Linearized Krylov subspace Bregman iteration with nonnegativity constraint

Alessandro Buccini[1] (ORCID) **· Mirjeta Pasha[2] · Lothar Reichel[3]**

Abstract

Bregman-type iterative methods have received considerable attention in recent years due to their ease of implementation and the high quality of the computed solutions they deliver. However, these iterative methods may require a large number of iterations and this reduces their usefulness. This paper develops a computationally attractive linearized Bregman algorithm by projecting the problem to be solved into an appropriately chosen low-dimensional Krylov subspace. The projection reduces the computational effort required for each iteration. A variant of this solution method, in which nonnegativity of each computed iterate is imposed, also is described. Extensive numerical examples illustrate the performance of the proposed methods.

Keywords Linearized Bregman iteration · Ill-posed problem · Krylov subspace · Nonnegativity constraint

Mathematics Subject Classification (2010) 65F22 · 65K10 · 65R32

✉ Alessandro Buccini
alessandro.buccini@unica.it

Mirjeta Pasha
mpasha3@asu.edu

Lothar Reichel
reichel@math.kent.edu

1 Department of Mathematics and Computer Science, University of Cagliari, Via Ospedale 72, 09124 Cagliari, Italy

2 School of Mathematical and Statistical Sciences, Arizona State University, Tempe, 85281 AZ, USA

3 Department of Mathematical Sciences, Kent State University, 1300 Lefton Esplanade, Kent, 44242 OH, USA

1 Introduction

Many applications in science and engineering require the solution of minimization problems of the form:

$$\min_{u \in \mathbb{R}^n} \|Au - b\|_2, \tag{1}$$

where $A \in \mathbb{R}^{m \times n}$ is a matrix, whose singular values gradually decay to zero with no significant gap; the matrix may be rank deficient. Throughout this paper, $\|\cdot\|_2$ stands for the Euclidean vector norm. The vector $b \in \mathbb{R}^m$ represents data that is corrupted by an unknown error $e \in \mathbb{R}^m$, which may stem from measurement or discretization inaccuracies. We will refer to the error e as "noise." Problems of this kind arise, for instance, from the discretization of Fredholm integral equations of the first kind and are commonly referred to as linear discrete inverse ill-posed problems; see, e.g., [17, 22, 23] for discussions on discrete inverse ill-posed problems. We are primarily interested in the situation when $m \leq n$, but the methods described also can be applied when $m > n$.

Let b_{true} denote the unknown error-free vector associated with b, i.e.:

$$b = b_{\text{true}} + e,$$

and let $\mathcal{R}(A)$ denote the range of A. We will assume that $b_{\text{true}} \in \mathcal{R}(A)$ and that a fairly accurate bound for the error:

$$\|e\|_2 \leq \epsilon, \tag{2}$$

is known. This allows application of the discrepancy principle; see below.

Let $A^\dagger$ denote the Moore–Penrose pseudo-inverse of A. We would like to determine an accurate approximation of $u_{\text{true}} = A^\dagger b_{\text{true}}$, i.e., of the minimum-norm solution of:

$$\min_{u \in \mathbb{R}^n} \|Au - b_{\text{true}}\|_2.$$

We remark that the exact solution of (1) can be expressed as $A^\dagger b$. Due to the error e in b and the clustering of the singular values of A at the origin, the vector:

$$A^\dagger b = A^\dagger b_{\text{true}} + A^\dagger e = u_{\text{true}} + A^\dagger e$$

typically is dominated by the propagated error $A^\dagger e$ and, therefore, is not a useful approximation of u_{true}.

A possible approach to reducing the propagation of the error e into the computed solution when $m < n$ and $b \in \mathcal{R}(A)$ is to seek a sparse solution of (1). This can be achieved by minimizing the ℓ_1-norm of the computed solution, i.e., by solving the constrained minimization problem:

$$\min_{\text{subject to } Au=b} \|u\|_1, \tag{3}$$

where

$$\|u\|_1 = \sum_{i=1}^n |u_i|, \qquad u = [u_1, u_2, \ldots, u_n]^T.$$

Here and throughout this paper, the superscript ^{T} denotes transposition. The solution of (3) might not be unique, since the ℓ_1-norm is not strictly convex. Uniqueness of the solution can be ensured by replacing (3) by:

$$u_\mu = \arg \min_{u \in \mathbb{R}^n} \left\{ \mu \|u\|_1 + \frac{1}{2\delta} \|u\|_2^2 : Au = b \right\}, \tag{4}$$

where $\mu > 0$ and $0 < \delta < 1/\rho(A^T A)$ are user-defined constants, with $\rho(M)$ denoting the spectral radius of the square matrix M. The upper bound for δ secures convergence of the Bregman iterations defined below. We will refer to the parameter $\mu > 0$ in (4) as the regularization parameter. One of the most popular algorithms for the solution of (4) is the linearized Bregman algorithm, which is an iterative method; see, e.g., [3, 12, 13, 32, 35] or below.

The solutions of many linear discrete ill-posed problems (1) have a sparse representation in a suitably chosen basis. For instance, in image restoration problems, images usually have sparse representations in terms of wavelets or framelets. To make use of the sparsity, we transform the problem (4) so that its solution has a sparse representation.

The iterates generated by the linearized Bregman algorithm might converge only slowly to the solution of (4). Its application therefore may be quite expensive for some problems. This paper describes an approach to reduce the computational cost of Bregman iteration. Specifically, the problem to be solved is projected into a Krylov subspace of small dimension $d \ll \min\{m, n\}$ by applying d steps of Golub–Kahan bidiagonalization to the matrix A. The dimension d is chosen large enough so that the space contains a vector $u^* \in \mathbb{R}^n$ that satisfies the *discrepancy principle*, that is:

$$\|Au^* - b\|_2 \leq \tau\epsilon, \tag{5}$$

where ϵ satisfies (2) and $\tau \geq 1$ is a user-supplied constant that is independent of ϵ. Once we have constructed the appropriate subspace, we solve (4) there. Since the dimension d of the subspace selected is usually much smaller than $\min\{m, n\}$, the iterations with the Bregman algorithm in this space are cheaper to carry out than in $\mathbb{R}^n$.

In many applications, it is known that the desired solution u_{true} lies in a closed and convex set. In this situation, it is generally beneficial to impose constraints on the Bregman algorithm such that the generated iterates lie in this closed and convex set. For instance, in image restoration problems, the entries of the desired solution represent pixel values of the image. Pixel values are nonnegative; therefore, it is generally meaningful to solve the constraint minimization problem:

$$u_\mu^+ = \arg \min_{u \geq 0} \left\{ \mu \|u\|_1 + \frac{1}{2\delta} \|u\|_2^2 : Au = b \right\} \tag{6}$$

instead of (4). When the desired solution u_{true} is nonnegative, it is usually beneficial to impose a nonnegativity constraint on the computed solution of (1); see [1, 6, 15, 29, 33, 36] for illustrations. In particular, the vector u_μ^+ is usually a more accurate approximation of u_{true} than u_μ. We remark that a closed form for u_μ^+ is generally not available. Section 3 describes a solution method for the problem (6), based on

first projecting the problem into a Krylov subspace of fairly small dimension and then projecting computed approximate solutions into the nonnegative cone.

This paper is organized as follows. Section 2 presents the linearized Bregman algorithm and discusses how to include a transformation to the framelet domain. In Section 3, we describe our projected method and discuss the use of a nonnegativity constraint. A faster converging variant of the unconstrained Bregman iterations is briefly discussed in Section 4, and Section 5 presents a few numerical examples that illustrate the performance of the methods described in this paper. Finally, Section 6 contains concluding remarks.

2 The linearized Bregman algorithm

The linearized Bregman algorithm was introduced in [10, 35]. It is designed to solve problems of the form:

$$\min_{u \in \mathbb{R}^n} \{J(u) : Au = b\},$$

where $J(u)$ is a continuous, convex functional. We briefly review some properties of the linearized Bregman algorithm. Let $D_J^p(u, v)$ denote the (nonnegative) quantity:

$$D_J^p(u, v) = J(u) - J(v) - \langle p, u - v \rangle,$$

where $p \in \partial J(v)$ is an element of the subgradient of the functional J at the point v, and $\langle u, w \rangle$ denotes the standard inner product of elements $u, w \in \mathbb{R}^n$. The quantity $D_J^p(u, v)$ is commonly referred to as the Bregman distance [5, 27] based on the convex functional J between the points u and v. Note that the Bregman distance in general is not a metric in the usual sense, since it does not satisfy the triangle inequality and it is not symmetric, i.e., $D_J^p(u, v)$ may be different from $D_J^p(v, u)$. Nevertheless, the Bregman distance measures the closeness between u and v, since $D_J^p(u, v) \geq 0$, and $D_J^p(u, v) = 0$ if $u = v$.[1]

We will assume that $m \leq n$ and that A is a surjective matrix. Then, the linear system of equations $Au = b$ has at least one solution for any right-hand side b. Given $u^0 = v^0 = 0$, the linearized Bregman iteration can be expressed as:

$$\begin{cases} u^{k+1} = \arg\min_{u \in \mathbb{R}^n} \{\frac{1}{2\delta} \|u - (u^k - \delta A^T (Au^k - b))\|_2^2 + \mu D_J^{p_k}(u, u^k)\}, \\ v^{k+1} = v^k - \frac{1}{\mu\delta}(u^{k+1} - u^k) - \frac{1}{\mu} A^T (Au^k - b) \end{cases}$$

(7)

for $k = 0, 1, \ldots$. These iterations can be written as:

$$\begin{cases} v^{k+1} = v^k - A^T (Au^k - b), \\ u^{k+1} = \delta \text{prox}_J(v^{k+1}, \mu), \end{cases}$$

(8)

for $k = 0, 1, \ldots$, where $\text{prox}_J(v^{k+1}, \mu)$ denotes the proximal operator of the functional J, i.e.:

$$\text{prox}_J(v, \mu) = \arg\min_u \|v - u\|_2^2 + \mu J(u);$$

---

[1]$D_J^p(u, v) = 0$ if and only if $u = v$ when J is a strictly convex functional.

see [11, 12] for details. The special case when $J(u) = \|u\|_1$ is of particular interest to us. Then:

$$\text{prox}_J(v, \mu) = T_\mu(v),$$

where $T_\mu(v)$ denotes the soft-thresholding operator, i.e.:

$$T_\mu(v) = [t_\mu(v_1), t_\mu(v_2), \ldots, t_\mu(v_n)]^T \in \mathbb{R}^n \quad \text{for} \quad v = [v_1, v_2, \ldots, v_n]^T,$$

where

$$t_\mu(x) := \begin{cases} 0 & \text{if } |x| \leq \mu, \\ \text{sign}(x)(|x| - \mu) & \text{if } |x| > \mu. \end{cases}$$

We summarize the linearized Bregman algorithm for $J(u) = \|u\|_1$ in Algorithm 1.

Algorithm 1 Linearized Bregman iteration.

1: **Input:** Surjective matrix $A \in \mathbb{R}^{m \times n}$ with $m \leq n$. Parameters $\mu > 0$ and
2:      $0 < \delta < 1/\rho(A^T A)$. Soft-thresholding operator T_μ.
3: $u^0 = v^0 = 0$.
4: **for** $k = 1, 2, \ldots$ until convergence
5:      $v^{k+1} = v^k - A^T(Au^k - b)$
6:      $u^{k+1} = \delta T_\mu(v^{k+1})$
7: **end for**
8: **Output:** Approximate solution u^{k+1}.

Algorithm 1 is concise, simple to program, and requires only matrix-vector product evaluations, vector additions, and soft-thresholding. The iterations are terminated when two consecutive iterates are sufficiently close; see Section 5 for details.

We note that Algorithm 1 may require many iterations to give an accurate approximation of the solution of (4); see below. Moreover, in some applications, the matrix-vector product evaluations with A and A^T may be expensive. Applications of the algorithm include basis pursuit problems, which arise in compressed sensing and allow images and signals to be reconstructed from small amounts of data.

The condition number of A with respect to the spectral matrix norm is defined as the ratio of the largest to smallest singular values of A. A matrix is said to be ill-conditioned when this ratio is large. When the matrix A is ill-conditioned, the convergence of the sequence $u_1, u_2, \ldots$ generated by Algorithm 1 to the solution of (4) may be very slow. This prompted Cai et al. [13] to propose the use of a preconditioner. The preconditioner described in [13] is attractive to use for certain matrices A. An approach based on replacing the stationary iteration in Algorithm 1 by a non-stationary one to speed up the convergence has recently been discussed by Huang et al. [26]. Numerical aspects of the latter kind of iterations are considered in [7]. Moreover, extensions of the method proposed in [26] were considered in [4, 14]. We also note that when A is severely ill-conditioned and the vector b is contaminated by noise, the iterates generated by Algorithm 1 do not converge to u_{exact} with increasing iteration number. More precisely, there is an index ℓ such that the first ℓ iterates u_i approach u_{exact} as i increases and is bounded by ℓ, but the iterates u_i move away

from u_{exact} as $i > \ell$ increases. This behavior of the iterates is commonly referred to as semiconvergence. This difficulty can be remedied by terminating the iteration process sufficiently early. Early termination of the iterations can be achieved with the aid of the discrepancy principle, but other stopping criteria also can be used.

We conclude this section with a discussion on the application of framelets to represent the solution. Many solutions of interest have a sparse representation in terms of framelets; see, e.g., [8, 12, 13, 26]. Framelets are frames with local support.

Definition 1 Let $W \in \mathbb{R}^{r \times n}$ with $n \le r$. The set of rows of W is said to be a tight frame for $\mathbb{R}^n$ if $\forall u \in \mathbb{R}^n$ it holds:

$$\|u\|^2 = \sum_{j=1}^{r} (w_j^T u)^2, \tag{9}$$

where $w_j \in \mathbb{R}^n$ is the j-th row of W (written as a column vector), i.e., $W = [w_1, w_2, \ldots, w_r]^T$. The matrix W is called an analysis operator and W^T a synthesis operator.

Equation (9) is equivalent to the perfect reconstruction formula $x = W^T y$, $y = Wx$, i.e., W is a tight frame if and only if $W^T W = I$. In general, $WW^T \ne I$, unless $r = n$ and the framelets are orthogonal.

We can transform our given linear discrete ill-posed problem into the framelet domain by using the identity $W^T W = I$: inserting this identity into $Au = b$ yields:

$$AW^T Wu = b.$$

Let $Z = AW^T$ and $y = Wu$. Then, the above equation can be written as

$$Zy = b.$$

The entries of the vector y are the framelet coefficients of the solution. In many applications, the vector y is sparse. Linearized Bregman-type iteration, which aims to determine a sparse solution, is a suitable iterative solution method. Note that the matrix Z is not explicitly formed when applying Algorithm 1. Since typically the matrix W is very sparse, the evaluations of matrix-vector products with W and W^T are very cheap. Thus, the computational cost of transforming a linear discrete ill-posed problem to a framelet domain is almost negligible.

3 Projected linearized Bregman and nonnegative linearized Bregman algorithms

This section first introduces a projected Bregman algorithm, which is based on projecting the given linear discrete ill-posed problem into a Krylov subspace of fairly small dimension. Subsequently, we introduce a projected Bregman algorithm with nonnegativity constraint.

3.1 Projection into a Krylov subspace

We seek an approximate solution of the problem (1) in a Krylov subspace:

$$\mathcal{K}_d(A^T A, A^T b) = \text{span}\{A^T b, A^T A A^T b, (A^T A)^2 A^T b, \ldots, (A^T A)^{d-1} A^T b\} \quad (10)$$

of low-dimension $d \ll \min\{m, n\}$. An orthonormal basis for this space is constructed with the Golub–Kahan bidiagonalization method applied to the matrix A with initial vector b; see, e.g., [21]. Application of ℓ steps of Golub–Kahan bidiagonalization gives the decompositions:

$$AV_\ell = U_{\ell+1} B_{\ell+1,\ell}, \quad A^T U_{\ell+1} = V_\ell B_{\ell,\ell}^T, \quad (11)$$

where $U_{\ell+1} \in \mathbb{R}^{m \times (\ell+1)}$ and $V_\ell \in \mathbb{R}^{n \times \ell}$ have orthonormal columns, the first column of $U_{\ell+1}$ is $b/\|b\|_2$, and the matrices $B_{\ell+1,\ell} \in \mathbb{R}^{(\ell+1) \times \ell}$ and $B_{\ell,\ell} \in \mathbb{R}^{\ell \times \ell}$ are lower bidiagonal; the matrix $B_{\ell,\ell}$ is the leading $\ell \times \ell$ submatrix of $B_{\ell+1,\ell}$. The columns of V_ℓ, for $\ell = d$, span the Krylov subspace (10); in particular, $v_1 = A^T b/\|A^T b\|_2$. We tacitly assume that d is sufficiently small so that the decompositions (11) exist for $\ell = d$. This is the generic situation. If this condition does not hold, then the computations simplify. This situation is rare and we will therefore not dwell on it further.

We would like to choose d as small as possible, but such that the discrepancy principle (5) can be satisfied by an element in $\mathcal{K}_d(A^T A, A^T b)$, i.e., we would like d to satisfy:

$$d = \arg\min \left\{ \ell : \min_{u \in \mathcal{K}_\ell(A^T A, A^T b)} \|Au - b\|_2 \leq \tau\epsilon \right\}.$$

Once we have determined d, we substitute (11) with $\ell = d$ into (1) to obtain:

$$\min_{u \in \mathcal{K}_d(A^T A, A^T b)} \|Au - b\|_2^2 = \min_{y \in \mathbb{R}^d} \|AV_d y - b\|_2^2 \overset{(a)}{=} \min_{y \in \mathbb{R}^d} \|U_{d+1} B_{d+1,d} y - b\|_2^2$$

$$\overset{(b)}{=} \min_{y \in \mathbb{R}^d} \|B_{d+1,d} y - \|b\|_2 e_1\|_2^2$$

where $e_1 = [1, 0, \ldots, 0]^T$ denotes the first axis vector. Here, (a) follows by substituting the decomposition (11), and (b) follows from the facts that the columns of U_{d+1} are orthonormal and $U_{d+1}^T b = \|b\|e_1$.

Let $W \in \mathbb{R}^{r \times n}$ be a framelet analysis operator, define:

$$K = B_{d+1,d} V_d^T W^T,$$

and observe that $V_d^T W^T W V_d = I$. We would like to apply linearized Bregman iteration to solve:

$$\min_{z \in \mathbb{R}^r} \left\{ \mu\|z\|_1 + \frac{1}{\delta}\|z\|_2^2 : z \in \arg\min_z \|Kz - \|b\|_2 e_1\|_2^2 \right\}, \quad (12)$$

where $z = W V_d y$. These iterations can be expressed as:

$$\begin{cases} v^{k+1} = v^k - K^T(Kz^k - \|b\|_2 e_1), \\ z^{k+1} = \delta T_\mu(v^{k+1}), \end{cases} \quad k = 0, 1, 2, \ldots,$$

which can be written in the form:

$$
\begin{cases}
v^{k+1} = v^k - W V_d B_{d+1,d}^T (B_{d+1,d} V_d^T W^T z^k - \|b\|_2 e_1), \\
z^{k+1} = \delta T_\mu(v^{k+1}),
\end{cases}
\quad k = 0, 1, 2, \dots .
$$

(13)

The sequence $z^1, z^2, \dots$ converges to the solution of (12). Since the original (large) problem has been projected into a subspace of small dimension, the iterations generally do not display semiconvergence. We observe that the large matrix K does not have to be explicitly formed to carry out the iterations (13). In fact, all required matrix-vector product evaluations involve only fairly small or sparse matrices. We refer to the algorithm defined by (13) as the Projected Linearized Bregman (PLB) algorithm. Convergence is secured for $0 < \delta < 1/\rho(K^T K)$, where we note that $\rho(K^T K) = \rho(B_{d+1,d}^T B_{d+1,d}) \leq \rho(A^T A)$. This means that the PLB algorithm allows a wider range of δ-values than the linearized Bregman iteration (Algorithm 1). In particular, the PLB method may give a convergent sequence of iterates also when linearized Bregman iteration does not. Moreover, the small size of $B_{d+1,d}$ makes it easy to compute $\rho(B_{d+1,d}^T B_{d+1,d})$ and thereby to determine a bound for the parameter δ. We summarize the computations of the PLB iterations in Algorithm 2, where, with slight abuse of notation, we use u^k instead of z^k. Iterations with the algorithm are terminated when consecutive iterates are sufficiently close; see Section 5.

Algorithm 2 PLB.

1: **Input:** Surjective matrix $A \in \mathbb{R}^{m \times n}$ with $m \leq n$. Noise-corrupted data
2:           vector $b \in \mathbb{R}^m$. Bound ϵ for the norm of the noise in b. Parameters $\tau > 1$,
3:           $0 < \delta < 1/\rho(B_{d+1,d}^T B_{d+1,d})$, and $\mu > 0$. Analysis operator $W \in \mathbb{R}^{r \times n}$ and
4:           soft-thresholding operator T_μ.
5: **for** $k = 1, 2, \dots$
6:       Compute the decompositions (11)
7:       **if** $\min_y \|B_{k+1,k} y - \|b\|_2 e_1\|_2 \leq \tau \epsilon$
8:           $d = k$
9:           exit
10:      **end if**
11: **end for**
12: **for** $k = 1, 2, \dots$ until convergence
13:      $v^{k+1} = v^k - W(V_d(B_{d+1,d}^T(B_{d+1,d}(V_d^T(W^T u^k)) - \|b\|_2 e_1)))$
14:      $u^{k+1} = \delta T_\mu(v^{k+1})$
15: **end for**
16: **Output:** Approximate solution u^{k+1}

3.2 Nonnegativity constraint

In many applications, such as medical imaging and astronomy, the exact solution of (1) is known to live in a closed and convex set Ω. Often approximations of higher accuracy of the desired solution can be determined by constraining the iterates of the

PLB algorithm to Ω. In this section, as well as in the computed examples presented in Section 5, the set Ω is chosen to be the nonnegative cone:

$$\Omega_0 = \{u \in \mathbb{R}^n : u_i \geq 0, \ i = 1, 2, \ldots, n\}.$$

However, the theory developed in the following easily can be adapted to more general closed and convex sets.

Define the indicator function i_0 for Ω_0:

$$i_0(x) = \begin{cases} 0 & \text{if } x \in \Omega_0, \\ \infty & \text{else.} \end{cases}$$

We insert the nonnegativity constraint on $u = W^T z$ into (12) to obtain:

$$\arg\min_{z \in \mathbb{R}^r} \left\{ \mu\|z\|_1 + \frac{1}{\delta}\|z\|_2^2 + i_0(W^T z) : z \in \arg\min_z \|Kz - \|b\|_2 e_1\|_2^2 \right\}.$$

To solve this problem, we introduce the proximal operator for

$$J(z) = \mu\|z\|_1 + i_0(W^T z). \tag{14}$$

Definition 2 Let $\Omega \subseteq \mathbb{R}^n$ be a closed and convex set, and let $u \in \mathbb{R}^n$. The metric projection of u onto Ω is given by:

$$P_\Omega(u) = \arg\min_{z \in \Omega} \|u - z\|_2.$$

In particular, the metric projection onto Ω_0 can be obtained by

$$(P_{\Omega_0}(u))_i = \begin{cases} u_i & \text{if } u_i \geq 0, \\ 0 & \text{else.} \end{cases}$$

If $W = I$, then the proximal operator for J in (14) can be expressed as:

$$\operatorname{prox}_J(z) = P_{\Omega_0}(T_\mu(z)). \tag{15}$$

Eq. (15) is a known result from [30] and can be derived with the help of proximal operator theory.

We now derive the proximal operator for J in (14) when $W \neq I$. Let

$$\Omega_W = \{y; \ y = W\tilde{y}, \ \tilde{y} \geq 0\} \tag{16}$$

and let i_W be the indicator function for the set Ω_W. Then

$$\begin{aligned} \operatorname{prox}_J(z) &= \arg\min_v \left\{ \|v - z\|_2^2 + \mu\|v\|_1 + i_0(W^T v) \right\} \\ &= \arg\min_v \left\{ \|v - z\|_2^2 + \mu\|v\|_1 + i_W(v) \right\} \\ &= P_{\Omega_W}(T_\mu(z)). \end{aligned}$$

It remains to discuss the evaluation of the operator $P_{\Omega_W}(z)$. We have:

$$
\begin{aligned}
P_{\Omega_W}(z) &= \arg\min_{x \in \Omega_W} \|x - z\|_2^2 \\
&= \arg\min_{x \in \Omega_W} \|(WW^T)(x - z) + (I - WW^T)(x - z)\|_2^2 \\
&\overset{(a)}{=} \arg\min_{x \in \Omega_W} \{\|(WW^T)(x - z)\|_2^2 + \|(I - WW^T)(x - z)\|_2^2\} \\
&= \arg\min_{x \in \Omega_W} \{\|(WW^T)(x - z)\|_2^2 + \|W\tilde{x} - WW^T W\tilde{x} - (I - WW^T)z\|_2^2\} \\
&= \arg\min_{x \in \Omega_W} \{\|(WW^T)(x - z)\|_2^2\} + \|(I - WW^T)z\|_2^2 \\
&= \arg\min_{x \in \Omega_W} \|(W^T)(x - z)\|_2^2 \\
&= W \arg\min_{\tilde{x} \in \Omega_0} \|\tilde{x} - W^T z\|_2^2 \\
&= W P_{\Omega_0}(W^T z),
\end{aligned}
$$

where (a) follows from the fact that the columns of the matrix W are orthonormal. Hence, WW^T and $I - WW^T$ are orthogonal projectors onto complementary subspaces of $\mathbb{R}^r$.

We are now in a position to formulate the Projected Nonnegative Linearized Bregman (PNBL) algorithm, which is presented as Algorithm 3. The stopping criterion for the algorithm is the same as for the other algorithms.

Algorithm 3 PNLB.

1: **Input:** Surjective matrix $A \in \mathbb{R}^{m \times n}$ with $m \le n$. Noise-corrupted data

2:     vector $b \in \mathbb{R}^m$. Bound ϵ for the norm of the noise in b. Parameters $\tau > 1$,

3:     $0 < \delta < 1/\rho(B_{d+1,d}^T B_{d+1,d})$, and $\mu > 0$. Analysis operator $W \in \mathbb{R}^{r \times n}$ and

4:     soft-thresholding operator T_μ.

5: **for** $k = 1, 2, \ldots$

6:     Compute the factorization (11)

7:     **if** $\min_y \|B_{k+1,k} y - \|b\|_2 e_1\|_2 \le \tau\epsilon$

8:         $d = k$

9:         exit

10:     **end if**

11: **end for**

12: **for** $k = 1, 2, \ldots$ until convergence

13:     $v^{k+1} = v^k - W(V_d(B_{d+1,d}^T(B_{d+1,d}(V_d^T(W^T u^k)) - \|b\|_2 e_1)))$

14:     $u^{k+1/2} = \delta T_\mu(v^{k+1})$

15:     $u^{k+1} = W P_{\Omega_0}(W^T u^{k+1/2})$

16: **end for**

17: **Output:** Approximate solution u^{k+1}

It is shown in [34] that linearized Bregman iteration is equivalent to gradient descent applied to the dual problem. This result ensures the convergence of our method to the solution of

$$\arg\min \left\{ \mu \|u\|_1 + \frac{1}{\delta}\|u\|_2 + i_W(u) : u \in \arg\min \|Ku - \|b\|_2 e_1\|_2 \right\}, \qquad (17)$$

where $K = B_{d+1,d} V_d^T W^T$.

4 Further acceleration of the iterations

As mentioned above, the PLB algorithm is computationally cheaper than the LB algorithm. The rate of convergence of the iterates generated with the latter algorithm is $\mathcal{O}(1/k)$, where k denotes the number of iterations. Huang et al. [25] proposed an accelerated version of the LB algorithm with a rate of convergence of $\mathcal{O}(1/k^2)$. This section describes a PLB algorithm that incorporates the acceleration approach of Huang et al. [25]. We refer to this method as the Accelerated PLB (APLB) algorithm. Its performance is illustrated in Section 5.

Algorithm 4 Accelerated PLB.

1: **Input:** Surjective matrix $A \in \mathbb{R}^{m \times n}$ with $m \leq n$. Noise-corrupted data

2: vector $b \in \mathbb{R}^m$. Bound ϵ for the norm of the noise in b. Parameters $\tau > 1$,

3: $0 < \delta < 1/\rho(B_{d+1,d}^T B_{d+1,d})$, and $\mu > 0$. Analysis operator $W \in \mathbb{R}^{r \times n}$ and

4: soft-thresholding operator T_μ.

5: **for** $k = 1, 2, \ldots$

6: Compute the factorization (11)

7: **if** $\min_y \|B_k y - \|b\|_2 e_1\| \leq \tau\epsilon$

8: $d = k$

9: exit

10: **end if**

11: **end for**

12: $\theta_0 = 1$

13: **for** $k = 1, 2, \ldots$ until convergence

14: $v^{k+1} = z^k - W(V_d(B_{d+1,d}^T(B_{d+1,d}(V_d^T(W^T u^k)) - \|b\|_2 e_1)))$

15: $\theta_k = \frac{2}{k+2}$

16: $\alpha_k = 1 + \theta_k(\theta_{k-1}^{-1} - 1)$

17: $z^{k+1} = \alpha_k v^{k+1} + (1 - \alpha_k)v^k$

18: $u^{k+1} = \delta T_\mu(z^{k+1})$

19: **end for**

20: **Output:** Approximate solution u^{k+1}

The convergence results in [25] can be easily extended to the APLB algorithm. We therefore do not present a proof of its convergence properties. Furthermore, the

acceleration approach due to Huang et al. [25] also can be applied in the PNLB algorithm. This gives the Accelerated PNLB (APNLB) algorithm. The latter algorithm is obtained by replacing $u^{k+1} = \delta T_\mu(z^{k+1})$ by $u^{k+1} = \delta P_{\Omega_W}(T_\mu(z^{k+1}))$ in the APLB algorithm.

5 Numerical experiments

This section presents a few numerical examples that illustrate the performance of the methods discussed in the previous sections. We consider the restoration of images that have been contaminated by blur and noise. Continuous space-invariant image deblurring can be formulated as a Fredholm integral of the first kind, i.e., as an integral equation of the form:

$$g(x, y) = \int_\Omega K(s - x, t - y) f(s, t) \, ds \, dt, \qquad (x, y) \in \Omega, \qquad (18)$$

where g represents a blurred, but noise-free, image, f is the unknown image that we would like to recover, and K is a smooth kernel with compact support. The integral operator in (18) is compact. Therefore, the solution of (18) is an ill-posed problem. Discretizing (18) gives a problem of the form (1), with a matrix A that is the sum of a block Toeplitz with Toeplitz block matrix and a correction of small norm due to the boundary conditions that are imposed; see, e.g., [24] for more details on image deblurring.

We use the same tight frames as in [7, 13, 26], 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 $W_1, W_2 \in \mathbb{R}^{n \times n}$, whose corresponding masks are:

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

The analysis operator W in one space-dimension is derived from these masks and by imposing reflexive boundary conditions to ensure that $W^T W = I$. The so-determined filter matrices are:

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

and

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

The corresponding analysis operator W in two space-dimensions is given by:

$$W = \begin{bmatrix} W_0 \otimes W_0 \\ W_0 \otimes W_1 \\ W_0 \otimes W_2 \\ W_1 \otimes W_0 \\ \vdots \\ W_2 \otimes W_2 \end{bmatrix},$$

where $\otimes$ denotes the Kronecker product. This matrix is not explicitly formed. We note that the evaluation of matrix-vector products with W and W^T is inexpensive because the matrix W is very sparse.

In our numerical tests, we fix $\delta = 0.9/\rho(B_{d+1,d}^T B_{d+1,d})$, where $\rho(B_{d+1,d}^T B_{d+1,d})$ can be computed inexpensively since the matrix $B_{d+1,d}$ of our projected problem is of fairly small size. We run the algorithms for different values of the parameter μ, and choose the μ-value that yields the smallest relative restoration error (RRE),

$$\mathrm{RRE}(u) = \frac{\|u - u_{\text{true}}\|_2}{\|u_{\text{true}}\|_2}.$$

This makes it easy to compare the performance of the algorithms of the present paper with other methods discussed in the literature. However, it is possible to determine the parameter μ adaptively during the computations; see [26]. A discussion on the roles of the parameters μ and δ can be found in [7].

We terminate the iterations with Algorithms 2 and 3 when two consecutive iterates are sufficiently close, i.e., when

$$\frac{\|u^{(k+1)} - u^{(k)}\|_2}{\|u^{(k)}\|_2} < 10^{-4}.$$

The error e in the data vector b is modeled by white Gaussian noise. We refer to the ratio:

$$\sigma = \frac{\|e\|_2}{\|A u_{\text{true}}\|_2}$$

as the noise level. We set the parameter τ in the discrepancy principle (5) to 1.01.

Algorithms 2 and 3, and their accelerated variants, are compared to the methods IRfista, IRirn, IRhybrid_fgmres, IRnnfcgls, and IRhtv using MATLAB codes provided in [18]. IRfista is a first-order optimization method that solves a minimization problem of the form:

$$\min_{u \in C}\{\|Au - b\|_2^2 + \mu\|u - u^{(0)}\|_1\}, \tag{19}$$

where C denotes a constrained set defined, e.g., by box or energy constraints, and $u^{(0)}$ is the initial approximate solution vector. For simplicity, it is set to be the zero vector, i.e., $u^{(0)} = 0$; see [2]. IRirn implements an iteratively reweighted norm approach with penalized restarted iterations for computing a 1-norm penalized solution; see [31]. IRhybrid_fgmres applies a flexible version of the solution subspace used in IRhybrid_gmres, and incorporates an iteration-dependent preconditioner that aims to minimize the ℓ_1-norm of the computed solution; see [18] for more details. IRnnfcgls

is a flexible conjugate gradient least squares method for solving nonnegatively constrained least squares problems; the method is proposed in [20]. IRhtv is a penalized restarted iteration method that incorporates a heuristic total variation penalization term described in [19].

To ensure a fair comparison, we provide all the considered methods with the same information, including the noise level and the optimal value of the regularization parameter μ.

Since IRnnfcgls semiconverges, one may need to tune the number of the iterations and force the iterations to stop before the iterates converge to an unregularized least squares solution that might be a poor reconstruction of the desired solution. We stop the iterations as soon as the discrepancy principle is satisfied. All computations are carried out in MATLAB R2018a with about 15 significant decimal digits running on a desktop computer with core CPU Intel(R) Core(TM)i7-4470 @3.40GHz with 8.00GB of RAM.

Comparison of the LB and PLB algorithms We first illustrate that projection into a Krylov subspace can be beneficial both in terms of computational efficiency and quality of the computed restoration especially of large-scale problems. Consider the exact telescope image in Fig. 1a. It is made up of 986×986 pixels. We blur this image with a Gaussian PSF (shown in Fig. 1b). We then add 1% white Gaussian noise and obtain the blur and noise contaminated telescope image in Fig. 1c. The iterates computed by the standard LB method, without projection to a Krylov subspace, show semiconvergence. We therefore terminate the iterations with this method with the discrepancy principle, i.e., we terminate the iterations as soon as:

$$\|Au^k - b\|_2 \leq \tau\epsilon.$$

Figure 2a shows the reconstruction with of telescope image in Fig. 1c with LB algorithm, while the reconstruction by PLB is shown in Fig. 2b. Table 1 reports results obtained with the LB and PLB methods for different noise levels. The table shows that projection into the Krylov subspace significantly accelerates the method and improves the quality of the reconstructed image.
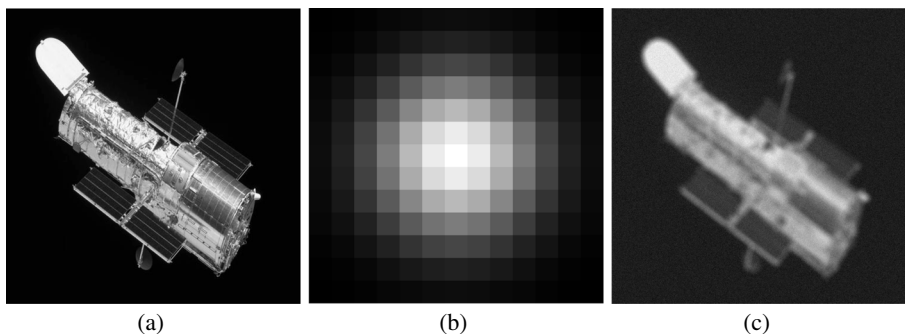

(a)   (b)   (c)

Fig. 1 Comparison of the LB and PLB algorithms: **a** True image (986×986 pixels), **b** PSF (13×13 pixels), **c** blurred and noise-contaminated image with $\sigma = 0.01\|b\|$
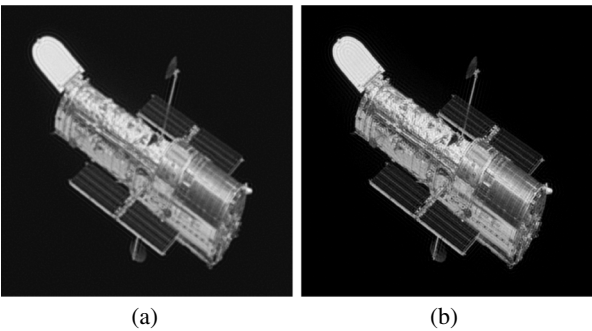
<center>(a)   (b)</center>

Fig. 2 Comparison of the reconstructions obtained by LB and PLB algorithms.

Comparison of the PLB and PNLB methods We would like to illustrate the effect of projecting onto the nonnegative cone. With this aim, we apply the PLB and PNLB methods to restore the contaminated satellite image of Fig. 3c. It is obtained by applying motion blur described by a motion PSF (see Fig. 3b) to the "exact" image of Fig. 3a and then adding 5% white Gaussian noise.

Table 2 displays results for the PLB and PNLB methods. We can observe that the use of nonnegativity constraints improves the quality of the reconstruction. The Krylov subspaces used for both methods are the same. We note that the number of iterations needed for the PNLB method is slightly larger than for the PLB method. This is usually the case for methods with nonnegativity constraints.

Figure 4 shows blowups of the reconstructions obtained with the two methods. Visual inspection of the images shows the PNLB method to be able to provide a uniform reconstruction of the black sky behind the satellite. Moreover, the PLB method generates ringing effects around the edges of the satellite that are not present in the reconstruction computed by the PNLB method.

Table 1 Comparison of the LB and PLB algorithms in terms of RRE, number of iterations, and CPU time for different noise levels

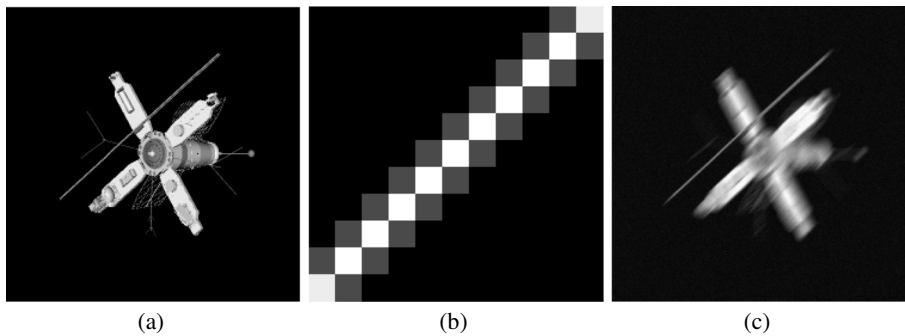|	LB		
	RRE	Number of iterations	CPU time (seconds)
Telescope 1%	0.0899	145	197
Telescope 5%	0.1316	27	25
	PLB		
	RRE	Number of iterations	CPU time (seconds)
Telescope 1%	0.0867	189	42
Telescope 5%	0.1205	30	7.8

Fig. 3 Comparison of the LB and PLB algorithms: **a** true image (246×246 pixels), **b** PSF (11×11 pixels), **c** blurred and noisy image with $\epsilon = 0.05\|b\|$

Since the PLNB method is more accurate than the PLB method, we focus on the former method in the following.

Barbara We consider the Barbara image in Fig. 5a and blur it with the motion PSF shown in Fig. 5b. We add 1% white Gaussian noise to the blurred image. This gives the blurred and noise-contaminated image in Fig. 5c.

Table 3 displays results obtained with the PNLB and other methods described in [18] for several noise levels from 0.1 to 15%. The PNLB algorithm can be seen to outperform the other methods in terms of quality of the reconstruction. This is confirmed by visual inspection of the reconstructions shown in Fig. 6 with 1% noise added.

We can observe that the PNLB method is able to accurately reconstruct the exact image.

Cameraman We turn to the cameraman image shown in Fig. 7a. The exact image is blurred by a PSF that models out-of-focus blur; the PSF is shown in Fig. 7b. The blurred and noise-contaminated image with 2% white Gaussian noise is shown in Fig. 7c.

In Table 4, we report the RRE of the reconstructions obtained with the PNLB and other methods. The PNLB method consistently outperforms the other methods, in particular for higher noise levels. This is confirmed by visual inspection of the reconstructions shown in Fig. 8. For anisotropic blurs, the choice of the Krylov subspace as in IRhybrid_fgmres may not suitable. A more relevant choice of the Krylov subspace for these kinds of blurs is described in [16].

Tomography In this example, we consider a synthetic tomography problem, where the data are the Radon transform of the attenuation coefficients of some scanned object; for details on computerized tomography, see, e.g., [9]. We consider parallel beam tomography, where K parallel X-ray beams are sent through an object at different angles ϕ_k with $k = 1, 2, \ldots, K$. The measured data $b_{j,k}$, that are known as the sinogram, is the line integral of the attenuation coefficient of the object along the j-th beam at angle ϕ_k. We generate the synthetic data using the Matlab program package

Table 2 Comparison of PLB and PNLB in terms of RRE, number of iterations, and CPU time

PLB

Noise level	Dim. of the Krylov subsp.	RRE	Number of iter.	Total CPU time	Krylov CPU time
1%	16	0.1337	300	60.56	1.25
5%	6	0.2126	87	17.56	0.359
15%	3	0.2703	23	4.656	0.187

PNLB

Noise level	Dim. of the Krylov subsp.	RRE	Number of iter.	Total CPU time	Krylov CPU time
1%	16	0.1276	300	58.75	1.4
5%	6	0.2033	98	19.32	0.468
15%	3	0.2618	29	5.921	0.125

We also report the dimension of the considered Krylov subspace and we show the fraction of time required to its computation
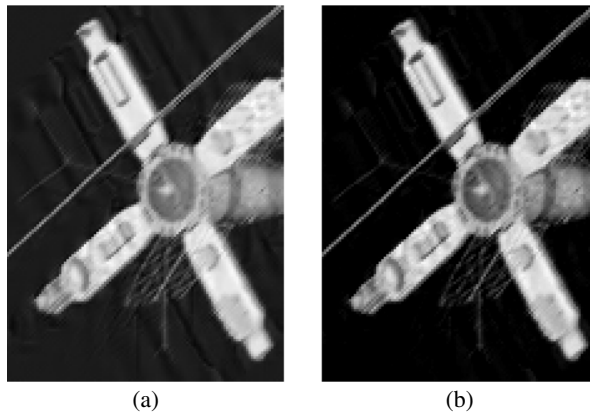
Fig. 4 Comparison of the PLB and PNLB algorithms: Blowups of the reconstructions determined by **a** PLB, **b** PNLB
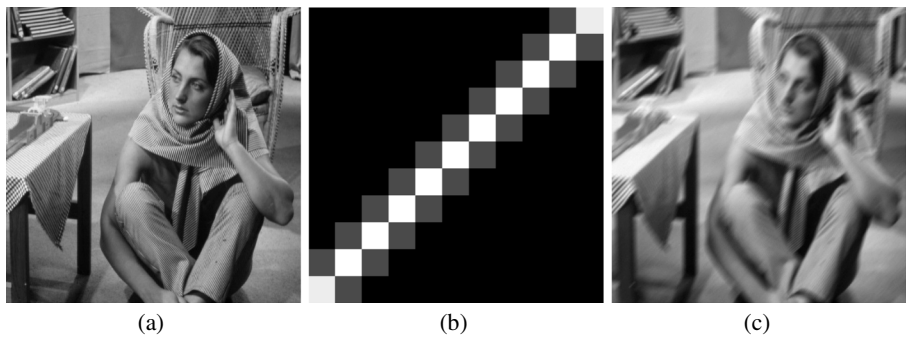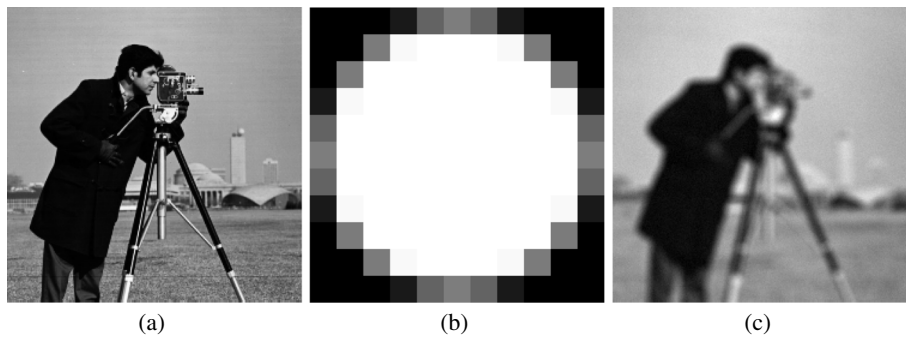

Fig. 5 Barbara test problem: **a** true image (246 × 246 pixels), **b** the PSF (11 × 11 pixels), **c** blurred and noisy image ($\epsilon = 0.01\|b\|$)

Table 3 Barbara test problem: RRE for each noise level and each tested method

	Noise level						
	0.001	0.002	0.005	0.01	0.05	0.1	0.15
PNLB	*0.0832*	*0.0849*	*0.0887*	*0.0912*	*0.1126*	*0.1221*	*0.1302*
IRfista	0.1623	0.1712	0.1917	0.0999	0.1144	0.1231	0.1334
IRhybrid_fgmres	0.1376	0.1366	0.1369	0.1379	0.1607	0.1932	0.2182
IRirn	0.1369	0.1359	0.1362	0.1372	0.1602	0.1932	0.2177
IRhtv	0.1336	0.1346	0.1348	0.1353	0.1447	0.1531	0.1592
IRnnfcgls	0.1336	0.1337	0.1336	0.1337	0.1341	0.1354	0.1376

We report the best result in italics

Fig. 6 Barbara test problem restorations computed by **a** PNLB, **b** IRfista, **c** IRirn

Fig. 7 Cameraman test problem: **a** true image (246 × 246 pixels), **b** PSF (11 × 11 pixels), **c** blurred and noisy image ($\epsilon = 0.002\|b\|$)

Table 4 Cameraman test problem: RRE for each noise level and for each method

Method	Noise level						
	0.001	0.002	0.005	0.01	0.05	0.1	0.15
PNLB	*0.1014*	*0.1014*	*0.1015*	*0.1098*	*0.1522*	*0.1587*	*0.1608*
IRfista	0.1091	0.1129	0.1015	0.1155	0.1511	0.1654	0.1673
IRhybrid_fgmres	0.1512	0.1511	0.1516	0.1528	0.1753	0.2085	0.3651
IRirn	0.1505	0.1506	0.1509	0.1527	0.1648	0.1820	0.2031
IRhtv	0.1481	0.1482	0.1486	0.1490	0.1591	0.1665	0.1724
IRnnfcgls	0.1704	0.1704	0.1704	0.1704	0.1705	0.1715	0.1739

(a)   (b)   (c)
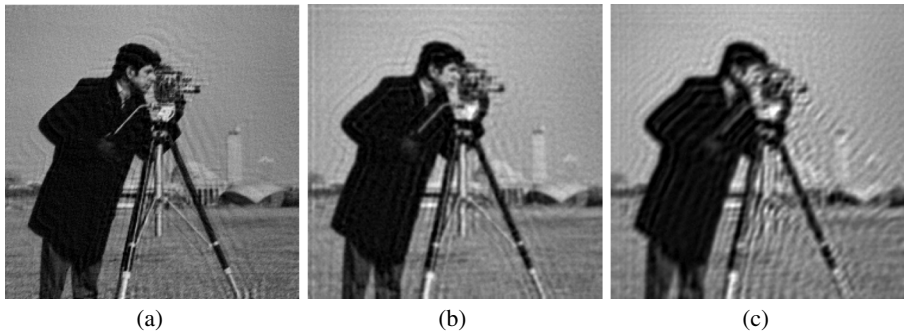
Fig. 8 Cameraman test problem restoration by **a** PNLB, **b** IRfista, **c** IRirn

IR Tools [18]. More specifically, we use the command `PRtomo`(n, options), set the dimension n of the image to 256×256, and consider 90 angles equispaced between $0°$ and $179°$, and 362 beams. This gives an underdetermined system of equations with a matrix $A \in \mathbb{R}^{32580 \times 65536}$.

We comment on the choice of the dimension of the Krylov subspace used in the algorithms described in this paper. The algorithms illustrate the computational benefits of determining an approximate solution of the problem (1) by solving (12) in a Krylov subspace of fairly small dimension d. In all computed examples above, we used the discrepancy principle to determine the value d, which we here will refer to as d_{dp}. It is illustrated in [28] that in the context of Tikhonov regularization with the regularization parameter determined by the discrepancy principle, and the low-dimenional Krylov subspace is determined by a few steps by the Arnoldi process (instead of by Golub–Kahan bidiagonalization), the quality of the computed solution may be increased somewhat by carrying out a few more Arnoldi steps than the smallest number necessary to satisfy the discrepancy principle. Similarly,

(a)   (b)
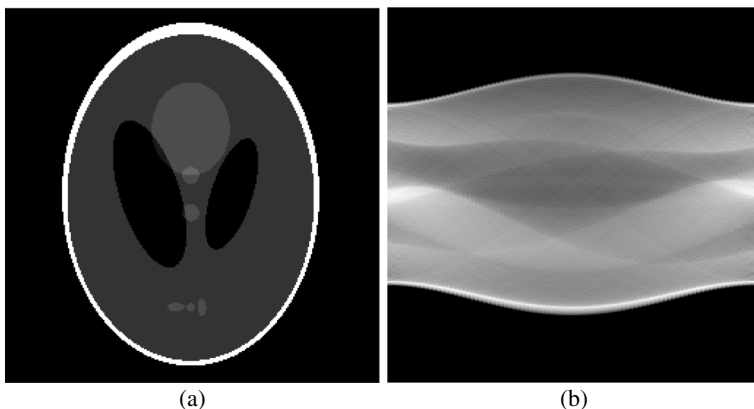
Fig. 9 Tomography test problem: **a** true image (256×256 pixels), **b** noise-free sinogram (362×90 pixels)

Table 5 Tomo test problem: RRE for two noise level for $d = d_{dp}$ and $d = d_{dp} + 5$

	Noise level	
Method	1% ($d_{dp} = 13$)	5% ($d_{dp} = 8$)
PNLB ($d = d_{dp}$)	0.1712	0.2552
PNLB ($d = d_{dp} + 5$)	0.1691	0.2324

it may be possible to improve the quality of the computed solution determined by the algorithms of the present paper somewhat by taking a few more than $d = d_{dp}$ bidiagonalization steps. However, it is difficult to provide simple guidelines for how much larger the number of steps, d, should be than d_{dp}. Moreover, the improvement in quality of the computed solution by letting $d > d$ is modest. The following example provides an illustration. Figure 9a shows the exact attenuation coefficients and Fig. 9b displays the associated noise-free sinogram. We add 1% and 5% white Gaussian noise in the numerical examples reported in Table 5. The table shows small improvements in the quality of the reconstruction by increasing the dimension of the solution subspace. The number of iterations shown in parentheses close to the error level shows (d_{dp}), the number of iterations needed to satisfy the discrepancy principle. Figure 10 depicts the reconstructions of the solution obtained determined by the PNLB method with $d = d_{dp}$ when adding 1% noise to the sinogram and compares this reconstruction with the one determined by the PNLB method with $d = d_{dp} + 5$. Since the gain by choosing $d > d_{dp}$ is not very large, and it is poorly understood how much larger d should be chosen than d_{dp}, we propose to choose $d = d_{dp}$.

Comparison of PLB (PNLB) and accelerated PLB (accelerated PNLB) The accelerated PLB method is described by Algorithm 4. We also apply the PNLB and APNLB methods; the latter is described in Section 4. These four methods are compared in Table 6 for the Barbara, Cameraman, and Satellite test problems with 1% noise. For the Satellite test problem, the PSF is the same as shown in Fig. 3. We can observe
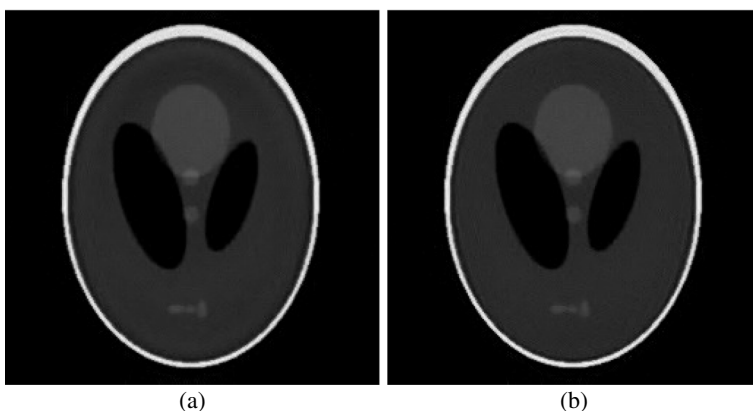

(a)   (b)

Fig. 10 Tomography test problem restoration by **a** PNLB($d = d_{dp}$), **b** PNLB($d = d_{dp} + 5$) with 1% noise

Table 6 Comparison of PNLB and Accelerated PNLB in terms of RRE and CPU time for several test problems

Example	Method	RRE	Iterations	CPU time (seconds)
Barbara	PLB	0.0918	141	105
	Accelerated PLB	0.0916	55	47
	PNLB	0.0912	88	70.04
	Accelerated PNLB	0.091	48	41.82
Cameraman	PLB	0.1277	282	192
	Accelerated PLB	0.1236	91	73
	PNLB	0.1098	116	91.54
	Accelerated PNLB	0.1082	56	49.25
Satellite	PLB	0.1337	300	60.56
	Accelerated PLB	13.27	83	16.5
	PNLB	0.1276	300	58.75
	Accelerated PNLB	0.1234	97	21

that the acceleration strategy reduces the computing time significantly. The RRE is reduced as well, but not by much.

6 Conclusions

This paper proposes that the linearized Bregman method be a projected problem into a Krylov subspace of fairly low dimension. This is shown to reduce the computing time significantly, and to increase the quality of the computed solution somewhat. Unconstrained iterations as well as iterations constrained to a convex set are considered. The imposition of convex constraints may increase the quality of the computed solutions, and in our experience such constraints do not increase the computational burden significantly. The constrained projected linearized Bregman iterative method of this paper is compared with several methods from IR Tools [18] for the restoration of 2D images and was found to be competitive.

References

1. Bai, Z.-Z., Buccini, A., Hayami, K., Reichel, L., Yin, J.-F., Zheng, N.: Modulus-based iterative methods for constrained Tikhonov regularization. J. Comput. Appl. Math. **319**, 1–13 (2017)
2. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**, 183–202 (2009)
3. Benning, M., Burger, M.: Modern regularization methods for inverse problems. Acta Numer. **27**, 1–111 (2018)
4. Bianchi, D., Buccini, A., Donatelli, M.: Structure preserving preconditioning for frame-based image deblurring. In Computational Methods for Inverse Problems in Imaging, page In press. Springer (2019)
5. Bregman, L.M.: The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. USSR Comput. Math. Math. Phys. **7**(3), 200–217 (1967)
6. Buccini, A.: Regularizing preconditioners by non-stationary iterated Tikhonov with general penalty term. Appl. Numer. Math. **116**, 64–81 (2017)
7. Buccini, A., Park, Y., Reichel, L.: Numerical aspects of the nonstationary modified linearized bregman algorithm. Appl. Math. Comput. **337**, 386–398 (2018)
8. Buccini, A., Reichel, L.: An ℓ^2-ℓ^q regularization method for large discrete ill-posed problems. J. Sci. Comput. **78**, 1526–1549 (2019)
9. Buzug, T.M.: Computed tomography. In: Kramme, R., Hoffmann, K.-P., Pozos, R.S. (eds.) Springer Handbook of Medical Technology, pp. 311–342. Springer, Berlin (2011)
10. Cai, J.-F., Osher, S., Shen, Z.: Fast linearized Bregman iteration for compressed sensing and sparse denoising. In: UCLA CAM Report, pp. 08–06 (2008)
11. Cai, J.-F., Osher, S., Shen, Z.: Convergence of the linearized Bregman iteration for ℓ_1-norm minimization. Math. Comput. **78**, 2127–2136 (2009)
12. Cai, J.-F., Osher, S., Shen, Z.: Linearized Bregman iterations for compressed sensing. Math. Comput. **78**, 1515–1536 (2009)
13. Cai, J.-F., Osher, S., Shen, Z.: Linearized Bregman iterations for frame-based image deblurring. SIAM J. Imaging Sci. **2**, 226–252 (2009)
14. Cai, Y., Donatelli, M., Bianchi, D., Huang, T.-Z.: Regularization preconditioners for frame-based image deblurring with reduced boundary artifacts. SIAM J. Sci. Comput. **38**(1), B164–B189 (2016)
15. Calvetti, D., Lewis, B., Reichel, L., Sgallari, F.: Tikhonov regularization with nonnegativity constraint. Electron. Trans. Numer. Anal. **18**, 153–173 (2004)
16. Chung, J., Gazzola, S.: Flexible Krylov methods for ℓ_p regularization. SIAM J. Sci. Comput. **41**(5), S149–S171 (2019)
17. Engl, H.W., Hanke, M., Neubauer, A.: Regularization of Inverse Problems. Kluwer, Dordrecht (1996)
18. Gazzola, S., Hansen, P.C., James, G.N.: IR TOols: A Matlab package of iterative regularization methods and large-scale test problems. Numer. Algorithm. **81**, 773–811 (2019)
19. Gazzola, S., Nagy, J.G.: Generalized Arnoldi–Tikhonov method for sparse reconstruction. SIAM J. Sci. Comput. **36**, B225–B247 (2014)
20. Gazzola, S., Wiaux, Y.: Fast nonnegative least squares through flexible Krylov subspaces. SIAM J. Sci. Comput. **39**(2), A655–A679 (2017)
21. Golub, G.H., Van Loan, C.F. Matrix Computations, 4th edn. Johns Hopkins University Press, Baltimore (2013)

22. Hansen, P.C.: Regularization Tools: A Matlab package for analysis and solution of discrete ill-posed problems. Numer. Algorithm. **6**, 1–35 (1994)
23. Hansen, P.C.: Rank-Deficient and Discrete III-Posed Problems: Numerical Aspects of Linear Inversion. SIAM, Philadelphia (1998)
24. Hansen, P.C., Nagy, J.G., O'Leary, D.P.: Deblurring Images: Matrices, Spectra, and Filtering. SIAM, Philadelphia (2006)
25. Bo, H., Ma, S., Goldfarb, D.: Accelerated linearized Bregman method. J. Sci. Comput. **54**, 428–453 (2013)
26. Huang, J., Donatelli, M., Chan, R.H.: Nonstationary iterated thresholding algorithms for image deblurring. Inverse Probl. Imaging **7**, 717–736 (2013)
27. Kiwiel, K.C.: Proximal minimization methods with generalized Bregman functions. SIAM J. Control Optim. **35**(4), 1142–1168 (1997)
28. Lewis, B., Lothar, R.: Arnoldi–Tikhonov regularization methods. J. Comput. Appl. Math. **226**, 92–102 (2009)
29. Nagy, J.G., Strakos̄, Z.: Enforcing nonnegativity in image reconstruction algorithms. In: Mathematical Modeling, Estimation, and Imaging, vol. 4121, pp. 182–191. International Society for Optics and Photonics (2000)
30. Rapin, J., Bobin, J., Larue, A., Starck, J.-L.: Sparse and non-negative BSS for noisy data. IEEE Trans. Signal Process. **61**, 5620–5632 (2013)
31. Rodriguez, P., Wohlberg, B.: An efficient algorithm for sparse representations with lp data fidelity term. In: Proceedings of 4th IEEE Andean Technical Conference (ANDESCON) (2008)
32. Teboulle, M.: Entropic proximal mappings with applications to nonlinear programming. Math. Oper. Res. **17**(3), 670–690 (1992)
33. Yi, Y., Möller, M., Osher, S.: A dual split Bregman method for fast ℓ^1 minimization. Math. Comput. **82**, 2061–2085 (2013)
34. Yin, W.: Analysis and generalizations of the linearized Bregman method. SIAM J. Imaging Sci. **3**(4), 856–877 (2010)
35. Yin, W., Osher, S., Goldfarb, D., Jerome, D.: Bregman Iterative algorithms for ℓ_1-minimization with applications to compressed sensing. SIAM J. Imaging Sci. **1**, 143–168 (2008)
36. Zheng, N., Hayami, K., Yin, J.-F.: Modulus-type inner outer iteration methods for nonnegative constrained least squares problems. SIAM J. Matrix Anal. Appl. **37**, 1250–1278 (2016)