Generalized Cross Validation for ℓ^p-ℓ^q Minimization

Alessandro Buccini · Lothar Reichel

Abstract Discrete ill-posed inverse problems arise in various areas of science and engineering. The presence of noise in the data often makes it difficult to compute an accurate approximate solution. To reduce the sensitivity of the computed solution to the noise, one replaces the original problem by a nearby well-posed minimization problem, whose solution is less sensitive to the noise in the data than the solution of the original problem. This replacement is known as regularization. We consider the situation when the minimization problem consists of a fidelity term, that is defined in terms of a p-norm, and a regularization term, that is defined in terms of a q-norm. We allow $0 < p, q \leq 2$. The relative importance of the fidelity and regularization terms is determined by a regularization parameter. This paper develops an automatic strategy for determining the regularization parameter for these minimization problems. The proposed approach is based on a new application of generalized cross validation. Computed examples illustrate the performance of the method proposed.

Keywords ℓ^p-ℓ^q minimization, regularization parameter, generalized cross validation, inverse problem, iterative method

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

A. Buccini
Department of Mathematics and Computer Science
University of Cagliari
09124 Cagliari, Italy
E-mail: alessandro.buccini@unica.it
L. Reichel
Department of Mathematical Sciences
Kent State University
Kent, 44242 OH
E-mail: reichel@math.kent.edu

1 Introduction

In many areas of science and engineering one is faced with the problem of having to compute a meaningful approximate solution, defined in an appropriate way, of linear systems of equations of the form

$$A\mathbf{x} \approx \mathbf{b}, \tag{1}$$

where $A \in \mathbb{R}^{m \times n}$ is a given matrix, $\mathbf{x} \in \mathbb{R}^n$ is the desired solution, and $\mathbf{b} \in \mathbb{R}^m$ is a data vector. The symbol $\approx$ indicates that we would like to determine a vector $\mathbf{x}$ such that $A\mathbf{x}$ approximates $\mathbf{b}$ in a suitable way. We consider the case when the singular values of A decrease to zero quickly without a significant gap. Then the matrix A is severely ill-conditioned and, moreover, it is not meaningful to define the rank of A. Approximation problems (1) with a matrix of this kind are commonly referred to as *discrete ill-posed problems*; see, e.g., [17, 26, 27] for discussions.

In applications of interest to us, the data vector $\mathbf{b}$ is not available; instead a measured vector $\mathbf{b}^\delta \in \mathbb{R}^m$, which is contaminated by noise $\boldsymbol{\eta} \in \mathbb{R}^m$, is known, i.e.,

$$\mathbf{b}^\delta = [b_1^\delta, b_2^\delta, \ldots, b_m^\delta]^T = \mathbf{b} + \boldsymbol{\eta}$$

replaces the vector $\mathbf{b}$ in (1). Here and throughout this paper the superscript ^{T} denotes transposition. We assume that $\boldsymbol{\eta}$ is made up of white Gaussian noise, impulse noise and/or salt and pepper noise. Impulse noise modifies only some entries of $\mathbf{b}$ and leaves the other entries unchanged. In detail, we have for $1 \le i \le m$,

$$b_i^\delta = \begin{cases} d_i & \text{with probability } \sigma, \\ b_i & \text{with probability } 1 - \sigma, \end{cases}$$

where the d_i are realizations of a random variable with uniform distribution in the interval $[d_{\min}, d_{\max}]$, which is the dynamic range of b_i. If $d_i \in \{d_{\min}, d_{\max}\}$, i.e., if the d_i only attain their maximal or minimal achievable values, then impulse noise is commonly referred to as salt-and-pepper noise. Impulse noise simulates the effect of broken sensors on a measuring device such as a CCD camera.

Since A is ill-conditioned and $\mathbf{b}^\delta$ is contaminated by noise, the least-squares solution $A^\dagger \mathbf{b}^\delta$ of minimal norm of (1) with $\mathbf{b}$ replaced by $\mathbf{b}^\delta$ is meaningless due to propagation and severe amplification of the error $\boldsymbol{\eta}$ in $\mathbf{b}^\delta$ into the solution; here $A^\dagger$ denotes the Moore-Penrose pseudo-inverse of A. To reduce the propagated error in the computed solution, one typically modifies the problem to be solved. This modification is commonly referred to as *regularization*.

A regularization method that recently has received considerable attention is the ℓ^p-ℓ^q minimization method; see [14, 18, 30, 35]. This method solves the minimization problem

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

where $L \in \mathbb{R}^{r \times n}$ is a regularization matrix, $\mu > 0$ a regularization parameter, and $\|\mathbf{z}\|_p^p = \sum_{i=1}^k |z_i|^p$ for $\mathbf{z} = [z_1, z_2, \ldots, z_k]^T \in \mathbb{R}^k$ for $k \in \{m, n\}$. We are interested in parameters p and q in the interval $(0, 2]$. Observe that $\mathbf{z} \mapsto \|\mathbf{z}\|_p$ is a norm only if $p \geq 1$; however, for notational convenience, we will refer to the quantity $\|\mathbf{z}\|_p$ as an ℓ^p-norm also for $0 < p < 1$.

The regularization parameter $\mu > 0$ determines the trade-off between the the first and second terms in (2), and decides how sensitive the solution of (2) is to the error $\boldsymbol{\eta}$ in $\mathbf{b}^\delta$. Moreover, the choice of μ affects how close the solution is to the desired vector $\mathbf{x}_{\text{true}} = A^\dagger \mathbf{b}$. An imprudent choice of μ may result in that the solution of (2) is a poor approximation of $\mathbf{x}_{\text{true}}$. It therefore is important to develop methods that are able to determine a suitable value of μ. This value will depend on the matrices A and L, the vectors $\mathbf{b}^\delta$ and $\boldsymbol{\eta}$, as well as the parameters p and q. If $p = q = 2$, then the problem (2) reduces to Tikhonov regularization in general form; see, e.g., [17,22,27,38] for discussions on Tikhonov regularization.

We now briefly discuss different choices for p and q, and first comment on the choice of q, which affects the second term in (2). This terms is commonly referred to as the *regularization term*. In many situations the desired solution $\mathbf{x}_{\text{true}}$ of (1) is sparse after some transformation. For instance, if we represent $\mathbf{x}_{\text{true}}$ in terms of framelets or wavelets, then $\mathbf{x}_{\text{true}}$ typically has a sparse representation, i.e., many coefficients in this representation vanish. Moreover, when L is a discretized gradient operator, the vector $L\mathbf{x}_{\text{true}}$ generally has many vanishing entries. To promote sparsity of the computed solution of (2), we may consider letting $L = I$ and $q = 0$. Then the regularization term measures the size of the computed solution by the ℓ^0-norm. This "norm" counts the number of nonzero entries in the vector $\mathbf{x}$ in (2). Note that this norm is not convex. Similarly, to promote sparsity in the vector $L\mathbf{x}$, we may consider letting $q = 0$ in (2). However, the minimization problems so obtained are extremely difficult to solve. Therefore, it is common to approximate the ℓ^0-norm by the ℓ^1-norm. This approximation has the advantage that the ℓ^1-norm is convex, which makes it easier to solve (2) than when using the ℓ^0-norm. However, ℓ^q-norms with $0 < q < 1$ are better approximations of the ℓ^0-norm than the ℓ^1-norm. In particular, smaller values of $q > 0$ yield better approximations of the ℓ^0-norm than larger values; see Figure 1 for an illustration. For $0 < q < 1$ the resulting minimization problem (2) is not convex and its solution may be difficult to determine; see Lanza et al. [36] for a recent discussion on the choice of q in the context of image restoration.

The choice of p affects the first term in (2), known as the *fidelity term*, and should depend on the type of noise $\boldsymbol{\eta}$ in the data vector $\mathbf{b}^\delta$. If the noise $\boldsymbol{\eta}$ is white Gaussian, then $p = 2$ is appropriate and a method for determining μ for this kind of noise, based on the discrepancy principle, is described in [7]. This method requires that a fairly accurate estimate of the norm of $\boldsymbol{\eta}$ be available and allows $0 < q < 1$. However, for impulse noise, choosing $p = 2$ usually produces computed solutions of poor quality. It has been shown, see, e.g., [8,30,35], that the choice $0 < p < 1$ leads to accurate restorations in the case of salt-and-pepper noise. In [8], the authors developed two strategies based
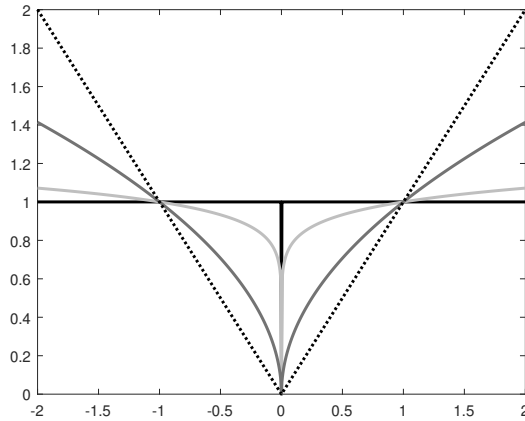
Fig. 1: Comparison of different ℓ^q-norms. The solid black graph represents the ℓ^0-norm, the dotted black graph shows the ℓ^1-norm, the dark gray solid graph displays the $\ell^{0.5}$-norm, and the light gray solid graph depicts the $\ell^{0.1}$-norm.

on cross-validation for determining a suitable value of μ for any $0 < p \leq 2$ without any knowledge of the noise. This method is further commented on below.

Let $\mathcal{N}(M)$ denote the null space of the matrix M. We will assume that the matrices A and L are such that

$$\mathcal{N}(A) \cap \mathcal{N}(L) = \{\mathbf{0}\}. \tag{3}$$

Then the matrix $A^T A + \mu L^T L$ is nonsingular for any $\mu > 0$, and the minimization problem (2) for $p = q = 2$ has the unique solution

$$\mathbf{x}_\mu = (A^T A + \mu L^T L)^{-1} A^T \mathbf{b}^\delta. \tag{4}$$

We remark that both the selection of L and μ are important for the quality of the computed solution and have been widely discussed; see, e.g., [15, 19, 31–33, 37].

Generalized cross validation (GCV) introduced by Golub et al. [23] is a popular approach to choosing the regularization parameter for Tikhonov regularization; more recent discussions on this method can be found in [19, 20, 24, 29]. The GCV method is statistically based and chooses a regularization parameter that minimizes the GCV functional

$$G(\mu) = \frac{\left\| \mathbf{b}^\delta - A\mathbf{x}_\mu \right\|_2^2}{\left(\mathsf{trace} \left(I - A \left(A^T A + \mu L^T L \right)^{-1} A^T \right) \right)^2}. \tag{5}$$

Denote the minimizer of $G(\mu)$ by μ_{GCV}. In rare events, when the minimizer is not unique, we choose the largest one.

The minimization of $G(\mu)$ requires the computation, or at least estimation, of the trace of the matrix $I - A(A^T A + \mu L^T L)^{-1} A^T$ that is not explicitly known. This can make the computation of $G(\mu)$ expensive when the matrices A and L are large. Different ways to speed up the computations by application of a global block Lanczos method or Golub-Kahan bidiagonalization are described in [19, 20]. However, these works are limited to the situation when $L = I$.

In this paper, we combine ideas in [19] with the techniques described in [30] to derive an algorithm for computing the GCV parameter for (2) for general values of $0 < p, q \leq 2$. Specifically, we consider the AMM-GKS algorithm in [30] for the solution of (2) (see Section 3.1 for more details), and modify it so that at each iteration the regularization parameter is chosen adaptively by minimizing a reduced version of the function (5). The computation of the GCV parameter is fully automatic in the sense that no information about the noise $\boldsymbol{\eta}$ is required. The GCV method for determining the regularization parameter is a so-called heuristic method, see Kindermann [32], and therefore may fail for certain data vectors $\mathbf{b}$. In our experience, failure is extremely rare. The GCV method is related to cross validation (CV), another heuristic method for determining a suitable regularization parameter; see [8, 39]. We will illustrate in Section 4 that GCV typically gives about the same value of the regularization parameter as CV for a much lower computational effort.

This paper is organized as follows: Section 2 reviews how the GCV parameter for Tikhonov regularization in general form can be computed fairly inexpensively by projection into a Krylov subspace, and in Section 3 we discuss how to determine the GCV parameter for large-scale minimization problems (2). A smoothed variant of GCV that can give better regularization parameter values also is presented. Section 4 contains numerical examples that illustrate the performance of the proposed approach. Finally, Section 5 contains some concluding remarks.

2 GCV for Tikhonov regularization in general form

This section discusses the computation of the GCV parameter for Tikhonov regularization in general form, i.e., when solving the minimization problem (2) with $p = q = 2$. We first describe an approach that requires the generalized singular value decomposition (GSVD) of the matrix pair $\{A, L\}$. This approach is computationally feasible only for small to medium-sized problems since the computation of the GSVD for a large matrix pair is very expensive. We then discuss how the computational burden for large-scale problems can be reduced by projecting the matrix pair $\{A, L\}$ into a Krylov subspace of fairly small dimension.

2.1 GCV computation by using the GSVD

This subsection reviews material in [27], where further details can be found. Consider the Tikhonov regularization problem in general form with $\mu > 0$,

$$\mathbf{x}_\mu = \arg \min_{\mathbf{x} \in \mathbb{R}^n} \left\{ \left\| A\mathbf{x} - \mathbf{b}^\delta \right\|_2^2 + \mu \left\| L\mathbf{x} \right\|_2^2 \right\}, \tag{6}$$

and introduce the GSVD of the matrix pair $\{A, L\}$,

$$\begin{cases} A = U\Sigma_A Y^T, \\ L = V\Sigma_L Y^T, \end{cases} \tag{7}$$

where Σ_A and Σ_L are diagonal, possibly rectangular, matrices, the matrices U and V are orthogonal, and the matrix Y is square and non-singular. Further details of these matrices, including their sizes, can be found, e.g., in [16, 25, 27]. The evaluation of the GCV function $G(\mu)$, defined by (5), requires the computation of the quantities

$$r_\mu = \left\| A\mathbf{x}_\mu - \mathbf{b}^\delta \right\|_2^2,$$
$$t_\mu = \mathsf{trace} \left(I - A \left(A^T A + \mu L^T L \right)^{-1} A^T \right),$$

for several values of $\mu > 0$, where the vector $\mathbf{x}_\mu$ is given by (4). These quantities can be evaluated inexpensively when the factorizations (7) are available as follows. For notational simplicity, the formulas below assume that all matrices are square. This restriction easily can be removed; see, e.g., [16, 25, 27] for details.

The quantity r_μ can be expressed as

$$\begin{aligned} r_\mu &= \left\| \mathbf{b}^\delta - A\mathbf{x}_\mu \right\|_2^2 \\ &\overset{(a)}{=} \left\| \mathbf{b}^\delta - A(A^T A + \mu L^T L)^{-1} A^T \mathbf{b}^\delta \right\|_2^2 \\ &\overset{(b)}{=} \left\| \tilde{\mathbf{b}}^\delta - \Sigma_A (\Sigma_A^T \Sigma_A + \mu \Sigma_L^T \Sigma_L)^{-1} \Sigma_A^T \tilde{\mathbf{b}}^\delta \right\|_2^2, \end{aligned}$$

where (a) is obtained by using the definition (4) of $\mathbf{x}_\mu$, and (b) is derived using the factorizations (7) with $\tilde{\mathbf{b}}^\delta = U^T \mathbf{b}^\delta$. Analogously,

$$\begin{aligned} t_\mu &= \mathsf{trace} \left(I - A \left(A^T A + \mu L^T L \right)^{-1} A^T \right) \\ &= \mathsf{trace} \left(I - \Sigma_A (\Sigma_A^T \Sigma_A + \mu \Sigma_L^T \Sigma_L)^{-1} \Sigma_A^T \right), \end{aligned}$$

where we have used the factorizations (7) and the fact that trace is invariant under similarity transformation. We have

$$\mu_{\mathrm{GCV}} = \arg \min_\mu \frac{r_\mu}{t_\mu},$$

where we in rare cases of nonunicity let μ_{GCV} be the largest of the minimizers.

It is clear from the above expressions for r_μ and t_μ that the main computational cost for evaluating (5) is the computation of the GSVD (7) of the matrix pair $\{A, L\}$. Observe that, to determine μ_{GCV} we typically have to evaluate $G(\mu)$ for many values of μ. However, it suffices to compute the factorizations (7) only once, since they are independent of μ.

2.2 GCV computation in a Krylov subspace

When the matrices A and L are large, the computation of the GSVD (7) is unattractive due to its high cost; for instance, Paige's algorithm generally requires at least $35.3n^3$ arithmetic floating point operations when $A, L \in \mathbb{R}^{n \times n}$; see [1, Table 5.1] for details. We therefore now describe a more computational attractive approach that is based on projecting the problem into a Krylov subspace of fairly small dimension. Our projection method extends the method used in [19] by allowing the matrix L to be different from the identity.

Let $A \in \mathbb{R}^{m \times n}$ and $L \in \mathbb{R}^{r \times n}$ and let the columns of $V \in \mathbb{R}^{n \times d}$ form an orthonormal basis for the Krylov subspace $\mathcal{V}$ of dimension $1 \le d \ll \min\{m, n, r\}$,

$$\mathcal{V} = \mathcal{K}_d\left(A^T A, A^T \mathbf{b}^\delta\right) = \mathsf{span}\left\{A^T \mathbf{b}^\delta, A^T A A^T \mathbf{b}^\delta, \ldots, (A^T A)^{d-1} A^T \mathbf{b}^\delta\right\}.$$

We assume here that d is small enough so that $\dim(\mathcal{K}_d\left(A^T A, A^T \mathbf{b}^\delta\right)) = d$. This is the generic situation. It is convenient to compute the columns of V by applying d steps of Golub-Kahan bidiagonalization to A with initial vector $\mathbf{b}^\delta$. This gives the decomposition

$$AV = [U, u_{d+1}]B^{(A)}, \tag{8}$$

where the matrices $[U, u_{d+1}] \in \mathbb{R}^{m \times (d+1)}$ and $V \in \mathbb{R}^{n \times d}$ have orthonormal columns and $B^{(A)} \in \mathbb{R}^{(d+1) \times d}$ is lower bidiagonal; see, e.g., [25]. We assume that d is small enough so that this decomposition exists.

We would like to compute an approximate solution of the Tikhonov minimization problem (6) in $\mathcal{V}$. The restriction of the approximate solution to the subspace $\mathcal{V}$ of low dimension d reduces the computational cost of significantly. Thus, we would like to solve

$$\mathbf{x} = \arg\min_{\mathbf{x} \in \mathcal{V}}\left\{\left\|A\mathbf{x} - \mathbf{b}^\delta\right\|_2^2 + \mu\left\|L\mathbf{x}\right\|_2^2\right\}.$$

Substituting $\mathbf{x} = V\mathbf{y}$ for some $\mathbf{y} \in \mathbb{R}^d$ into the above equation gives

$$\mathbf{y} = \arg\min_{\mathbf{y} \in \mathbb{R}^d}\left\{\left\|AV\mathbf{y} - \mathbf{b}^\delta\right\|_2^2 + \mu\left\|LV\mathbf{y}\right\|_2^2\right\}. \tag{9}$$

Substitute the QR factorization

$$LV = Q^{(L)}R^{(L)},$$

where the matrix $Q^{(L)} \in \mathbb{R}^{r \times d}$ has orthonormal columns and $R^{(L)} \in \mathbb{R}^{d \times d}$ is upper triangular, and the decomposition (8) into (9) gives

$$\mathbf{y} = \arg \min_{\mathbf{y} \in \mathbb{R}^d} \left\{ \left\| B^{(A)} \mathbf{y} - \mathbf{e}_1 \|\mathbf{b}^\delta\|_2^2 \right\|_2^2 + \mu \left\| R^{(L)} \mathbf{y} \right\|_2^2 \right\}, \tag{10}$$

with $\mathbf{e}_1 = [1, 0, \ldots, 0]^T$ denoting the first axis vector. The projected problem (10) is of quite inexpensive to compute and of fairly small dimension. In particular, it is feasible to compute the GSVD of the matrix pair $\{B^{(A)}, R^{(L)}\}$ and use it to calculate an approximation of the GCV function $G(\mu)$.

3 GCV for ℓ^p-ℓ^q regularization

We describe how to combine the computations discussed in Section 2 for determining the regularization parameter by the GCV method with the algorithms presented in [30] for minimizing the expression (2).

3.1 Solution of the ℓ^p-ℓ^q minimization problem for fixed $\mu > 0$

For convenience of the reader, we outline the AMM-GKS method for solving (2) proposed in [30]. Introduce a smoothed version of the q-norm $\|\mathbf{x}\|_q^q$ for $0 < q \le 1$ as follows. Let $\Phi_q : \mathbb{R} \to \mathbb{R}$ be defined by

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

and observe that, if $0 < q \le 1$, then $t \mapsto \Phi_q(t)$ is not differentiable at $t = 0$. We approximate Φ_q by the differentiable function

$$\Phi_{q,\varepsilon}(t) = \left(\sqrt{t^2 + \varepsilon^2} \right)^q, \tag{11}$$

where $\varepsilon > 0$ is a (small) constant. Similarly, we define for $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^n$ the smoothed version of $\|\mathbf{x}\|_q^q$ as

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

Finally, introduce the smoothed version of the functional that is minimized in (2),

$$\mathcal{J}_\varepsilon(\mathbf{x}) := \frac{1}{p} \sum_{i=1}^m \Phi_{p,\varepsilon}((A\mathbf{x} - \mathbf{b}^\delta)_i) + \frac{\mu}{q} \sum_{i=1}^r \Phi_{q,\varepsilon}((L\mathbf{x})_i). \tag{12}$$

This gives us the smoothed version

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

of the minimization problem (2).

The AMM-GKS method described in [30] is a majorization-minimization method for computing a stationary point of the functional (12). This method determines a sequence of iterates $\mathbf{x}^{(k)}$, $k = 1, 2, \ldots$, that converge to a stationary point of (12). Each iterate is computed in by two steps: first the functional (12) is majorized by a quadratic tangent majorant $\mathbf{x} \to \mathcal{Q}(\mathbf{x}, \mathbf{x}^{(k)})$ for $\mathcal{J}_\varepsilon$ at $\mathbf{x}^{(k)}$. Then the unique minimizer of $\mathbf{x} \to \mathcal{Q}(\mathbf{x}, \mathbf{x}^{(k)})$ is calculated. This minimizer is the next iterate $\mathbf{x}^{(k+1)}$. We outline this method in the remainder of this subsection.

Definition 1 ([30]) Consider the differentiable functional $\mathcal{J}_\varepsilon(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$. We say that the functional $\mathbf{x} \mapsto \mathcal{Q}(\mathbf{x}, \mathbf{y}) : \mathbb{R}^n \to \mathbb{R}$ is a *quadratic tangent majorant* for $\mathcal{J}_\varepsilon(\mathbf{x})$ at $\mathbf{y} \in \mathbb{R}^n$ if the following conditions are satisfied:

– $\mathcal{Q}(\mathbf{x}, \mathbf{y})$ 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 $\nabla_{\mathbf{x}} \mathcal{Q}(\mathbf{y}, \mathbf{y}) = \nabla_{\mathbf{x}} \mathcal{J}(\mathbf{y})$,

where $\nabla_{\mathbf{x}}$ denotes the gradient with respect to the first argument of Q.

Majorization step. The quadratic tangent majorant at the point $\mathbf{x}^{(k)}$ constructed by the AMM-GKS method is determined as follows. Let

$$\mathbf{v}^{(k)} := A\mathbf{x}^{(k)} - \mathbf{b}^\delta,$$
$$\mathbf{u}^{(k)} := L\mathbf{x}^{(k)},$$

and compute the vectors

$$\boldsymbol{\omega}_{\text{fid}}^{(k)} := ((\mathbf{v}^{(k)})^2 + \varepsilon^2)^{p/2-1},$$
$$\boldsymbol{\omega}_{\text{reg}}^{(k)} := ((\mathbf{u}^{(k)})^2 + \varepsilon^2)^{q/2-1},$$

where all the operations, including squaring, are meant element-wise. Define the diagonal matrices

$$W_{\text{fid}}^{(k)} = \text{diag}(\boldsymbol{\omega}_{\text{fid}}^{(k)}), \quad W_{\text{reg}}^{(k)} = \text{diag}(\boldsymbol{\omega}_{\text{reg}}^{(k)}).$$

It is shown in [30] that the functional

$$\mathcal{Q}(\mathbf{x}, \mathbf{x}^{(k)}) = \frac{1}{2} \left\| (W_{\text{fid}}^{(k)})^{1/2} (A\mathbf{x} - \mathbf{b}^\delta) \right\|_2^2 + \frac{\mu}{2} \left\| (W_{\text{reg}}^{(k)})^{1/2} L\mathbf{x} \right\|_2^2 + c, \qquad (13)$$

with c a suitable constant that is independent of $\mathbf{x}$, is a quadratic tangent majorant for $\mathcal{J}_\varepsilon$ at $\mathbf{x}^{(k)}$; see [30] for details on the derivation of $\boldsymbol{\omega}_{\text{fid}}^{(k)}$, $\boldsymbol{\omega}_{\text{reg}}^{(k)}$, and c.

Minimization step. We describe the computation of the minimizer $\mathbf{x}^{(k+1)}$ of (13). Since $\mathcal{Q}$ is quadratic, $\mathbf{x}^{(k+1)}$ can be determined as the zero of the gradient, i.e., by solving

$$(A^T W_{\mathrm{fid}}^{(k)} A + \mu L^T W_{\mathrm{reg}}^{(k)} L)\mathbf{x} = A^T W_{\mathrm{fid}}^{(k)} \mathbf{b}^\delta. \tag{14}$$

The system matrix is nonsingular and positive definite for all $\mu > 0$ if and only if

$$\mathcal{N}(A^T W_{\mathrm{fid}}^{(k)} A) \cap \mathcal{N}(L^T W_{\mathrm{reg}}^{(k)} L) = \{\mathbf{0}\}.$$

This condition typically is satisfied. Then the solution $\mathbf{x}^{(k+1)}$ of (14) is the unique minimizer of $\mathbf{x} \to \mathcal{Q}(\mathbf{x}, \mathbf{x}^{(k)})$.

The computation of the solution of (14) can be demanding when the matrices A and L are large. An approximate solution can be computed efficiently by projecting the problem into a low-dimensional subspace $\mathcal{V}_d$ of dimension $1 \leq d \ll \min\{m, n, r\}$. Let the columns of the matrix $V_d \in \mathbb{R}^{n \times d}$ form an orthonormal basis for $\mathcal{V}_d$ and calculate the skinny QR factorizations

$$(W_{\mathrm{fid}}^{(k)})^{1/2} A V_d = Q_A R_A \quad \text{with} \quad Q_A \in \mathbb{R}^{m \times d}, \quad R_A \in \mathbb{R}^{d \times d},$$
$$(W_{\mathrm{reg}}^{(k)})^{1/2} L V_d = Q_L R_L \quad \text{with} \quad Q_L \in \mathbb{R}^{r \times d}, \quad R_L \in \mathbb{R}^{d \times d},$$

where the matrices Q_A and Q_L have orthonormal columns and the matrices R_A and R_L are upper triangular. Using these factorizations, we obtain the reduced analogue of equation (14),

$$(R_A^T R_A + \mu R_L^T R_L)\mathbf{y} = R_A^T Q_A^T (W_{\mathrm{fid}}^{(k)})^{1/2} \mathbf{b}^\delta. \tag{15}$$

The solution $\mathbf{y}^{(k+1)}$ of this equation gives, with a slight abuse of notation, the approximate solution

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

of (14). Substituting this expression into (14) gives the residual vector

$$\mathbf{r}^{(k+1)} := A^T W_{\mathrm{fid}}^{(k)} (A V_d \mathbf{y}^{(k+1)} - \mathbf{b}^\delta) + \mu L^T W_{\mathrm{reg}}^{(k)} L V_d \mathbf{y}^{(k+1)}.$$

Observe that this vector in general is nonvanishing, since we are solving (14) in the subspace $\mathcal{V}_d$ of $\mathbb{R}^n$. We enlarge at each step the solution subspace by adding the scaled residual vector $\mathbf{v}_{\mathrm{new}} = \mathbf{r}^{(k+1)}/\|\mathbf{r}^{(k+1)}\|$ to the solution subspace. This vector is orthogonal to the columns of the matrix V_d. Define the matrix $V_{d+1} = [V_d, \mathbf{v}_{\mathrm{new}}] \in \mathbb{R}^{n \times (d+1)}$. Its columns form an orthonormal basis for the new solution subspace $\mathcal{V}_{d+1}$. The so determined solution subspace is referred to as a *Generalized Krylov subspace* (GKS); see, e.g., [34] for another application of this kind of subspaces. The computations are initialized by choosing a d_0-dimensional solution subspace of $\mathbb{R}^n$ and letting the columns of the matrix V_{d_0} form an orthonormal basis for this subspace. Typically, setting $d_0 = 1$ and $V_0 = A^T \mathbf{b}^\delta / \|A^T \mathbf{b}^\delta\|_2$ is appropriate. An algorithm that describes these computations and determines the regularization parameter μ by the GCV method is described in the following subsection.

3.2 GCV and MM-GKS

In the computations outlined in the previous subsection, the regularization parameter is kept fixed. We now describe how a value μ_k of the regularization parameter can be determined in a fairly simple manner before computing the next approximate solution $\mathbf{x}^{(k+1)}$ of (2).

Equation (15) is the normal equation associated with the minimization problem

$$\mathbf{y}^{(k+1)} = \arg \min_{\mathbf{y} \in \mathbb{R}^d} \left\{ \left\| R_A \mathbf{y} - Q_A^T W_{\text{fid}}^{(k)} \mathbf{b}^\delta \right\|_2^2 + \mu \left\| R_L \mathbf{y} \right\|_2^2 \right\}.$$

This minimization problem is analogous to (10). Since it stems from projecting a large problem into a generalized Krylov subspace of modest dimension d, the matrices R_A and R_L are small enough to allow the computation of the GSVD of the matrix pair $\{R_A, R_L\}$ at moderate expense. We therefore can use the technique described in Section 2.1 to determine a suitable value of the regularization parameter by GCV. In detail, let the GSVD of the matrix pair $\{R_A, R_L\}$ be given by

$$R_A = U \Sigma_A Y^T,$$
$$R_L = V \Sigma_L Y^T.$$

Similarly as in Section 2.1, we define, for any $\mu > 0$, the quantities

$$r_\mu^{(k)} = \left\| U^T Q_A^T \mathbf{b}^\delta - \Sigma_A^2 (\Sigma_A^2 + \mu \Sigma_L^2)^{-1} U^T Q_A^T W_{\text{fid}}^{(k)} \mathbf{b}^\delta \right\|_2^2,$$
$$t_\mu^{(k)} = \mathsf{trace} \left(I - \Sigma_A^2 (\Sigma_A^2 + \mu \Sigma_L^2)^{-1} \right). \tag{16}$$

This derivation uses that both matrices R_A and R_L are square. We compute the GCV parameter $\mu^{(k)}$ as the solution of

$$\mu^{(k)} = \arg \min_\mu \frac{r_\mu^{(k)}}{t_\mu^{(k)}}.$$

If the minimum is not unique, then we choose the largest one. Typically, the minimum is unique. This is assumed in Algorithm 1, which describes the computations. The parameter $\mu^{(k)}$ is updated in each iteration.

Algorithm 1 (The MM-GKS-GCV method) *Let $0 < p, q \le 2$. Consider $A \in \mathbb{R}^{m \times n}$ and $L \in \mathbb{R}^{r \times n}$ such that (3) holds. Fix $\varepsilon > 0$ and $k_0 > 0$, and choose the initial vector $\mathbf{x}_0 = A^T \mathbf{b}^\delta$;*

> *Generate the initial subspace basis: $V_{d_0} \in \mathbb{R}^{n \times d_0}$ such that $V_{d_0}^T V_{d_0} = I$.*
> *Let $d = d_0$. In the computed examples, we let $d_0 = 1$ and*
> *$V_{d_0} = A^T \mathbf{b}^\delta / \|A^T \mathbf{b}^\delta\|$;*
> *Compute and store AV_0 and LV_0;*
> *$\mathbf{y}^{(0)} = V_d^T \mathbf{x}^{(0)}$;*
> **for** $k = 0, 1, \dots$ **do**
>> *$\mathbf{v}^{(k)} = A\mathbf{x}^{(k)} - \mathbf{b}^\delta$;*
>> *$\mathbf{u}^{(k)} = LV_d \mathbf{y}^{(k)}$;*
>> *$\boldsymbol{\omega}_{\text{fid}}^{(k)} := ((\mathbf{v}^{(k)})^2 + \varepsilon^2)^{p/2-1}$;*
>> *$\boldsymbol{\omega}_{\text{reg}}^{(k)} = ((\mathbf{u}^{(k)})^2 + \varepsilon^2)^{q/2-1}$;*
>> *$W_{\text{fid}}^{(k)} = \text{diag}(\boldsymbol{\omega}_{\text{fid}}^{(k)})$;*
>> *$W_{\text{reg}}^{(k)} = \text{diag}(\boldsymbol{\omega}_{\text{reg}}^{(k)})$;*
>> *Compute the QR factorization $Q_A R_A = (W_{\text{fid}}^{(k)})^{1/2} AV_d$;*
>> *Compute the QR factorization $Q_L R_L = (W_{\text{ref}}^{(k)})^{1/2} LV_d$;*
>> *Compute the GSVD of the pair $\{R_A, R_L\}$;*
>> *$\mu^k = \arg\min_\mu \frac{r_\mu^{(k)}}{t_\mu^{(k)}}$, where $r_\mu^{(k)}$ and $t_\mu^{(k)}$ are defined in (16);*
>> *$\mathbf{y}^{(k+1)} = (R_A^T R_A + \mu R_L^T R_L)^{-1} R_A^T Q_A^T (W_{\text{fid}}^{(k)})^{1/2} \mathbf{b}^\delta$;*
>> *$\mathbf{r} = A^T W_{\text{fid}}^{(k)} (AV_d \mathbf{y}^{(k+1)} - \mathbf{b}^\delta) + \mu L^T W_{\text{reg}}^{(k)} LV_d \mathbf{y}^{(k+1)}$;*
>> *$\mathbf{v}_{\text{new}} = \mathbf{r} / \|\mathbf{r}\|_2$; $V_{d+1} = [V_d, \mathbf{v}_{\text{new}}]$; $d = d + 1$;*
>> *Compute and store AV_d and LV_d;*
>> *$\mathbf{x}^{(k+1)} = V_d \mathbf{y}^{(k+1)}$;*
> **end**

Algorithm 1 requires the computation of the GSVD of the matrix pair $\{R_A, R_L\}$ in each iteration. Since the matrices R_A and R_L are not very large, this is not very expensive.

3.3 Smoothed GCV for impulse noise

We describe a modified version of the GCV that, as we will see in Section 4, is well suited for impulse noise. The Euclidean norm of impulse noise is infinite. Therefore, the numerator in $G(\mu)$, see (5), may be infinite. It therefore may be difficult to compute the minimum of $G(\mu)$. In computations in finite precision arithmetic, the Euclidean norm of impulse noise is extremely large and this leads to that it may be difficult to determine the minimum of $G(\mu)$. To remedy this issue, we smooth the data vector $\mathbf{b}^\delta$ by convolving it with a Gaussian filter with small variance. Let $\mathbf{k}_{\nu^2} \in \mathbb{R}^m$ represent a sampling of a Gaussian function with mean zero and variance ν^2, and define the smoothed vector

$$\mathbf{b}_{\text{smooth}}^\delta = \mathbf{k}_{\nu^2} * \mathbf{b}^\delta,$$

where $*$ denotes the convolution operator. It is well known that if we assume periodic boundary conditions, then convolution can be carried out easily with the aid of the Fourier transform $\mathfrak{F}$. We have

$$\mathfrak{F}(\mathbf{b}_{\text{smooth}}^{\delta}) = \mathfrak{F}(\mathbf{k}_{\nu^2}) \odot \mathfrak{F}(\mathbf{b}^{\delta}),$$

where $\odot$ denotes element-wise multiplication. Moreover, if $\mathbf{b}^{\delta}$ represents the vectorization of a multi-dimensional array, e.g., an image, it is beneficial to let $\mathbf{k}_{\nu^2}$ be the vectorization of a discretized multi-dimensional Gaussian function.

Using $\mathbf{b}_{\text{smooth}}^{\delta}$ we can define the smoothed version of the GCV functional by

$$G_{\text{smooth}}(\mu) = \frac{\left\| \mathbf{b}_{\text{smooth}}^{\delta} - A\mathbf{x}_{\mu} \right\|_2^2}{\left(\mathsf{trace}\left(I - A\left(A^T A + \mu L^T L\right)^{-1} A^T \right) \right)^2}.$$

This functional is the standard GCV functional (5) with the data vector $\mathbf{b}^{\delta}$ replaced by the smoothed data vector $\mathbf{b}_{\text{smooth}}^{\delta}$. The MM-GKS-GCV-Smooth algorithm is obtained by minimizing G_{smooth} at each iteration in Algorithm 1 instead of G.

4 Numerical examples

This section shows some numerical examples that illustrate the performance of the proposed methods. The examples are concerned with the restoration of images that have been contaminated by noise and space-invariant blur, as well as with computerized tomography; see [28] for details on image restoration and [9] for a description of computerized tomography.

We set $q = 0.1$ in all examples, while the value of p depends on the type of noise the data vector $\mathbf{b}^{\delta}$ is corrupted by. For white Gaussian noise, we set $p = 2$; for other kinds of noise, we let $p = 0.8$.

We would like to compute a sparse solution in the framelet domain 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 $\mathcal{W} \in \mathbb{R}^{r \times n}$ with $n \leq r$. The set of the rows of $\mathcal{W}$ is a *framelet system* for $\mathbb{R}^n$ if $\forall \mathbf{x} \in \mathbb{R}^n$ it holds

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

where $\mathbf{y}_j \in \mathbb{R}^n$ is the j-th row of $\mathcal{W}$ (written as a column vector), i.e., $\mathcal{W} = [\mathbf{y}_1, \ldots, \mathbf{y}_r]^T$. The matrix $\mathcal{W}$ is referred to as an *analysis operator* and $\mathcal{W}^T$ as a *synthesis operator*. The matrix $\mathcal{W}$ defines a tight frame if and only if $\mathcal{W}^T \mathcal{W} = I$.

We remark that in general $\mathcal{W}\mathcal{W}^T \neq I$, unless $r = n$ and the framelets are orthonormal. Note that $\mathcal{N}(\mathcal{W}) = \{\mathbf{0}\}$. Therefore, property (3) holds. Tight frames have been used in many image restoration applications including inpainting and deblurring; see, e.g., [2–6,10–13].

We use framelets determined by linear B-splines. In one space-dimension they are made up of a low-pass filter W_0 and two high-pass filters W_1 and W_2, whose masks are

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

respectively. We can derive the synthesis operator $\mathcal{W}$ starting from these masks by imposing reflexive boundary conditions. These boundary conditions secure that $\mathcal{W}^T \mathcal{W} = I$. Defining

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

and

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

The synthesis operator is obtained by stacking the three matrices above

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

We can extend the operator above to two space-dimension by means of tensor products

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

This yields the analysis operator

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

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. In our examples, we set $L = \mathcal{W}$.

We compare generalized cross validation with two other approaches to selecting the regularization parameter in (2), including cross validation as described in [8] and the discrepancy principle as discussed in [7]. We also compare the quality of the computed solutions determined by these method with the solution obtained with the optimal choice of the regularization parameter. The latter choice of the regularization parameter is not practical, because it requires that the desired solution $\mathbf{x}_{\text{true}}$ be known; the regularization parameter is then determined to yield a computed solution that minimizes the restoration error

$$\text{RRE}(\mathbf{x}) = \frac{\|\mathbf{x} - \mathbf{x}_{\text{true}}\|_2}{\|\mathbf{x}_{\text{true}}\|_2}, \tag{17}$$

where $\mathbf{x}_{\text{true}}$ denotes the desired solution.

The method described in [7] determines a sequence of regularization parameters, in a similar fashion as in this work, by requiring that the computed approximate solution in each iteration satisfies the discrepancy principle (DP). To perform well, this method requires that a fairly accurate estimate of the norm of the error $\|\boldsymbol{\eta}\|_2$ in the data vector $\mathbf{b}^\delta$ be known. This norm is not meaningful for impulse noise. Therefore, the discrepancy principle cannot be used when $\mathbf{b}^\delta$ is contaminated by impulse noise.

Two methods for determining the regularization parameter by cross-validation are described in [8]. The first method determines the regularization parameter so that missing data is well predicted. This classical approach of carrying out cross-validation is referred to as CV in the tables. The other method, which we refer to as modified cross-validation (MCV), seeks to determine the regularization parameter so that missing entries of the computed solution are well predicted. The MCV method typically gives approximations of the desired solution $\mathbf{x}_{\text{true}}$ of somewhat higher quality than the CV method; see [8] for illustrations. This paper compares different parameter choice rules for the minimization problem (2). A comparison with other methods for solving the minimization problem (2), that require the regularization parameter to be hand-tuned, can be found in [30, 35].

The restoration error (17) and the structural and similarity index (SSIM) are compared in our tests. The definition of the latter is somewhat involved; here we just note that the SSIM measures how well the overall structure of an image is recovered, and that the larger the index the better the reconstruction. In particular, the highest value achievable is 1; see [40] for details.

For all examples we set $\varepsilon = 1$ in (11); see [7] for a discussion on the choice of ε. We terminate the iterations either after 100 iterations or when

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

Finally, for the computations of $\mathbf{b}^\delta_{\text{smooth}}$, we set $\nu^2 = 1$ in all examples.

All the computations were performed using Matlab R2018b running on a Windows 10 laptop computer with a i7-8750H @2.20 GHz CPU and 16GB of RAM.

Fig. 2: Tomography test case: (a) True phantom (256×256 pixels), (b) sinogram (90 equispaced angles in $[0, \pi($, 362 measurements per angle, and 1% white Gaussian noise).

Tomography. The first example considers a tomography problem. The example is constructed with the aid of the IR Tools Matlab package [21]. The phantom has 256×256 pixels, and we consider 90 equispaced angles in the interval $[0, \pi($, and 362 rays per angle. This results in a matrix $A \in \mathbb{R}^{32580 \times 65536}$. We added 1% of white Gaussian noise to the sinogram, i.e., $\|\boldsymbol{\eta}\|_2 = 0.01 \|\mathbf{b}\|_2$. Since the noise is Gaussian, we set $p = 2$ and, like in all considered examples, $q = 0.1$. We show the true phantom and the sinogram in Figure 2.

Table 1 reports the RRE and SSIM for the computed solutions determined by all the parameter choice rules considered in Table 1. All the rules can be seen to perform very similarly. Obviously, the optimal parameter provides the best reconstruction in terms of the RRE. This parameter choice rule requires knowledge of the exact solution $\mathbf{x}_{\text{true}}$ and therefore cannot be used in real application. The DP provides the second best approximation of $\mathbf{x}_{\text{true}}$, however, it demands knowledge of the norm of the noise $\boldsymbol{\eta}$. The CV, MCV, and GCV methods do not require any knowledge of the noise. We can observe, that although the MCV method slightly outperforms the GCV method in terms of the RRE (but not in terms of the SSIM), the GCV method demands much less CPU time than the CV and MCV methods. This is due to the fact that both the CV and MCV methods require several runs of the MM-GKS algorithm, while the GCV method only carries out a single run. Finally, we note that the GCV-Smooth method does not perform well. This is due to the fact that the noise in the data $\mathbf{b}^\delta$ is purely Gaussian. The smoothing of $\mathbf{b}^\delta$ leads to an underestimation of the amount of noise in $\mathbf{b}^\delta$, which results in a too small value of the regularization parameter; the computed solution is under-regularized. These considerations are confirmed by the visual comparison of the reconstructions shown in Figure 3.

Peppers. Our second example considers an image deblurring problem. We blur the image in Figure 4(a) with the PSF shown in Figure 4(b), and add 25% of impulse noise. This yields the blurred and noisy image displayed in Figure 4(c). When we construct this example, we cut the boundaries of the blurred image
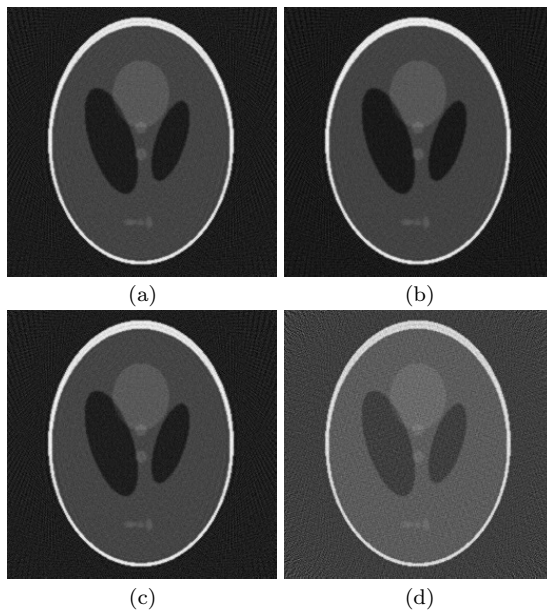
Fig. 3: Tomography test case reconstructions: (a) Optimal, (b) DP, (c) GCV, (d) GCV-Smooth.

to simulate realistic data; see [28] for more details. Since the image is generic, we impose reflexive boundary conditions. Moreover, since the blurred image is corrupted by impulse noise, we set $p = 0.8$.

We report the results obtained in Table 1. Since the noise is not Gaussian, we cannot apply the discrepancy principle to determine the regularization parameter. However, the CV, MCV, and GCV methods can be applied. We also report the RRE and SSIM for the optimal choice of the regularization parameter. We can see that the GCV and GCV-Smooth methods provide the best reconstructions. In particular, the reconstruction obtained with GCV is slightly better than the optimal one, and the one obtained with GCV-Smooth is significantly better. While this may appear strange, it is due to the construction of the generalized Krylov subspace. In particular, since the solution subspace depends on the approximate solutions generated during the computations, the solution subspaces determined by the various methods differ, and this may lead to different results. We remark that, while we for stationary methods (with μ fixed) can show converge of the computed iterates, we have no such result for the iterates determined by the CV and GCV methods. Finally, we observe that the use of the smoothed GCV function yields a very accurate reconstruction which, in particular, is more accurate than the one obtained with the standard GCV method. These observations are confirmed by visual inspection of the reconstructions displayed in Figure 5.
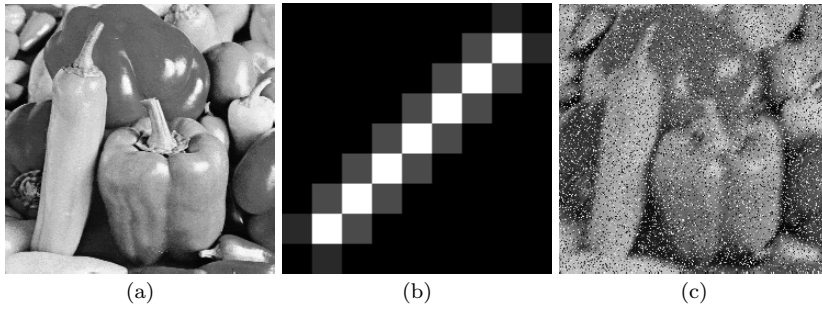
Fig. 4: Peppers test case: (a) True image (246 × 246 pixels), (b) PSF (9 × 9 pixels), (c) Blurred and noisy image (25% of impulse noise).
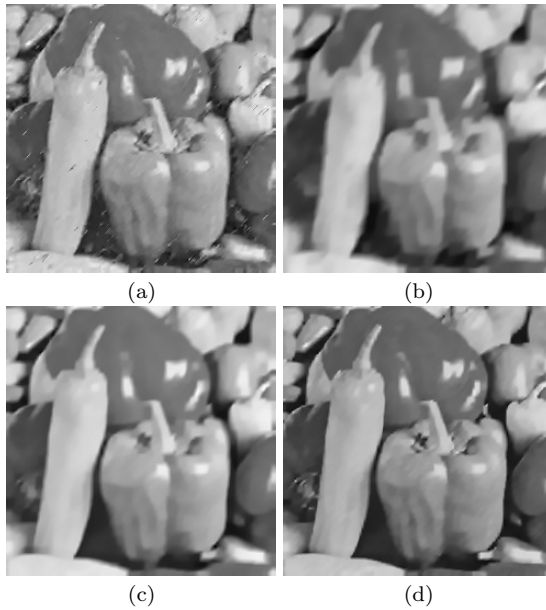

Fig. 5: Peppers test case reconstructions: (a) Optimal, (b) MCV, (c) GCV, (d) GCV-Smooth.

Cameraman. The last example describes another image deblurring problem. We consider the exact image in Figure 6(a) and blur it with the PSF in Figure 6(b). Adding 20% of salt-and-pepper noise produces the blurred and noisy image in Figure 6(c). As in the previous example, we cut the boundary of the blurred image to simulate realistic data. In view of that the image is generic, we impose reflexive boundary conditions. Since the PSF is quadrantically symmetric, the matrix A obtained can be diagonalized by the discrete cosine transform. Because $L^T L = I$, we therefore could avoid the application of generalized Krylov subspaces in the MM-GKS method. However, for consis-
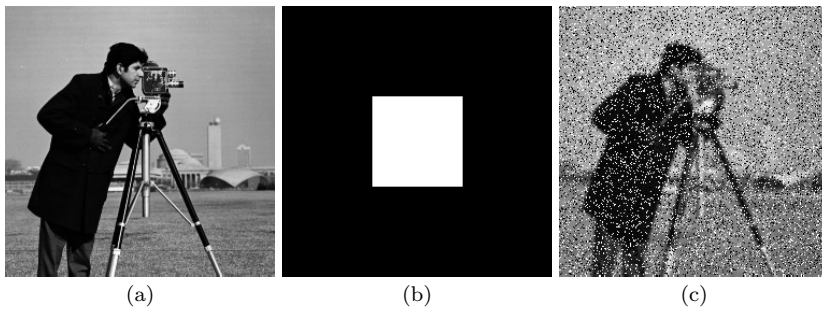
Fig. 6: Cameraman test case: (a) True image (252×252 pixels), (b) PSF (12×12 pixels), (c) Blurred and noisy image (20% of salt and pepper noise).

tency we use the same approach as in the previous examples. Since the noise is not Gaussian, we set $p = 0.8$, similarly as in the example above.

Table 1 shows the RRE and SSIM values obtained with the different parameter choice rules. We observe that the GCV and GCV-smooth methods provide more accurate approximations of $\mathbf{x}_{\text{true}}$ in terms of RRE, than both the CV and MCV, for a much lower computational cost. Moreover, the RRE obtained with the GCV method is close to the optimal one, and the RRE for the restoration determined by the GCV-Smooth method is significantly smaller. Figure 7 displays the obtained reconstructions. Visual inspection shows that the reconstruction obtained with the GCV-Smooth method is very detailed, but "grayish". The latter is due to the presence of some pixels, whose values are significantly larger than 255 (which is the value corresponding to white, the value 0 corresponds to black), which causes a rescaling before plotting. Projecting the reconstruction orthogonally into the cube $[0, 255]^n$ gives the image in Figure 8. We can observe that the gray-scale level is now correct and the reconstruction is very close to the original image.

5 Conclusions

We propose the GCV method for choosing the regularization parameter for the ℓ^p-ℓ^q regularization method. This method is easy to implement and computationally fairly inexpensive thanks to that the original minimization problem (2) is projected into a generalized Krylov subspace of small to moderate size. A comparison with some other available methods shows that the proposed approach is competitive and provides accurate approximations of the desired solution. Moreover, the proposed approach is completely automatic and does not need hand-tuning of any parameter. Thus, it can be considered for real world applications.
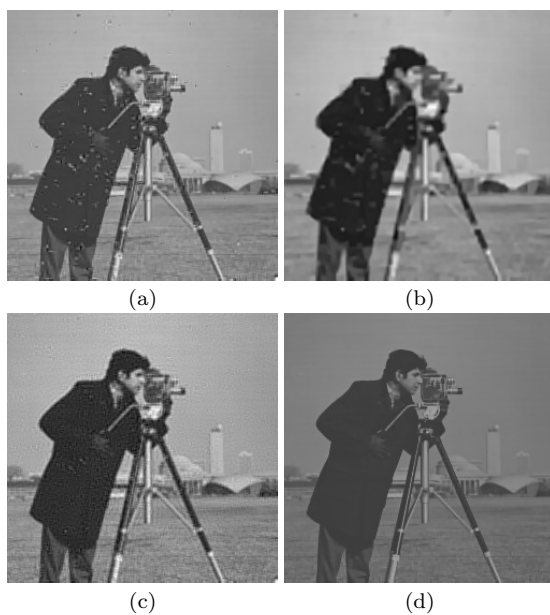
Fig. 7: Cameraman test case reconstructions: (a) Optimal, (b) MCV, (c) GCV, (d) GCV-Smooth.

Fig. 8: Cameraman test case reconstruction obtained using the MM-GKS-GCV-Smoothed method and projecting the approximation into the cube $[0, 255]^n$.

Acknowledgments

Table 1: RRE, SSIM, and CPU time in seconds obtained with the different choice rule for the regularization parameter considered.

Test case	Choice rule	RRE	SSIM	CPU time (sec.)
Tomograpy	Optimal	0.19614	0.58851	5.975
	DP	0.19897	0.62124	7.345
	CV	0.20239	0.53228	1.012×10^3
	MCV	0.19687	0.57299	2.072×10^3
	GCV	0.19945	0.55018	8.605×10^1
	GCV-Smooth	0.28796	0.32062	1.036×10^2
Peppers	Optimal	0.10166	0.76351	1.164×10^1
	CV	0.18149	0.58044	9.570×10^2
	MCV	0.15015	0.64031	1.847×10^3
	GCV	0.099600	0.74446	9.375×10^1
	GCV-Smooth	0.077343	0.81442	8.915×10^1
Cameraman	Optimal	0.092446	0.84164	1.121×10^1
	CV	0.16119	0.64255	1.050×10^3
	MCV	0.12752	0.75052	2.020×10^3
	GCV	0.098815	0.73118	1.182×10^2
	GCV-Smooth	0.047421	0.93320	9.317×10^1

References

1. Bai, Z.: The CSD, GSVD, their applications and computation **958** (1992)
2. Bianchi, D., Buccini, A.: Generalized structure preserving preconditioners for frame-based image deblurring. Mathematics **8**(4), 468 (2020)
3. Bianchi, D., Buccini, A., Donatelli, M.: Structure preserving preconditioning for frame-based image deblurring. In: M. Donatelli, S. Serra-Capizzano (eds.) Computational Methods for Inverse Problems in Imaging, pp. 33–49. Springer, Cham (2019)
4. Buccini, A., Donatelli, M.: A multigrid frame based method for image deblurring. Electronic Transactions on Numerical Analysis **53**, 283–312 (2020)
5. Buccini, A., Park, Y., Reichel, L.: Numerical aspects of the nonstationary modified linearized Bregman algorithm. Applied Mathematics and Computation **337**, 386–398 (2018)
6. Buccini, A., Pasha, M., Reichel, L.: Linearized Krylov subspace Bregman iteration with nonnegativity constraint. Numerical Algorithms (In press)
7. Buccini, A., Reichel, L.: An ℓ^2-ℓ^q regularization method for large discrete ill-posed problems. Journal of Scientific Computing **78**(3), 1526–1549 (2019)
8. Buccini, A., Reichel, L.: An ℓ^p-ℓ^q minimization method with cross-validation for the restoration of impulse noise contaminated images. Journal of Computational and Applied Mathematics **375**, 112,824 (2020)
9. Buzug, T.M.: Computed tomography. In: R. Kramme, K.P. Hoffmann, R.S. Pozos (eds.) Springer Handbook of Medical Technology, pp. 311–342. Springer, Berlin (2011)
10. Cai, J.F., Chan, R.H., Shen, L., Shen, Z.: Simultaneously inpainting in image and transformed domains. Numerische Mathematik **112**(4), 509–533 (2009)
11. Cai, J.F., Chan, R.H., Shen, Z.: A framelet-based image inpainting algorithm. Applied and Computational Harmonic Analysis **24**(2), 131–149 (2008)
12. Cai, J.F., Osher, S., Shen, Z.: Linearized Bregman iterations for frame-based image deblurring. SIAM Journal on Imaging Sciences **2**(1), 226–252 (2009)

13. Cai, Y., Donatelli, M., Bianchi, D., Huang, T.Z.: Regularization preconditioners for frame-based image deblurring with reduced boundary artifacts. SIAM Journal on Scientific Computing **38**(1), B164–B189 (2016)
14. Chan, R.H., Liang, H.X.: Half-quadratic algorithm for ℓ_p-ℓ_q problems with applications to TV-ℓ_1 image restoration and compressive sensing. In: Efficient Algorithms for Global Optimization Methods in Computer Vision, Lecture Notes in Computer Sci. # 8293, pp. 78–103. Springer (2014)
15. Donatelli, M., Neuman, A., Reichel, L.: Square regularization matrices for large linear discrete ill-posed problems. Numerical Linear Algebra with Applications **19**(6), 896–913 (2012)
16. Dykes, L., Noschese, S., Reichel, L.: Rescaling the GSVD with application to ill-posed problems. Numerical Algorithms **68**, 531–545 (2015)
17. Engl, H.W., Hanke, M., Neubauer, A.: Regularization of Inverse Problems. Kluwer, Doordrecht (1996)
18. Estatico, C., Gratton, S., Lenti, F., Titley-Peloquin, D.: A conjugate gradient like method for p-norm minimization in functional spaces. Numerische Mathematik **137**(4), 895–922 (2017)
19. Fenu, C., Reichel, L., Rodriguez, G.: GCV for Tikhonov regularization via global Golub-Kahan decomposition. Numerical Linear Algebra with Applications **23**(3), 467–484 (2016)
20. Fenu, C., Reichel, L., Rodriguez, G., Sadok, H.: GCV for Tikhonov regularization by partial SVD. BIT Numerical Mathematics **57**, 1019–1039 (2017)
21. Gazzola, S., Hansen, P.C., Nagy, J.G.: IR Tools: a MATLAB package of iterative regularization methods and large-scale test problems. Numerical Algorithms **81**(3), 773–811 (2019)
22. Gazzola, S., Novati, P., Russo, M.R.: On Krylov projection methods and Tikhonov regularization. Electronic Transactions on Numerical Analysis **44**, 83–123 (2015)
23. Golub, G.H., Heath, M., Wahba, G.: Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics **21**(2), 215–223 (1979)
24. Golub, G.H., von Matt, U.: Generalized cross-validation for large-scale problems. Journal of Computational and Graphical Statistics **6**, 1–34 (1997)
25. Golub, G.H., Van Loan, C.F.: Matrix Computations, 4th. edition. Johns Hopkins University Press, Baltimore (2013)
26. Hanke, M., Hansen, P.C.: Regularization methods for large-scale problems. Surveys on Mathematics for Industry **3**(4), 253–315 (1993)
27. Hansen, P.C.: Rank Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion. SIAM, Philadelphia (1998)
28. Hansen, P.C., Nagy, J.G., O'Leary, D.P.: Deblurring Images: Matrices, Spectra, and Filtering. SIAM, Philadelphia (2006)
29. Hansen, P.C., Nagy, J.G., Tigkos, K.: Rotational image deblurring with sparse matrices. BIT Numerical Mathematics **54**, 649–671 (2014)
30. Huang, G., Lanza, A., Morigi, S., Reichel, L., Sgallari, F.: Majorization-minimization generalized Krylov subspace methods for ℓ_p-ℓ_q optimization applied to image restoration. BIT Numerical Mathematics **57**, 351–378 (2017)
31. Huang, G., Reichel, L., Yin, F.: On the choice of solution subspace for nonstationary iterated Tikhonov regularization. Numerical Algorithms **81**, 33–55 (2019)
32. Kindermann, S.: Convergence analysis of minimization-based noise level-free parameter choice rules for linear ill-posed problems. Electron. Trans. Numer. Anal. **38**, 233–257 (2011)
33. Kindermann, S., Raik, K.: A simplified L-curve as error estimator. Electron. Trans. Numer. Anal. **53**, 213–238 (2020)
34. Lampe, J., Reichel, L., Voss, H.: Large-scale Tikhonov regularization via reduction by orthogonal projection. Linear Algebra and its Applications **436**, 2845–2865 (2012)
35. Lanza, A., Morigi, S., Reichel, L., Sgallari, F.: A generalized Krylov subspace method for ℓ_p-ℓ_q minimization. SIAM Journal on Scientific Computing **37**(5), S30–S50 (2015)
36. Lanza, A., Morigi, S., Sgallari, F.: Constrained TV_p-ℓ_2 model for image restoration. Journal of Scientific Computing **68**(1), 64–91 (2016)
37. Reichel, L., Rodriguez, G.: Old and new parameter choice rules for discrete ill-posed problems. Numerical Algorithms **63**(1), 65–87 (2013)

38. Reichel, L., Yu, X.: Matrix decompositions for Tikhonov regularization. Electronic Transactions on Numerical Analysis **43**, 223–243 (2015)
39. Stone, M.: Cross-validatory choice and assessment of statistical prediction. Journal of the Royal Statistical Society, series B **36**, 111–147 (1977)
40. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004)