Modulus-Based Iterative Methods for Constrained Tikhonov Regularization

Zhong-Zhi Bai[a], Alessandro Buccini[b], Ken Hayami[c], Lothar Reichel[d], Jun-Feng Yin[e], Ning Zheng[c]

[a]*State Key Laboratory of Scientific/Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, P.O. Box 2719, Beijing 100190, P.R. China.*
[b]*Dipartimento di Scienza e Alta Tecnologia, Università dell'Insubria, Como, 22100, Italy.*
[c]*National Institute of Informatics, and Department of Informatics, School of Multidisciplinary Sciences, SOKENDAI (The Graduate University for Advanced Studies), 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan.*
[d]*Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA.*
[e]*Department of Mathematics, Tongji University, 1239 Siping Road, Shanghai 200092, P.R. China.*

Abstract

Tikhonov regularization is one of the most popular methods for the solution of linear discrete ill-posed problems. In many applications the desired solution is known to lie in the nonnegative cone. It is then natural to require that the approximate solution determined by Tikhonov regularization also lies in this cone. The present paper describes two iterative methods, that employ modulus-based iterative methods, to compute approximate solutions in the nonnegative cone of large-scale Tikhonov regularization problems. The first method considered consists of two steps: first the given linear discrete ill-posed problem is reduced to a small problem by a Krylov subspace method, and then the reduced Tikhonov regularization problems so obtained is solved. The second method described explores the structure of certain image restoration problems. Computed examples illustrate the performances of these methods.

Keywords: Discrete ill-posed problem, Regularization method, Constrained minimization
2000 MSC: 65F22, 15A29, 65F10, 90C20

Email addresses: bzz@lsec.cc.ac.cn (Zhong-Zhi Bai), alessandro.buccini@uninsubria.it (Alessandro Buccini), hayami@nii.ac.jp (Ken Hayami), reichel@math.kent.edu (Lothar Reichel), yinjf@tongji.edu.cn (Jun-Feng Yin), nzheng@nii.ac.jp (Ning Zheng)

1. Introduction

Many applications in science and engineering require the solution of least-squares problems of the form

$$\min_{x\in\mathbb{R}^n} \|Ax - b\|, \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$ is a large matrix whose singular values "cluster" at the origin and the vector $b \in \mathbb{R}^m$ is contaminated by error. In particular, the matrix is severely ill-conditioned and may be rank-deficient. Matrices of this kind arise from the discretization of linear ill-posed problems such as Fredholm integral equations of the first kind and, therefore, the minimization problem (1) is referred to as a linear discrete ill-posed problem. Applications include remote sensing and image restoration. In the latter application the kernel of the integral equation is known as the point-spread function (PSF) and describes the blur-contamination of an unavailable image that one would like to restore. The vector b represents known error-contaminated data and can be expressed as

$$b = b_{\text{true}} + \eta, \tag{2}$$

where b_{true} is an unknown error-free vector associated with b and η represents the error in b. We will refer to η as *noise*. In image restoration applications, b_{true} represents an unavailable blur-contaminated, but noise-free, image, while b represents an available image that has been contaminated by both blur and noise. The noise may stem from measurement and/or discretization errors. We will assume that a fairly sharp bound,

$$\|\eta\| \leq \delta, \tag{3}$$

for the error is available. Throughout this paper $\|\cdot\|$ denotes the Euclidean vector norm or the spectral matrix norm. The error-free system of equations, $Ax = b_{\text{true}}$, is assumed to be consistent. We will comment on this requirement, as well as on the bound (3), further below.

Let $A^\dagger$ denote the Moore–Penrose pseudoinverse of A. We would like to determine $x_{\text{true}} = A^\dagger b_{\text{true}}$. The minimum norm solution of (1) can be expressed as $A^\dagger b$. Due to the severe ill-conditioning of A and the presence of the error η in b, the vector

$$A^\dagger b = A^\dagger b_{\text{true}} + A^\dagger \eta = x_{\text{true}} + A^\dagger \eta$$

typically is dominated by the propagated error $A^\dagger\eta$ and then is not a useful approximation of x_{true}. Generally, a much better approximation of x_{true} can be determined by first replacing the least-squares problem (1) by a nearby minimization problem that is less sensitive to the error η in b. One of the most popular replacement methods is *Tikhonov regularization*, which in its simplest form yields the minimization problem

$$\min_{x\in\mathbb{R}^n} \left\{ \|Ax - b\|^2 + \mu \|x\|^2 \right\}, \tag{4}$$

where the scalar $\mu > 0$ is referred to as the *regularization parameter*. The normal equations associated with this minimization problem are given by

$$(A^t A + \mu I_n)x = A^t b, \tag{5}$$

which shows that (4) has the unique solution

$$x_\mu = \left(A^t A + \mu I_n\right)^{-1} A^t b. \tag{6}$$

for any fixed $\mu > 0$; see, e.g., [14, 17, 19] for properties of Tikhonov regularization. The superscript ^{t} denotes transposition and I_n is the identity matrix of order n.

The Tikhonov minimization problem (4) has two terms: the first one is a *fidelity term* that ensures that the solution x_μ approximately fits the observed data, and the second one is a *regularization term* that penalizes the Euclidean norm of x_μ. The balance between data fitting and penalization is determined by the regularization parameter μ. An imprudent choice of μ makes x_μ a poor approximation of x_{true}: if μ is too small, then the error η will be propagated and amplified in x_μ, while if μ is too large, then x_μ will be over-smoothed without displaying details that x_{true} may possess.

The *discrepancy principle* provides an approach to determine a suitable value of μ. It prescribes that $\mu > 0$ be chosen so that

$$\|Ax_\mu - b\| = \tau\delta, \tag{7}$$

where $\tau > 1$ is a user-chosen parameter independent of δ. This is a nonlinear equation for μ, which can be solved, e.g., by Newton's method; see Section 3. The discrepancy principle requires a bound (3) for the error η to be available, as well as the error-free minimization problem associated with (1) to be consistent; see [14, 20] for discussions. We will use the discrepancy principle in the computed examples reported in Section 4. However, the solution methods discussed in this paper also can be applied in conjunction with other techniques for determining a suitable value of $\mu > 0$, including the L-curve criterion and generalized cross validation; see [15, 20, 23, 31] for discussions and comparisons of many methods for determining a suitable value of the regularization parameter.

Due to the fact that many singular values of the matrix A cluster at the origin, the least-squares problem (1) may be numerically rank-deficient. Therefore, it is generally beneficial to impose constraints on the computed solution that the desired solution x_{true} is known to satisfy. For instance, in image restoration problems the entries of the vector (6) represent pixel values of the image. Pixel values are nonnegative and, therefore, it is generally meaningful to solve the constraint minimization problem

$$x_\mu^+ = \arg\min_{x \geq 0} \left\{ \|Ax - b\|^2 + \mu \|x\|^2 \right\} \tag{8}$$

instead of (4). Here $x \geq 0$ is intended component-wise. A closed form of the solution x_μ^+ generally is not available.

Let Ω denote the nonnegative cone and let P_Ω be the orthogonal projector from $\mathbb{R}^n$ to Ω. Thus, we determine $P_\Omega(z)$ for $z \in \mathbb{R}^n$ by setting all negative entries of z to zero. An approximation of x_μ^+ is furnished by

$$x_\mu^\Omega = P_\Omega(x_\mu) = P_\Omega\left(\left(A^tA + \mu I\right)^{-1} A^t b\right). \tag{9}$$

When $x_{\text{true}} \geq 0$, the vector x_μ^Ω generally is a better approximation of x_{true} than x_μ. However, typically x_μ^+ is a much more accurate approximation of x_{true} than x_μ^Ω. This depends, at least in part, on the fact that the condition number of the matrix A restricted to Ω typically is smaller than the condition number of A; the latter is defined as the ratio between the largest and smallest singular values of A; see, e.g., [16].

One of the aims of the present paper is to discuss the solution of the constrained Tikhonov regularization problem (8) by the modulus-based iterative method described by Zheng et al. [34], who discuss the application of this kind of method to the solution of nonnegative constrained least-squares (NNLS) problems,

$$\min_{x\geq 0} \|Ax - b\|, \tag{10}$$

with a matrix $A \in \mathbb{R}^{m\times n}$ that is either well conditioned or ill conditioned. We restrict our attention to matrices whose singular values cluster at the origin. This allows us to determine accurate approximate solutions of (8) in a Krylov subspace of, generally, fairly small dimension. The computational effort therefore generally is smaller than when solving minimization problem (10) with a general rectangular matrix $A \in \mathbb{R}^{m\times n}$ with $m \geq n$.

We also consider the situation when A is a block circulant with circulant blocks (BCCB) matrix. Such matrices arise in certain image restoration problems. We describe a modulus-based iterative method for the NNLS problem (10) that exploits the structure of A. The fast Fourier transform (FFT) is applied to obtain an efficient solution method.

This paper is organized as follows: Section 2 reviews results about modulus-based iterative methods discussed in [3, 12, 18, 34]. We apply a modulus-based iterative method to the solution of large-scale constrained Tikhonov regularization problems of the form (8). These problems are reduced to small size ones by a Krylov subspace method. This reduction decreases the computational effort required for the solution of the constrained Tikhonov regularization problem considerably. Section 3 describes our Krylov subspace-based method for the solution of (8) and Section 4 contains a few computed examples. The latter section also illustrates how the BCCB structure of the matrix A can be exploited. Concluding remarks can be found in Section 5.

We conclude this section with some comments on related work on the computation of a nonnegative approximate solution of problem (10) with a large matrix A, whose singular values cluster at the origin. The importance of being able to solve this kind of problem has spurred the development of a variety of methods. Nagy and Strakoš [27] described a curtailed steepest descent method that determines nonnegative solutions. Active set methods based on Tikhonov

regularization were developed in [24, 26] and barrier methods for Tikhonov regularization were discussed in [8, 25, 32]. A discussion of many optimization methods, including active set and barrier methods, was provided by Nocedal and Wright [29]. In our experience, it is beneficial to use methods that exploit special properties or structure of the matrix A. The fact that the matrix A can be approximated well by a matrix of low rank makes our Krylov subspace modulus-based method competitive with available Krylov subspace methods, because it only requires the computation of one Krylov subspace of modest dimension. This subspace then is used repeatedly. When $A \in \mathbb{R}^{n\times n}$ is a BCCB matrix, fast solution methods are based on the fact that matrix-vector products with A can be evaluated in only $\mathcal{O}(n \log n)$ arithmetic floating point operations (flops) with the aid of the FFT.

2. Modulus-based iterative methods for constrained least-squares problems

This section summarizes results discussed in [12, 18, 34] of interest for the solution methods of the present paper. Other recent discussions on modulus-based iterative methods were provided by Bai [3] and Bai and Zhang [5], where further references can be found.

We reduce the constrained least-squares problem (10) to a linear complementarity problem, which we will solve by a modulus-based iterative method. The following result can be found in Cottle et al. [9, Page 5, Definition 3.3.1 and Theorem 3.3.7]. It is also shown in [34, Theorem 2.1].

Theorem 1. *Let M be a symmetric positive semidefinite matrix. Then the nonnegatively constrained quadratic programming problem,*

$$\min_{x\geq 0}\left(\frac{1}{2}x^tMx + c^tx\right),$$

denoted by NNQP(M, c), is equivalent to the linear complementarity problem,

$$x \geq 0, \quad Mx + c \geq 0, \quad \textit{and} \quad x^t\left(Mx + c\right) = 0,$$

denoted by LCP(M, c).

The results below, shown in [12, 18, 34], are consequences of the above theorem.

Corollary 2. *Let $M \in \mathbb{R}^{n\times n}$ be symmetric and positive definite and let $c \in \mathbb{R}^n$. Then the problems NNQP(M, c) and LCP(M, c) have the same unique solution.*

Corollary 3. *The NNLS problem* (10) *is equivalent to LCP($A^tA, -A^tb$),*

$$x \geq 0, \quad r = A^tAx - A^tb \geq 0, \quad \textit{and} \quad x^tr = 0.$$

It has a unique solution when A is of full column rank.

Theorem 4. *Let D be a positive definite diagonal matrix and define for $y = [y_1, y_2, \ldots, y_n]^t \in \mathbb{R}^n$ the vector $|y| = [|y_1|, |y_2|, \ldots, |y_n|]^t \in \mathbb{R}^n$.*

(i) If (x, r) is a solution of LCP($A^tA, -A^tb$), then $y = (x - D^{-1}r)/2$ satisfies

$$\left(D + A^tA\right) y = \left(D - A^tA\right) |y| + A^tb. \tag{11}$$

(ii) If y satisfies (11), then

$$x = |y| + y \quad \textit{and} \quad r = D(|y| - y)$$

is a solution of LCP($A^tA, -A^tb$).

Proof. The results can be shown by using [3, Theorem 2.1]. □

We will in the remainder of this section assume that the matrix A has full column rank. This requirement is satisfied by the matrix $\tilde{A}$ used in the following section; see (15). Theorem 4, and in particular (11), suggest the fixed-point iteration

$$(D + A^tA)y_{k+1} = (D - A^tA)\,|y_k| + A^tb, \tag{12}$$

which is the basis for the following algorithm.

Algorithm 1. *Let $y_0 \in \mathbb{R}^n$ be an initial approximate solution of (11) and let D be a positive definite diagonal matrix.*

$x_0 = y_0 + |y_0|$
for $k = 0, 1, 2, \ldots$
  $y_{k+1} = (D + A^tA)^{-1}\left((D - A^tA)\,|y_k| + A^tb\right)$
  $x_{k+1} = y_{k+1} + |y_{k+1}|$
end

This algorithm is a special case of the modulus-based matrix splitting iterative methods proposed in [3]. Its convergence was investigated in [34] based on the analysis of HSS methods by Bai et al. [4]. The case of interest to us is when $D = \alpha I_n$ with $\alpha > 0$. This iterative method is analyzed in [12, 18]. We discuss the convergence of the iterates y_k for completeness. Let y^* denote the solution of (11) for $D = \alpha I_n$, Then

$$y_{k+1} - y^* = (\alpha I_n + A^tA)^{-1}(\alpha I_n - A^tA)(|y_k| - |y^*|)$$

and we obtain

$$\begin{aligned}\|y_{k+1} - y^*\| &\le \left\|(\alpha I_n + A^tA)^{-1}(\alpha I_n - A^tA)\right\| \||y_k| - |y^*|\| \\ &\le \left\|(\alpha I_n + A^tA)^{-1}(\alpha I_n - A^tA)\right\| \|y_k - y^*\|.\end{aligned}$$

The matrix $(\alpha I_n + A^tA)^{-1}(\alpha I_n - A^tA)$ is symmetric. Therefore,

$$\left\|(\alpha I_n + A^tA)^{-1}(\alpha I_n - A^tA)\right\| = \max_{\lambda_j \in \lambda(A^tA)} \left|\frac{\alpha - \lambda_j}{\alpha + \lambda_j}\right|, \tag{13}$$

where $\lambda(A^tA)$ denotes the spectrum of A^tA. Since A is of full rank, $\lambda_j > 0$ for all j. Therefore,

$$\left|\frac{\alpha-\lambda_j}{\alpha+\lambda_j}\right| < 1 \qquad \forall j.$$

Hence,

$$\left\|(\alpha I_n + A^tA)^{-1}(\alpha I_n - A^tA)\right\| < 1,$$

which shows convergence of the iteration (12) for $D = \alpha I_n$ with $\alpha > 0$. The rate of convergence generally increases when (13) decreases. Replacing $\lambda(A^tA)$ in the right-hand side of (13) by its convex hull gives an optimization problem whose solution easily can be determined,

$$\alpha^* = \arg\min_{\alpha\in\mathbb{R}} \left\{ \max_{\lambda_{\min}\leq\lambda\leq\lambda_{\max}} \left|\frac{\alpha-\lambda}{\alpha+\lambda}\right| \right\} = \sqrt{\lambda_{\min}\lambda_{\max}}. \tag{14}$$

Here $\lambda_{\min}$ and $\lambda_{\max}$ denote the smallest and largest eigenvalues of A^tA, respectively. Thus, the relaxation parameter α^* gives a near-optimal rate of convergence.

3. Krylov subspace methods for nonnegative Tikhonov regularization

This section describes the application of the modulus-based iterative method to Tikhonov regularization with nonnegativity constraint (8). We discuss how the computational effort for large-scale problems can be reduced by using a Krylov subspace method with a fixed Krylov subspace. Comments on how to exploit BCCB structure of A conclude this section.

We first rewrite the minimization problem (8) in the form of the previous section. Thus,

$$\begin{aligned} & \min_{x\geq 0}\left\{\|Ax-b\|^2 + \mu\|x\|^2\right\} \\ = \; & \min_{x\geq 0}\left\| \begin{bmatrix} A \\ \sqrt{\mu}I_n \end{bmatrix} x - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|^2 \\ = \; & \min_{x\geq 0}\left\|\tilde{A}x-\tilde{b}\right\|^2, \end{aligned} \tag{15}$$

where we assume that $\mu > 0$. Then the matrix $\tilde{A} \in \mathbb{R}^{(m+n)\times n}$ is of full column rank and the minimization problem (15) satisfies the conditions in Section 2. Therefore, the iterates determined by Algorithm 1 will converge.

When the matrix A is large and without exploitable structure, the computation with Algorithm 1 with $D = \alpha I_n$ may be expensive. In particular, factoring the matrix $\alpha I_n + \tilde{A}^t\tilde{A}$ in order to solve the linear systems of equations with this matrix required by Algorithm 1 may be unattractive or infeasible. We are interested in trying to reduce the computational effort required for solving these linear systems of equations. One way to achieve this is to solve them by the conjugate gradient method. It is convenient to use the CGLS implementation [7]. This solution approach is discussed in [34] and also is illustrated in Section 4.

The present paper describes an alternative way to reduce the computational effort. We first determine an initial reduction of A to a small bidiagonal matrix with the aid of Golub–Kahan bidiagonalization. The Krylov solution subspace generated by this reduction method then is reused for all linear systems of equations with the matrix $\alpha I_n + \tilde{A}^t\tilde{A}$ that have to be solved.

Application of $\ell \ll \min\{m,n\}$ steps of Golub–Kahan bidiagonalization to A with initial vector $u_1 = b/\|b\|$ gives the decompositions

$$AV_\ell = U_{\ell+1}B_{\ell+1,\ell}, \qquad A^tU_\ell = V_\ell B_{\ell,\ell}^t, \tag{16}$$

where $U_{\ell+1} = [u_1, u_2, \ldots, u_{\ell+1}] \in \mathbb{R}^{m\times(\ell+1)}$ and $V_\ell = [v_1, v_2, \ldots, v_\ell] \in \mathbb{R}^{n\times\ell}$ have orthonormal columns, $U_\ell \in \mathbb{R}^{m\times\ell}$ is made up of the first ℓ columns of $U_{\ell+1}$, $B_{\ell+1,\ell} \in \mathbb{R}^{(\ell+1)\times\ell}$ is lower bidiagonal with positive diagonal and subdiagonal entries, and $B_{\ell,\ell}$ is the leading $\ell\times\ell$ submatrix of $B_{\ell+1,\ell}$; see, e.g., [16] for details on the decompositions (16). We assume for now that ℓ is chosen small enough so that the decompositions (16) with the stated properties exist. For future reference, we note that the columns of V_ℓ span the Krylov subspace

$$\mathcal{K}_\ell(A^tA, A^tb) = \text{span}\{A^tb, (A^tA)A^tb, \ldots, (A^tA)^{\ell-1}A^tb\}. \tag{17}$$

Substituting $x = V_\ell y$, $y \in \mathbb{R}^\ell$, into (5) and determining an approximate solution by a Galerkin method gives the equation

$$V_\ell^t(A^tA + \mu I_n)V_\ell y = V_\ell^t A^t b,$$

which with the aid of the decompositions (16) can be expressed as

$$(B_{\ell+1,\ell}^t B_{\ell+1,\ell} + \mu I_\ell)y = e_1 \left\|A^tb\right\|. \tag{18}$$

Here and below e_j denotes the jth column of an identity matrix of appropriate order. The reduced Tikhonov equation (18) is the normal equation associated with the least-squares problem

$$\min_{y\in\mathbb{R}^\ell} \left\| \begin{bmatrix} B_{\ell+1,\ell} \\ \mu^{1/2} I_\ell \end{bmatrix} y - \mu^{-1/2} e_{\ell+2}\|A^tb\| \right\|. \tag{19}$$

We solve the latter instead of (18) for $y = y_\mu$ for reasons of numerical stability. For each fixed $\mu > 0$, the least-squares problem (19) can be solved in only $\mathcal{O}(\ell)$ arithmetic floating-point operations; see Eldén [13] for details on the solution of least-squares problems of the form (19).

We turn to the determination of $\mu > 0$ by the discrepancy principle. Substituting $x_\mu = V_\ell y$ into (7) and using (16) gives the reduced problem

$$\|B_{\ell+1,\ell}y - e_1\|b\|\,\| = \tau\delta, \tag{20}$$

where y solves (19).

Proposition 5. *Introduce the function*

$$\phi_\ell(\mu) = \|b\|^2 e_1^t(\mu^{-1}B_{\ell+1,\ell}B_{\ell+1,\ell}^t + I_{\ell+1})^{-2}e_1. \tag{21}$$

Then the solution $\mu > 0$ of

$$\phi_\ell(\mu) = \tau^2\delta^2 \tag{22}$$

determines a solution $y = y_\mu$ of (19) that solves (20). The vector $x_{\mu,\ell} = V_\ell y_\mu$ satisfies (7).

Proof. It follows from (16) that

$$A^t b = A^t U_\ell e_1 \|b\| = v_1\, e_1^t B_{\ell+1,\ell}^t e_1 \|b\|.$$

Substituting this expression and the solution of (18) into the left-hand side of (20) gives

$$\begin{aligned} & \|B_{\ell+1,\ell}(B_{\ell+1,\ell}^t B_{\ell+1,\ell} + \mu I_\ell)^{-1} e_1 \|A^t b\| - e_1 \|b\| \,\|^2 \\ = \;& \|B_{\ell+1,\ell}(B_{\ell+1,\ell}^t B_{\ell+1,\ell} + \mu I_\ell)^{-1} B_{\ell+1,\ell}^t e_1 - e_1\|^2 \,\|b\|^2. \end{aligned} \tag{23}$$

The identity

$$B_{\ell+1,\ell}(B_{\ell+1,\ell}^t B_{\ell+1,\ell} + \mu I_\ell)^{-1} B_{\ell+1,\ell}^t - I_{\ell+1} = -(\mu^{-1} B_{\ell+1,\ell} B_{\ell+1,\ell}^t + I_{\ell+1})^{-1}$$

can be shown, e.g., by multiplication by $B_{\ell+1,\ell}B_{\ell+1,\ell}^t + \mu I_{\ell+1}$ from the right-hand side. Substitution into (23) gives

$$\|B_{\ell+1,\ell} y_\mu - e_1 \|b\| \,\|^2 = \|b\|^2 e_1^t (\mu^{-1} B_{\ell+1,\ell} B_{\ell+1,\ell}^t + I_{\ell+1})^{-2} e_1.$$

This shows (21). The fact that the vector $x_{\mu,\ell}$ satisfies (7) follows from (16) and (20). □

Proposition 6. *Let $\phi_\ell(\mu)$ be defined by (21). Then the function $\nu \to \phi_\ell(1/\nu)$ is strictly decreasing and convex for $\nu > 0$. Moreover,*

$$\lim_{\mu\to\infty} \phi_\ell(\mu) = \|b\|^2.$$

In particular, Newton's method applied to the solution of the equation $\phi_\ell(1/\nu) = \tau^2\delta^2$ with initial approximate solution ν_0 to the left of the solution converges monotonically and quadratically.

Proof. The decrease, convexity, and limit follows from the representation

$$\phi_\ell(1/\nu) = \|b\|^2 e_1^t (\nu B_{\ell+1,\ell} B_{\ell+1,\ell}^t + I_{\ell+1})^{-2} e_1.$$

Newton's method converges monotonically and quadratically for decreasing convex functions when the initial iterate is smaller than the solution. The initial iterate ν_0 can be chosen to be zero with

$$\lim_{\nu\searrow 0} \phi_\ell(1/\nu) = \|b\|^2, \qquad \lim_{\nu\searrow 0} \frac{d}{d\nu}\phi_\ell(1/\nu) = -2\|b\|^2 \|B_{\ell+1,\ell}^t e_1\|^2.$$

□

In actual computations it typically suffices to choose $\ell \ll \min\{m, n\}$. We apply Algorithm 1 to the reduced Tikhonov minimization problem (18). Thus, we replace $A^t A$ in the algorithm by

$$T_{\ell,\mu} = B_{\ell+1,\ell}^t B_{\ell+1,\ell} + \mu I_\ell. \tag{24}$$

Since the matrix $B_{\ell+1,\ell}$ is small, we can easily determine its largest singular value $\sigma_{\max}$. Typically, zero is a quite sharp lower bound for the smallest singular value. The largest eigenvalue of $T_{\ell,\mu}$ is $\sigma_{\max}^2 + \mu$ and the smallest eigenvalue is bounded below by, and is generally close to, μ. We will use the relaxation parameter

$$\alpha = \sqrt{(\sigma_{\max}^2 + \mu)\mu} \tag{25}$$

for the algorithm; cf. (14). This yields the following scheme.

Algorithm 2. *Choose the number of Golub–Kahan bidiagonal steps, ℓ, and compute the decompositions* (16)*. Determine a regularization parameter μ that satisfies* (22) *as described in Proposition 6. Compute the solution $y = y_\mu$ of* (19) *and define the initial approximate solution $x_0 = P_\Omega(V_\ell y_\mu)$ of* (8)*. Determine the largest singular value of the matrix $B_{\ell+1,\ell}$ and define the relaxation parameter* (25)*. Let $T_{\ell,\mu}$ be given by* (24)*.*

$$\begin{aligned}
&\hat{b} = e_1 \|A^t b\| \\
&y_0 = V_\ell^t x_0 \\
&\tilde{y}_0 = V_\ell^t |V_\ell y_0| \\
&\textit{for } k = 0, 1, 2, \ldots \textit{ until convergence} \\
&\quad y_{k+1} = (\alpha I_\ell + T_{\ell,\mu})^{-1} \left((\alpha I_\ell - T_{\ell,\mu}) \tilde{y}_k + \hat{b} \right) \\
&\quad \tilde{y}_{k+1} = V_\ell^t |V_\ell y_{k+1}| \\
&\textit{end} \\
&x = V_\ell \tilde{y}_{k+1} + |V_\ell \tilde{y}_{k+1}|
\end{aligned}$$

The above algorithm computes the magnitude of every entry of an n-vector at each step. Therefore, a transformation from the ℓ-dimensional subspace, where the vectors y_k live, to $\mathbb{R}^n$ is required. Every step demands the solution of a linear system of equations of the form

$$(\alpha I_\ell + T_{\ell,\mu})\, z = d$$

for some vector d. The solution z can be computed by solving a least-squares problem analogous to (19).

We remark that the Krylov subspace (17) is invariant under shifts of $A^t A$ by a multiple of the identity, i.e.,

$$\mathcal{K}_\ell(A^t A, A^t b) = \mathcal{K}_\ell(A^t A + \alpha I_n, A^t b).$$

It follows that the shifted matrix $\alpha I_\ell + T_{\ell,\mu}$ in Algorithm 2 corresponds to the shifted matrix $\alpha I_n + (\mu I_n + A^t A)$.

We described above how to determine the regularization parameter μ by first reducing equation (7) to an equation with a small matrix (20). When restoring images we sometimes may impose periodic boundary conditions without affecting the quality of the computed restoration significantly. This yields a BCCB blurring matrix $A \in \mathbb{R}^{n\times n}$, which can be diagonalized by the unitary Fourier matrix $F \in \mathbb{C}^{n\times n}$,

$$A = F^*\Sigma F. \tag{26}$$

Here the matrix Σ is diagonal, possibly with complex entries, and the superscript * denotes transposition and complex conjugation; see, e.g., [22] for details. We can transform the Tikhonov minimization problem (4) to a minimization problem with a diagonal matrix, and we also can transform (7) to an equation with a diagonal matrix. These transformations allow easy computation of the regularization parameter $\mu > 0$ such that the solution (6) satisfies (7) by Newton's method analogously as described above. Moreover, Algorithm 1 with $D = \alpha I_n$ can be executed efficiently when A has the factorization (26). This is illustrated in the following section.

4. Numerical examples

This section presents a few numerical examples that illustrate the performance of the algorithms described. An example in one space-dimension from the Regularization Tools MATLAB package [21] and examples in two space-dimensions obtained with the Restore Tools MATLAB package [6] will be discussed. We compare four methods: classical unconstrained Tikhonov regularization (6), projected Tikhonov regularization (9), and Algorithms 1 and 2. In Algorithm 1, we solve linear systems of equations with the matrix $D + A^tA$ by the CGLS method; see [7]. In all examples, $D = \alpha I_n$ with $\alpha > 0$. Algorithm 3 below illustrates how the availability of a factorization of the form (26) can be exploited.

At step k of Algorithm 1, we have to solve the linear system of equations

$$(A^tA + \mu I_n + \alpha I_n)y_{k+1} = (\alpha I_n - A^tA - \mu I_n)|y_k| + A^tb, \tag{27}$$

which is equivalent to the least-squares problem

$$y_{k+1} = \arg\min_y \left\| \begin{bmatrix} A \\ \sqrt{\mu} I_n \\ \sqrt{\alpha} I_n \end{bmatrix} y - \begin{bmatrix} b - A|y_k| \\ -\sqrt{\mu}|y_k| \\ \sqrt{\alpha}|y_k| \end{bmatrix} \right\|^2 = \arg\min_y \left\| \bar{A}y - \bar{y}_k \right\|,$$

for a suitably defined matrix $\bar{A} \in \mathbb{R}^{(m+2n)\times n}$ and vector $\bar{y}_k \in \mathbb{R}^{m+2n}$. We terminate the iterations with the CGLS method at iteration k of Algorithm 1 as soon as

$$\left\| \bar{A}^t(\bar{A}y - \bar{b}) \right\| < \frac{10^{-2}}{k} \left\| \bar{A}^t\bar{b} \right\|, \text{ for } k = 0, 1, \ldots.$$

This stopping criterion takes the scalings of A and b into account. Both execution times and accuracy of the methods in our comparison are tabulated. The

accuracy of a computed approximation x of x_{true} is measured by the relative error

$$\text{RE}(x) = \frac{\|x - x_{\text{true}}\|}{\|x_{\text{true}}\|}.$$

The iterations with Algorithms 1 and 2 are terminated when two consecutive iterates y_k and y_{k+1} are close enough, i.e., as soon as

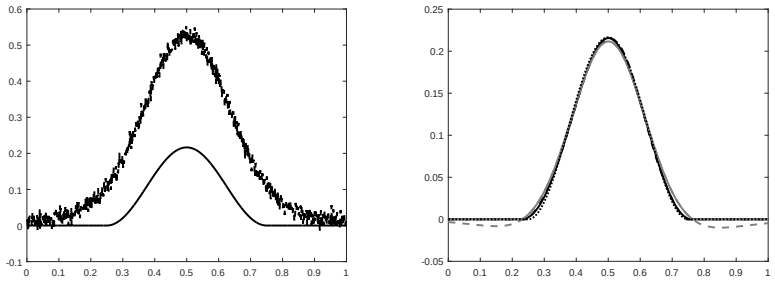
$$\frac{\|y_{k+1} - y_k\|}{\|y_k\|} < s,$$

where s is a user-supplied constant. We set $s = 10^{-4}$ in all examples. The regularization parameter μ is determined by the discrepancy principle, i.e., μ is chosen such that (7) holds, where $\delta = \|\eta\|$ and $\tau = 1.01$.

Determination of a near-optimal relaxation parameter α for Algorithms 2 is straightforward. We have to compute the largest and smallest eigenvalues of the matrix $T_{\ell,\mu}$ in the algorithm; see the discussion preceding (25). We turn to Algorithm 1. This algorithm requires the estimation of the largest and smallest eigenvalues of the matrix $A^tA+\mu I_n$; see (27). We briefly comment on how these eigenvalues can be computed when the matrix A is large. Let A be scaled to have norm about unity. Since A stems from the discretization of an ill-posed problem, it has many singular values close to the origin. It follows that an accurate estimate of the smallest eigenvalue of the matrix $A^tA + \mu I_n$ is given by μ. An estimate of the largest eigenvalue of this matrix can be determined by computing an estimate of the largest singular value of A. This can be done quite inexpensively with the implicitly restarted Golub–Kahan bidiagonalization algorithm `irbla` described in [2]. The dominant computational work with this algorithm consists of the evaluation of a few matrix-vector products with the matrices A and A^t. The discussion and computed examples presented in [30] indicate that the `irbla` algorithm typically only requires the evaluation of a few of these matrix-vector products to determine the largest singular value of a matrix of a linear discrete ill-posed problem (1). The computation of an estimate of the largest singular value of A, and hence of the largest eigenvalue of $A^tA + \mu I_n$, is quite inexpensive.

In image restoration problems A is a blurring matrix. For a typical row j, blurring matrices satisfy $e_j^tAe = 1$, where $e = [1, 1, \ldots, 1]^t$. However, both e_j^tAe and $e_j^tA^tAe$ may differ from one for certain j. In particular, $e_j^tA^tAe$ may be significantly larger than one for some j values. The size of $\max_{1\leq j\leq n} |e_j^tA^tAe|$ depends on the boundary conditions used; see, e.g., [11] for a discussion. We conclude that for some image restoration problems, the largest singular value of A is close to unity and does not have to be computed. However, certain image restoration problems, in particular problems with anti-reflective boundary conditions, may require that the largest singular value of the blurring matrix be computed as described above.

All the computations in this section were carried out in MATLAB version 9.0.0.341360 (R2016a) on a laptop computer with an Intel i7-6700HQ @ 2.60 Ghz CPU and 8 GB of RAM. The computations were done with about 15 significant decimal digits.



(a) (b)

Figure 1: Shaw test problem: (a) desired solution x_{true} (solid curve) and error-contaminated data vector b_{true} (dashed curve), (b) computed solutions obtained with classical Tikhonov x_μ (dashed gray curve), projected Tikhonov x_μ^Ω (solid gray curve), Algorithm 1 (dashed black curve), and Algorithm 2 (solid black curve).

Method	RE	CPU time	Iterations
Tikhonov	0.073600	0.39441	–
Projected Tikhonov	0.052816	0.40583	–
Algorithm 1	0.029923	0.61289	36
Algorithm 2	**0.024316**	0.062824	37

Table 1: Shaw test problem: relative errors (RE) and CPU times in seconds for standard Tikhonov (4), projected Tikhonov (9), Algorithm 1, and Algorithm 2. For the last two algorithms also the number of iterations is displayed. The smallest error is shown in boldface.

Shaw. We consider a modified version of the `shaw` example in [21]. To show the effectiveness of our methods, we use the discretized integral operator from `shaw` and the exact solution x_{true} from the `phillips` example, also from [21]. We choose this solution vector because it is nonnegative with many vanishing components. The discretized operator is represented by a matrix $A \in \mathbb{R}^{1024\times 1024}$. Thus, $x_{\text{true}} \in \mathbb{R}^{1024}$ represents the desired solution and the noise-free data vector is given by $b_{\text{true}} = Ax_{\text{true}}$. We add 5% white Gaussian noise to b_{true} to obtain the noise-contaminated vector b in (1); cf. (2).

Figure 1(a) shows the vectors x_{true} and b. For Algorithm 2, we use a Krylov subspace of dimension $\ell = 30$. Table 1 displays CPU times as well as the relative errors in the computed approximations of x_{true} determined by the different methods. Algorithm 2 is seen to require fewer iterations and less CPU time than the other methods. The standard Tikhonov method is implemented by first computing the singular value decomposition of A. We remark that Algorithm 2 performs particularly well for discrete ill-posed problems (1) with a matrix A whose singular values decay to zero fairly quickly, because in this situation the dimension ℓ of the Krylov subspace can be chosen fairly small. When the singular values decay slowly and, therefore, ℓ has to be chosen rather large, Algorithm 1 may be competitive with Algorithm 2.

We now compare Algorithm 2 with an active set method designed for the solution of nonnegatively constrained linear discrete ill-posed problems (10). Our comparison is with the method described in [26]. The performances of this active set method and the one discussed in [24] are quite similar. We therefore only compare with the former. It is based on repeatedly reducing the large problem (1) to a problem of small size with the aid of a few steps of Golub–Kahan bidiagonalization of the matrix A or of a matrix AD. Here D is a diagonal matrix with diagonal entries one or zero. The diagonal entries are zero if the corresponding variable is in the active set. The reduction of A or AD by Golub–Kahan bidiagonalization proceeds until an approximate solution that satisfies the discrepancy principle has been found. If the computed approximate solution satisfies the constraints, then we are done; otherwise those variables that violate the constraint are projected into the feasible set and the active set is updated. This means that the matrix D is updated. If the projected solution satisfies the discrepancy principle, then we also are done; otherwise a partial Golub–Kahan bidiagonalization of the new matrix AD is computed. The computations proceed in this manner until a feasible approximate solution of (10) that satisfies the discrepancy principle has been found. Updating the active set only when the discrepancy principle holds gives a much faster method than if the active set were updated as soon as a constraint is violated. However, this updating strategy may allow "cycling". It is discussed in [24] how cycling can be detected and avoided. Computed examples in [24, 26] show the active set methods to perform well when the noise level is not small. However, for small noise levels many partial Golub-Kahan bidiagonalizations may have to be computed. This requires the evaluation of many matrix-vector products (MvPs) and can make the methods slow. We remark that the evaluation of these MvPs is the dominating work for large-scale problems.

Noise Level	Method	RE	MvPs
0.1%	Active set method	0.018188	36
	Algorithm 2	0.013495	30 ($\ell = 15$)
0.05%	Active set method	0.0093832	47
	Algorithm 2	0.013320	30 ($\ell = 15$)

Table 2: Shaw test problem: relative errors (RE) and number of MvPs for the active set method [24] and for Algorithm 2. Results are shown for two noise levels.

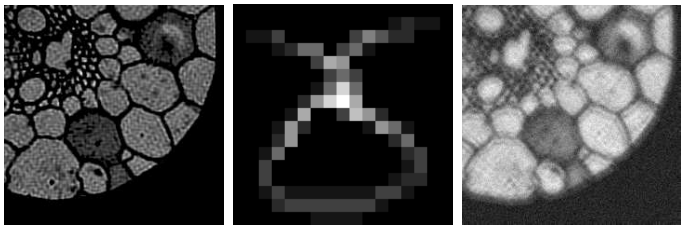
Table 2 illustrates that, differently from the active set method [26], Algorithm 2 does not require more computational effort when the noise level is reduced. We compare Algorithm 2 in terms of accuracy of the computed restoration and in terms of the number of MvP evaluations required. For Algorithm 2 the number of MvPs needed depends only on the dimension of the Krylov solution subspace used. Since we use Golub–Kahan bidiagonalization, the computation of a solution subspace of dimension ℓ requires the evaluation of 2ℓ MvPs. Table 2 displays results for two noise levels. The table shows that when the noise level decreases the number of MvP evaluations required by the active set method [26] increases, while it does not for Algorithm 2. In fact, it may be possible to choose a Krylov subspace of smaller dimension for Algorithm 2 for small noise levels and this would reduce the number of MvP evaluations required.

Grain. We turn to image deblurring in two space-dimensions. We blur the true image Grain from [6] using a non-symmetric PSF and add 10% white Gaussian noise; see Figure 2. Anti-reflective boundary conditions are imposed; see [33] for details. There is no fast transformation that can be applied to diagonalize the blurring matrix A. Therefore, we use Algorithm 2 to compute a restoration. We compare this algorithm to standard and projected Tikhonov regularization (4) and (9), respectively, and to Algorithm 1, in which the inner linear systems of equations are solved by the CGLS method; cf. the discussion at the beginning of Section 4. In Algorithm 2, we apply a Krylov subspace of dimension $\ell = 100$. Table 3 displays CPU times and the errors in the computed restorations determined by these methods. We see that Algorithm 1 requires about the same computing time as standard and projected Tikhonov regularization, and that Algorithm 2 is much faster than Algorithm 1. Moreover, Algorithm 2 gives the most accurate restoration. This is confirmed by visual inspection of Figure 3.

In this and the following examples, Tikhonov regularization (5) is implemented by solving the least-squares problem

$$\min_{y\in\mathbb{R}^n} \left\| \begin{bmatrix} A \\ \sqrt{\mu} I_n \end{bmatrix} y - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|,$$

by the CGLS method. Here $0 \in \mathbb{R}^n$ denotes the zero vector. The μ-value is determined as follows. Let C denote the blurring matrix obtained by using periodic boundary conditions. We compute μ that satisfies (22), where we substitute the matrix $B_{\ell+1,\ell}$ in (21) by C and exploit the factorization (26).



(a) (b) (c)

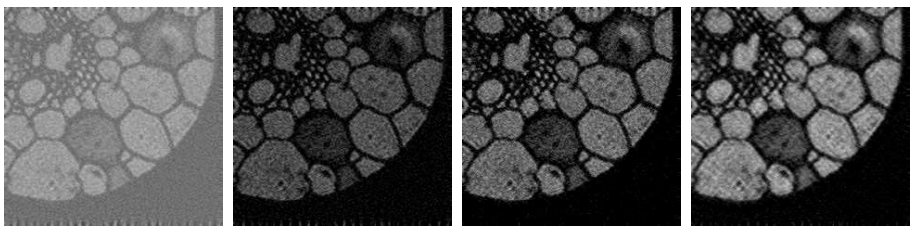
Figure 2: Grain test problem: (a) true image (238 × 238 pixels), (b) non-symmetric PSF (17 × 17 pixels), (c) blurred and noisy image with RE=0.41503.

Method	RE	CPU time	Iterations
Tikhonov	0.33043	9.2475	–
Projected Tikhonov	0.30239	9.2726	–
Algorithm 1	0.27633	8.1002	24
Algorithm 2	**0.27491**	4.7601	18

Table 3: Grain test problem: relative errors (RE) and CPU times in seconds for standard Tikhonov (4), projected Tikhonov (9), Algorithm 1, and Algorithm 2. For the last two algorithms also the number of iterations is displayed.

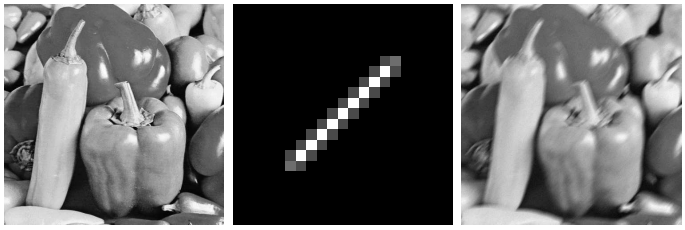
Proceeding in this way yields a suitable value of the regularization parameter μ in a computationally efficient manner. We remark that we are primarily interested in the errors in the solutions determined by the different methods. Therefore it is not necessary to implement the standard Tikhonov method as a black box method.

Peppers. We now present an example with a larger image. This example illustrates that Algorithm 2 may require much less CPU time than Algorithm 1. Figure 4 displays the true image, the motion PSF used for blurring, and the blurred and noise-contaminated image. The noise is 3% white Gaussian. Since the image is generic, we impose anti-reflective boundary conditions; see [33] for



(a) (b) (c) (d)

Figure 3: Grain restorations by (a) standard Tikhonov, (b) projected Tikhonov, (c) Algorithm 1, (d) Algorithm 2.



(a) (b) (c)

Figure 4: Peppers test problem: (a) true image (496×496 pixels), (b) motion PSF (21×21 pixels), (c) blurred and noisy image with RE=0.10793.

Method	RE	CPU time	Iterations
Tikhonov	0.31718	39.667	–
Projected Tikhonov	0.28645	44.192	–
Algorithm 1	0.16160	1.1621×10^3	55
Algorithm 2	**0.095639**	33.573	32

Table 4: Peppers test problem: relative errors (RE) and CPU times in seconds for standard Tikhonov (4), projected Tikhonov (9), Algorithm 1, and Algorithm 2. For the last two algorithms also the number of iterations is displayed.

a discussion.

Similarly as above, we compare standard Tikhonov regularization (4), the projected version (9), and Algorithms 1 and 2. We set $\ell = 100$ in the latter algorithm. Table 4 provides the relative errors of the computed restorations and the CPU times for the methods. Algorithm 2 can be seen to outperform all the other methods both with respect to accuracy in the computed restoration and computing time. In particular, while Algorithm 1 yields a restoration of high quality, it requires too much CPU time to be attractive. Figure 5 displays the restorations. The imposition of the nonnegativity constraint during the computations can be seen to give a restoration of higher quality than standard and projected Tikhonov regularizations (4) and (9).

(a) (b) (c) (d)

Figure 5: Peppers restorations by (a) standard Tikhonov, (b) projected Tikhonov, (c) Algorithm 1, (d) Algorithm 2.

Atmospheric Blur. Our last example considers the test data `AtmoshpericBlur50` from [6]. Figure 6 shows the true image, the PSF, and the observed image. Using the knowledge of the true image, we are able to determine an approximation of the noise level in the data, which turns out to be a little more than 1%. Due to the large black area near the boundary, we may impose periodic boundary conditions on the matrix A without significantly reducing the quality of the computed restoration. This makes A a BCCB matrix and matrix-vector products with matrices of the form $A + cI_n$, where c is a scalar, can be computed in only $\mathcal{O}(n \log(n))$ flops with the aid of the FFT. The FFT also can be applied to solve linear systems of equations with a matrix of the form $A + cI_n$ in only $\mathcal{O}(n \log(n))$ flops. We remark that when the available contaminated image is represented by $m \times m$ pixels, each circulant matrix that makes up A is of size $m \times m$, and A has m circulants blocks along the diagonal. Thus, $A \in \mathbb{R}^{n \times n}$ with $n = m^2$.

The spectral factorization (26) of A can be computed in $\mathcal{O}(n \log(n))$ flops. This factorization allows the solution of (7) for $\mu > 0$ by Newton's method with each iteration requiring only $\mathcal{O}(n \log(n))$ flops. Using (26), we obtain

$$A^t A + \mu I = F^*(|\Sigma|^2 + \mu I_n)F.$$

The smallest eigenvalue of this matrix generally is μ or very close to μ. Let $\sigma_{\max} = \|\Sigma\|$. We will use the acceleration parameter

$$\alpha = \sqrt{(\sigma_{\max}^2 + \mu)\mu}, \tag{28}$$

which is close to the optimal one (14).

The following algorithm is a modification of Algorithm 1 that exploits the BCCB structure of A. It uses the matrix

$$S_\mu = |\Sigma|^2 + \mu I_n.$$

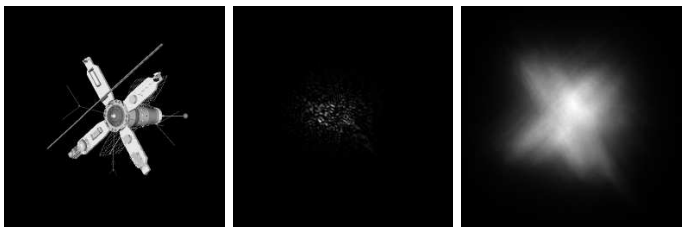
Algorithm 3. *Compute the decomposition* (26) *and determine a regularization parameter* μ *that satisfies* (22) *as outlined above. Determine the relaxation parameter* (28) *and an initial approximate solution* x_0 *of* (8).

$\hat{b} = \overline{\Sigma} F b$
$y_0 = F x_0$
$\tilde{y}_0 = F|F^* y_0|$
for $k = 0, 1, 2, \ldots$ *until convergence*
  $y_{k+1} = (\alpha I_n + S_\mu)^{-1} \left((\alpha I_n - S_\mu)\, \tilde{y}_k + \hat{b} \right)$
  $\tilde{y}_{k+1} = F|F^* y_{k+1}|$
end
$x = F^* \tilde{y}_{k+1} + |F^* \tilde{y}_{k+1}|$

Table 5 compares the CPU time required and accuracy achieved with Algorithm 3 to those for standard and projected Tikhonov regularizations (4) and (9), respectively, and to those for Algorithm 1. Algorithm 3 imposes periodic

Method	RE	CPU time	Iterations
Tikhonov	0.28354	4.4644	–
Projected Tikhonov	0.26822	4.4766	–
Algorithm 1	0.22901	98.349	131
Algorithm 3	**0.22757**	1.4647	109

Table 5: Satellite test problem: relative errors (RE) and CPU times in seconds for standard Tikhonov (4), projected Tikhonov (9), and Algorithm 3. For the last algorithm also the number of iterations is displayed.



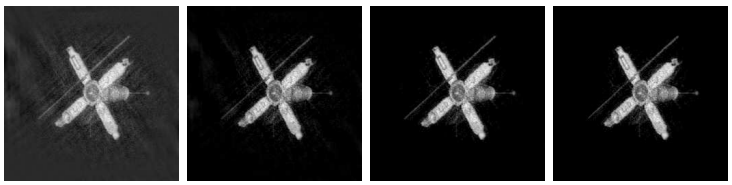
(a) (b) (c)

Figure 6: Satellite test problem from [6]: (a) true image (256×256 pixels), (b) PSF defined by atmospheric blur (256×256 pixels), (c) blurred and noisy image (256×256 pixels) with RE=0.76267.

boundary conditions, while the other methods are implemented with zero Dirichlet boundary conditions. The table shows Algorithm 3 to be the fastest and to give the most accurate restoration. The superior quality of the restoration delivered by Algorithm 3 is confirmed by Figure 7, which displays the restorations. Algorithm 3 can be seen to yield a restoration with a more homogeneous black background than the other methods. Figure 8 displays a detail of the lower-right corner of the restored images in a different color map.

We do not compare with Algorithm 2 in this example, because Algorithm 3 yields a more accurate restoration faster than the former. While Algorithm 2 performs well for many linear discrete ill-posed problems, Algorithm 3 gives superior restorations when the image is such that periodic boundary conditions can be imposed without creating significant boundary artifacts.

Algorithms analogous to Algorithm 3 can be developed for reflective and anti-reflective boundary conditions when the PSF is quadrantally symmetric, i.e., when the PSF is symmetric with regard to the horizontal and vertical axes. As an example, the PSF for Gaussian blur is quadrantally symmetric. For reflective boundary conditions the algorithm can be based on the discrete cosine transform [28] and for anti-reflective boundary conditions on the discrete sine transform [1, 10].



(a) (b) (c) (d)

Figure 7: Satellite restorations by (a) Tikhonov, (b) projected Tikhonov, (c) Algorithm 1, (d) Algorithm 3.

(a) (b) (c) (d)

Figure 8: Satellite restoration details (lower right corner) by (a) Tikhonov, (b) projected Tikhonov, (c) Algorithm 1, (d) Algorithm 3.

5. Conclusions

This paper applies modulus-based iterative methods to nonnegative Tikhonov regularization. The discrepancy principle is used to determine the regularization parameter. Efficient solution methods are described. Several numerical examples in one and two space-dimensions illustrate the efficacy of the proposed methods.

Acknowledgments

The work of the first author is supported by the National Natural Science Foundation (No. 11671393), P.R. China. And the work of the second author is supported in part by MIUR - PRIN 2012 N. 2012MTE38N and by a grant of the group GNCS of INdAM.

[1] A. Aricó, M. Donatelli, J. G. Nagy, and S. Serra–Capizzano, The anti-reflective transform and regularization by filtering, in Numerical Linear Algebra in Signals, Systems and Control, Lecture Notes in Electrical Engineering, Vol. 80, Springer, 2011, pp. 1–21.

[2] J. Baglama and L. Reichel, Augmented implicitly restarted Lanczos bidiagonalization methods, SIAM J. Sci. Comput., 27 (2005), pp. 19–42.

[3] Z.-Z. Bai, Modulus-based matrix splitting iteration methods for linear complementarity problems, Numer. Linear Algebra Appl., 17 (2010), pp. 917–933.

[4] Z.-Z. Bai, G. H. Golub, and M. K. Ng, Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 603–624.

[5] Z.-Z. Bai and L.-L. Zhang, Modulus-based synchronous multisplitting iteration methods for linear complementarity problems, Numer. Linear Algebra Appl., 20 (2013), pp. 425–439.

[6] S. Berisha and J. G. Nagy, Iterative Methods for Image Restoration, Technical Report, Department of Mathematics and Computer Science, Emory University, 2012;
`http://www.mathcs.emory.edu/~nagy/RestoreTools/IR.pdf`

[7] Å. Björck, Numerical Methods for Least Squares Problems, SIAM, Philadelphia, 1996.

[8] D. Calvetti, B. Lewis, L. Reichel, and F. Sgallari, Tikhonov regularization with nonnegativity constraint, Electron. Trans. Numer. Anal., 18 (2004), pp. 153–173.

[9] R. W. Cottle, J.-S. Pang, and R. E. Stone, The Linear Complementarity Problem, Academic Press, San Diego, 1992.

[10] M. Donatelli, C. Estatico, A. Martinelli, and S. Serra–Capizzano, Improved image deblurring with anti-reflective boundary conditions and re-blurring. Inverse Problems, 22 (2006), pp. 2035–2053.

[11] M. Donatelli, D. Martin, and L. Reichel, Arnoldi methods for image deblurring with anti-reflective boundary conditions, Appl. Math. Comput., 253 (2015), pp. 135–150.

[12] J.-L. Dong and M.-Q. Jiang, A modified modulus method for symmetric positive-definite linear complementarity problems, Numer. Linear Algebra Appl., 16 (2009), pp. 129–143.

[13] L. Eldén, Algorithms for the regularization of ill-conditioned least squares problems, BIT, 17 (1977), pp. 134–145.

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

[15] C. Fenu, L. Reichel, and G. Rodriguez, GCV for Tikhonov regularization via global Golub–Kahan decomposition, Numer. Linear Algebra Appl., 23 (2016), pp. 467–484.

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

[17] C. W. Groetsch, The Theory of Tikhonov Regularization for Fredholm Equations of the First Kind, Pitman, Boston, 1984.

[18] A. Hadjidimos and M. Tzoumas, Nonstationary extrapolated modulus algorithms for the solution of the linear complementarity problem, Linear Algebra Appl., 431 (2009), pp. 197–210.

[19] M. Hanke and P. C. Hansen, Regularization methods for large-scale problems, Surveys Math. Indust., 3 (1993), pp. 253–315.

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

[21] P. C. Hansen, Regularization tools version 4.0 for Matlab 7.3, Numer. Algorithms, 46 (2007), pp. 189–194.

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

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

[24] S. Morigi, R. Plemmons, L. Reichel, and F. Sgallari, A hybrid multilevel-active set method for large box-constrained linear discrete ill-posed problems, Calcolo, 48 (2011), pp. 89–105.

[25] S. Morigi, L. Reichel, and F. Sgallari, An interior-point method for large constrained discrete ill-posed problems, J. Comput. Appl. Math., 233 (2010), pp. 1288–1297.

[26] S. Morigi, L. Reichel, F. Sgallari, and F. Zama, An iterative method for linear discrete ill-posed problems with box constraints, J. Comput. Appl. Math., 198 (2007), pp. 505–520.

[27] J. G. Nagy and Z. Strakoš, Enforcing nonnegativity in image reconstruction algorithms, in Mathematical Modeling, Estimation and Imaging, D. C. Wilson et al., eds., Society of Photo-Optical Instrumentation Engineers (SPIE), 4121, The International Society for Optical Engineering, Bellingham, WA, 2000, pp. 182–190.

[28] M. K. Ng, R. H. Chan, W. C. Tang, A fast algorithm for deblurring models with Neumann boundary conditions, SIAM J. Sci. Comp., 21 (1999), pp. 851–866.

[29] J. Nocedal and S. J. Wright, Numerical Optimization, 2nd ed., Springer, New York, 2006.

[30] E. Onunwor and L. Reichel, On the computation of a truncated SVD of a large linear discrete ill-posed problem, Numer. Algorithms, in press.

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

[32] M. Rojas and T. Steihaug, An interior-point trust-region-based method for large-scale non-negative regularization, Inverse Problems, 18 (2002), pp. 1291–1307.

[33] S. Serra–Capizzano, A note on antireflective boundary conditions and fast deblurring models, SIAM J. Sci. Comput., 25 (2004), pp. 1307–1325.

[34] N. Zheng, K. Hayami, and J.-F. Yin, Modulus-type inner outer iteration methods for nonnegative constrained least squares problems, SIAM J. Matrix Anal. Appl., 37 (2016), pp. 1250-1278.