [1].

We turn to the roles of q and ε in the MM-GKS-DP method. To gain some insight into how these parameters should be chosen, we apply the MM-GKS-DP method to a fairly small test problem. Specifically, we consider the Boat image blurred by an average PSF of size 4×4 pixels, and contaminated by white Gaussian noise such that $\delta = 0.02 \|\mathbf{b}\|_2$; see Figure 2. Figure 3 shows results obtained for different values of q and ε . We observe that for small values of ε , the method converges in fairly few iterations, but the computed restoration is not so accurate. The best restorations are achieved when $\varepsilon = 1$ and q is small (in this case $q = 10^{-1}$). We also observe that, fixing $\varepsilon = 1$, the error decreases with q , but the number of iterations required by the method grows sharply as q decreases.

Figure 4 displays three restorations determined by the MM-GKS-DP method for three different choices of q and ε . Specifically, we show restorations obtained for $q = 10^{-1}$ and $\varepsilon = 10^{-3}$, for $q = 1$ and $\varepsilon = 1$, and for $q = 10^{-1}$ and $\varepsilon = 1$. Visual inspection of the images shows the first parameter pair to yield a restoration with some noise and ringing near the edges, the second parameter pair gives a restoration with less noise, but still shows some ringing, while the third parameter pair gives the best restoration; it appears noise-free and does not show significant ringing.

Example 2 We would like to illustrate that the restorations obtained by using the smoothed functional \mathcal{J}_ε defined by (8) are very close to the restoration determined by the functional \mathcal{J} defined by (4). To this aim, we iteratively apply Algorithm 3 while progressively decreasing the value of ε . Let $\{\varepsilon_j\}_j$ denote a decreasing sequence such that $\varepsilon_j \rightarrow 0$ as $j \rightarrow \infty$. For each ε_j , we run Algorithm 3 until convergence. To speed

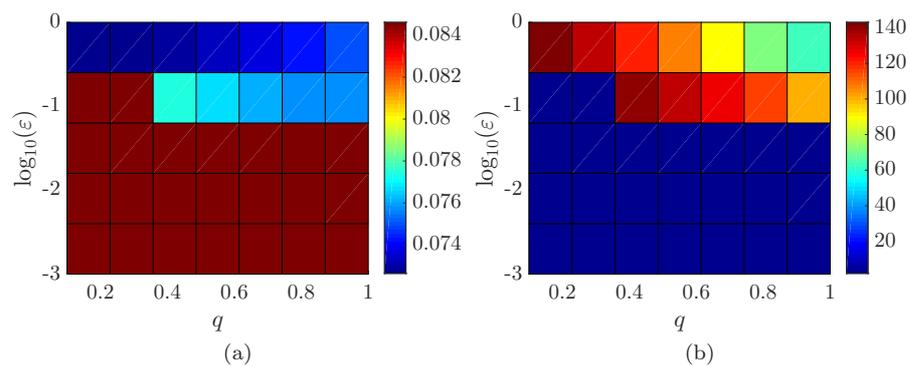


Fig. 3: Boat test case: (a) RRE obtained with MM-GKS-DP for different values of q and ϵ , (b) Number of iterations required to reach convergence with MM-GKS-DP for different values of q and ϵ .

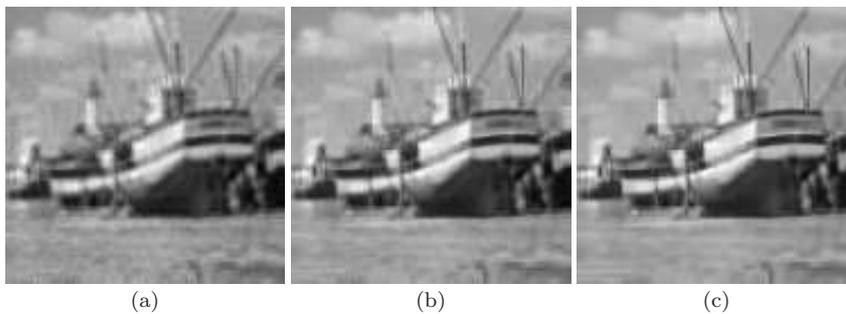


Fig. 4: Boat test image restorations obtained with MM-GKS-DP with different values of q and ϵ : (a) $q = 10^{-1}$ and $\epsilon = 10^{-3}$, (b) $q = 1$ and $\epsilon = 1$, (c) $q = 10^{-1}$ and $\epsilon = 1$.

up the computations, we use as initial vector the restoration provided by Algorithm 3 at the previous step, i.e., when ϵ_{j-1} is used as smoothing parameter. The procedure is summarized by Algorithm 4.

Algorithm 4 (The restarted MM-GKS-DP method) Let $0 < q \leq 2$. Consider $A \in \mathbb{R}^{m \times n}$ and $L \in \mathbb{R}^{\ell \times n}$ such that (5) holds. Let $\{\varepsilon_j\}_j$ denote a decreasing sequence such that $\varepsilon_j \rightarrow 0$ as $j \rightarrow \infty$ and fix $k_0 > 0$.

```

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  $LV_0$ ;
Compute the QR factorizations  $AV_0 = Q_A R_A$  and  $LV_0 = Q_L R_L$ ;
 $\mathbf{y}^{(0)} = V_0^t \mathbf{x}^{(0)}$ ;
for  $j = 0, 1, \dots$  do
  for  $k = 0, 1, \dots$  do
     $\mathbf{u}^{(k)} = LV_k \mathbf{y}^{(k)}$ ;
     $\boldsymbol{\omega}^{(k)} = \mathbf{u}^{(k)} \left( 1 - \left( \frac{(\mathbf{u}^{(k)})^2 + \varepsilon_j^2}{\varepsilon_j^2} \right)^{q/2-1} \right)$ ;
     $\eta^{(k)} = \mu^{(k)} \varepsilon_j^{q-2}$ ;
    where  $\mu^{(k)}$  is such that  $\|AV_k \mathbf{y}^{(k+1)} - \mathbf{b}^\delta\|_2 = \tau \delta$ ;
     $\mathbf{y}^{(k+1)} = (R_A^t R_A + \eta^{(k)} R_L^t R_L)^{-1} (R_A^t Q_A^t \mathbf{b}^\delta + \eta^{(k)} R_L^t Q_L^t \boldsymbol{\omega}^{(k)})$ ;
     $\mathbf{r} = A^t (AV_k \mathbf{y}^{(k+1)} - \mathbf{b}^\delta) + \eta L^t (LV_k \mathbf{y}^{(k+1)} - \boldsymbol{\omega}^{(k)})$ ;
    Reorthogonalize, if needed:  $\mathbf{r} = \mathbf{r} - V_k V_k^t \mathbf{r}$ ;
     $\mathbf{v}_{\text{new}} = \mathbf{r} / \|\mathbf{r}\|_2$ ;  $V_{k+1} = [V_k, \mathbf{v}_{\text{new}}]$ ;
    Update the QR factorizations:  $AV_{k+1} = Q_A R_A$  and  $LV_{k+1} = Q_L R_L$ ;
     $\mathbf{x}^{(k+1)} = V_{k+1} \mathbf{y}^{(k+1)}$ ;
  end
end
end

```

Similarly as in the previous algorithms, the residual vector \mathbf{r} is defined by (15).

The computations described by Algorithm 4 can be demanding both in terms of computing time and memory consumption. We therefore only apply the algorithm to the restoration of a fairly small image. In particular, we consider a 122×122 -pixel satellite image, see Figure 5(a), blur it with a 13×13 -pixel motion blur PSF, and add 3% white Gaussian noise. The contaminated image is shown in Figure 5(c).

We choose a logarithmically spaced sequence ε_j , $j = 1, 2, \dots, 5$, between 1 and 10^{-6} . Table 1 shows the restoration errors obtained and the number of iterations required to reach convergence by Algorithms 4 and 3. The table shows the restorations determined by these algorithms to be very close. It is not surprising to see that the restarted MM-GKS-DP method is able to produce a more accurate restoration. However, the difference in quality of the restorations obtained by Algorithms 4 and 3 is insignificant. In particular, the small improvement in accuracy does not justify the additional computational effort required by Algorithm 4; see Table 1 and Figure 6. In the following numerical examples we therefore are not going to consider the restarted MM-GKS-DP method.

Example 3 We consider the cameraman image, blur it with a non-symmetric PSF, and add white Gaussian noise such that $\delta = 0.05 \|\mathbf{b}\|_2$; see Figure 7. Anti-reflective boundary conditions are imposed; see [25].

We first analyze how the choice of μ affects the quality of the restoration determined by the MM-GKS method, and compare it to restorations obtained by the MM-GKS-MD and MM-GKS-DP methods. Let $q = 0.1$ and run the MM-GKS method for different

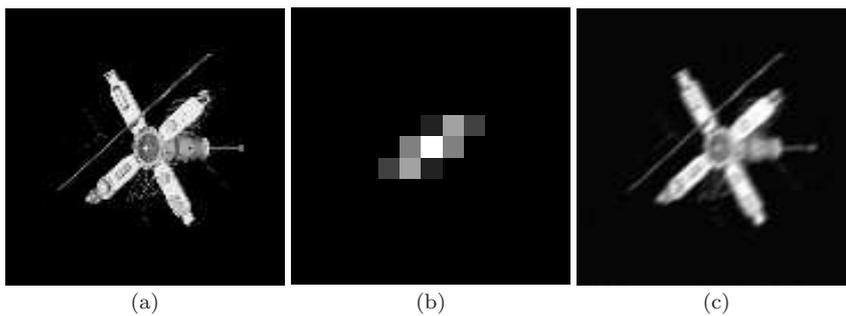


Fig. 5: Satellite test image: (a) True image (122×122 pixels), (b) PSF (13×13 pixels), (c) blurred and noisy image ($\delta = 0.03 \|\mathbf{b}\|_2$).

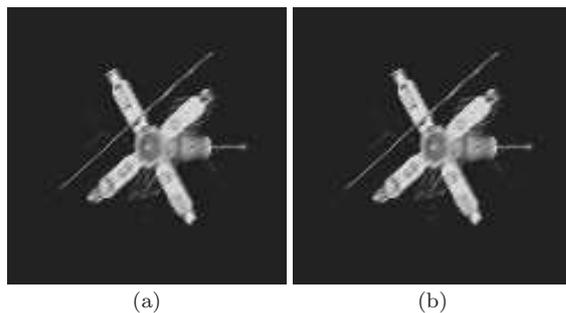


Fig. 6: Satellite test image restoration: (a) MM-GKS-DP, (b) Restarted MM-GKS-DP.

Method	RRE	Iterations
Restarted MM-GKS-DP	0.13274	256
MM-GKS-DP	0.13655	186

Table 1: Satellite test image: Relative restoration errors of the restorations obtained with the different methods and number of iterations required to reach convergence. The smallest error is in boldface.

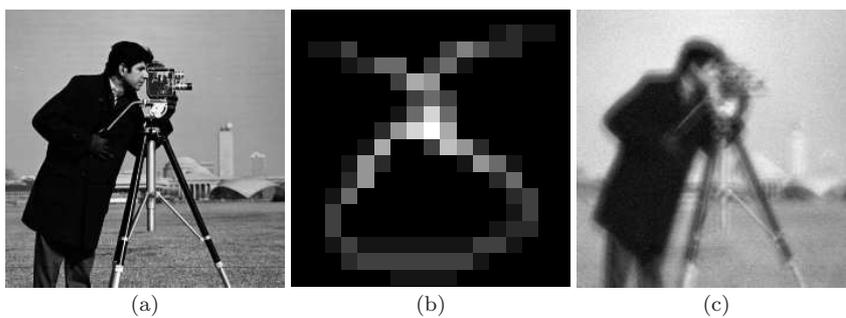


Fig. 7: Cameraman test image: (a) True image (238×238 pixels), (b) PSF (17×17 pixels), (c) blurred and noisy image ($\delta = 0.03 \|\mathbf{b}\|_2$).

choices of μ . Figure 8(a) shows graphs for the RRE as a function of μ for the three methods. The graphs for the MM-GKS-MD and MM-GKS-DP methods are obviously constant since they are independent of the choice of μ . Visual inspection of the graph for the MM-GKS method shows, as one would expect, that the RRE obtained for the optimal μ -value to be smaller than the RRE for the MM-GKS-MD and MM-GKS-DP methods. However, the optimal RRE-value is very close to the error obtained by the MM-GKS-DP method and not too far away from the error obtained by the MM-GKS-MD method. Moreover, if μ is chosen too large or too small, then the restoration error for the MM-GKS method can be very large. Since the optimal value of the regularization parameter μ for the MM-GKS method is close to the value determined by the MM-GKS-DP method, we will use the latter in the following computed examples with the MM-GKS method.

Starting from this observation, we define a new method, denoted by MM-GKS-R, for majorization-minimization in generalized Krylov subspaces, repeatedly. This method combines the MM-GKS-DP and MM-GKS methods and is defined by Algorithm 5.

Algorithm 5 (MM-GKS-R) *Let $0 < q \leq 2$. Consider $A \in \mathbb{R}^{m \times n}$ and $L \in \mathbb{R}^{\ell \times n}$ such that (5) holds. Fix $\varepsilon > 0$ and $k_0 > 0$.*

*Run MM-GKS-DP until convergence;
Denote by μ^* the last parameter used in MM-GKS-DP;
Run MM-GKS with $\mu = \mu^*$ until convergence;*

This algorithm satisfies all the properties shown in [19], i.e., Theorem 1 holds. Moreover, it can be shown experimentally that the limit point of this algorithm approximately satisfies the discrepancy principle.

We next discuss the effect of varying q . Figure 8(b) shows the RRE obtained with the three methods when q is varied. We observe that the RRE obtained by both the MM-GKS-R and MM-GKS-DP methods increases with q , while the RRE obtained by the MM-GKS-MD method does not change much with q (and is larger than for the other two methods). While this comparison suggests that it is good to choose a very small q -value, it must be stressed that, if q is too small, then all the three methods might become numerically unstable.

Let $q = 0.1$. Table 2 reports the RRE obtained with the different methods as well as the number of iterations required to reach convergence. Figure 9 shows some of the restorations obtained.

Table 2 shows the MM-GKS-R, MM-GKS-MD, and MM-GKS-DP methods to outperform the AIT, AIT-GP, APIT, and APIT-GP algorithms in terms of accuracy. However, the MM-GKS-type methods require more iterations to converge. Moreover, among the MM-GKS-type methods, the MM-GKS-MD method is the fastest, but the computed restorations are less accurate than those determined with the MM-GKS-R and MM-GKS-DP methods.

Visual inspection of the restorations in Figure 9 shows that the restorations obtained with AIT-type methods are more noisy than the ones determined with MM-GKS-type methods. MM-GKS-type methods are seen to be able to restore details of the image more accurately with less noise propagation.

Example 4 We turn to the restoration of a Clock image, which has been contaminated by out-of-focus blur and white Gaussian noise such that $\delta = 0.01 \|\mathbf{b}\|_2$; see Figure 10.

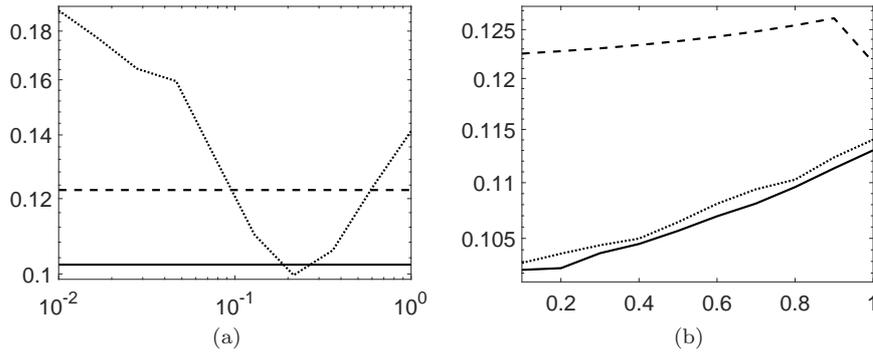


Fig. 8: Cameraman test image: Behavior of the RRE as function of μ and q . Figure (a) shows the RRE obtained for different μ -values and $q = 0.1$. The dashed curve displays the RRE obtained with MM-GKS-MD, the solid curve displays the RRE determined by MM-GKS-DP, and the dotted line depicts the RRE obtained with MM-GKS. The first two methods have constant error since they do not need an estimate of the parameter μ . Figure (b) displays the error obtained with the three methods when changing q . For the MM-GKS method, we used the μ -value determined in the last iteration of MM-GKS-DP, i.e., we applied the MM-GKS-R method. The dashed curve shows the RRE determined by MM-GKS-MD, the solid curve depicts the RRE obtained by MM-GKS-DP, and the dotted curve shows the RRE generated by MM-GKS.

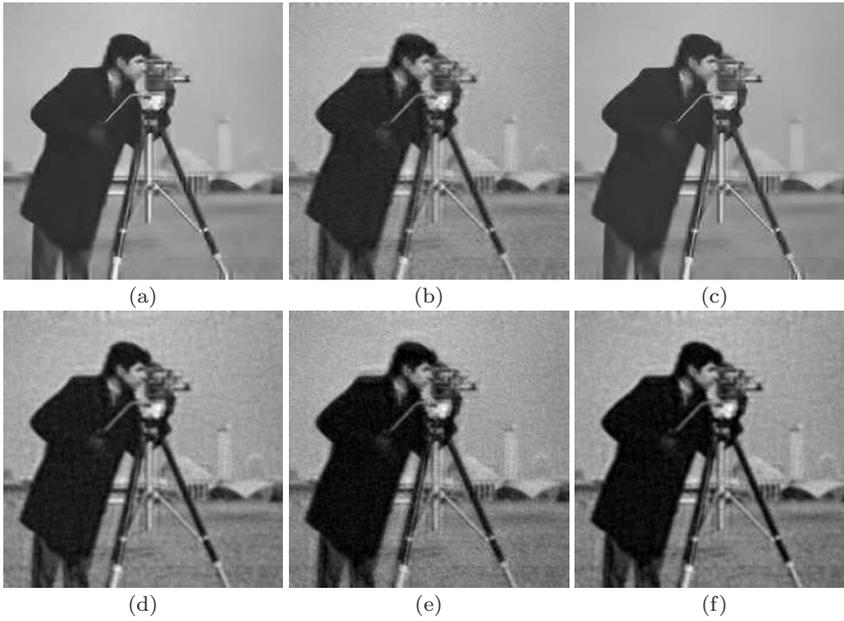


Fig. 9: Cameraman test image restorations: (a) MM-GKS-R, (b) MM-GKS-MD, (c) MM-GKS-DP, (d) AIT-GP, (e) APIT, (f) APIT-GP.

Method	RRE	Iterations
MM-GKS-R	0.10287	97
MM-GKS-MD	0.12252	7
MM-GKS-DP	0.10228	100
AIT	0.13061	3
AIT-GP	0.12371	3
APIT	0.13025	3
APIT-GP	0.12346	3

Table 2: Cameraman test image: Relative restoration errors of the restorations obtained with the different methods and number of iterations required to reach convergence. The smallest error is in boldface.

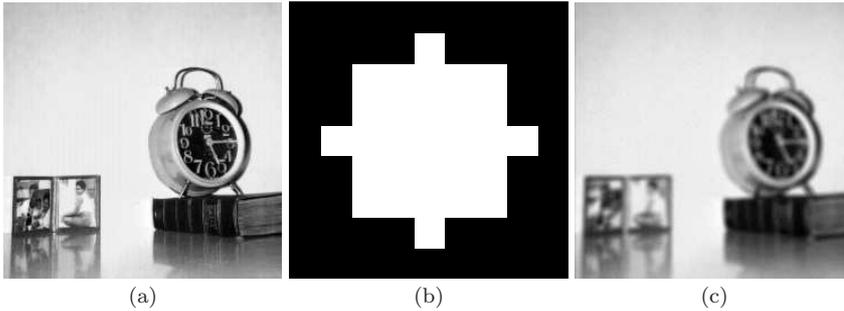


Fig. 10: Clock test image: (a) True image (246×246 pixels), (b) PSF (9×9 pixels), (c) blurred and noisy image ($\delta = 0.01 \|\mathbf{b}\|_2$).

Method	RRE	Iterations
MM-GKS-R	0.032887	29
MM-GKS-MD	0.036663	5
MM-GKS-DP	0.032828	28
AIT	0.051979	5
AIT-GP	0.050436	5
APIT	0.051979	5
APIT-GP	0.050436	5

Table 3: Clock test image: Relative errors of the restorations determined by different methods and number of iterations required to reach convergence. The smallest error is in boldface.

The PSF has a radius of 4 pixels. We impose anti-reflective boundary conditions and set $q = 0.7$. Figure 11 displays some of the computed restorations and Table 3 reports the RRE and the number of iterations for the different methods.

As in the previous examples, MM-GKS-type methods out-perform the other methods, even though for this example the difference in performance is smaller than for the previous example. This is due to the fact that there is less noise-contamination than in the previous example. In term of efficiency, we can see that the MM-GKS-MD method requires fewer iterations than the other two MM-GKS methods, but gives a less accurate restoration.

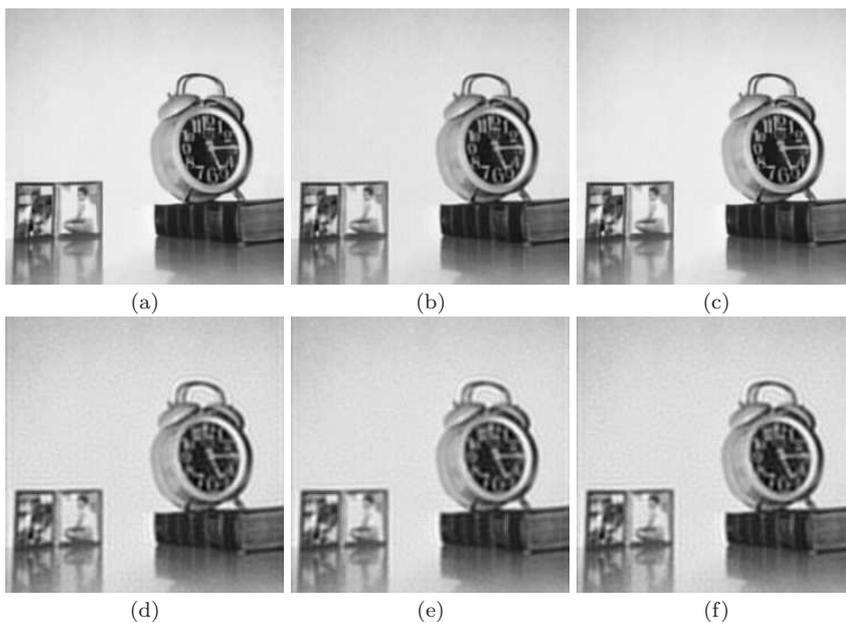


Fig. 11: Clock test image restorations: (a) MM-GKS-R, (b) MM-GKS-MD, (c) MM-GKS-DP, (d) AIT-GP, (e) APIT, (f) APIT-GP.

5 Conclusions

The image restoration methods described in [19, 23] require a user to provide a suitable value of the regularization parameter. This paper develops two approaches to determine such a value with the aid of the discrepancy principle. This enhances the usefulness of the methods. They can be applied without user-interaction to real-world applications for which an estimate of the norm of the error in \mathbf{b}^δ is available.

Acknowledgments

The authors would like to thank the referees for their insightful comments that improved the presentation. The first author is a member of the INdAM Research group GNCS and his work is partially funded by the group. The second author is supported in part by NSF grants DMS-1720259 and DMS-1729509.

References

1. A. Buccini, Regularizing preconditioners by non-stationary iterated Tikhonov with general penalty term, *Appl. Numer. Math.*, 116 (2017), pp. 64–81.
2. J.-F. Cai, R. H. Chan, L. Shen, and Z. Shen, Simultaneously inpainting in image and transformed domains, *Numer. Math.*, 112 (2009), pp. 509–533.
3. J.-F. Cai, R. H. Chan, and Z. Shen, A framelet-based image inpainting algorithm, *Appl. Comput. Harmonic Anal.*, 24 (2008), pp. 131–149.

4. J.-F. Cai, R. H. Chan, and Z. Shen, Linearized Bregman iterations for frame-based image deblurring, *SIAM J. Imaging Sci.*, 2 (2009), pp. 226–252.
5. J.-F. Cai, S. Osher, and Z. Shen, Split Bregman methods and frame based image restoration, *Multiscale Model. Simul.*, 8 (2009), pp. 337–369.
6. R. H. Chan and H. X. Liang, Half-quadratic algorithm for ℓ_p - ℓ_q problems with applications to TV- ℓ_1 image restoration and compressive sensing, in *Proceedings of Efficient Algorithms for Global Optimization Methods in Computer Vision*, Lecture Notes in Comput. Sci. # 8293, Springer, Berlin, 2014, pp. 78–103.
7. J. W. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart, Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization, *Math. Comp.*, 30 (1976), pp. 772–795.
8. M. Donatelli and M. Hanke, Fast nonstationary preconditioned iterative methods for ill-posed problems with application to image deblurring, *Inverse Problems*, 29 (2013), 095008.
9. M. Donatelli, T. Hucke, M. Mazza, and D. Sesana, Image deblurring by sparsity constraint on the Fourier coefficients, *Numer. Algorithms*, 72 (2016), pp. 341–361.
10. H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, Kluwer, Dordrecht, 1996.
11. C. Estatico, S. Gratton, F. Lenti, and D. Titley-Peloquin, A conjugate gradient like method for p -norm minimization in functional spaces, *Numer. Math.*, 137 (2017), pp. 895–922.
12. S. Gazzola and J. G. Nagy, Generalized Arnoldi–Tikhonov method for sparse reconstruction, *SIAM J. Sci. Comput.*, 36 (2014), pp. B225–B247.
13. S. Gazzola, P. Novati, and M. R. Russo, On Krylov projection methods and Tikhonov regularization, *Electron. Trans. Numer. Anal.*, 44 (2015), pp. 83–123.
14. M. Hanke and P. C. Hansen, Regularization methods for large-scale problems, *Surv. Math. Ind.*, 3 (1993), pp. 253–315.
15. M. Hanke and C. W. Groetsch, Nonstationary iterated Tikhonov regularization, *J. Optim. Theory Appl.*, 98 (1998), pp. 37–53.
16. P. C. Hansen, *Rank Deficient and Discrete Ill-Posed Problems*, SIAM, Philadelphia, 1998.
17. P. C. Hansen, J. G. Nagy, and D. P. O’Leary, *Deblurring Images: Matrices, Spectra, and Filtering*, SIAM, Philadelphia, 2006.
18. J.-P. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of Convex Analysis*, Springer, New York, 2004.
19. G. Huang, A. Lanza, S. Morigi, L. Reichel, and F. Sgallari, Majorization-minimization generalized Krylov subspace methods for ℓ_p - ℓ_q optimization applied to image restoration, *BIT Numer. Math.*, 57 (2017), pp. 351–378.
20. G. Huang, L. Reichel, and F. Yin, Projected nonstationary iterated Tikhonov regularization, *BIT Numer. Math.*, 56 (2016), pp. 467–487.
21. J. Huang, M. Donatelli, and R. H. Chan, Nonstationary iterated thresholding algorithms for image deblurring, *Inverse Problems and Imaging*, 7 (2013), pp. 717–736.
22. J. Lampe, L. Reichel, and H. Voss, Large-scale Tikhonov regularization via reduction by orthogonal projection, *Linear Algebra Appl.*, 436 (2012), pp. 2845–2865.
23. A. Lanza, S. Morigi, L. Reichel, and F. Sgallari, A generalized Krylov subspace method for ℓ_p - ℓ_q minimization, *SIAM J. Sci. Comput.*, 37 (2015), pp. S30–S50.
24. P. Rodríguez and B. Wohlberg, Efficient minimization method for a generalized total variation functional, *IEEE Trans. Image Process.*, 18 (2009), pp. 322–332.
25. S. Serra-Capizzano, A note on antireflective boundary conditions and fast deblurring models, *SIAM J. Sci. Comput.*, 25 (2004), pp. 1307–1325.
26. R. Wolke and H. Schwetlick, Iteratively reweighted least squares: algorithms, convergence analysis, and numerical comparisons, *SIAM J. Statist. Comput.*, 9 (1988), pp. 907–921.