Numerical aspects of the nonstationary modified linearized Bregman algorithm

Alessandro Buccini[a], Yonggi Park[a], Lothar Reichel[a]

[a]*Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA.*

Abstract

The solution of discrete ill-posed problems has been a subject of research for many years. Among the many methods described in the literature, the Bregman algorithm has attracted a great deal attention and been widely investigated. Recently, a nonstationary preconditioned version of this algorithm, referred to as the nonstationary modified linearized Bregman algorithm, was proposed. The aim of this paper is to discuss numerical aspects of this algorithm and to compare computed results with known theoretical properties. We also discuss the effect of several parameters required by the algorithm on the computed solution.

Keywords: Ill-posed problem, Bregman iteration, Preconditioning, Regularization
2010 MSC: 65F22, 65F10, 65R32

1. Introduction

We consider systems of equations of the form

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

where $A \in \mathbb{R}^{m\times n}$ is a large matrix, whose singular values decrease to zero gradually with no significant gap, and the vector $\mathbf{b}^{\varepsilon} \in \mathbb{R}^m$ represents measured error-contaminated data. We will refer to the error $\boldsymbol{\eta}$ in $\mathbf{b}^{\varepsilon}$ as "noise." It is assumed not to be known. Problems of the kind (1) are known as *linear discrete ill-posed problems*. They typically arise from the discretization

Email addresses: abuccini@kent.edu (Alessandro Buccini), ypark7@kent.edu (Yonggi Park), reichel@math.kent.edu (Lothar Reichel)

of linear ill-posed problems, such as Fredholm integral equations of the first kind with a smooth kernel; see, e.g., [1, 2] for discussions on ill-posed and discrete ill-posed problems.

We would like to determine an approximation of the solution $\mathbf{x}_{\text{true}} = A^\dagger(\mathbf{b}^\varepsilon - \boldsymbol{\eta})$ of the unknown noise-free system associated with (1), where $A^\dagger$ denotes the Moore-Penrose pseudoinverse of A. Due to the noise $\boldsymbol{\eta}$ in $\mathbf{b}^\varepsilon$ and the presence of "tiny" positive singular values of A, the vector $A^\dagger\mathbf{b}^\varepsilon$ generally is not a useful approximation of $\mathbf{x}_{\text{true}}$.

Many approaches have been proposed in the literature for computing a useful approximation of $\mathbf{x}_{\text{true}}$; see, e.g., [1, 3, 2]. We will consider the nonstationary modified linearized Bregman (NMLB) algorithm proposed by Huang et al. [4]. This method is a variant of the modified linearized Bregman (MLB) algorithm described by Cai et al. [5] and is designed to yield faster convergence than the latter. The MLB algorithm is an iterative method for solving the minimization problem

$$\arg\min_{\mathbf{x}\in\mathbb{R}^n} \left\{ \mu \|\mathbf{x}\|_1 + \frac{1}{2\delta}\|\mathbf{x}\|_2^2 : A\mathbf{x} = \mathbf{b}^\varepsilon \right\}, \tag{2}$$

where $\mu > 0$ and $0 < \delta < 1/\rho(A^TA)$ are user-supplied constants. Throughout this paper $\rho(M)$ denotes the spectral radius of the square matrix M, and $\|\cdot\|_1$ and $\|\cdot\|_2$ stand for the ℓ_1 and ℓ_2 vector norms, respectively. In the following, we will refer to μ as the *regularization parameter*. The MLB algorithm, which is reviewed in Section 2, is typically applied when the desired solution $\mathbf{x}_{\text{true}}$ is known to be "sparse," i.e., to have many zero entries, and we would like to determine an approximation of $\mathbf{x}_{\text{true}}$ with the same property. Sparse solutions may be desirable when $m \ll n$ or when in some basis, such as a framelet basis, $\mathbf{x}_{\text{true}}$ is known to be sparse.

The purpose of the ℓ_1-norm in the minimization problem (2) is to force the computed solution to be sparse, i.e., to have many vanishing components. The parameter $\mu \geq 0$ determines the amount of shrinkage. Its choice is important for the performance of the solution methods. This is illustrated in Section 4. The ℓ_2-norm in (2) makes the minimization problem strictly convex.

Denote the iterates determined by the MLB algorithm by $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \ldots$. Since we are not interested in the solution $A^\dagger\mathbf{b}^\varepsilon$ of the available system $A\mathbf{x} = \mathbf{b}^\varepsilon$, we terminate the iterations before an accurate approximation of this solution has been determined. Specifically, we terminate the iterations

as soon as an iterate that satisfies the *discrepancy principle* has been found, i.e., as soon as

$$\left\| A\mathbf{x}^k - \mathbf{b}^\varepsilon \right\|_2 \leq \tau\varepsilon, \tag{3}$$

where ε is a bound for the error in $\mathbf{b}^\varepsilon$, which is assumed to be known. Thus,

$$\|\boldsymbol{\eta}\|_2 \leq \varepsilon. \tag{4}$$

The parameter τ in (3) is a user-supplied constant larger than one and is independent of ε; see, e.g., [1] for details on the discrepancy principle.

In many applications the desired vector $\mathbf{x}_{\text{true}}$ represents a signal that is sparse in a suitable basis, such as in the framelet domain. Tight frames have been used in many applications, see, e.g., [6, 5, 7], because many signals of interest have a sparse representation in the framelet domain. We will provide details about the transformation of (1) to the framelet domain in Section 2.

This work is structured as follows: in Section 2 we recall the main results on the convergence of the NMLB algorithm, and Section 3 discusses the choice of the parameter δ in (2) in the situation when $\mu = 0$. Section 4 is concerned with the choice of several parameters, including δ and the regularization parameter μ, required by the NMLB algorithm. Numerical examples illustrate the performance of the algorithm for different choices of these parameters. Concluding remarks can be found in Section 5.

2. The nonstationary modified linearized Bregman algorithm

This section collects the main results in [4]. We first derive the NMLB algorithm from the linearized Bregman (LB) algorithm. Then we summarize its theoretical properties and, finally, describe how to combine the LB algorithm with tight frames.

Let $A \in \mathbb{R}^{m\times n}$, with $m \leq n$, be a surjective matrix, i.e., all its singular values are positive. We will comment on below how the situation when A has singular values that are numerically vanishing can be handled.

The aim of linearized Bregman iteration is to find an approximation of the solution of (1) of minimal ℓ_1 norm, i.e., one seeks to solve

$$\min_{\mathbf{s}\in\mathbb{R}^n} \left\{ \|\mathbf{s}\|_1 : A\mathbf{s} = \mathbf{b}^\varepsilon \right\}. \tag{5}$$

Note that this minimization problem is not guaranteed to have a unique solution. The iterations of the LB algorithm can be written as

$$\begin{cases} \mathbf{z}^{k+1} &= \mathbf{z}^k + A^T(\mathbf{b}^\varepsilon - A\mathbf{s}^k), \\ \mathbf{s}^{k+1} &= \delta\mathbb{S}_\mu(\mathbf{z}^{k+1}), \end{cases} \tag{6}$$

for $k = 0, 1, \ldots$ with $\mathbf{s}^0 = \mathbf{z}^0 = \mathbf{0}$. Here $\mathbb{S}_\mu(\mathbf{x})$ denotes the soft-thresholding operator,

$$\mathbb{S}_\mu(\mathbf{x}) := \text{sign}(\mathbf{x})(|\mathbf{x}| - \mu)_+,$$

where all the operations are element-wise and $(x)_+ := \max\{0, x\}$ denotes the non-negative part of $x \in \mathbb{R}$.

The iterations (6) can be easily implemented. They require only matrix-vector multiplications, vector additions, scalar multiplication of vectors, and soft-thresholding. Applications of the LB algorithm include basis pursuit problems, which arise in compressed sensing; see [8, 9]. In this, as well as in many other applications of the LB algorithm, the matrix A is sparse, and matrix-vector products can be evaluated cheaply. The algorithm is designed for the approximate solution of problems (5) for which the desired solution, $\mathbf{x}_{\text{true}}$, is sparse. It is shown in [10, 11] that the limit of the sequence $\{\mathbf{s}^k\}_k$ generated by (6) converges to a solution of (5).

When the matrix A is ill-conditioned, i.e., when the ratio of the largest to smallest singular values of A is large, convergence of the sequence $\mathbf{s}^1, \mathbf{s}^2, \ldots$ generated by the LB algorithm may be very slow. Therefore, it may be necessary to carry out many iterations (6) until an accurate approximation of $\mathbf{x}_{\text{true}}$ has been found. To alleviate this difficulty, Cai et al. [5] proposed the use of a preconditioner $P \in \mathbb{R}^{m\times m}$ in (6). This yields the MLB algorithm,

$$\begin{cases} \mathbf{z}^{k+1} = \mathbf{z}^k + A^T P(\mathbf{b}^\varepsilon - A\mathbf{s}^k), \\ \mathbf{s}^{k+1} = \delta\mathbb{S}_\mu(\mathbf{z}^{k+1}), \end{cases} \tag{7}$$

for $k = 0, 1, \ldots$, with $\mathbf{s}^0 = \mathbf{z}^0 = \mathbf{0}$.

Theorem 1 ([5]). *Assume that $A \in \mathbb{R}^{m\times n}$, $m \leq n$, is surjective, let $P = (AA^T)^{-1}$, and let $0 < \delta < 1$ be a fixed constant. Then the sequence $\mathbf{s}^1, \mathbf{s}^2, \ldots$, generated by the MLB algorithm (7) converges to a solution of (2) for any $\mu > 0$. Furthermore, as $\mu \to \infty$, the limit of the sequence $\mathbf{s}^1, \mathbf{s}^2, \ldots$ converges to the solution of (5) that is closest to the minimal ℓ_2-norm solution among all solutions of (5).*

The main difficulty with the iterations described by the above theorem is that when the matrix A is ill-conditioned, the preconditioner $P = (AA^T)^{-1}$ may be of very large norm. This may cause numerical difficulties. Moreover, in some applications of interest, the matrix A is rank deficient and then this preconditioner is not defined.

To avoid these difficulties, Cai et al. [5] generalized Theorem 1 to allow the preconditioner P to be an arbitrary symmetric positive definite matrix. This extension is described in the following theorem. We need the following definition. Let the matrix $M \in \mathbb{R}^{m\times m}$ be symmetric positive definite. Then $\|\cdot\|_M$ denotes the vector norm induced by the matrix M, i.e.,

$$\|\mathbf{v}\|_M = (\mathbf{v}^T M\mathbf{v})^{1/2}, \qquad \mathbf{v} \in \mathbb{R}^m.$$

Theorem 2 ([5]). *Let $P \in \mathbb{R}^{m\times m}$ be a symmetric positive definite matrix and assume that $0 < \delta < 1/\rho(A^T PA)$. Then the sequence $\mathbf{s}^1, \mathbf{s}^2, \ldots$ generated by the iterations (7) converges to the unique solution of*

$$\arg\min_{\mathbf{x}} \left\{ \mu \|\mathbf{x}\|_1 + \frac{1}{2\delta}\|\mathbf{x}\|_2^2 : \mathbf{x} = \arg\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}^\varepsilon\|_P \right\}.$$

Furthermore, as $\mu \to \infty$, the limit of the sequence $\mathbf{s}^1, \mathbf{s}^2, \ldots$ converges to the solution of

$$\arg\min_{\mathbf{x}} \left\{ \mu \|\mathbf{x}\|_1 : \mathbf{x} = \arg\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}^\varepsilon\|_P \right\} \tag{8}$$

of minimal ℓ_2-norm among all solutions of (8).

Inspired by Tikhonov regularization, Cai et al. [5] considered the application of preconditioners of the form

$$P = (AA^T + \alpha I)^{-1}, \tag{9}$$

where $\alpha > 0$ is a fixed user-specified parameter. With this preconditioner the iterations (7) can be written as

$$\begin{cases} \mathbf{z}^{k+1} = \mathbf{z}^k + A^T(AA^T + \alpha I)^{-1}(\mathbf{b}^\varepsilon - A\mathbf{s}^k), \\ \mathbf{s}^{k+1} = \delta \mathbb{S}_\mu(\mathbf{z}^{k+1}), \end{cases} \tag{10}$$

for $k = 0, 1, \ldots$, where $\mathbf{s}^0 = \mathbf{z}^0 = \mathbf{0}$. Theorem 2 yields that the iterates $\mathbf{s}^1, \mathbf{s}^2, \ldots$ generated by (10) converge to the unique solution of

$$\arg\min_{\mathbf{x}} \left\{ \mu \|\mathbf{x}\|_1 + \frac{1}{2\delta}\|\mathbf{x}\|_2^2 : \mathbf{x} = \arg\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}^\varepsilon\|_{(AA^T+\alpha I)^{-1}} \right\}.$$

Huang et al. [4] observed that the iterates (10) can be sensitive to the choice of $\alpha > 0$, i.e., the quality of the computed solution may deteriorate

significantly when α is chosen slightly off an optimal value. Determining an accurate estimate of the optimal α-value can be difficult, and is not possible in most applications. To circumvent this difficulty, Huang et al. [4] replaced the parameter α in (9) by a sequence of parameter values $\alpha_0, \alpha_1, \ldots$, similarly to a strategy suggested in [12]. In other words, the parameter α in (10) is changed in each iteration. This defines a nonstationary preconditioning approach. Since $\rho(A^T P A) < 1$ for all $\alpha > 0$, Huang et al. [4] let $\delta = 1$ in (10). Summarizing, the iterations become

$$\left\{ \begin{array}{l} \mathbf{z}^{k+1} = \mathbf{z}^k + A^T (AA^T + \alpha_k I)^{-1} (\mathbf{b}^\varepsilon - A\mathbf{s}^k), \\ \mathbf{s}^{k+1} = \mathbb{S}_\mu(\mathbf{z}^{k+1}), \end{array} \right. \tag{11}$$

for $k = 0, 1, \ldots$, where $\mathbf{s}^0 = \mathbf{z}^0 = \mathbf{0}$. This scheme is in [4] referred to as the NMLB algorithm. The following convergence results are shown by Huang et al. [4].

Theorem 3. *Assume that $\alpha_k \to \bar{\alpha}$ as $k \to \infty$ for some $0 < \bar{\alpha} < \infty$. Let $\mathbf{s}^1, \mathbf{s}^2, \ldots$ denote the iterates determined by (11). Then, as k increases, the $\mathbf{s}^k$ converge to the unique solution of*

$$\arg\min_{\mathbf{s}} \left\{ \mu \left\| \mathbf{s} \right\|_1 + \frac{1}{2} \left\| \mathbf{s} \right\|_2^2 : \mathbf{s} = \arg\min_{\mathbf{s}} \left\| A\mathbf{s} - \mathbf{b}^\varepsilon \right\|_{(AA^T + \bar{\alpha} I)^{-1}} \right\}. \tag{12}$$

Furthermore, as $\mu \to \infty$, the limit of the iterates $\mathbf{s}_k$ as $k \to \infty$ is the solution of (8), with P given by (9) and α replaced by $\bar{\alpha}$, of minimal ℓ_2-norm.

The parameter $\bar{\alpha}$ in the above theorem has to be positive for theoretical purposes. It is "tiny" and a lower bound for the α_k in the computed examples of Section 4. In these examples the α_k are a decreasing function of k, and the iterations are terminated well before α_k is close to $\bar{\alpha}$.

Huang et al. [4] illustrate that the iterates determined by the NMLB algorithm with a suitable decreasing parameter sequence $\alpha_0, \alpha_1, \ldots$ are less sensitive to the choice of the parameters α_k than the iterates generated by the MLB algorithm (10) are to the choice of the single parameter α. Moreover, Huang et al. [4] found that the NMLB algorithm may determine more accurate approximations of $\mathbf{x}_{\text{true}}$ than the MLB algorithm.

The application of the preconditioner P defined by (9) is attractive when the matrix AA^T is not too large. Then we can explicitly form this matrix, compute the Choleski factorization of $AA^T + \alpha I$, and use the latter when

evaluating matrix-vector products with P. When AA^T is large, the preconditioner should be chosen so that it approximates $(AA^T + \alpha I)^{-1}$ in a suitable manner. For instance, in image restoration applications, the matrix A often is a large square block-Toeplitz-Toeplitz-block matrix. It may then be attractive to approximate the preconditioner (9) by a matrix of the form $(CC^T + \alpha I)^{-1}$, where C is a block-circulant-circulant-block matrix that approximates A. Techniques for determining such preconditioners are described in, e.g., [13, 14, 15]. A recent discussion on how to determine approximations of the preconditioner (9) and numerical illustrations are provided by Cai et al. [16]; see Section 5 for further comments.

In many applications the desired solution, $\mathbf{x}_{\text{true}}$, is not sparse in the canonical basis for $\mathbb{R}^n$, but it is sparse in a framelet basis. Framelets are frames with local support. We will review how to combine tight frames and the NMLB algorithm. Applications of tight frames are described, e.g., in [5, 4]. Computed examples with tight frames are presented in Section 4. First, we define tight frames:

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

$$\|\mathbf{x}\|_2^2 = \sum_{j=1}^{r} (\mathbf{w}_j^T \mathbf{x})^2, \tag{13}$$

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

Equation (13) is equivalent to the perfect reconstruction formula

$$\mathbf{x} = W^T \mathbf{y}, \qquad \mathbf{y} = W\mathbf{x}.$$

In other words

$$W \text{ is a tight frame} \quad \Leftrightarrow \quad W^T W = I.$$

Note that, in general, $WW^T \neq I$, unless $r = n$ and the frames are orthogonal.

One of the interesting properties of tight frames is that many signals that arise in applications have a sparse representation in the framelet domain. Since the NMLB algorithm seeks to compute a sparse solution, we would like to modify (1) so that the unknowns are framelet coefficients. Let W denote

an analysis operator. Inserting $W^TW = I$ into (1) and ignoring $\boldsymbol{\eta}$ in the right-hand side yields the system of equations

$$AW^TW\mathbf{x} = \mathbf{b}^\varepsilon.$$

Let $K = AW^T$ and $\mathbf{y} = W\mathbf{x}$. Then the above equation can be expressed as

$$K\mathbf{y} = \mathbf{b}^\varepsilon. \tag{14}$$

The entries of the unknown vector $\mathbf{y}$ are framelet coefficients of the solution. In many applications the vector $\mathbf{y}$ is very sparse. Transformation to the framelet domain allows us to take advantage of the sparsity of solutions computed by the NMLB algorithm, even when the desired solution $\mathbf{x}_{\text{true}}$ is not sparse in the canonical basis for $\mathbb{R}^n$. Thus, we first apply the NMLB algorithm to (14) to determine the framelet coefficient vector $\mathbf{y}$, and then compute an approximation of $\mathbf{x}_{\text{true}}$ by applying the synthesis operator W^T to $\mathbf{y}$. Note that, generally, the matrix W is very sparse and, therefore, the evaluation of matrix-vector products with W and W^T is very cheap. It follows that the computational cost of the transformation of (1) to the framelet domain and back typically is negligible. Note that the preconditioner P is not affected by the transformation to the framelet domain; we have

$$\begin{aligned}(KK^T + \alpha_k I)^{-1} &= (AW^T(AW^T)^T + \alpha_k I)^{-1} \\ &= (AA^T + \alpha_k I)^{-1}.\end{aligned}$$

We turn to the stopping criterion for the NMLB algorithm. From Theorem 3, we know that the limit point of the iterates determined by the NMLB algorithm is a solution of (12). However, when the vector $\mathbf{b}^\varepsilon$ is contaminated by noise and the matrix A is very ill-conditioned (i.e., the ratio of the largest and smallest singular values of A is very large), solutions of (12) are not meaningful approximations of $\mathbf{x}_{\text{true}}$. A fairly accurate approximation of $\mathbf{x}_{\text{true}}$ often can be determined by terminating the iterations with the NMLB algorithm before convergence is achieved. Huang et al. [4] employed the discrepancy principle to determine when to terminate the iterations (11). Assume that a fairly sharp bound (4) for the norm of the error in the data vector $\mathbf{b}^\varepsilon$ is available. We then terminate the iterations with the NMLB algorithm when the discrepancy principle (3) is satisfied, or equivalently, as soon as an iterate $\mathbf{s}^k$ satisfies

$$\left\| AW^T\mathbf{s}^k - \mathbf{b}^\varepsilon \right\|_2 \leq \tau\varepsilon. \tag{15}$$

Algorithm 1 summarizes the computations of the NMLB method applied to (14).

Algorithm 1 The NMLB Algorithm

Input: $A \in \mathbb{R}^{m\times n}$, $\mathbf{b}^\varepsilon \in \mathbb{R}^m$, $\{\alpha_k\}_k$ such that $\alpha_k \to \bar{\alpha}$ with $0 < \bar{\alpha} < \infty$, $\mu > 0$, $\tau > 1$, and $W \in \mathbb{R}^{r\times n}$ an analysis operator

Output: regularized solution $\mathbf{x}^*$

1: $\mathbf{z}^0 = \mathbf{0}, \mathbf{s}^0 = \mathbf{0}, k = 0$
2: **repeat**
3: $\quad k = k + 1$
4: $\quad \mathbf{z}^k = \mathbf{z}^{k-1} + WA^T(AA^T + \alpha_n I)^{-1}(\mathbf{b}^\varepsilon - AW^T\mathbf{s}^{k-1})$
5: $\quad \mathbf{s}^k = \mathbb{S}_\mu(\mathbf{z}^k)$
6: **until** $\left\|AW^T\mathbf{s}^k - \mathbf{b}^\varepsilon\right\|_2 \le \tau\varepsilon$
7: $\mathbf{x}^* = W^T\mathbf{s}^k$

3. Landweber iteration

This section seeks to shed light on how the parameter $\delta > 0$ in (6) and (10) affects the rate of convergence of the iterates. To simplify the analysis, we set $\mu = 0$. Then the soft-thresholding operator $\mathbb{S}_\mu$ becomes the identity operator, and the iterations (6) and (10) turn into Landweber iteration

$$\mathbf{s}^{k+1} = \mathbf{s}^k + \delta A^T(\mathbf{b}^\varepsilon - A\mathbf{s}^k), \qquad k = 0, 1, \dots , \tag{16}$$

and preconditioned Landweber iteration

$$\mathbf{s}^{k+1} = \mathbf{s}^k + \delta A^T(AA^T + \alpha I)^{-1}(\mathbf{b}^\varepsilon - A\mathbf{s}^k), \qquad k = 0, 1, \dots , \tag{17}$$

respectively. The parameter α is assumed to be positive. Analyses of these iterations related to our analysis below can be found in, e.g., Elfving et al. [17] and Engl et al. [1].

Let the matrix $M \in \mathbb{R}^{m\times m}$ be symmetric. Then its eigenvalues are real and we may choose the eigenvectors to be orthogonal. We will refer to the eigenvectors associated with the largest eigenvalues as the largest eigenvectors.

Proposition 1. *Let* $\mathbf{s}^0 = \mathbf{0}$ *and assume that* $0 < \delta < 2/\rho(A^TA)$. *Then the iterates* (16) *converge to the solution* $\mathbf{s}^*$ *of minimal* ℓ_2*-norm of the least-squares problem*

$$\min_{\mathbf{s}\in\mathbb{R}^n} \|As - \mathbf{b}^\varepsilon\|_2. \tag{18}$$

Let $\{\mathbf{u}_j\}_{j=1}^n$ *denote the set of orthonormal eigenvectors of* A^TA *and express the difference* $\mathbf{s}^k - \mathbf{s}^*$ *in terms of these eigenvectors,*

$$\mathbf{s}^k - \mathbf{s}^* = \sum_{j=1}^{n} \gamma_j^k \mathbf{u}_j, \qquad \gamma_j^k \in \mathbb{R}. \tag{19}$$

Then the choice $\delta = 1/\rho(A^TA)$ *makes nonvanishing coefficients* γ_j^k *associated with the largest eigenvectors* $\mathbf{u}_j$ *converge to zero faster as* k *increases than nonvanishing coefficients* γ_j^k *associated with other eigenvectors.*

Proof. The convergence of the iterates (16) when $0 < \delta < 2/\rho(A^TA)$ is well known. It follows by substituting the spectral factorization of A^TA into (16). The fact that all iterates live in the range of A^T makes them orthogonal to the null space of A. Therefore, they converge to the solution of minimal Euclidean norm; see, e.g., [17, 1] for details. The rate of convergence of the coefficients γ_j^k as k increases follows by studying how the components in the right-hand side of (19) are damped during the iterations. □

Proposition 2. *Let* $\mathbf{s}^0 = \mathbf{0}$, $\alpha > 0$, *and assume that* $0 < \delta < 2(1 + \alpha/\rho(A^TA))$. *Then the iterates* (17) *converge to the solution* $\mathbf{s}^*$ *of the minimization problem* (18) *of minimal* ℓ_2*-norm. Consider the differences* (19) *with the iterates* $\mathbf{s}^k$ *defined by* (17). *Then the choice*

$$\delta = 1 + \alpha/\rho(A^TA) \tag{20}$$

makes nonvanishing coefficients γ_j^k *in* (19) *associated with the largest eigenvectors* $\mathbf{u}_j$ *converge to zero faster as* k *increases than nonvanishing coefficients* γ_j^k *associated with other eigenvectors.*

Proof. By (17), the iterates $\mathbf{s}^k$, $k = 1, 2, \ldots$, live in the range of A^T. Therefore, if they converge, then they converge to the solution of (18) of minimal Euclidean norm. The convergence of the sequence $\mathbf{s}^k$, $k = 1, 2, \ldots$, can be established similarly as in the proof of Proposition 1, i.e., by investigating how the error $\mathbf{e}_k = \mathbf{s}^k - \mathbf{s}^*$ is damped during the iterations. We have

$$\mathbf{e}^{k+1} = \mathbf{e}^k - \delta A^T(AA^T + \alpha I)^{-1}A\mathbf{e}^k. \tag{21}$$

Using the identity

$$A^T(AA^T + \alpha I)^{-1}A = (A^TA + \alpha I)^{-1}A^TA$$

and the spectral factorization

$$A^T A = U\Lambda U^T, \quad \Lambda = \mathrm{diag}[\lambda_1, \lambda_2, \ldots, \lambda_n], \quad U = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n],$$

we obtain from (21) that

$$\widetilde{\mathbf{e}}_j^{k+1} = (I - \delta(\Lambda + \alpha I)^{-1}\Lambda)\widetilde{\mathbf{e}}^k, \qquad \widetilde{\mathbf{e}}^j := U^T \mathbf{e}^j. \tag{22}$$

The observation that $t \to t/(\alpha + t)$ is an increasing function of $t \geq 0$ shows convergence of the errors $\mathbf{e}^k$ to zero as k increases when $0 < \delta < 2(1 + \alpha/\rho(A^T A))$. The rate of convergence of the coefficients γ_j^k in the expansion (21) to zero as k increases follows by studying the components of errors in (22). □

We remark that it is easy to show that the vector $\mathbf{s}^*$ of Proposition 2 also is the solution of minimal Euclidean norm of

$$\min_{\mathbf{s}\in\mathbb{R}^n} \|A\mathbf{s} - \mathbf{b}^\varepsilon\|_{(AA^T+\alpha I)^{-1}},$$

which is the expression in (12).

We are interested in damping the largest eigenvectors in the difference $\mathbf{s}^k - \mathbf{s}^*$ of Proposition 2, because these eigenvectors are the most important components of $\mathbf{x}_{\mathrm{true}}$; the smallest eigenvectors model noise and generally should not be included in the computed approximation of $\mathbf{x}_{\mathrm{true}}$. Proposition 2 suggests that when the parameter α is not "tiny" and an estimate of $\rho(A^T A)$ that is not "huge" is available, a value of δ based on an estimate of the right-hand side of (20) should be used, because this may result in faster convergence of the iterates than $\delta = 1$.

Thus, Proposition 2 indicates that δ should be chosen larger than unity for the iterations (17) to achieve a higher rate of convergence. While the proposition does not apply to the iterates (11) with $\mu > 0$, we, nevertheless, would expect the latter iterates to converge faster for $\delta > 1$ than for $\delta = 1$, at least for some problems and when μ is not too large. Computed examples reported in the following section illustrate that this indeed is the case.

4. Numerical aspects of the NMLB algorithm

This section discusses the performance of the NMLB algorithm when applied to the solution of a few linear discrete ill-posed problems from Regularization Tools by Hansen [18]. In particular, we are interested in studying the influence of user-specified parameters on the computed solution.

Following Huang et al. [4], we chose the sequence of parameters

$$\alpha_k = \alpha_0 q^k + 10^{-15} \tag{23}$$

for the preconditioners P in (11), where $\alpha_0 > 0$ and $0 < q < 1$. Thus, $\alpha_k \to \bar{\alpha} = 10^{-15}$ as $k \to \infty$. We set $\alpha_0 = 0.5$ in all experiments. This leaves us with the determination of the parameters μ, q, and δ.

We use the discrepancy principle as a stopping criterion with $\tau = 1.01$ in (3) and (15). The maximum number of allowed iterations is set to 7000. We will investigate the number of iterations required to satisfy the discrepancy principle as a function of μ, q, and δ. Also the relative restoration error (RRE), defined by

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

is studied as a function of these parameters. Finally, we will consider the norm of the residual at the final iteration, i.e., $\left\|A\mathbf{x}^{k^*} - \mathbf{b}^{\varepsilon}\right\|_2$, where k^* denotes the number of iterations carried out by the NMLB algorithm.

We use the same tight frame system as Huang et al. [4], i.e., the system of linear B-splines. This system is formed by a low-pass filter $W_0 \in \mathbb{R}^{n\times n}$ and two high-pass filters $W_1 \in \mathbb{R}^{n\times n}$ and $W_2 \in \mathbb{R}^{n\times n}$, whose corresponding masks are

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

The analysis operator W is derived from these masks and by imposing reflexive boundary conditions. These boundary conditions are such that $W^TW = I$. We obtain

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

and

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

Thus,

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

The matrix W is very sparse. Therefore, the evaluation of matrix-vector products with W and W^T is inexpensive.

Since we would like to investigate the performance of the NMLB algorithm for many different choices of μ, q, and δ, we choose the dimensions m and n in all examples to be fairly small; specifically, we set $n = m = 200$. The error $\boldsymbol{\eta}$ in $\mathbf{b}^{\varepsilon}$ is modeled by white Gaussian noise and we refer to the ratio

$$\sigma = \frac{\|\boldsymbol{\eta}\|_2}{\|A\mathbf{x}_{\text{true}}\|_2}$$

as the noise level. We use the test problems baart, phillips, and heat from Regularization Tools [18]. They are discretizations of Fredholm integral of the first kind. In all examples, the desired solution $\mathbf{x}_{\text{true}}$ has a sparse representation in terms of the framelet basis used.

All computations are carried out using MATLAB 8.6 (R2015a) on a laptop computer with an Intel(R) Core(TM) i5-3337U CPU @ 1.80 GHz and 16 GB of memory. The floating-point precision is 10^{-16}. Following Huang et al. [4], we let $\delta = 1$ in subsections 4.1-4.3. The influence of the value of δ on the convergence rate is illustrated in subsection 4.4.

4.1. The number of iterations

We first discuss the number of iterations required by the NMLB algorithm to reach convergence (i.e., to satisfy the discrepancy principle). This is of particular importance since, if too many iterations are carried out, then the algorithm becomes unstable. Consider the sequence α_k defined by (23). Let $\sigma_{\max}$ and $\sigma_{\min}$ denote the largest and smallest singular values, respectively, of A. Thus, $\sigma_{\max}^2 = \rho(A^T A)$. Then the condition number of the preconditioner $P = (AA^T + \alpha_k I)^{-1}$ is given by

$$\kappa_2(P) = \frac{\sigma_{\max}^2 + \alpha_k}{\sigma_{\min}^2 + \alpha_k}.$$

We are interested in the situation when A is severely ill-conditioned, i.e., when $\sigma_{\max} \gg \sigma_{\min}$. Assume for the moment that A is scaled so that $\sigma_{\max} = 1$.

Then we obtain

$$\kappa_2(AA^T + \alpha_k I) \approx \frac{1+\alpha_k}{\alpha_k} \to \frac{1+10^{-15}}{10^{-15}} \approx 10^{15} \text{ as } k \to \infty.$$

Thus, if many iteration are carried out, i.e., if k is large, then the matrix $AA^T + \alpha_k I$ becomes very ill-conditioned. In this situation, the evaluation of $P\mathbf{u}$ for a vector $\mathbf{u} \in \mathbb{R}^m$ may suffer from severely propagated error stemming from round-off errors that are introduced during the computations.

Figure 1 displays the number of iterations required for the NMBL algorithm to satisfy the discrepancy principle. Visual inspection of the graphs shows the number of iterations to increase with μ. Moreover, we can observe that, the larger q is, the more iterations are needed. The latter is to be expected, since for large q-values and modest k the preconditioner $(AA^T + \alpha_k I)^{-1}$ is a poor approximation of the matrix $(AA^T)^\dagger$.

These observations show that μ should not be chosen too large, because a large μ-value may lead to that the NMBL algorithm requires a large number of iterations to satisfy the discrepancy principle. This makes the algorithm expensive and, moreover, unstable. The poor performance of the NMBL algorithm is evident in Figure 1. We can observe that, starting from a certain μ-value, the number of iterations sharply increases as μ increases. We remark that this behavior is less evident for the phillips test problem, because the matrix A of this problem is less ill-conditioned than the matrices of the other problems.

4.2. The residual norm

We turn to the behavior of the norm of the residual at the final iteration. This analysis tells us when, in practical applications, the discrepancy principle is able to effectively terminate the iterations. In theory, it follows from Theorem 3 that the discrepancy principle should effectively stop the algorithm after finitely many iterations, independently of the choice of μ and q. However, if the norm of the residual does not decrease fast enough, then the NMLB algorithm fails to converge due to numerical instability.

Figure 2 shows the norm of the residual at the last iteration for different values of μ and q. We observe that for small values of μ, the norm of the residual behaves as expected, i.e., it is constant and equal to $\tau\varepsilon$. This implies that the iterations were terminated by the discrepancy principle. However, if μ is too large, then we can see that, especially for small noise levels, the discrepancy principle is not able to stop the iterations. This is due to the

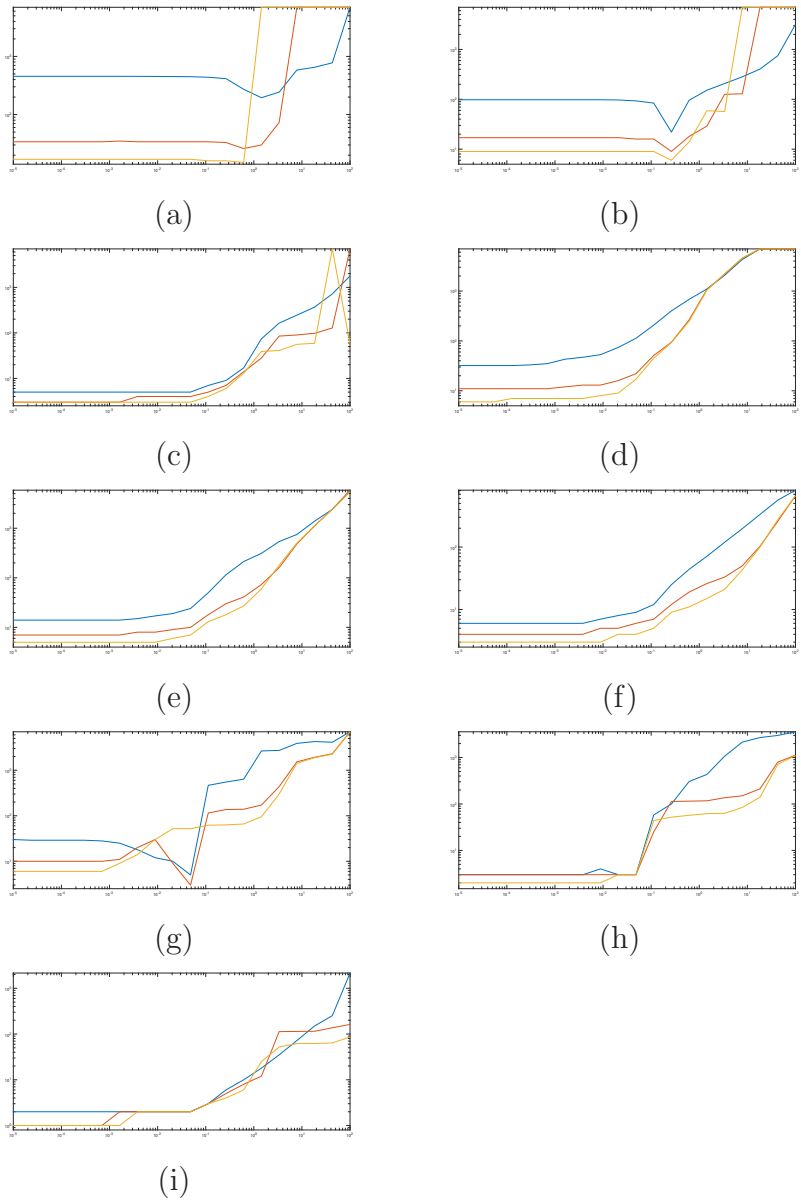

Figure 1: Number of iterations required to reach convergence for different choices of μ and q. The different graphs represent the number of iterations versus μ. The yellow graph is for $q = 0.6$, the red graph for $q = 0.8$, and the blue graph for $q = 0.9$. Panels (a)-(c) report results for the baart test problem, panels (d)-(f) for the heat test problem, and panels (g)-(i) for the phillips test problem. The panels (a), (d), and (g) show results for the noise level $\sigma = 10^{-3}$, the panels (b), (e), and (h) for $\sigma = 10^{-2}$, and the panels (c), (f), and (i) for $\sigma = 10^{-1}$.

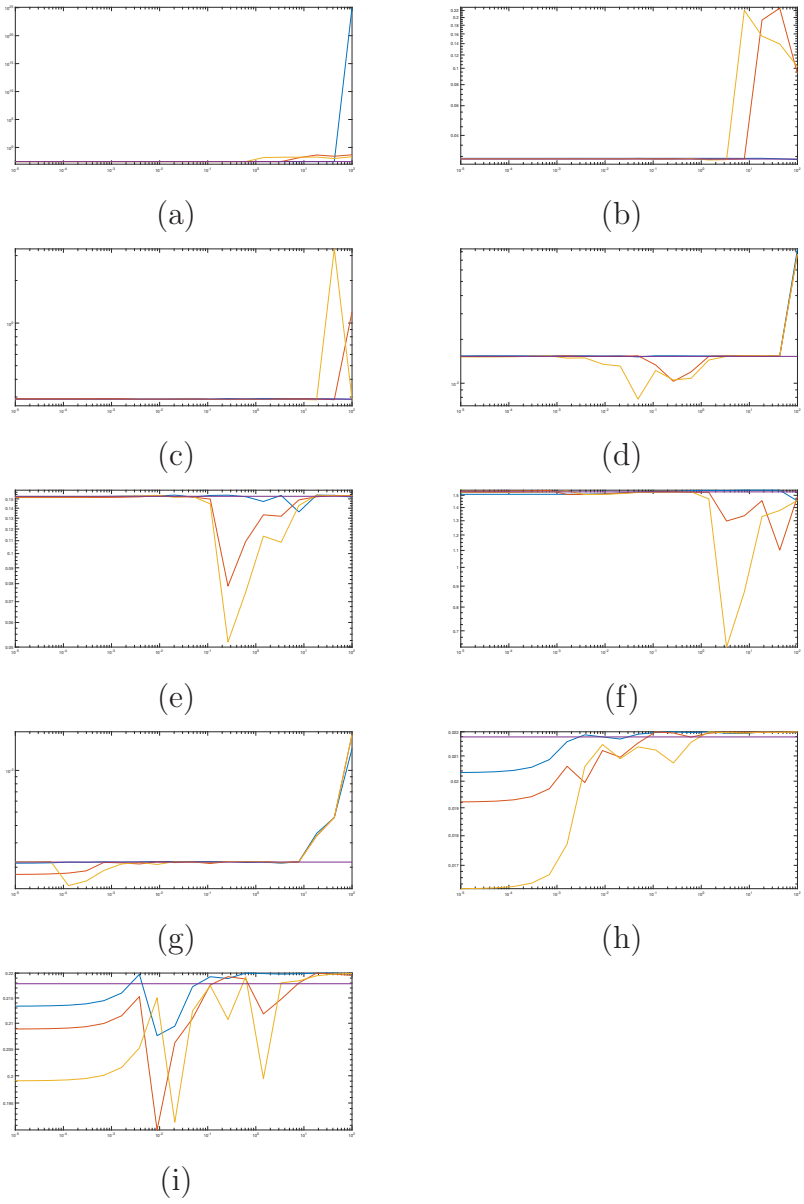

Figure 2: Norm of the residual at the final iteration for different choices of μ and q. The different graphs display the norm of the residual at the final iteration versus μ. The yellow graph is for $q = 0.6$, the red graph for $q = 0.8$, and the blue graph for $q = 0.9$. Panels (a)-(c) report results for the baart test problem, panels (d)-(f) for the heat test problem, and panels (g)-(i) for the phillips test problem. The panels (a), (d), and (g) show results for the noise level $\sigma = 10^{-3}$, the panels (b), (e), and (h) for $\sigma = 10^{-2}$, and the panels (c), (f), and (i) for $\sigma = 10^{-1}$.

large number of iterations performed and the consequent ill-conditioning of the preconditioner with small $\alpha_k > 0$. Severely propagated round-off errors prevent the NMLB algorithm from terminating.

4.3. The relative restoration error

We would like to analyze the behavior of the RRE as functions of the parameters μ and q. Figure 3 shows the RRE obtained for different choices of μ and q. We observe that for small values of μ, the RRE is almost constant. In fact, for μ small, the NMLB algorithm essentially becomes a nonstationary preconditioned Landweber iteration method; see Section 3. As μ increases, the RRE starts to decrease until a minimum is reached. This behavior is particularly evident for the test problem heat. When μ becomes large, the error increases with μ and this increase can be very sharp. This effect is due to the fact that the NMLB algorithm is unstable for large values of μ. Figure 4 displays magnifications of the graphs of Figure 3 around the value μ that gives the smallest RRE.

4.4. The choice of δ

Let $\mathbf{s}^0 = \mathbf{z}^0 = \mathbf{0}$. This subsection illustrates how the iterates $\mathbf{s}^1, \mathbf{s}^2, \dots$, defined by

$$\begin{cases} \mathbf{z}^{k+1} = \mathbf{z}^k + A^T(AA^T + \alpha_k I)^{-1}(\mathbf{b}^\varepsilon - A\mathbf{s}^k), \\ \mathbf{s}^{k+1} = \delta \mathbb{S}_\mu(\mathbf{z}^{k+1}), \end{cases} \tag{24}$$

for $k = 0, 1, \dots$, depend on the choice of δ. Huang et al. [4] let $\delta = 1$. This choice secures convergence. However, the analysis in Section 3 suggests that a larger value of δ may give faster convergence. The computations reported in this subsection show that this indeed may be the case, i.e., the iterates (24) for $\delta > 1$ display faster convergence than the iterates (11). We will illustrate this with a few representative computations. In all computations for this subsection, the noise level is $\sigma = 1 \cdot 10^{-2}$.

We first consider the baart test problem. For $\mu = 6.9 \cdot 10^{-4}$ and $\mu = 4.8 \cdot 10^{-2}$, the iterations (24) for $\delta = 1$ and $\delta = 1.5$ are terminated by the discrepancy principle for all q-values reported in Table 1; the errors in the computed approximate solutions are essentially independent of q. Table 1 shows the number of iterations required to satisfy the discrepancy principle for several q-values and two δ-values. As expected, the number of iterations decreases as q is decreased for both values of δ. We also see that for fixed μ and q, the number of iterations required is smaller for the larger value of δ.

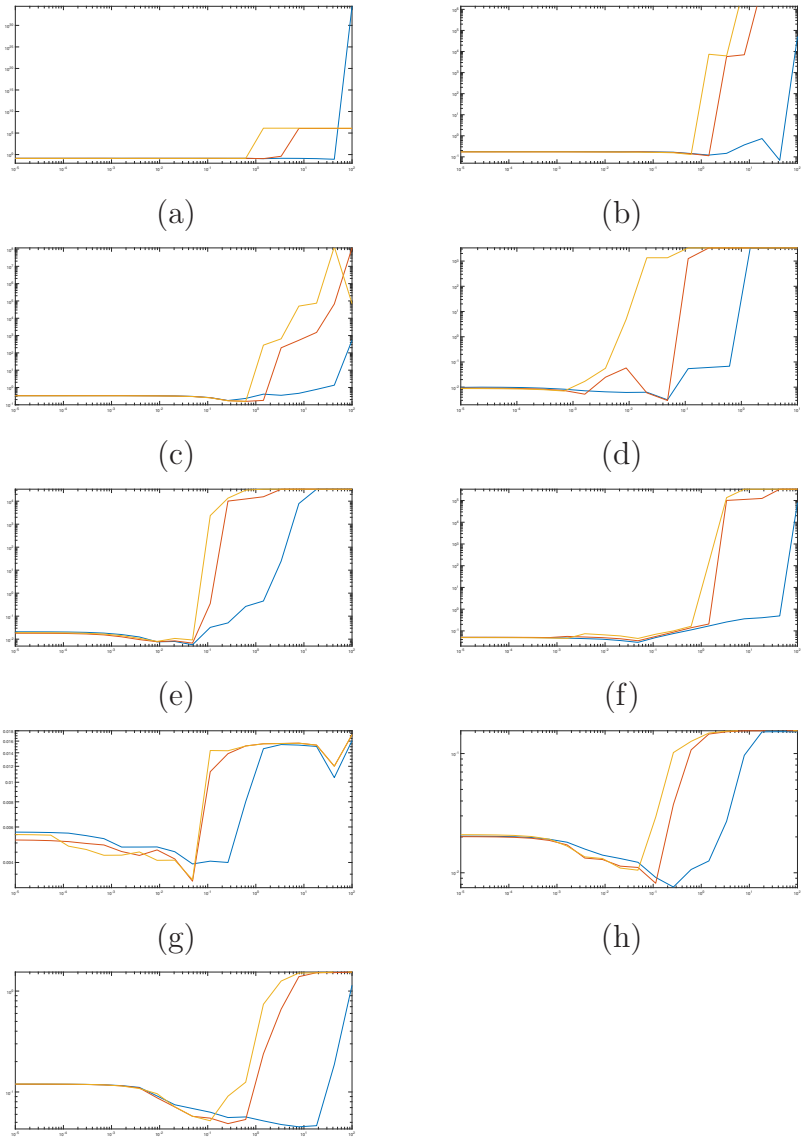

Figure 3: RRE obtained for several choices of μ and q. The different graphs display the RRE versus μ. The yellow graph is for $q = 0.6$, the red graph for $q = 0.8$, and the blue graph for $q = 0.9$. Panels (a)-(c) report results for the baart test problem, panels (d)-(f) for the heat test problem, and panels (g)-(i) for the phillips test problem. The panels (a), (d), and (g) show results for the noise level $\sigma = 10^{-3}$, the panels (b), (e), and (h) for $\sigma = 10^{-2}$, and the panels (c), (f), and (i) for $\sigma = 10^{-1}$.

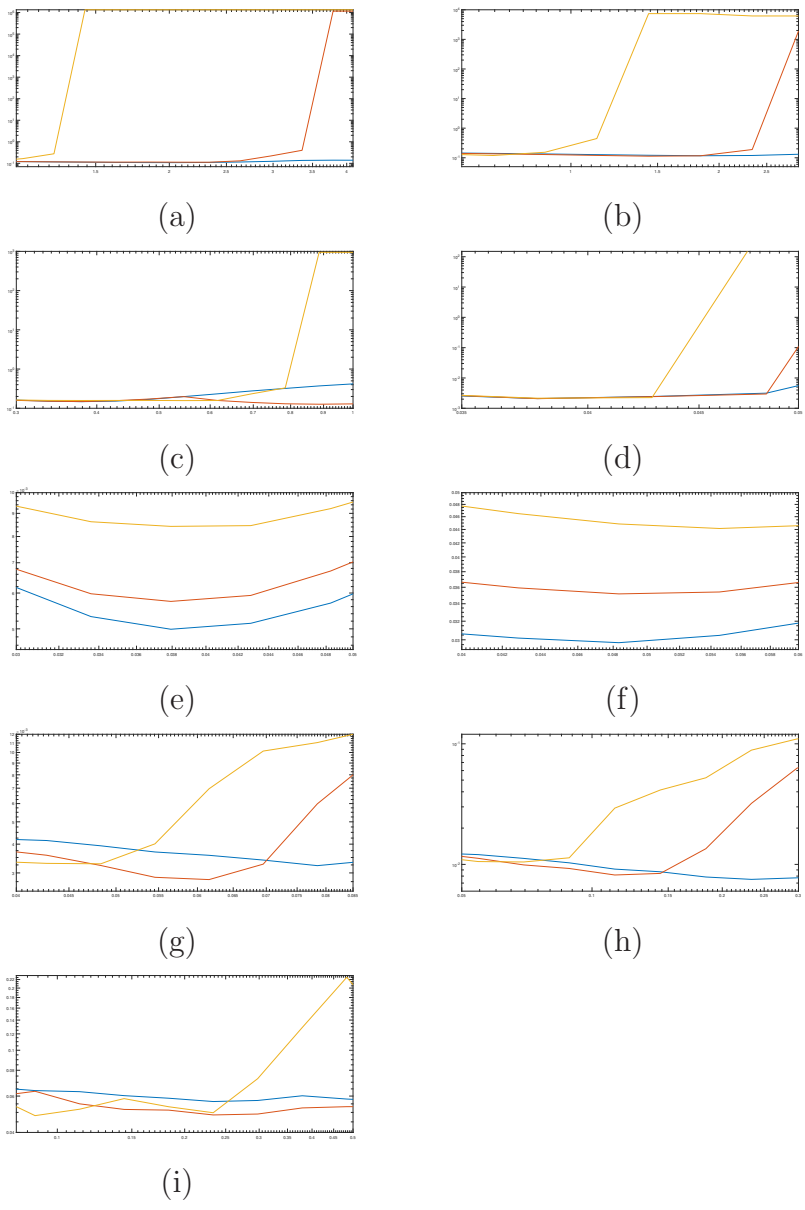

Figure 4: RRE for different choices of μ and q. The panels are magnifications of the the panels of Figure 3 around the optimal value of μ, i.e., the μ-value that gives the smallest RRE. The different graphs show the RRE versus μ. The yellow graph is for $q = 0.6$, the red graph for $q = 0.8$, and the blue graph for $q = 0.9$. Panels (a)-(c) report results for the baart test problem, panels (d)-(f) for the heat test problem, and panels (g)-(i) for the phillips test problem. The panels (a), (d), and (g) show results for the noise level $\sigma = 10^{-3}$, the panels (b), (e), and (h) for $\sigma = 10^{-2}$, and the panels (c), (f), and (i) for $\sigma = 10^{-1}$.

Table 1: Number of iterations for the baart test problem for two values of δ.

δ	μ	$q = 0.99$	$q = 0.95$	$q = 0.90$	$q = 0.85$	$q = 0.80$
1.0	$6.9 \cdot 10^{-4}$	98	44	28	21	17
	$4.8 \cdot 10^{-2}$	93	43	27	20	16
1.5	$6.9 \cdot 10^{-4}$	74	37	24	18	15
	$4.8 \cdot 10^{-2}$	71	36	24	18	14

Table 2: Number of iterations for the heat test problem for two values of δ.

δ	μ	$q = 0.99$	$q = 0.95$	$q = 0.90$	$q = 0.85$	$q = 0.80$
1.0	$6.9 \cdot 10^{-4}$	14	11	9	8	7
	$4.8 \cdot 10^{-2}$	24	18	14	11	10
1.5	$6.9 \cdot 10^{-4}$	9	8	7	6	5
	$4.8 \cdot 10^{-2}$	16	13	10	9	8

Indeed, we have observed the number of iterations to decrease as δ increases until δ is too large. The value of δ that gives the least number of iterations depends on the problem.

We turn to the heat test problem for the same values of μ, q, and δ, and the same noise level. The errors in the computed approximate solutions are essentially independent of q also for this test problem. Table 2 shows the number of iterations required to satisfy the discrepancy principle. Similarly as in Table 1, the number of iterations decreases as q decreases for fixed δ, and decreases as δ increases for fixed q.

Letting $\delta > 1$ does not reduce the number of iterations required to satisfy the discrepancy principle for the phillips test problem. Hence, the choice of δ that results in the least number of iterations is problem dependent. We conclude that $\delta = 1$ is a safe choice, but the number of iterations may be reduced by choosing a larger δ-value for some problems.

4.5. Final considerations

We observed the value of the parameter μ to be important for the performance of the NMLB algorithm. This parameter affects both the rate of convergence and the quality of the computed approximate solution. In particular, if μ is chosen too large, then the NMLB algorithm slows down to the point of becoming unstable. Moreover, if the value of μ is far from the value that results in the smallest RRE, the NMLB algorithm may determine an

approximation of $\mathbf{x}_{\text{true}}$ of poor quality. The value of the parameter q has a lesser effect on the quality of the computed approximation of $\mathbf{x}_{\text{true}}$. However, this parameter affects the rate of convergence. Using a large q-value is both advantageous and disadvantageous. The main advantage is that the NMLB algorithm is more stable, since the convergence towards $\bar{\alpha}$ of the sequence α_k is slower. Therefore, more iterations can be carried out before the method becomes unstable due to severe ill-conditioning of the preconditioner P. On the other hand, the number of iterations required to satisfy the discrepancy principle increases with q. This increase may be rapid. We note that, if a large μ-value is required, then we have to choose a large value of q. This is due to the fact that when μ is large, convergence typically is slow and many iterations are required. The latter leads to instability if q is not large enough. Finally, the examples of the previous subsection illustrate that letting δ be strictly larger than one may increase the rate of convergence.

5. Conclusions

A numerical investigation of the NMBL algorithm is presented. We have elucidated how the choice of some user-specified parameters affect the results obtained with this algorithm. In particular, we show that the choice of the regularization parameter μ is of importance, and that an imprudent choice of this parameter may result in computed solutions of poor quality. The iterates are found to converge faster when the parameter δ in (24) is chosen larger than one. The exact choice is not critical, but a too large value of δ prevents convergence. Our examples illustrate that if the user-specified parameters in (11) are not chosen carefully, then the theoretical results for this method shown in [4] may not hold in finite precision arithmetic.

In our experiments we considered fairly small examples in one space-dimension so that the preconditioner can be applied repeatedly by using a direct factorization in reasonable computational time. However, in many real-world applications the number of space-dimensions is larger than one and, if the matrix A does not have an exploitable structure, application of the preconditioner by a direct factorization method may not be feasible. To reduce this difficulty Cai et al. [16] recently proposed a modification of the NMLB algorithm, in which preconditioners of the form (9) are approximated by nearby ones with a structure that allows their fairly inexpensive application also for large-scale problems. The numerical analysis presented in this

paper can be extended to the iterative scheme described by Cai et al. [16]. This is a topic of future research.

Acknowledgment

The first author is member the group GNCS of INdAM that partially founded this work with the grant "Non linear numercal methods for inverse problems and applications" , while the research of the third author is supported in part by NSF grants DMS-1720259 and DMS-1729509. The authors would like to thank Marco Donatelli for useful discussions and suggestions and the referees for comments.

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

[2] P. C. Hansen, Rank Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion, SIAM, Philadelphia, 1998.

[3] S. Gazzola, P. Novati, M. R. Russo, On Krylov projection methods and Tikhonov regularization, Electronic Transactions on Numerical Analysis 44 (2015) 83–123.

[4] J. Huang, M. Donatelli, R. H. Chan, Nonstationary iterated thresholding algorithms for image deblurring, Inverse Problems & Imaging 7 (3) (2013) 717–736.

[5] J.-F. Cai, S. Osher, Z. Shen, Linearized Bregman iterations for frame-based image deblurring, SIAM Journal on Imaging Sciences 2 (1) (2009) 226–252.

[6] J.-F. Cai, R. H. Chan, Z. Shen, A framelet-based image inpainting algorithm, Applied and Computational Harmonic Analysis 24 (2) (2008) 131–149.

[7] J.-F. Cai, S. Osher, Z. Shen, Split Bregman methods and frame based image restoration, Multiscale Modeling & Simulation 8 (2) (2009) 337–369.

[8] J.-F. Cai, S. Osher, Z. Shen, Linearized Bregman iterations for compressed sensing, Mathematics of Computation 78 (267) (2009) 1515–1536.

[9] S. Osher, Y. Mao, B. Dong, W. Yin, Fast linearized Bregman iteration for compressed sensing and sparse denoising, Communications in Mathematical Sciences 8 (2010) 93–111.

[10] J. Cai, S. Osher, Z. Shen, Convergence of the linearized Bregman iteration for ℓ_1-norm minimization, Mathematics of Computation 78 (2009) 2127–2136.

[11] D. L. Donoho, Compressed sensing, IEEE Transactions on Information Theory 52 (4) (2006) 1289–1306.

[12] M. Hanke, C. W. Groetsch, Nonstationary iterated Tikhonov regularization, Journal of Optimization Theory and Applications 98 (1) (1998) 37–53.

[13] L. Dykes, S. Noschese, L. Reichel, Circulant preconditioners for discrete ill-posed Toeplitz systems, Numerical Algorithms 75 (2017) 477–490.

[14] M. K. Ng, Iterative Methods for Toeplitz Systems, Oxford University Press, Oxford, 2004.

[15] S. Noschese, L. Reichel, Generalized circulant Strang-type preconditioners, Numerical Linear Algebra With Applications 19 (2012) 3–17.

[16] Y. Cai, M. Donatelli, D. Bianchi, T.-Z. Huang, Regularization preconditioners for frame-based image deblurring with reduced boundary artifacts, SIAM Journal on Scientific Computing 38 (1) (2016) B164–B189.

[17] T. Elfving, T. Nikazad, P. C. Hansen, Semi-convergence and relaxation parameters for a class of SIRT algorithms, Electronic Transactions on Numerical Analysis 37 (2010) 321–336.

[18] P. C. Hansen, Regularization tools version 4.3 for Matlab 7.3, Numerical Algorithms 46 (2007) 189–194.