Applied Numerical Mathematics 164 (2021) 211–221

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

On the choice of regularization matrix for an ℓ_2-ℓ_q minimization method for image restoration

Alessandro Buccini[a], Guangxin Huang[b,*], Lothar Reichel[c], Feng Yin[d]

[a] Department of Mathematics and Computer Science, University of Cagliari, Via Ospedale 72, 09124 Cagliari, Italy
[b] Department of Information and Computing, Geomathematics Sichuan Key Laboratory, Chengdu University of Technology, Chengdu, 610059, PR China
[c] Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA
[d] School of Mathematics and Statistics, Sichuan University of Science and Engineering, Zigong, 643000, PR China

ARTICLE INFO

Article history:
Received 4 March 2020
Received in revised form 20 October 2020
Accepted 7 November 2020
Available online 12 November 2020

Keywords:
ℓ_2-ℓ_q minimization
Ill-posed problem
Iterative method
Generalized Krylov subspace method
Reordering

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. To reduce this sensitivity, the original problem may be replaced by a minimization problem with a fidelity term and a regularization term. We consider minimization problems of this kind, in which the fidelity term is the square of the ℓ_2-norm of a discrepancy and the regularization term is the qth power of the ℓ_q-norm of the size of the computed solution measured in some manner. We are interested in the situation when $0 < q \leq 1$, because such a choice of q promotes sparsity of the computed solution. The regularization term is determined by a regularization matrix. Novati and Russo let $q = 2$ and proposed in (2014) [13] a regularization matrix that is a finite difference approximation of a differential operator applied to the computed approximate solution after reordering. This gives a Tikhonov regularization problem in general form. We show that this choice of regularization matrix also is well suited for minimization problems with $0 < q \leq 1$. Applications to image restoration are presented.

1. Introduction

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

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

where $A \in \mathbb{R}^{m\times n}$ is a large matrix, whose singular values decrease to zero gradually with no significant gap. In particular, A is severely ill-conditioned and may be rank-deficient. The vector $\mathbf{b}^\delta \in \mathbb{R}^m$ represents measured error-contaminated data. We will refer to the error in $\mathbf{b}^\delta$ as “noise.” The norm $\|\cdot\|_2$ in (1) denotes the Euclidean vector norm. We will allow $m \geq n$ as well as $m < n$. Unicity of the computed solution is secured by adding a regularization term to (1); see below.

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., [7,8] for discussions on ill-posed and discrete ill-posed problems. We are primarily interested in image deblurring. Then

[*] Corresponding author.
E-mail addresses: alessandro.buccini@unica.it (A. Buccini), huangx@cdut.edu.cn (G. Huang), reichel@math.kent.edu (L. Reichel), fyin@suse.edu.cn (F. Yin).

https://doi.org/10.1016/j.apnum.2020.11.004

the vector $\mathbf{b}^\delta$ represents a blur- and noise-contaminated image, and the matrix A is a blurring operator. We would like to determine the unknown blur- and noise-free image associated with the available image represented by $\mathbf{b}^\delta$.

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

$$\|\mathbf{b} - \mathbf{b}^\delta\|_2 \leq \delta. \tag{2}$$

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

Let $A^\dagger$ denote the Moore–Penrose pseudoinverse of A. Then

$$\widehat{\mathbf{x}} := A^\dagger \mathbf{b} \tag{3}$$

represents the blur- and noise-free image associated with $\mathbf{b}^\delta$, which we would like to determine. Since A is ill-conditioned and $\mathbf{b}^\delta$ is contaminated by error, the naïve solution of (1) of minimal Euclidean norm, given by

$$A^\dagger \mathbf{b}^\delta = \widehat{\mathbf{x}} + A^\dagger(\mathbf{b}^\delta - \mathbf{b}),$$

typically is a useless approximation of $\widehat{\mathbf{x}}$, because generally $\|A^\dagger(\mathbf{b}^\delta - \mathbf{b})\|_2 \gg \|\widehat{\mathbf{x}}\|_2$. To achieve a more accurate approximation, the discrete ill-posed problem (1) is replaced by a nearby problem, whose solution is less sensitive to the error in $\mathbf{b}^\delta$. This replacement is known as *regularization*.

A regularization technique that recently has received considerable attention is the replacement of (1) by a minimization problem of the form

$$\mathbf{x}^* := \arg\min_{\mathbf{x} \in \mathbb{R}^n} \mathcal{J}(\mathbf{x}), \tag{4}$$

where

$$\mathcal{J}(\mathbf{x}) := \left\{ \frac{1}{2}\|A\mathbf{x} - \mathbf{b}^\delta\|_2^2 + \frac{\mu}{q}\|L\mathbf{x}\|_q^q \right\} \tag{5}$$

for some $q > 0$. The regularization matrix $L \in \mathbb{R}^{\ell \times n}$ is such that

$$\mathcal{N}(A) \cap \mathcal{N}(L) = \{\mathbf{0}\}. \tag{6}$$

Then there are no nontrivial vectors that are in both $\mathcal{N}(A)$ and $\mathcal{N}(L)$. Here $\mathcal{N}(M)$ denotes the null space of the matrix M. Moreover,

$$\|\mathbf{z}\|_q := \left(\sum_{j=1}^{\ell} |z_j|^q \right)^{1/q}, \qquad \mathbf{z} = [z_1, \ldots, z_\ell]^T \in \mathbb{R}^\ell.$$

We will refer to $\|\mathbf{z}\|_q$ as the q-norm of $\mathbf{z}$ also for $0 < q < 1$, even though the mapping $\mathbf{z} \to \|\mathbf{z}\|_q$ does not satisfy the triangle inequality and, therefore, is not a norm for these q-values. Throughout this paper the superscript ^{T} denotes transposition.

The regularization parameter $\mu > 0$ in (5) balances the relative influence of the first term (the *fidelity term*) and the second term (the *regularization term*). When $q = 2$ the minimization problem (4) is known as a Tikhonov regularization problem in general form [8].

The use of $0 < q < 2$ has received considerable attention; see, e.g., [1–5,9,12] and references therein. Note that if $0 < q < 1$, then the functional (4) generally is not convex. We will determine $\mu > 0$ so that the computed solution $\mathbf{x}_{\text{comp}}$ of (4) satisfies the discrepancy principle, i.e., such that

$$\|A\mathbf{x}_{\text{comp}} - \mathbf{b}^\delta\|_2 = \tau\delta, \tag{7}$$

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

We briefly comment on the choice of q. In many situations it is known that the desired vector (3) is sparse in some basis, i.e., the vector $\widehat{\mathbf{x}}$ has many vanishing coefficients when expressed in a suitable basis. To enhance sparsity in the solution of (4), we use $0 < q \leq 1$. Generally, the smaller $q > 0$, the sparser is the solution and the more difficult is its computation; see [9,12] for illustrations. We remark that the “norm” $\|\mathbf{z}\|_0$, which counts the number of nonvanishing entries of the vector $\mathbf{z}$, promotes sparsity but is difficult to compute with. Therefore, this “norm” generally, is not used.

We will solve the minimization problem (4) by approximating the ℓ_q-norm for some $0 < q \leq 1$ 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. It has been applied in several different situations with good results; see [3,9,12,15,16]. 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 q smaller than unity are described in [3,9,12].

We turn to the choice of the regularization matrix L in (4). The choice of this matrix has not received much attention in the literature. We propose to use a matrix L_1 that represents a discrete first order derivative operator in one space-dimension in combination with a reordering of the unknowns. Let P be a permutation matrix that describes the reordering. Then we let $L := L_1 P$ in (4). An adaptive reordering method is described. The construction of this kind of regularization matrices for the minimization problem (4) is an adaption of a method proposed by Novati and Russo [13] for choosing a regularization matrix for Tikhonov regularization in general form.

This paper is organized as follows: Section 2 outlines a majorization-minimization method for the solution of (4) described in [9]. An algorithm for determining the regularization parameter μ and the regularization matrix L by reordering the entries of the approximate solution are discussed in Section 3, and a few numerical examples are presented in Section 4. Finally, Section 5 contains concluding remarks.

2. A majorization-minimization method

We review a majorization-minimization method for the solution of (4). This method was first described in [9], where it is referred to as the FMM-GKS method. Introduce the function

$$\Phi_{q,\varepsilon}(t) := \begin{cases} |t|^q, & q > 1, \\ \left(t^2 + \varepsilon^2\right)^{q/2}, & 0 < q \le 1, \end{cases} \tag{8}$$

where $\varepsilon > 0$ is a small constant. We will comment on the choice of ε in Section 4. The function $t \to \Phi_{q,\varepsilon}(t)$ is everywhere differentiable for any $q > 0$. This function is used to define a smoothed version of $\|\mathbf{z}\|_q^q$ by

$$\|\mathbf{z}\|_q^q \approx \sum_{i=1}^{\ell} \Phi_{q,\varepsilon}(z_i), \qquad \mathbf{z} = [z_1, z_2, \dots, z_\ell]^T \in \mathbb{R}^\ell.$$

Substituting this smoothed norm into the functional (4) yields

$$\mathcal{J}_\varepsilon(\mathbf{x}) := \frac{1}{2}\|A\mathbf{x} - \mathbf{b}^\delta\|_2^2 + \frac{\mu}{q}\sum_{i=1}^{\ell} \Phi_{q,\varepsilon}((L\mathbf{x})_i). \tag{9}$$

Instead of minimizing (4), we seek to solve

$$\mathbf{x}^* := \arg\min_{\mathbf{x}\in\mathbb{R}^n} \mathcal{J}_\varepsilon(\mathbf{x}). \tag{10}$$

When $q < 1$, the functional (10) is not convex. The majorization-minimization method for the solution of (10) described in [9] constructs a sequence of iterates $\mathbf{x}^{(k)}$, $k = 1, 2, \dots$, that converge to a stationary point of $\mathcal{J}_\varepsilon$. At each step, the functional $\mathcal{J}_\varepsilon$ is majorized by a quadratic functional $\mathbf{x} \to \mathcal{Q}(\mathbf{x}, \mathbf{x}^{(k)})$ that is tangent to $\mathcal{J}_\varepsilon$ at $\mathbf{x}^{(k)}$. Thus, i) $\mathbf{x} \to \mathcal{Q}(\mathbf{x}, \mathbf{x}^{(k)})$ is quadratic, ii) $\mathcal{Q}(\mathbf{x}, \mathbf{x}^{(k)}) \ge \mathcal{J}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^n$, and iii) $\mathcal{Q}(\mathbf{x}^{(k)}, \mathbf{x}^{(k)}) = \mathcal{J}(\mathbf{x}^{(k)})$ and $\nabla_{\mathbf{x}}\mathcal{Q}(\mathbf{x}^{(k)}, \mathbf{x}^{(k)}) = \nabla\mathcal{J}(\mathbf{x}^{(k)})$, where $\nabla_{\mathbf{x}}$ denotes the gradient with respect to the first argument of $\mathcal{Q}$.

We outline the construction of a quadratic tangent majorant at the point $\mathbf{x}^{(k)}$ for $0 < q < 2$. Let

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

and introduce the vector

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

where all the operations, including squaring, are meant element-wise. It is shown in [9] that the functional

$$\mathcal{Q}(\mathbf{x}, \mathbf{x}^{(k)}) := \frac{1}{2}\|A\mathbf{x} - \mathbf{b}^\delta\|_2^2 + \frac{\mu\varepsilon^{q-2}}{2}\left(\|L\mathbf{x}\|_2^2 - 2\langle\boldsymbol{\omega}^{(k)}, L\mathbf{x}\rangle\right) + c, \tag{11}$$

with c a suitable constant that is independent of $\mathbf{x}$, is a quadratic tangent majorant for $\mathcal{J}_\varepsilon$ at $\mathbf{x}^{(k)}$. In (11) and below $\langle\mathbf{u}, \mathbf{v}\rangle := \mathbf{u}^T\mathbf{v}$ denotes the standard inner product of two real vectors $\mathbf{u}$ and $\mathbf{v}$.

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

$$(A^T A + \eta L^T L)\mathbf{x}^{(k+1)} = A^T\mathbf{b}^\delta + \eta L^T\boldsymbol{\omega}^{(k)}, \qquad \eta := \mu\varepsilon^{q-2}. \tag{12}$$

Due to the requirement (6), the above linear system of equations has a unique solution for any $\eta > 0$.

An approximate solution of (12) 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 (12). We determine $\mathbf{x}^{(k+1)}$ as the solution of the minimization problem

$$\mathbf{y}^{(k+1)} := \arg\min_{\mathbf{y}} \left\| \begin{bmatrix} AV_k \\ \eta^{1/2} LV_k \end{bmatrix} \mathbf{y} - \begin{bmatrix} \mathbf{b}^\delta \\ \eta^{1/2}\boldsymbol{\omega}^{(k)} \end{bmatrix} \right\|_2^2 \tag{13}$$

and let

$$\mathbf{x}^{(k+1)} := V_k \mathbf{y}^{(k+1)}. \tag{14}$$

However, instead of explicitly solving the problem (13), we compute the solution of a reduced problem that is obtained by using the QR factorizations

$$\begin{array}{rcll} AV_k & = & Q_A R_A \text{ with } Q_A \in \mathbb{R}^{m\times d}, & R_A \in \mathbb{R}^{d\times d}, \\ LV_k & = & Q_L R_L \text{ with } Q_L \in \mathbb{R}^{\ell\times d}, & R_L \in \mathbb{R}^{d\times d}. \end{array} \tag{15}$$

Thus, the matrices Q_A and Q_L have orthonormal columns and the matrices R_A and R_L are upper triangular. Inserting the factorizations (15) into (13) yields the low-dimensional minimization problem

$$\mathbf{y}^{(k+1)} := \arg\min_{\mathbf{y}} \left\| \begin{bmatrix} R_A \\ \eta^{1/2} R_L \end{bmatrix} \mathbf{y} - \begin{bmatrix} Q_A^T\mathbf{b}^\delta \\ \eta^{1/2} Q_L^T\boldsymbol{\omega}^{(k)} \end{bmatrix} \right\|_2^2, \tag{16}$$

which we solve numerically. The solution $\mathbf{y}^{(k+1)}$ is computed by QR factorization of the matrix

$$\begin{bmatrix} R_A \\ \eta^{1/2} R_L \end{bmatrix}, \tag{17}$$

where we may use the fact that both submatrices R_A and $\eta^{1/2}R_L$ are upper triangular and of the same size.

Substituting (14) into (12) gives the residual vector

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

This expression can be simplified by using the factorizations (15). We expand the solution subspace by including the scaled residual vector $\mathbf{v}_{\text{new}} := \mathbf{r}/\|\mathbf{r}\|$ in it. This vector is orthogonal to the columns of the matrix V_k, and we define the new matrix $V_{k+1} = [V_k, \mathbf{v}_{\text{new}}] \in \mathbb{R}^{n\times(d+1)}$. Its columns form an orthonormal basis for the expanded solution subspace. The solution subspaces determined in this manner are referred to as *generalized Krylov subspaces*.

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. The QR factorizations (15), as well as of (17), can be updated for increasing values of k as described in [6]. It can be shown that the approximate solutions $\mathbf{x}^{(k)}$ for $k = 1, 2, \ldots$, computed in the manner outlined, converge to a stationary point of (9); see [9] for details.

We turn to the determination of the regularization parameter $\mu > 0$. A bound (2) for the error in $\mathbf{b}^\delta$ is assumed to be known. This allows us to determine μ with the aid of the discrepancy principle (7). Thus, for each k, we determine the regularization parameter $\mu = \mu^{(k)}$ so that the computed approximate solution (14) satisfies the discrepancy principle, i.e.,

$$\|A\mathbf{x}^{(k+1)} - \mathbf{b}^\delta\|_2 = \tau\delta. \tag{18}$$

The computation of such a μ-value can be carried out quite efficiently by using the factorizations (15). Details are described in [3]; here we just provide an outline. Substituting (14) into (18) and using (15) gives

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

where we note that the vector $\mathbf{y}^{(k+1)}$, which is computed by solving (16), depends on μ, because η is a function of μ; cf. (12). Define the function

$$\varphi(\mu) = \|R_A\mathbf{y}^{(k+1)} - Q_Q^T\mathbf{b}^\delta\|_2^2$$

and solve

$$\varphi(\mu) = \tau^2\delta^2 \tag{19}$$

by Newton's method or a related root-finder. The regularization parameter μ is updated in each iteration, and this requires the solution of (16). Since the matrices R_A and R_L are fairly small, the computations can be carried out quite rapidly. We denote the solution of (19) by $\mu^{(k+1)}$.

We refer to [7] for a discussion on the discrepancy principle. We remark that other methods for choosing the regularization parameter also can be used; see, e.g., [4,8,10,11,14] for discussions.

The following algorithm outlines the computations for a fixed matrix L and a fixed regularization parameter $\mu > 0$. We will generalize this algorithm in Section 3.

Algorithm 1 *(The MM-GKS-DP method).* Let $0 < q < 2$ be fixed and let the matrices $A \in \mathbb{R}^{m \times n}$ and $L \in \mathbb{R}^{\ell \times n}$ satisfy (6). We are particularly interested in the regularization matrix (23) below. Let $\mathbf{x}^{(0)} = \mathbf{0}$ be the 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 $LV_0 = Q_L R_L$;
for $k = 0, 1, \dots$ do
    $\mathbf{u}^{(k)} = L\mathbf{x}^{(k)}$;
    $\boldsymbol{\omega}^{(k)} = \mathbf{u}^{(k)} \left(1 - \left(\frac{(\mathbf{u}^{(k)})^2 + \varepsilon^2}{\varepsilon^2}\right)^{q/2-1}\right)$;
    $\eta^{(k)} = \mu^{(k)} \varepsilon^{q-2}$;
    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 factorization (15) to obtain $AV_{k+1} = Q_A R_A$ and $LV_{k+1} = Q_L R_L$;
    $\mathbf{x}^{(k+1)} = V_{k+1} \mathbf{y}^{(k+1)}$;
end
```

The iterations with Algorithm 1 are terminated as soon as two consecutive approximate solutions are sufficiently close, i.e., as soon as

$$\frac{\|\mathbf{x}^{(k)} - \mathbf{x}^{(k+1)}\|_2}{\|\mathbf{x}^{(k)}\|_2} \leq 10^{-4}. \tag{20}$$

3. Determining the regularization matrix

This section describes a new method for adaptively determining a regularization matrix L. We first discuss how to determine a permutation matrix in each iteration of Algorithm 1.

Let $P \in \mathbb{R}^{n \times n}$ be a permutation matrix. Replacing $L\mathbf{x}$ by $L\mathbf{y}$ with $\mathbf{y} = P\mathbf{x}$ in (11) gives the functional

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

where $\mathbf{y}^{(k)} = P\mathbf{x}^{(k)}$. The next iterate, $\mathbf{y}^{(k+1)}$, satisfies the linear system of equations

$$(PA^TAP^T + \eta L^T L)\mathbf{y} = PA^T\mathbf{b}^\delta + \eta L^T \boldsymbol{\omega}^{(k)},$$

which is analogous to (12). Equivalently, $\mathbf{x}^{(k+1)}$ satisfies

$$(A^TA + \eta P^T L^T L P)\mathbf{x} = A^T\mathbf{b}^\delta + \eta P^T L^T \boldsymbol{\omega}^{(k)},$$

which are the normal equations associated with the least-squares problem

$$\min_{\mathbf{x} \in \mathbb{R}^n} \left\{ \|A\mathbf{x} - \mathbf{b}^\delta\|_2^2 + \eta \|LP\mathbf{x} - \boldsymbol{\omega}^{(k)}\|_2^2 \right\}. \tag{21}$$

The minimization problem (21) suggests that we determine a permutation matrix P such that the regularization term be small. However, the minimization of this term over all permutations for given vectors $\mathbf{x} \in \mathbb{R}^n$, $\boldsymbol{\omega}^{(k)} \in \mathbb{R}^\ell$, and matrix $L \in \mathbb{R}^{\ell \times n}$ is difficult. We therefore consider the simpler minimization problem

$$\min_{P \in \mathcal{P}} \|LP\mathbf{x}\| \tag{22}$$

for some distance measure $\|\cdot\|$, where $\mathcal{P}$ denotes the set of $n \times n$ permutation matrices. A permutation matrix $P^* \in \mathbb{R}^{n \times n}$ that satisfies (22) is said to be optimal with respect to the matrix L, the vector $\mathbf{x}$, and the norm $\|\cdot\|$. Novati and Russo [13] considered the minimization problem (22) for the Euclidean vector norm, but other vector norms also can be used. We are particularly interested in the situation when the matrix L in (22) is the bidiagonal matrix

$$L_1 := \begin{bmatrix} 1 & -1 & & & & 0 \\ & 1 & -1 & & & \\ & & 1 & -1 & & \\ & & & \ddots & \ddots & \\ 0 & & & & 1 & -1 \end{bmatrix} \in \mathbb{R}^{(n-1)\times n}, \tag{23}$$

which is a scaled finite difference approximation at equidistant points of the first derivative operator in one space-dimension.

Theorem 1. *Let the matrix L in (22) be given by (23), and let the permutation matrix $P^* \in \mathbb{R}^{n\times n}$ order the entries of the vector $\mathbf{x} \in \mathbb{R}^n$ in increasing or decreasing order. Then P^* solves (22) when $\|\cdot\| = \|\cdot\|_1$.*

Proof. Let $\mathbf{y} = [y_1, y_2, \dots, y_{n-1}]^T = LP\mathbf{x}$. Then, by the triangle inequality,

$$|y_1 + y_2 + \ldots + y_{n-1}| \le |y_1| + |y_2| + \ldots + |y_{n-1}| = \|\mathbf{y}\|_1,$$

with equality if all the y_i have the same sign. The latter occurs if the permutation P is such that all the entries of $LP\mathbf{x}$ are nonnegative or all entries are nonpositive. Thus, $\|LP\mathbf{x}\|_1$ achieves its lower bound if the permutation P orders the elements of $\mathbf{x}$ in increasing order or in decreasing order. □

The well-known relations

$$\|\mathbf{y}\|_2 \le \|\mathbf{y}\|_1 \le (n-1)^{1/2}\|\mathbf{y}\|_2, \qquad \mathbf{y} \in \mathbb{R}^{n-1}$$

suggests that the permutation P^* of Theorem 1 is suitable also when $\|\cdot\|$ is the Euclidean norm.

Theorem 2. *Let $\mathbf{y} = [y_1, y_2, \dots, y_{n-1}]^T = LP\mathbf{x}$ and consider the distance measure*

$$\|\mathbf{y}\| = \left(\sum_{i=1}^{n-1} |y_i|^q\right)^{1/q}$$

for some $q > 0$. Assume that all entries y_i are nonvanishing. Then $\|\mathbf{y}\|$ achieves its minimum when all $|y_i|$ are equal.

Proof. We will show the theorem by using a Lagrange multiplier. Let $z_i = |y_i|$. Since $z_i > 0$ for all i, we may scale all z_i by the same positive factor so that the scaled z_i satisfy

$$\prod_{i=1}^{n-1} z_i = 1. \tag{24}$$

We will assume this scaling has been carried out and consider the function

$$f(z_1, z_2, \dots, z_{n-1}) = z_1^q + z_2^q + \ldots + z_{n-1}^q + \lambda z_1 z_2 \ldots z_{n-1}.$$

The partial derivatives are given by

$$\frac{\partial f}{\partial z_i}(z_1, z_2, \dots, z_{n-1}) = q z_i^{q-1} + \lambda z_1 \ldots z_{i-1} z_{i+1} \ldots z_{n-1}, \qquad i = 1, 2, \dots, n-1.$$

Setting the partial derivatives to zero and multiplying by $z_i > 0$ gives

$$q z_i^q + \lambda z_1 z_2 \ldots z_{n-1} = q z_i^q + \lambda = 0, \qquad i = 1, 2, \dots, n-1.$$

We conclude that

$$z_i = (-\lambda/q)^{1/q}, \qquad i = 1, 2, \dots, n-1. \tag{25}$$

Thus, all the z_i are equal. This is the unique stationary point of the function f in its domain. Let $z_1 = t$ and $z_j = 1/t^{1/(n-2)}$ for $j = 2, 3, \dots, n-1$ and $t > 0$. Then these z_j satisfy the constraint (24) and it follows that $f(z_1, z_2, \dots, z_{n-1}) \to \infty$ as $t \to \infty$. We can similarly let any subset of at most $n-2$ variables z_j approach infinity so that the constraint (24) is satisfied. This shows that the function f is coercive. It therefore has (at least) one local minimum in its domain. It follows that the point defined by (25) therefore is the unique minimum of the function f under the constraint (24). The measure $\|\mathbf{z}\|$, and therefore $\|\mathbf{y}\|$, achieve their minima when f is minimal. We conclude that $\|\mathbf{y}\|$ is minimal when all nonvanishing $|y_i|$ are the same. This occurs, for instance, when the entries x_i of $\mathbf{x}$, for $i = 1, 2, \dots, n$, are equidistant. □

The vector $\mathbf{y} = [y_1, y_2, \ldots, y_{n-1}]^T = P^* L\mathbf{x}$, with the permutation P^* of Theorem 1, generally does not satisfy $|y_i| = |y_j|$ for all $1 \leq i, j < n$. Nevertheless, Theorem 2 indicates that the permutation P^* may be suitable to apply when the entries of $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ may be considered piece-wise smooth functions of their index, with only a few jump discontinuities.

Algorithm 2 solves the minimization problem (22) for every iterate $\mathbf{x} = \mathbf{x}^{(k+1)}$ generated during the execution of the algorithm. The initial permutation P_0 is the identity matrix. We refer to Algorithm 2 as the reordered (RR) MM-GKS-DP algorithm, because it complements the computations of Algorithm 1 with reordering of the entries of the approximate solutions determined in the inner loop.

Algorithm 2 *(The RR-MM-GKS-DP method).* Let $0 < q < 2$ be fixed and let the matrices $A \in \mathbb{R}^{m\times n}$ and $L_1 \in \mathbb{R}^{\ell\times n}$ satisfy (6). Let $\mathbf{x}_0 \in \mathbb{R}^n$ be an initial approximate solution, and let $\varepsilon > 0$, s and $k_0 > 0$ be constants.

```
for $t = 0, 1, \ldots$ do
    Compute $\mathbf{b}_t = \mathbf{b} - A\mathbf{x}_t$, let $\mathbf{x}_t^{(0)} = \mathbf{x}_t$;
    Generate the initial subspace basis on $A$ and $\mathbf{b}_t$: $V_0 \in \mathbb{R}^{n\times k_0}$ such that $V_0^T V_0 = I$;
    Define $L = L_1 P_t$;
    Compute the QR factorizations $AV_0 = Q_A R_A$ and $LV_0 = Q_L R_L$;
    for $k = 0, 1, \ldots$ do
        $\mathbf{u}^{(k)} = L\mathbf{x}_t^{(k)}$;
        $\boldsymbol{\omega}^{(k)} = \mathbf{u}^{(k)}\left(1 - \left(\frac{(\mathbf{u}^{(k)})^2+\varepsilon^2}{\varepsilon^2}\right)^{q/2-1}\right)$;
        $\eta_t^{(k)} = \mu_t^{(k)}\varepsilon^{q-2}$;
        where $\mu_t^{(k)}$ is such that $\|AV_k\mathbf{y}_t^{(k+1)} - \mathbf{b}^\delta\|_2 = \tau\delta$;
        $\mathbf{y}_t^{(k+1)} = (R_A^T R_A + \eta_t^{(k)} R_L^T R_L)^{-1}(R_A^T Q_A^T \mathbf{b}^\delta + \eta_t^{(k)} R_L^T Q_L^T \boldsymbol{\omega}^{(k)})$;
        $\mathbf{r} = A^T(AV_k\mathbf{y}_t^{(k+1)} - \mathbf{b}^\delta) + \eta L^T(LV_k\mathbf{y}_t^{(k+1)} - \boldsymbol{\omega}^{(k)})$;
        Reorthogonalize, if needed, $\mathbf{r} = \mathbf{r} - V_k V_k^T \mathbf{r}$;
        $\mathbf{v}_{\text{new}} = \mathbf{r}/\|\mathbf{r}\|$; $V_{k+1} = [V_k, \mathbf{v}_{\text{new}}]$;
        Update (15) to obtain $AV_{k+1} = Q_A R_A$ and $LV_{k+1} = Q_L R_L$;
        $\mathbf{x}_t^{(k+1)} = V_{k+1}\mathbf{y}_t^{(k+1)}$;
    end
    $\mathbf{x}_{t+1} = \mathbf{x}_t^{(k+1)}$;
    Determine new permutation matrix $P_{t+1}$;
end
```

The iterations in the inner loop (the k-loop) of Algorithm 2 are terminated as soon as two consecutive approximate solutions are sufficiently close, i.e., as soon as (20) holds. Then a new ordering of the entries of the available approximate solution $\mathbf{x}^{(k+1)}$ (i.e., a new permutation matrix P_{t+1}) is determined, and the computations with Algorithm 2 are continued. The outer loop is exited as soon as two consecutive approximate solutions $\mathbf{x}_{t+1}$ and $\mathbf{x}_t$ are sufficiently close, i.e., as soon as

$$\frac{\|\mathbf{x}_t - \mathbf{x}_{t+1}\|_2}{\|\mathbf{x}_t\|_2} \leq 10^{-4}. \tag{26}$$

This stopping criterion is analogous to (20).

4. Numerical examples

This section presents two image restoration examples that compare Algorithms 1 and 2. The examples illustrate that using the ordering of the entries of the computed solution described in Section 3 improves the quality of the computed restorations, when compared with the standard ordering of Algorithm 1 of Section 2.

The error $\mathbf{e} = \mathbf{b}^\delta - \mathbf{b}$ in the vector $\mathbf{b}^\delta$ in (4) models zero mean white Gaussian noise. We refer to the quotient

$$\sigma := \frac{\|\mathbf{e}\|}{\|\mathbf{b}\|}$$

as the *noise level*, and set $\delta := \sigma\|\mathbf{b}\|$ in (7) and in Algorithms 1 and 2. We let $\tau = 1.01$ in the algorithms. The parameter ε in (8) is set to one, as suggested in [3]. This choice of ε is appropriate for image restoration problems, because it is much smaller than the largest pixel value, which is 255. Choosing $0 < \varepsilon < 1$ does not improve the quality of the computed restorations, but increases the number of iterations required by the Algorithms 1 and 2, and therefore increases the computing time. Other applications of the minimization problem (4), such applications to linear regression described in [1], may require $0 < \varepsilon < 1$ to furnish good results.

Table 1
Example 4.1: Comparison of Algorithms 1 and 2 applied to the ℓ_2-ℓ_1 and ℓ_2-$\ell_{0.5}$ models for the restoration of MRI test images that have been corrupted by motion blur and Gaussian noise.

Noise	Iteration number		μ		Relative error		CPU time	
%	Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2
ℓ_2-$\ell_{0.5}$								
0.01	106	186	$4.9 \cdot 10^{-1}$	$7.0 \cdot 10^{-1}$	$1.10 \cdot 10^{-1}$	$1.02 \cdot 10^{-1}$	$4.99 \cdot 10^{1}$	$3.80 \cdot 10^{1}$
0.001	181	157	$1.9 \cdot 10^{-2}$	$6.6 \cdot 10^{-3}$	$5.25 \cdot 10^{-2}$	$4.63 \cdot 10^{-2}$	$2.26 \cdot 10^{2}$	$7.59 \cdot 10^{1}$
ℓ_2-ℓ_1								
0.01	74	186	$9.7 \cdot 10^{-3}$	$7.0 \cdot 10^{-1}$	$1.20 \cdot 10^{-1}$	$1.03 \cdot 10^{-1}$	$1.06 \cdot 10^{2}$	$2.94 \cdot 10^{1}$
0.001	181	186	$6.0 \cdot 10^{-4}$	$8.2 \cdot 10^{-2}$	$6.57 \cdot 10^{-2}$	$4.80 \cdot 10^{-2}$	$1.49 \cdot 10^{2}$	$6.03 \cdot 10^{1}$

The initial regularization matrix L in the algorithms is (23). We set the maximal number of iteration with Algorithm 1 and the maximal number of inner iterations with Algorithm 2 to 30, and the maximal number of outer iterations in the latter algorithm to 6. These choices do not affect the quality of the computed restorations significantly. In all examples we let the columns of $V_0 \in \mathbb{R}^{n \times k_0}$ in Algorithms 1 and 2 be the basis for the initial Krylov subspace

$$\mathcal{K}_{k_0}(A^T A, A^T \mathbf{b}_0) = \text{span}\{A^T \mathbf{b}_0, (A^T A) A^T \mathbf{b}_0, \ldots, (A^T A)^{k_0 - 1} A^T \mathbf{b}_0\} \tag{27}$$

with $k_0 = 10$, where $\mathbf{b}_0 = \mathbf{b}^\delta$ in Algorithm 1 or $\mathbf{b}_0 = \mathbf{b}_t$ in Algorithm 2.

We measure the quality of the restored image, represented by the vector $\mathbf{x}_{\text{comp}}$, with the relative restoration error

$$e_k := \frac{\|\mathbf{x}_{\text{comp}} - \widehat{\mathbf{x}}\|_2}{\|\widehat{\mathbf{x}}\|_2},$$

where the vector $\widehat{\mathbf{x}}$ represents the desired blur- and noise-free image. The computational efficiency is measured in terms of the total CPU time required by the algorithms to satisfy the stopping criterion. All experiments were carried out in MATLAB with about 15 significant decimal digits on an Intel Core i5-3230M @ 2.60 GHz computer with 8 GB RAM.

Example 4.1. We consider the restoration of the test image MRI, which is represented by an array of 256 × 256 pixels. The model (9) with $q = 1$ or $q = 0.5$ is applied. The pixels of the available blur- and noise-contaminated image are stored column-wise in the vector $\mathbf{b}^\delta \in \mathbb{R}^{256^2}$. The blurring matrix $A \in \mathbb{R}^{256^2 \times 256^2}$, which models horizontal motion blur is a block Toeplitz matrix defined by

$$A = I_{256} \otimes B,$$

where $\otimes$ stands for the Kronecker product, and $B = [b_{ij}] \in \mathbb{R}^{256 \times 256}$ is a symmetric banded matrix with half-bandwidth $d = 15$ and nontrivial entries

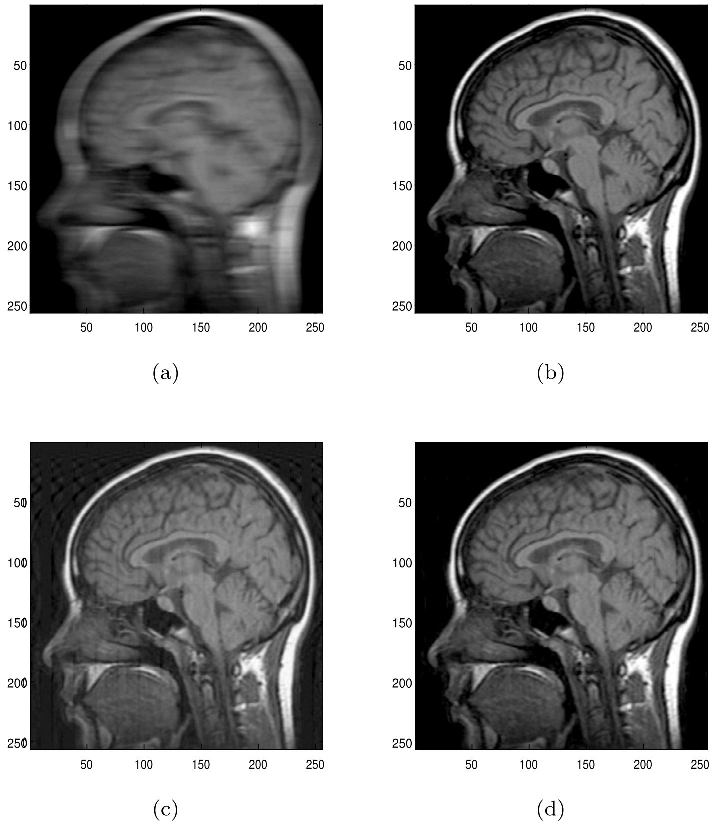
$$b_{ij} = \begin{cases} \frac{1}{2d-1}, & |i - j| \le d, \\ 0, & \text{otherwise.} \end{cases}$$

We show results for the noise levels $\nu = 1 \cdot 10^{-j}$, $j = 2, 3$. The available blur- and noise-contaminated image is displayed in Fig. 1(a) with noise corresponding to the noise level $\nu = 1 \cdot 10^{-3}$. Fig. 1(b) shows the desired blur- and noise-free image, which is represented by the vector $\widehat{\mathbf{x}} \in \mathbb{R}^{256^2}$. It is assumed not to be known.

Table 1 displays the regularization parameter values μ determined by Algorithms 1 and 2, as well as the total number of iterations, the relative error e_k in the computed restorations, and the CPU time required for two noise levels and two values of q. The iterations with the algorithms are terminated as soon as the stopping criteria (20) or (26) are satisfied. The table shows Algorithm 2 to determine approximations of $\widehat{\mathbf{x}}$ of higher quality with less CPU time than Algorithm 1 for all noise levels. We remark that the CPU time is not proportional to the number of iterations, because the count of the latter includes both inner and outer iterations; the inner iterations for the zero-finder are less expensive than the outer iterations that expand the solution subspace. Figs. 1(c) and (d) display the computed approximate solutions determined by Algorithm 1 and Algorithm 2, respectively, when the noise level in the available image $\mathbf{b}^\delta$ is $\nu = 1 \cdot 10^{-3}$.

Fig. 2 compares the relative error as a function of the iteration number in restorations computed with Algorithms 1 and 2 when $q = 1$ for the noise levels $\nu = 1 \cdot 10^{-j}$, $j = 2, 3$. For the purpose of comparison, we do not stop the iteration when the stopping criterion (20) and (26) are satisfied. The figure shows Algorithm 2 to determine restorations of higher quality than Algorithm 1 for both noise levels. □

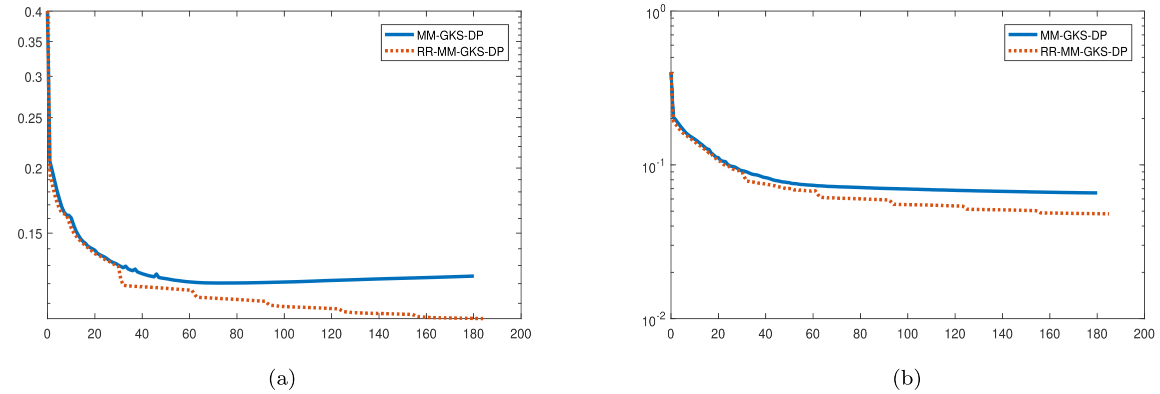
Example 4.2. We consider the restoration of the test image QRcode, which is represented by an array of 256 × 256 pixels. We let $q = 0.5$ or $q = 1$ in (9). The available image is represented by the vector $\mathbf{b}^\delta \in \mathbb{R}^{256^2}$. It is corrupted by motion



(a) (b)

(c) (d)

Fig. 1. Example 4.1: (a) Available blur- and noise-contaminated `MRI` image represented by the matrix b^δ, (b) desired image, restored images by (c) Algorithm 1, and (d) Algorithm 2.



(a) (b)

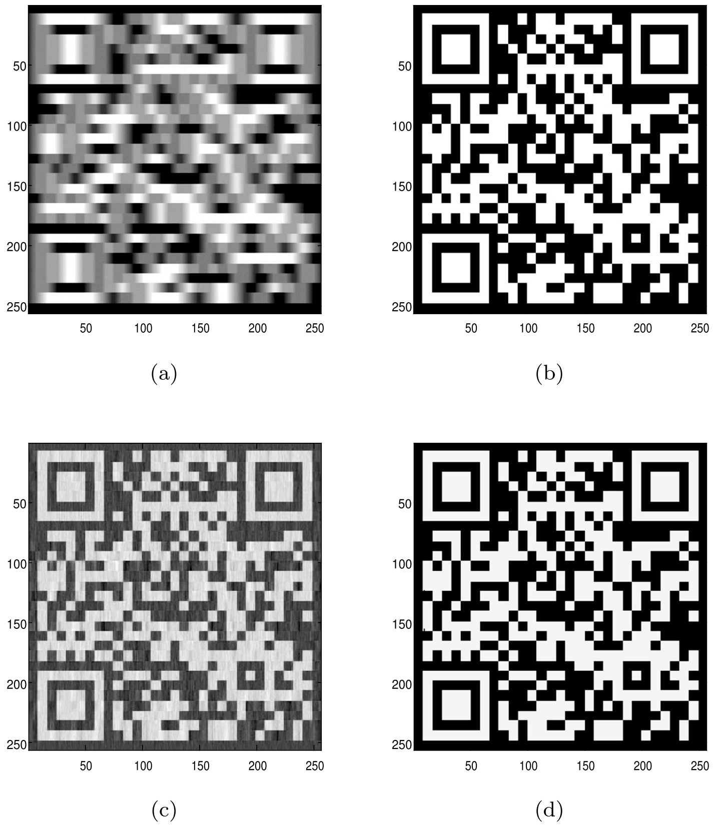
Fig. 2. Example 4.1: Convergence of the logarithm of the relative errors e_k as a function of the iteration number k in restorations determined by the MM-GKS-DP algorithm (Algorithm 1) and the RR-MM-GKS-DP algorithm (Algorithm 2) for ℓ_2-ℓ_1 minimization model with noise levels (a) $\nu = 1 \cdot 10^{-2}$ and (b) $\nu = 1 \cdot 10^{-3}$.

blur and additive zero-mean white Gaussian noise. The blurring matrix A is of the same type as in Example 4.1 with half-bandwidth $d = 15$. The contaminated image is determined similarly as in Example 4.1; it is shown in Fig. 3(a). Fig. 3(b) displays the desired blur- and noise-free image. It is represented by the vector $\widehat{\mathbf{x}} \in \mathbb{R}^{256^2}$, and is assumed not to be known. We computed restorations for the noise levels $\nu = 1 \cdot 10^{-j}$, $j = 1, 2, 3$.

Table 2 displays the values of the regularization parameter μ determined by the algorithms, the relative errors e_k in the computed approximate solutions, $\mathbf{x}_{\text{comp}}$, determined by Algorithms 1 or 2, and the CPU times required. The iterations are terminated as soon as the stopping criteria (20) or (26) are satisfied. Table 2 shows Algorithm 2 to determine approximations of $\widehat{\mathbf{x}}$ of higher quality with less CPU time than Algorithm 1. Figs. 3(c) and (d) depict the computed restorations determined by Algorithms 1 and 2, respectively, when $\nu = 1 \cdot 10^{-3}$.

Table 2
Example 4.2: Comparison of Algorithms 1 and 2 with $q = 0.5$ and $q = 1$ applied to the restoration of the `QRcode` test images that are corrupted by motion blur and Gaussian noise.

Noise	Iteration number		μ		Relative error		CPU time	
%	Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2	Algorithm 1	Algorithm 2
ℓ_2-$\ell_{0.5}$								
0.1	13	40	$1.2 \cdot 10^{-2}$	$4.9 \cdot 10^{-1}$	$3.02 \cdot 10^{-1}$	$2.47 \cdot 10^{-1}$	$5.25 \cdot 10^{1}$	$2.52 \cdot 10^{1}$
0.01	91	34	$6.8 \cdot 10^{-3}$	$4.9 \cdot 10^{-1}$	$1.51 \cdot 10^{-1}$	$1.83 \cdot 10^{-2}$	$1.06 \cdot 10^{2}$	$3.23 \cdot 10^{1}$
0.001	166	34	$3.3 \cdot 10^{-3}$	$7.0 \cdot 10^{-1}$	$1.24 \cdot 10^{-1}$	$5.60 \cdot 10^{-3}$	$1.20 \cdot 10^{2}$	$3.48 \cdot 10^{1}$
ℓ_2-ℓ_1								
0.1	14	41	$1.2 \cdot 10^{-1}$	$7.0 \cdot 10^{-1}$	$3.03 \cdot 10^{-1}$	$2.52 \cdot 10^{-1}$	$5.47 \cdot 10^{1}$	$2.52 \cdot 10^{1}$
0.01	107	36	$6.8 \cdot 10^{-3}$	$2.4 \cdot 10^{-1}$	$1.48 \cdot 10^{-1}$	$2.30 \cdot 10^{-2}$	$1.07 \cdot 10^{2}$	$3.74 \cdot 10^{1}$
0.001	180	34	$3.3 \cdot 10^{-1}$	$3.4 \cdot 10^{-1}$	$1.17 \cdot 10^{-1}$	$7.50 \cdot 10^{-3}$	$1.24 \cdot 10^{2}$	$3.78 \cdot 10^{1}$



(a) (b)

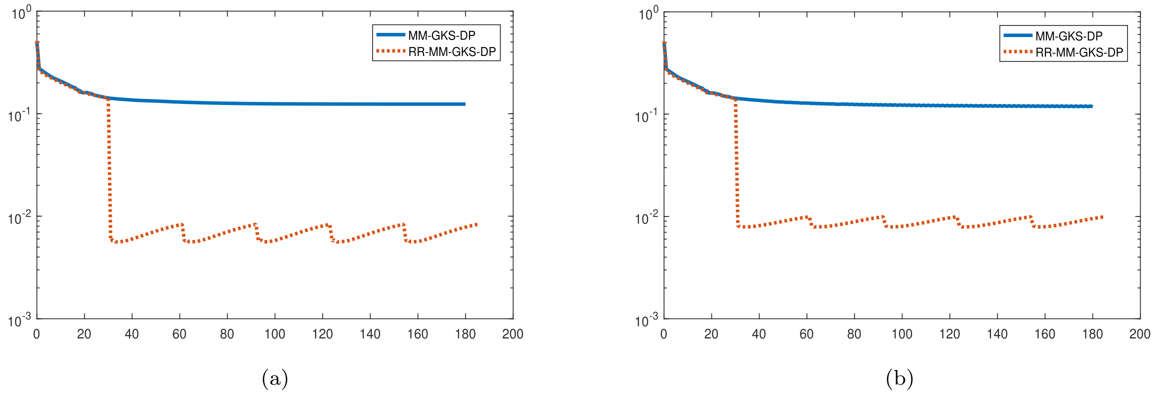
(c) (d)

Fig. 3. Example 4.2: (a) Available blur- and noise-contaminated `QRcode` image represented by the vector $\mathbf{b}^\delta$, (b) desired image, restored images determined by (c) Algorithm 1, and (d) Algorithm 2.

Fig. 4 shows $\log_{10}(e_k)$ for the restorations determined at step k of the algorithms with applied for noise level $\nu = 1 \cdot 10^{-3}$ with $q = 0.5$ and $q = 1$ in (9). The figures show the RR-MM-GKS-DP algorithm (Algorithm 2) to determine restorations of higher quality than the MM-GKS-DP algorithm (Algorithm 1). □

5. Conclusions

We apply a method for determining the regularization matrix based on reordering of the unknowns, proposed by Novati and Russo [13] for Tikhonov regularization, in a restarted ℓ_2-ℓ_q minimization method for image restoration. The restorations determined in this manner are of higher quality than when no reordering is carried out. The regularization parameter is determined by the discrepancy principle. This makes it possible to apply the method described without user interaction.



(a)

(b)

Fig. 4. Example 4.2: The logarithm of the relative error e_k as a function of the iteration number k for the MM-GKS-DP algorithm (Algorithm 1) and the RR-MMGKS-DP algorithm (Algorithm 2) with (a) $q = 0.5$ and (b) $q = 1$. The noise level in the contaminated image is $\nu = 1 \cdot 10^{-3}$.

Acknowledgements

The authors would like to thank the referees for comments. A.B. is a member of the GNCS-INdAM group that partially supported this work with the Young Researchers Project (Progetto Giovani Ricercatori) “Variational methods for the approximation of sparse data”. Moreover, 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). Research by G.H. was supported in part by Application Fundamentals Foundation of Science and Technology Department of Sichuan (2020YJ0366) and Key Laboratory of Bridge Nondestructive Testing and Engineering Calculation Open Fund projects (2020QZJ03), research by L.R. was partially supported by NSF (grants DMS-1729509 and DMS-1720259), and research by F.Y. was partially supported by NNSF (grant 11501392) and SUSE (grant 2019RC09).

References

[1] A. Buccini, O. De la Cruz Cabrera, M. Donatelli, A. Martinelli, L. Reichel, Large scale regression with non-convex loss and penalty, Appl. Numer. Math. 157 (2020) 590–601.

[2] A. Buccini, M. Pasha, L. Reichel, Modulus-based iterative methods for constrained ℓ_2-ℓ_q minimization, Inverse Probl. 36 (2020) 084001.

[3] A. Buccini, L. Reichel, An ℓ_2-ℓ_q regularization method for large discrete ill-posed problems, J. Sci. Comput. 78 (2019) 1526–1549.

[4] A. Buccini, L. Reichel, An ℓ^2-ℓ^q minimization method with cross-validation for the restoration of impulse noise contaminated images, J. Comput. Appl. Math. 375 (2020) 112824.

[5] R.H. Chan, 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, in: Lecture Notes in Comput. Sci., vol. 8293, Springer, Berlin, 2014, pp. 78–103.

[6] J.W. Daniel, W.B. Gragg, L. Kaufman, G.W. Stewart, Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization, Math. Comput. 30 (1976) 772–795.

[7] H.W. Engl, M. Hanke, A. Neubauer, Regularization of Inverse Problems, Kluwer, Dordrecht, 1996.

[8] P.C. Hansen, Rank Deficient and Discrete Ill-Posed Problems, SIAM, Philadelphia, 1998.

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

[10] S. Kindermann, Convergence analysis of minimization-based noise level-free parameter choice rules for linear ill-posed problems, Electron. Trans. Numer. Anal. 38 (2011) 233–257.

[11] S. Kindermann, K. Raik, A simplified L-curve method as error estimator, Electron. Trans. Numer. Anal. 53 (2020) 217–238.

[12] A. Lanza, S. Morigi, L. Reichel, F. Sgallari, A generalized Krylov subspace method for ℓ_p-ℓ_q minimization, SIAM J. Sci. Comput. 37 (2015) S30–S50.

[13] P. Novati, M.R. Russo, Adaptive Arnoldi–Tikhonov regularization for image restoration, Numer. Algorithms 65 (2014) 745–757.

[14] L. Reichel, G. Rodriguez, Old and new parameter choice rules for discrete ill-posed problems, Numer. Algorithms 63 (2013) 65–87.

[15] P. Rodríguez, B. Wohlberg, Efficient minimization method for a generalized total variation functional, IEEE Trans. Image Process. 18 (2009) 322–332.

[16] R. Wolke, H. Schwetlick, Iteratively reweighted least squares: algorithms, convergence analysis, and numerical comparisons, SIAM J. Sci. Stat. Comput. 9 (1988) 907–921.