

FIXING NONCONVERGENCE OF ALGEBRAIC ITERATIVE RECONSTRUCTION WITH AN UNMATCHED BACKPROJECTOR*

YIQIU DONG[†], PER CHRISTIAN HANSEN[†], MICHIEL E. HOCHSTENBACH[‡],
AND NICOLAI ANDRÉ BROGAARD RIIS[†]

Abstract. We consider algebraic iterative reconstruction methods with applications in image reconstruction. In particular, we are concerned with methods based on an unmatched projector/backprojector pair, i.e., the backprojector is not the exact adjoint or transpose of the forward projector. Such situations are common in large-scale computed tomography, and we consider the common situation where the method does not converge due to the nonsymmetry of the iteration matrix. We propose a modified algorithm that incorporates a small shift parameter, and we give the conditions that guarantee convergence of this method to a fixed point of a slightly perturbed problem. We also give perturbation bounds for this fixed point. Moreover, we discuss how to use Krylov subspace methods to efficiently estimate the leftmost eigenvalue of a certain matrix to select a proper shift parameter. The modified algorithm is illustrated with test problems from computed tomography.

Key words. unmatched transpose, algebraic iterative reconstruction, perturbation theory, leftmost eigenvalue estimation, computed tomography

AMS subject classifications. 65F10, 65F15, 65F22, 15A18, 15A60

DOI. 10.1137/18M1206448

1. Introduction. Algebraic iterative reconstruction techniques [10]—such as the methods by Kaczmarz and Cimmino—play an important role in solving inverse problems. In particular, they are popular in computed tomography (CT) due to their great flexibility with respect to the measurement geometry of the X-ray scanner and their ability to handle very underdetermined problems. This is in contrast to filtered backprojection and similar algorithms [16] that rely on specific geometries and a large amount of data. Algebraic iterative methods are also used successfully in other image reconstruction problems such as image deblurring.

The fundamental mechanism behind these methods is known as semiconvergence [16]. During the initial iterations, the iterates approach the exact (and unattainable) solution to the noise-free problem, while in later stages the iterates converge to the undesired noisy solution. The methods produce filtered, or regularized, solutions and the number of iterations plays the role of a regularization parameter [7].

The algebraic iterative methods, in their basic form as well as their block versions, lend themselves very well to implementations that utilize GPUs and other hardware accelerators, and where the coefficient matrix is never stored; rather, the matrix-vector multiplications are computed on the fly. This has led to an implementation paradigm that is routinely used in software packages for CT (such as ASTRA [22]

*Submitted to the journal's Methods and Algorithms for Scientific Computing section August 10, 2018; accepted for publication (in revised form) April 4, 2019; published electronically June 4, 2019.
<http://www.siam.org/journals/sisc/41-3/M120644.html>

Funding: This work was partially supported by Advanced Grant 291405 from the European Research Council. The work of the third author was supported by an NWO Vidi research grant.

[†]Department of Applied Mathematics and Computer Science, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark (yido@dtu.dk, <http://www.compute.dtu.dk/~yido/>, pcha@dtu.dk, <http://www.compute.dtu.dk/~pcha/>, nabr@dtu.dk, <http://www.compute.dtu.dk/~nabr/>).

[‡]Department of Mathematics and Computer Science, TU Eindhoven, 5600 MB Eindhoven, The Netherlands (M.E.Hochstenbach@tue.nl, <http://www.win.tue.nl/~hochsten/>).

and TIGRE [1]), namely, to focus on the computational speed of the matrix-vector multiplication. However, this introduces a convergence issue that has largely been ignored.

To set the stage, we formulate the noise-free problem as

$$(1.1) \quad A\bar{x} = \bar{b}, \quad A \in \mathbb{R}^{m \times n},$$

where the vectors \bar{x} and \bar{b} represent the exact image and the noise-free data, while A represents the forward model—known as the (forward) projection in CT where both $m \geq n$ and $m < n$ are common. The multiplication with A^T , the transpose of A , is known in CT as the backprojection. These two operations form the computational core of any algebraic iterative method and therefore—to optimize for computational speed—the software developers often choose different discretization schemes, and different model approximations, for these operations [24]. Consequently, in such software the backprojection corresponds to multiplication with a matrix \hat{A}^T that is not the exact transpose of A . We refer to this situation as having an *unmatched projector/backprojector pair*, and we call \hat{A}^T the *unmatched transpose*.

It appears that very little attention has been given to iterations based on such unmatched pairs; see [25] for an early reference and [2], [13] for two more recent ones. The latter paper has introduced a methodology for analyzing such iterations and formulated conditions for convergence, and it has been shown that an unmatched projector/backprojector pair deteriorates the best possible solution at the point of semiconvergence.

In the common situation of an unmatched projector/backprojector pair where the convergence criterion from [2] is not satisfied, the iterations will fail to converge for noise-free data (although some kind of semiconvergence may be observed experimentally for noisy data). In this paper we show that a small and cost-efficient modification of the basic algorithm can guarantee convergence to a solution of a slightly perturbed problem. This ensures that the fast implementations of the forward projections and backprojections can still be used without sacrificing convergence. Moreover, to provide a theoretical foundation we extend the convergence and perturbation analysis from [2] to the modified algorithm.

Our paper is organized as follows. Section 2 sets the stage by summarizing convergence results for a generic iterative algorithm (proposed in [2]) that allows an unmatched transpose. Section 3 introduces the modified algorithm, gives the associated convergence conditions, and discusses the perturbation theory for the underlying problem. The modified algorithm is based on the introduction of a shift parameter, and in section 4 we discuss how to efficiently estimate the leftmost eigenvalue of a certain matrix, which defines this shift. Finally, in section 5 we give numerical examples that illustrate the new method for solving inverse problems, followed by some conclusions in section 6.

We use the following notation: $\|\cdot\|$ denotes the vector and matrix 2-norm, $\mathcal{R}(A)$ and $\mathcal{N}(A)$ are the range and null space of A , respectively, and we split a complex eigenvalue λ_j into its real and imaginary parts denoted by $\operatorname{Re}(\lambda_j)$ and $\operatorname{Im}(\lambda_j)$, respectively. For a vector x , we use x^H for its conjugate transpose.

In section 2 we use notation and concepts associated with oblique projections and oblique pseudoinverses to obtain compact expression that would otherwise be quite lengthy. We refer to [9] for details and geometric interpretations of these quantities in relation to inverse problems. Assume we are given two complementary subspaces \mathcal{X} and \mathcal{Y} of \mathbb{R}^m that intersect trivially, and matrices X , Y , and Y_0 such that

$$\mathcal{X} = \mathcal{R}(X), \quad \mathcal{Y} = \mathcal{R}(Y), \quad \mathcal{Y}^\perp = \mathcal{R}(Y_0).$$

Then the $m \times m$ matrix $P_{\mathcal{X}, \mathcal{Y}}$ denotes the oblique projector onto \mathcal{X} along \mathcal{Y} which satisfies

$$\forall x \in \mathcal{X} : P_{\mathcal{X}, \mathcal{Y}} x = x, \quad \forall y \in \mathcal{Y} : P_{\mathcal{X}, \mathcal{Y}} y = 0, \quad \forall z \in \mathbb{R}^m : P_{\mathcal{X}, \mathcal{Y}} z \in \mathcal{X},$$

and the projection matrix can be written as

$$(1.2) \quad P_{\mathcal{X}, \mathcal{Y}} = X(Y_0^T X)^\dagger Y_0^T,$$

where † denotes the Moore–Penrose pseudoinverse. Moreover, if $X \in \mathbb{R}^{m \times n}$, then the $n \times m$ matrix $X_{\mathcal{Y}}^\dagger$ denotes the oblique pseudoinverse of X along \mathcal{Y} , given by

$$(1.3) \quad X_{\mathcal{Y}}^\dagger = \begin{cases} X^\dagger P_{\mathcal{X}, \mathcal{Y}} = (Y_0^T X)^\dagger Y_0^T, & m \geq n, \\ P_{\mathcal{Y}, \mathcal{N}(X)} X^\dagger = Y(XY)^\dagger, & m \leq n. \end{cases}$$

The case $m \geq n$ requires $\mathcal{R}(X)$ and $\mathcal{N}(Y)$ to be complementary, while the case $m \leq n$ requires $\mathcal{N}(X)$ and $\mathcal{R}(Y)$ to be complementary. If $\mathcal{Y} = \mathcal{X}^\perp$, then $P_{\mathcal{X}, \mathcal{Y}}$ is the orthogonal projector on \mathcal{X} and $X_{\mathcal{Y}}^\dagger$ is the ordinary pseudoinverse.

2. The BA ITERATION. When we consider noisy data $b = \bar{b} + e$, where the vector e represents the perturbation, then it is common to compute a (weighted) least squares solution. In the simplest case with unit weights we can compute the solution by means of the Landweber iteration (or gradient descent method) with initial guess $x^0 = 0$:

$$(2.1) \quad x^{k+1} = x^k + \omega A^T (b - A x^k), \quad k = 0, 1, 2, \dots,$$

where ω is a relaxation parameter satisfying $0 < \omega < 2 / \|A^T A\|$.

To analyze the behavior of this and similar algebraic iterative methods with an unmatched transpose, we follow [2] and consider the BA ITERATION defined by

$$(2.2) \quad x^{k+1} = x^k + \omega B (b - A x^k), \quad k = 0, 1, 2, \dots,$$

where different choices of the $n \times m$ matrix B give unmatched-transpose versions of known iterative methods. For example, B can be an unmatched transpose \hat{A}^T for Landweber's method, or B can be an unmatched approximation to $A^T \text{diag}(A A^T)^{-1}$ for Cimmino's method; see [10] for an overview of methods.

The convergence of the BA ITERATION is governed by the (complex) eigenvalues λ_j of the matrix BA : (2.2) converges if and only if the relaxation parameter ω and all nonzero λ_j satisfy

$$(2.3) \quad 0 < \omega < \frac{2 \operatorname{Re}(\lambda_j)}{|\lambda_j|^2} \quad \text{and} \quad \operatorname{Re}(\lambda_j) > 0;$$

see [2, Prop. 3.2] for details. Note that this specializes to the standard condition when $B = A^T$.

We will now investigate when the BA ITERATION (2.2) has a unique fixed point. From the definition of the BA ITERATION (2.2) with $x^0 = 0$ it follows that any fixed point x^* must satisfy $BAx^* = Bb$. Moreover, it is also clear from (2.2) that all iterates $x^k \in \mathcal{R}(B)$; in particular this holds for x^* . Therefore, we can write any fixed

point in the form $x^* = By$ for some vector $y \in \mathbb{R}^m$. (This vector y may not be unique, as one can add an arbitrary component $z \in \mathcal{N}(B)$, but this is irrelevant for what is to follow.) Inserting $x^* = By$ we obtain an equation for y :

$$(2.4) \quad B A B y = B b .$$

Here, BA is an operator from $\mathcal{R}(B)$ to itself. Recall that $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times m}$, where both $m \geq n$ and $m < n$ are possible in CT applications.

From (2.4) it follows there is a unique fixed point $x \in \mathcal{R}(B)$ if and only if one of the following eight equivalent conditions holds.

PROPOSITION 2.1. *Consider the two matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times m}$ with ranks $r_A, r_B \leq \min\{m, n\}$ and with the singular value decompositions (SVDs) $A = U_A \Sigma_A V_A^T$ and $B^T = U_B \Sigma_B V_B^T$. The following statements are equivalent:*

- (i) $BA : \mathcal{R}(B) \rightarrow \mathcal{R}(B)$ is nonsingular (meaning that $BAz = 0$ and $z \in \mathcal{R}(B)$ imply that $z = 0$);
- (ii) for every $b \in \mathbb{R}^m$, the equation $BAx = Bb$ has a unique solution $x \in \mathcal{R}(B)$;
- (iii) $\mathcal{R}(B) \cap \mathcal{N}(BA) = \{0\}$;
- (iv) $\mathcal{N}(BAB) = \mathcal{N}(B)$;
- (v) $\mathcal{R}(BAB) = \mathcal{R}(B)$;
- (vi) $\text{rank}(BAB) = \text{rank}(B)$;
- (vii) A is nonsingular on $\mathcal{R}(B)$ and B is nonsingular on $\mathcal{R}(AB)$;
- (viii) $\mathcal{R}(B) \cap \mathcal{N}(A) = \{0\}$ and $\mathcal{R}(AB) \cap \mathcal{N}(B) = \{0\}$.

Proof. The equivalences follow relatively straightforwardly from the (dimensions of) nullspaces and ranges of A , B , BA , and AB . For (iv) we have $\mathcal{N}(BAB) \supseteq \mathcal{N}(B)$ with equality if and only if (i) holds. For (v) one has $\mathcal{R}(BAB) \subseteq \mathcal{R}(B)$ with equality if and only if (i) holds, where the ranks in (vi) are the dimensions of the subspaces of (v). A nonzero vector in $\mathcal{R}(B)$ cannot be mapped to zero by the consecutive action of BA , which is stated in (vii) and (viii). \square

We assume from now on that there is a unique solution, and the next theorem provides specific expressions for this fixed point. We will see that, even in the absence of noise, the fixed point of (2.2) is *not* the exact solution \bar{x} . One way to understand this—following the discussion in [2]—is by the fact that the unmatched normal equations $BAx = Bb$ may be viewed as an oblique projection of $Ax = b$, instead of the common orthogonal projection underlying the normal equations $A^T Ax = A^T b$.

THEOREM 2.2. *Assume that A and B satisfy the criteria in Proposition 2.1. Then the fixed point x^* of the BA ITERATION (2.2) with starting vector $x^0 = 0$ can be expressed in three ways,*

$$(2.5) \quad x^* = (BA)_{\mathcal{R}(B)}^\dagger Bb = B(A B)_{\mathcal{N}(B)}^\dagger b = P_{\mathcal{R}(B), \mathcal{N}(BA)} A_{\mathcal{N}(B)}^\dagger b ,$$

and for noise-free data $\bar{b} = A\bar{x}$ the fixed point is given by

$$(2.6) \quad \bar{x}^* = P_{\mathcal{R}(BA), \mathcal{N}(BA)} \bar{x} .$$

If $m \geq n$ and A and B have full rank, then $x^* = (BA)^{-1} Bb$ and $\bar{x}^* = \bar{x}$.

Proof. By writing the fixed point as $x^* = By$ with $y \in \mathbb{R}^m$ it follows from (2.4) that $x^* = B(BAB)^\dagger Bb$, and the first two expressions in (2.5) are obtained by recognizing that $B((BA)B)^\dagger = (BA)_{\mathcal{R}(B)}^\dagger$ and $(B(AB))^\dagger B = (AB)_{\mathcal{N}(B)}^\dagger$; cf. (1.3). We now introduce the SVD

$$BA = U \Sigma V^T , \quad \Sigma \in \mathbb{R}^{p \times p} , \quad U, V \in \mathbb{R}^{n \times p} ,$$

where $p = \text{rank}(BA)$. Inserting this in (2.4) we get $U \Sigma V^T B y = B b$, and by multiplying from the left with $\Sigma^{-1} U^T$ we obtain $V^T B y = \Sigma^{-1} U^T B b$. The solution y of minimum norm is given by

$$y = (V^T B)^\dagger \Sigma^{-1} U^T B b,$$

and hence

$$x^* = B (V^T B)^\dagger \Sigma^{-1} U^T B b = B (V^T B)^\dagger V^T V \Sigma^{-1} U^T B b = B (V^T B)^\dagger V^T (B A)^\dagger B b.$$

By recognizing $P_{\mathcal{R}(B), \mathcal{N}(BA)} = B (V^T B)^\dagger V^T$ as the oblique projector onto $\mathcal{R}(B)$ along $\mathcal{N}(V^T) = \mathcal{N}(BA)$ (cf. (1.2)) and $A_{\mathcal{N}(B)}^\dagger = (B A)^\dagger B$ as the oblique pseudoinverse of A along $\mathcal{N}(B)$ (cf. (1.3)) we obtain the third expression in (2.5).

For the special case $b = \bar{b} = A \bar{x}$ the fixed point satisfies $\bar{x}^* \in \mathcal{R}(BA)$, and hence we can write it as $\bar{x}^* = B A \bar{y}$ for some vector $\bar{y} \in \mathbb{R}^n$. According to $B A \bar{x}^* = B \bar{b} = B A \bar{x}$, we obtain

$$\bar{x}^* = B A (B A B A)^\dagger B A \bar{x} = B A (((B A)^T)^T B A)^\dagger ((B A)^T)^T \bar{x},$$

and we recognize $P_{\mathcal{R}(BA), \mathcal{N}(BA)} = B A (((B A)^T)^T B A)^\dagger ((B A)^T)^T$ as the oblique projector onto $\mathcal{R}(BA)$ along $\mathcal{R}((BA)^T)^\perp = \mathcal{N}(BA)$. \square

The sensitivity of the fixed point to perturbations of the right-hand side can be characterized as follows.

COROLLARY 2.3. *Let \bar{x}^* and x^* denote the fixed point of the BA ITERATION when applied to the noise-free data \bar{b} and the noisy data $b = \bar{b} + e$, respectively. Then*

$$\|x^* - \bar{x}^*\| \leq \|P_{\mathcal{R}(B), \mathcal{N}(BA)}\| \|A_{\mathcal{N}(B)}^\dagger\| \|e\|.$$

If $m \geq n$ and A and B have full rank, then

$$\|x^* - \bar{x}^*\| \leq \|A_{\mathcal{N}(B)}^\dagger\| \|e\|.$$

When $B = A^T$ then the oblique pseudoinverse $A_{\mathcal{N}(B)}^\dagger$ is the ordinary pseudoinverse A^\dagger and we obtain the traditional least-squares perturbation bound.

Proof. The first bound is a direct consequence of (2.5), and the second bound follows from the fact that $P_{\mathcal{R}(B), \mathcal{N}(BA)} = I$ when $m \geq n = \text{rank}(A)$. \square

The conclusions to be drawn from the analysis in this section are that the conditions for the existence of a fixed point of the BA ITERATION are rather strict and that the fixed point is potentially very sensitive to data errors since the matrix A is ill-conditioned in inverse problems. Moreover, it is very difficult to check the existence conditions in a practical application, and it appears that they are very often violated in the available software systems. This motivates the development of a modified iterative method that is always guaranteed to have a fixed point, which we introduce and analyze in the rest of this paper.

3. A modified algorithm. In our numerical studies with ASTRA and other software packages for CT, we have found that very often the convergence condition in (2.3) is violated in that BA has one or more eigenvalues with negative real part. As a consequence the iteration has no fixed point and the typical situation is that the iterates x^k , after some iterations, start to diverge. We illustrate this in section 5.

3.1. The SHIFTED BA ITERATION. To remedy this nonconvergence issue, we propose a modified version of the BA ITERATION that has guaranteed convergence and whose fixed point approximates the exact solution \bar{x} . In addition, the modified method should exhibit semiconvergence properties similar to the BA ITERATION. We refer to the modified algorithm as the SHIFTED BA ITERATION, and it guarantees convergence of the iterations for appropriate choices of the two parameters $\alpha > 0$ and $\omega > 0$:

$$(3.1) \quad x^{k+1} = (1 - \alpha\omega)x^k + \omega B(b - Ax^k), \quad k = 0, 1, 2, \dots$$

This scheme is motivated by the Tikhonov problem,

$$(3.2) \quad x_\alpha = \arg \min_x \{ \|Ax - b\|^2 + \alpha \|x\|^2 \} = (A^T A + \alpha I)^{-1} A^T b,$$

for which a gradient descent step takes the form

$$x^{k+1} = x^k - \omega (A^T(b - Ax^k) + \alpha x^k) = (1 - \alpha\omega)x^k + \omega A^T(b - Ax^k).$$

Hence, if $B = A^T$, then with a properly chosen ω the iteration (3.1) converges to a Tikhonov solution x_α . Below we study the convergence properties of (3.1) with $B \neq A^T$.

The matrix that governs the iterations for the SHIFTED BA ITERATION (3.1) is $BA + \alpha I$ with I as the identity matrix, whose eigenvalues are $\lambda_j + \alpha$ (where λ_j are the eigenvalues of BA). Our key idea is that by a proper choice of the additional positive parameter α we can ensure that all these eigenvalues have a nonnegative real part—thus ensuring convergence. The shift needs to be just large enough that $\operatorname{Re}(\lambda_j) + \alpha > 0$ for those $\lambda_j \neq -\alpha$. At the same time, if α is small, then the fixed point will be an approximation to the exact solution \bar{x} . Hence, in contrast to the BA ITERATION the shifted version has a unique fixed point that is always attained for both noisy and noise-free data. Our new approach can therefore be viewed as a modification where both a regularization term and semiconvergence of the iterations are used for noisy data.

We note that the SHIFTED BA ITERATION is mathematically equivalent to applying the BA ITERATION to the augmented matrices and vector

$$(3.3) \quad \begin{bmatrix} A \\ \sqrt{\alpha}I \end{bmatrix}, \quad [B, \sqrt{\alpha}I], \quad \begin{bmatrix} b \\ 0 \end{bmatrix}.$$

A similar idea was used in [3, section 3.2], for the case $B = A^T$, to perform convergence analysis for the case $\operatorname{rank}(A) < n$. According to [2, Prop. 3.1 and 3.2], with the augmented matrices and vector defined in (3.3) we obtain the following convergence criterion for the SHIFTED BA ITERATION.

THEOREM 3.1. *Let λ_j denote those eigenvalues of BA that are different from $-\alpha$. Then the SHIFTED BA ITERATION (3.1) converges to a fixed point if and only if α and ω satisfy*

$$(3.4) \quad 0 < \omega < 2 \frac{\operatorname{Re}(\lambda_j) + \alpha}{|\lambda_j|^2 + \alpha(\alpha + 2\operatorname{Re}(\lambda_j))} \quad \text{and} \quad \operatorname{Re}(\lambda_j) + \alpha > 0.$$

Proof. Replacing the matrices B and A in the BA ITERATION with the augmented ones from (3.3), we define $C = BA + \alpha I$ and $T = I - \omega C$. Then T is the iteration matrix for the SHIFTED BA ITERATION (3.1), i.e.,

$$(3.5) \quad x^{k+1} = Tx^k + \omega Bb,$$

and any fixed point x_α^* of (3.1) satisfies the equation

$$(3.6) \quad C x_\alpha^* = B b .$$

Let $P = P_{\mathcal{R}(C^T)}$ be the orthogonal projector onto $\mathcal{R}(C^T) = \mathcal{N}(C)^\perp$. In view of the presence of the projection P , we consider the new coordinate system given by the orthogonal matrix $\begin{bmatrix} N & R \end{bmatrix}$, where N and R are matrices with orthonormal columns spanning $\mathcal{N}(C)$ and $\mathcal{R}(C^T)$, respectively. We examine PT in the new coordinates where $\begin{bmatrix} N & R \end{bmatrix}^T P \begin{bmatrix} N & R \end{bmatrix} \begin{bmatrix} N & R \end{bmatrix}^T T \begin{bmatrix} N & R \end{bmatrix}$ takes the form

$$\begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & -\omega N^T C R \\ 0 & I - \omega R^T C R \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & I - \omega R^T C R \end{bmatrix} .$$

For the first block row, it holds that

$$Tx = x \quad \Leftrightarrow \quad Cx = 0 \quad \Leftrightarrow \quad x \in \mathcal{N}(C) \quad \Leftrightarrow \quad -\alpha \text{ is an eigenvalue of } BA .$$

This shows the eigenvalue 1 of T is associated with the eigenspace $\mathcal{N}(C)$. We see that because of the projector operator, this eigenvalue, which corresponds to eigenvalues of BA equal to $-\alpha$, is irrelevant for convergence. From the second block row, it suffices to consider PT as operator $\mathcal{R}(C^T) \rightarrow \mathcal{R}(C^T)$, where it holds that $PT = T$. Combining these facts, we conclude that it is enough to consider the eigenvalues $1 - \omega(\lambda_j + \alpha)$ of T , where λ_j is not equal to $-\alpha$.

Then, applying the results in [2, Prop. 3.1 and 3.2] to the augmented matrices and vector (3.3), we obtain the sufficient and necessary condition of convergence with respect to α and ω . \square

When we have convergence, a fixed point x_α^* of the SHIFTED BA ITERATION satisfies

$$(3.7) \quad x_\alpha^* = (1 - \alpha\omega) x_\alpha^* + \omega B(b - Ax^*) \quad \Longleftrightarrow \quad (BA + \alpha I) x_\alpha^* = Bb$$

and, similar to Theorem 2.2, we can characterize this fixed point as follows.

THEOREM 3.2. *Assume that $BA + \alpha I$ is nonsingular. The fixed point of the SHIFTED BA ITERATION (3.1) applied to b , with starting vector $x^0 = 0$ and $\alpha > 0$, satisfies*

$$(3.8) \quad x_\alpha^* = (BA + \alpha I)^{-1} Bb = B(AB + \alpha I)^{-1} b, \quad x_\alpha^* \in \mathcal{R}(B) .$$

For noise-free data $\bar{b} = A\bar{x}$ the fixed point \bar{x}_α^* satisfies

$$\bar{x}_\alpha^* = (BA + \alpha I)^{-1} BA \bar{x}, \quad \bar{x}_\alpha^* \in \mathcal{R}(BA), \quad \bar{x} - \bar{x}_\alpha^* = \alpha (BA + \alpha I)^{-1} \bar{x} .$$

Proof. The first relation follows immediately from (3.7). To obtain the second relation we write $x_\alpha^* = By$ and note that $(BA + \alpha I)By = B(AB + \alpha I)y$. Hence y must satisfy

$$B((AB + \alpha I)y - b) = 0 ,$$

and therefore y has the general form with an arbitrary component in the null space of $B(AB + \alpha I)$ as well as B :

$$y = (AB + \alpha I)^{-1} b + z, \quad z \in \mathcal{N}(B) .$$

We note that $AB + \alpha I$ is nonsingular due to our assumption that $BA + \alpha I$ is nonsingular; cf. [11, Thm. 1.3.22]. Thus

$$x_\alpha^* = By = B((AB + \alpha I)^{-1}b + z) = B(AB + \alpha I)^{-1}b.$$

and it follows immediately that $x_\alpha^* \in \mathcal{R}(B)$. The first results for \bar{x}_α^* follows immediately from (3.8). To show the second result let BA have the eigendecomposition $BA = W \operatorname{diag}(\lambda_i) W^{-1}$; then $(BA + \alpha I)^{-1}BA = W \operatorname{diag}(\lambda_i/(\lambda_i + \alpha)) W^{-1}$ from which it follows that $\bar{x}_\alpha^* \in \operatorname{span}\{w_1, \dots, w_{\operatorname{rank}(BA)}\} = \mathcal{R}(BA)$. The third result follows from the relation

$$(BA + \alpha I)(\bar{x} - \bar{x}_\alpha^*) = (BA + \alpha I)(\bar{x} - (BA + \alpha I)^{-1}BA\bar{x}) = \alpha \bar{x}. \quad \square$$

Note that the results in (3.8) can also be derived by applying the augmented matrices and vector defined in (3.3) to (2.5).

To summarize, we formulated the convergence conditions for the SHIFTED BA ITERATION in terms of the shift α and the relaxation parameter ω , and we gave explicit expressions for the fixed point of this iterative method.

3.2. First-order perturbation analysis. Recall that the fixed point x_α^* of the SHIFTED BA ITERATION is the Tikhonov solution in (3.2) when $B = A^T$. Following [2] it is natural to give a general perturbation analysis of the Tikhonov problem when different perturbations are introduced in the matrices A and A^T in the corresponding normal equations $A^T A x = A^T b$. A special instance of this analysis is when B is an unmatched transpose of A , and where our analysis lets us bound the difference between the fixed point x_α^* and the exact solution \bar{x} .

We introduce the perturbed quantities

$$(3.9) \quad \tilde{A} = A + E_A, \quad \hat{A}^T = A^T + E_{A^T}, \quad b = \bar{b} + e,$$

with $E_A \in \mathbb{R}^{m \times n}$, $E_{A^T} \in \mathbb{R}^{n \times m}$, and $e \in \mathbb{R}^m$. Moreover we define \tilde{x}_α as the solution to the REGULARIZED UNMATCHED NORMAL EQUATIONS

$$(\hat{A}^T \tilde{A} + \alpha I)x = \hat{A}^T b.$$

We want to compare \tilde{x}_α to the exact solution \bar{x} . To do this, we introduce the Tikhonov solution $\bar{x}_\alpha = (A^T A + \alpha I)^{-1} A^T \bar{b}$ to the unperturbed problem (3.2).

We then split the error into a *perturbation error* $\tilde{x}_\alpha - \bar{x}_\alpha$ and a *regularization error* $\bar{x}_\alpha - \bar{x}$ as follows:

$$(3.10) \quad \tilde{x}_\alpha - \bar{x} = (\tilde{x}_\alpha - \bar{x}_\alpha) + (\bar{x}_\alpha - \bar{x}).$$

This approach allows us to study the effect of the matrix and right-hand side perturbations in isolation from the effect that Tikhonov regularization has on a noise-free system.

THEOREM 3.3. *With the definitions in (3.9) and (3.10) we have the following first-order error bounds obtained by omitting higher-order terms:*

$$\begin{aligned} \|\tilde{x}_\alpha - \bar{x}_\alpha\| &\lesssim \frac{1}{2\sqrt{\alpha}} \|e\| + \frac{1}{2\sqrt{\alpha}} \|E_A \bar{x}_\alpha\| + \frac{1}{\alpha} \|E_{A^T} (\bar{b} - A \bar{x}_\alpha)\|, \\ \frac{\|\tilde{x}_\alpha - \bar{x}_\alpha\|}{\|\bar{x}_\alpha\|} &\lesssim \frac{\|A\|}{2\sqrt{\alpha}} \frac{\|e\|}{\|A \bar{x}_\alpha\|} + \frac{\|A\|}{2\sqrt{\alpha}} \frac{\|E_A\|}{\|A\|} + \frac{\|A\|^2}{\alpha} \frac{\|E_{A^T}\|}{\|A\|} \frac{\|\bar{b} - A \bar{x}_\alpha\|}{\|A \bar{x}_\alpha\|}. \end{aligned}$$

Proof. Let $\tilde{x}_\alpha = \bar{x}_\alpha + \delta x_\alpha$ and consider the perturbed system

$$(3.11) \quad (\hat{A}^T \tilde{A} + \alpha I) (\bar{x}_\alpha + \delta x_\alpha) = \hat{A}^T b .$$

Moreover note that from (3.9) we have

$$\hat{A}^T b = (A^T + E_{A^T}) (\bar{b} + e) = A^T \bar{b} + A^T e + E_{A^T} b$$

and

$$\hat{A}^T \tilde{A} = (A^T + E_{A^T}) (A + E_A) = A^T A + E ,$$

where we have introduced

$$E = A^T E_A + E_{A^T} A + E_{A^T} E_A .$$

Inserting these equations in (3.11) we obtain

$$(A^T A + E + \alpha I) (\bar{x}_\alpha + \delta x_\alpha) = A^T \bar{b} + A^T e + E_{A^T} b ,$$

and rearranging we get

$$(A^T A + \alpha I) \bar{x}_\alpha + (A^T A + \alpha I) \delta x_\alpha = A^T \bar{b} + A^T e + E_{A^T} b - E \bar{x}_\alpha - E \delta x_\alpha .$$

Now using that $(A^T A + \alpha I) \bar{x}_\alpha = A^T \bar{b}$ and neglecting higher-order terms we get

$$\begin{aligned} (A^T A + \alpha I) \delta x_\alpha &\approx A^T e + E_{A^T} b - A^T E_A \bar{x}_\alpha - E_{A^T} A \bar{x}_\alpha \\ &= A^T (e - E_A \bar{x}_\alpha) + E_{A^T} (\bar{b} - A \bar{x}_\alpha) , \end{aligned}$$

which can also be obtained by using the augmented form in the proof of [2, Prop. 2.1], i.e., replacing A , E , and b with

$$\begin{bmatrix} A \\ \sqrt{\alpha} I \end{bmatrix}, \quad \begin{bmatrix} E \\ 0 \end{bmatrix}, \quad \begin{bmatrix} b \\ 0 \end{bmatrix} .$$

This leads to the bound

$$\begin{aligned} \|\delta x_\alpha\| &\lesssim \|(A^T A + \alpha I)^{-1} A^T\| \|(e - E_A \bar{x}_\alpha)\| + \\ &\quad \|(A^T A + \alpha I)^{-1}\| \|E_{A^T} (\bar{b} - A \bar{x}_\alpha)\| \\ &\leq \frac{1}{2\sqrt{\alpha}} \|e\| + \frac{1}{2\sqrt{\alpha}} \|E_A \bar{x}_\alpha\| + \frac{1}{\alpha} \|E_{A^T} (\bar{b} - A \bar{x}_\alpha)\| , \end{aligned}$$

where we use that, with σ_i being the i th singular value of A ,

$$\|(A^T A + \alpha I)^{-1} A^T\| = \max_i \frac{\sigma_i}{\sigma_i^2 + \alpha} \leq \frac{1}{2\sqrt{\alpha}}$$

and

$$\|(A^T A + \alpha I)^{-1}\| = \max_i \frac{1}{\sigma_i^2 + \alpha} \leq \frac{1}{\alpha} .$$

For the relative error we get

$$\begin{aligned} \frac{\|\delta x_\alpha\|}{\|\bar{x}_\alpha\|} &\lesssim \frac{1}{2\sqrt{\alpha}} \frac{\|e\|}{\|\bar{x}_\alpha\|} + \frac{1}{2\sqrt{\alpha}} \frac{\|E_A \bar{x}_\alpha\|}{\|\bar{x}_\alpha\|} + \frac{1}{\alpha} \frac{\|E_{A^T} (\bar{b} - A \bar{x}_\alpha)\|}{\|\bar{x}_\alpha\|} \\ &\leq \frac{\|A\|}{2\sqrt{\alpha}} \frac{\|e\|}{\|A \bar{x}_\alpha\|} + \frac{\|A\|}{2\sqrt{\alpha}} \frac{\|E_A\|}{\|A\|} + \frac{\|A\|^2}{\alpha} \frac{\|E_{A^T}\|}{\|A\|} \frac{\|\bar{b} - A \bar{x}_\alpha\|}{\|A \bar{x}_\alpha\|} , \end{aligned}$$

where we used that

$$\|A \bar{x}_\alpha\| \leq \|A\| \|\bar{x}_\alpha\| \iff \frac{1}{\|\bar{x}_\alpha\|} \leq \frac{\|A\|}{\|A \bar{x}_\alpha\|} .$$

□

To complete the analysis we need to bound the regularization error $\bar{x}_\alpha - \bar{x}$ associated with the noise-free system. To obtain a useful bound we need to incorporate the fact that we solve a discretized inverse problem. This is done in the following theorem from [5] (see also [7, Thm. 4.5.1]).

THEOREM 3.4. *Introduce SVD of A as $A = \sum_{i=1}^{\min(m,n)} u_i \sigma_i v_i^T$ and assume that the noise-free right-hand side \bar{b} is given by the model*

$$u_i^T \bar{b} = \sigma_i^\nu, \quad i = 1, 2, \dots, \min(m, n),$$

in which $\nu \geq 0$ is a model parameter that controls the decay of these coefficients. Then

$$\frac{\|\bar{x}_\alpha - \bar{x}\|}{\|\bar{x}\|} \leq \begin{cases} \sqrt{n}, & 0 \leq \nu < 1, \\ \sqrt{n} \left(\frac{\sqrt{\alpha}}{\|A\|} \right)^{\nu-1}, & 1 \leq \nu < 3, \\ \sqrt{n} \left(\frac{\sqrt{\alpha}}{\|A\|} \right)^2, & 3 \leq \nu. \end{cases}$$

In practice the Tikhonov regularization parameter α is always less than $\|A\|^2$ [7]. This theorem then says that the noise-free problem must satisfy the discrete Picard condition for Tikhonov regularization to produce a useful result—which is, of course, the case for the imaging problems we have in mind.

To summarize our results, the shift parameter α plays the following roles. The regularization error decreases as α decreases, and if the noise-free data satisfies the discrete Picard condition (as we expect), then a small nonzero α has little influence on the regularization error. On the other hand, as α decreases, then the perturbation error increases. We want to use an α just large enough to ensure convergence.

4. Eigenvalue estimation. The motivation behind the SHIFTED BA ITERATION is to introduce a small positive shift parameter α , just large enough to ensure that all the shifted eigenvalues have a positive real part, i.e., $\operatorname{Re}(\lambda_j) + \alpha > 0$. To turn this principle into an efficient working algorithm, we therefore need to be able to estimate the leftmost eigenvalue λ_{lm} of BA , the eigenvalue with the minimal real part. If $\operatorname{Re}(\lambda_{\text{lm}}) > 0$, then we just use the BA ITERATION; otherwise we use the SHIFTED BA ITERATION with α slightly larger than $|\operatorname{Re}(\lambda_{\text{lm}})|$. (In view of Theorem 3.1, we might theoretically even take α exactly equal to this quantity, but this is not important in practice, since the approximation to the smallest real part of the leftmost eigenvalue will usually be an upper bound.) It is important to note that we only have actions with A and B at our disposal, and no actions with A^T , B^T , or exact shift-and-invert transformations, are possible.

Various approaches have been investigated by Meerbergen and coauthors for the rightmost eigenvalue of a matrix C (or, equivalently, the leftmost of $-C$). Several of these are “matrix-free,” which means that only matrix-vector products are necessary. An approach based on Chebyshev polynomials has been proposed in [14]. In [15], the search space is expanded by an approximation to $\exp(C)z$, where z is the current Ritz vector. However, these methods usually take a considerable number of matrix-vector multiplications to expand the search space by one vector.

Since the leftmost eigenvalue is an eigenvalue located at the exterior of the spectrum, Krylov based methods are often well suited. Stewart’s Krylov–Schur method [20] is one of the most popular methods to compute such eigenvalues. This method is essentially equivalent to implicitly restarted Arnoldi [19], as, for instance, implemented in `eigs` in MATLAB, but has a particularly elegant and easy-to-understand

implementation. In our experiments, a custom-made implementation of the Krylov–Schur method proved to be, on average, a factor 1.2–1.3 faster than `eigs` in terms of matrix-vector multiplications. We give pseudocode for the Krylov–Schur method in Algorithm 1.

Algorithm 1. Krylov–Schur for the leftmost eigenvalue.

Input: Minimal and maximal subspace dimensions $\underline{\ell} < \bar{\ell}$, starting vector v_1 , tolerance `tol`, functions to perform matrix-vector multiplications with A and B .

Output: A pair (θ, v) that approximates the leftmost eigenpair $(\lambda_{\text{lm}}, v_{\text{lm}})$, with $\|(BA - \theta I)v\| \leq \text{tol}$.

- 1: Form the Krylov decomposition $BAV_{\bar{\ell}} = V_{\bar{\ell}}H_{\bar{\ell}} + h_{\bar{\ell}+1, \bar{\ell}}v_{\bar{\ell}+1}f_{\bar{\ell}}^H$
 - 2: **for** $k = 1, 2, \dots$
 - 3: Extract Schur pairs (θ_j, c_j) from $H_{\bar{\ell}}$ with $j = 1, \dots, \bar{\ell}$,
 sorted on nondecreasing real part
 - 4: **if** $|h_{\bar{\ell}+1, \bar{\ell}}f_{\bar{\ell}}^H c_1| \leq \text{tol}$
 - 5: Accept $\theta = \theta_1$ with leftmost Schur vector $v = V_{\bar{\ell}}c_1$, **stop**
 - 6: **end**
 - 7: Truncate decomposition to dimension $\underline{\ell}$ by selecting leftmost Schur vectors
 - 8: Expand the Krylov decomposition to dimension $\bar{\ell}$
 - 9: **end**
-

This algorithm uses the Krylov decomposition from [20] which is a generalization of the Arnoldi decomposition and which may have complex factors. In line 1, the first Krylov decomposition has $f_{\bar{\ell}} = e_{\bar{\ell}}$, the $\bar{\ell}$ th standard basis vector. In subsequent Krylov decompositions this vector is changed, as indicated below. In line 4, we exploit the fact that

$$\begin{aligned}(BA - \theta I)v &= (BA - \theta I)V_{\bar{\ell}}c_1 \\ &= V_{\bar{\ell}}(H_{\bar{\ell}}c_1 - \theta_1 c_1) + h_{\bar{\ell}+1, \bar{\ell}}v_{\bar{\ell}+1}f_{\bar{\ell}}^H c_1 = h_{\bar{\ell}+1, \bar{\ell}}v_{\bar{\ell}+1}f_{\bar{\ell}}^H c_1.\end{aligned}$$

As described in [20], the restart in lines 7–8 is performed as follows. Suppose $H_{\bar{\ell}} = QSQ^H$ is the Schur decomposition with the most relevant Schur vectors (corresponding to the leftmost Ritz values in S) sorted in the beginning of Q . Then the method is restarted with $V_{\underline{\ell}} := V_{\bar{\ell}}Q_{1:\underline{\ell}}$ instead of $V_{\bar{\ell}}$; $S_{\underline{\ell}}$ instead of $H_{\bar{\ell}}$; and $Q_{1:\underline{\ell}}^H f_{\bar{\ell}}$ instead of $f_{\bar{\ell}}$. We present numerical experiments with this method in section 5.2.

An alternative approach that uses inexact shift-and-invert operators by carrying out inner iterations is Jacobi–Davidson [18]. This inexact inner-outer type of method may be worthwhile when the leftmost eigenvalue is not well separated from neighboring eigenvalues. In our applications this does not seem the case, and Jacobi–Davidson performs usually worse than Krylov–Schur. In addition, it is not obvious to generate a preconditioner for a shifted version of BA , which would be very helpful for Jacobi–Davidson.

In our experiments with examples from CT, we find that the matrix BA is often close to normal (in fact, even close to symmetric; cf. section 5.2). For such problems, another alternative approach to approximate the leftmost eigenvalue is the following. Let

$$W(C) = \{z^H C z : z \in \mathbb{C}^n, \|z\| = 1\}$$

be the field of values (or numerical range) of C . Then we can expect the quantity

$$\nu(BA) := \min \operatorname{Re}(W(BA))$$

to be close to the leftmost eigenvalue of BA . This $\nu(BA)$ would in principle be relatively easy to approximate, since it is equal to $\frac{1}{2} \min \lambda(BA + (BA)^T)$, which results in computing an exterior eigenvalue of a symmetric eigenproblem. This would mean that we can use a symmetric version of Krylov–Schur, which saves roughly half of the reorthogonalization costs.

Unfortunately, in our applications we do not have the action with A^T or B^T , and the described approach is not an option. However, as an alternative, we can still approximate $\nu(BA)$ by

$$\nu(BA) \approx \nu(H_{\bar{\ell}}),$$

where $H_{\bar{\ell}}$ is the matrix in the Krylov decomposition obtained with BA after several iterations. The computation of this quantity only requires the known $H_{\bar{\ell}}^H$ and therefore bypasses the need of the transposes of A and B . Although there are usually no error bounds for this type of approximation, it may in practice be of very good quality. This algorithm is summarized below.

Algorithm 2. A field of values approximation for the leftmost eigenvalue.

Input: Minimal and maximal subspace dimensions $\underline{\ell} < \bar{\ell}$, starting vector v_1 , maximum iterations **maxit**, functions to perform actions with A and B .

Output: θ , the leftmost point of a projected field of values, which approximates the leftmost point of $W(BA)$.

- 1: Form the Krylov decomposition $BAV_{\bar{\ell}} = V_{\bar{\ell}}H_{\bar{\ell}} + h_{\bar{\ell}+1,\bar{\ell}}v_{\bar{\ell}+1}f_{\bar{\ell}}^H$
 - 2: **for** $k = 1, 2, \dots, \text{maxit}$
 - 3: Extract Schur pairs (θ_j, c_j) from $H_{\bar{\ell}}$ with $j = 1, \dots, \bar{\ell}$,
 sorted on nondecreasing real part
 - 4: Truncate decomposition to dimension $\underline{\ell}$ by selecting leftmost Schur vectors
 - 5: Expand the Krylov decomposition to dimension $\bar{\ell}$
 - 6: **end**
 - 7: Accept $\theta = \min \text{Re}W(H_{\bar{\ell}}) = \text{real part of leftmost eigenvalue of } \frac{1}{2}(H_{\bar{\ell}} + H_{\bar{\ell}}^H)$
-

Note that in a type of method as in Algorithm 2, there is typically no error estimate available; there is only a user-chosen parameter **maxit**, which often can be modest (see section 5). A main advantage of Algorithm 2 over Algorithm 1 is that it may be possible to stop the iterations improving the Krylov decomposition earlier, before the eigenpair of Algorithm 1 has converged to the desired tolerance. We test both approaches in the next section.

5. Numerical examples. We present numerical examples with two different test problems, in order to demonstrate the performance of our new algorithm. All computations are carried out in MATLAB.

5.1. Small illustrative test problem. The first test problem is quite small, with $m = n = 64$, such that we can explicitly compute the desired eigenvalues and other quantities that allow us to analyze the algorithms' performance related to the above theory. The matrix A is full, and it is generated by means of the function **regutm** from REGULARIZATION TOOLS [8] by which we can generate random test matrices with specified singular values, while the singular vectors have the characteristic spectral behavior of inverse problems [6]. We generate a well-conditioned matrix A_{well} and a more ill-conditioned matrix A_{ill} with the following distribution of singular values:

$$A_{\text{well}} : \text{logspace}(0, -2, m), \quad A_{\text{ill}} : \text{logspace}(0, -4, m).$$

We then generate a corresponding unmatched transpose $\hat{A}^T = A^T + E_{A^T}$ by adding random Gaussian elements to A^T ; all elements are from $\mathcal{N}(0, \sigma_A^2)$ where the variance is chosen such that $\|E_{A^T}\| / \|A\| = 0.05$. Both A and \hat{A}^T have full rank.

The exact solution \bar{x} is the one from the **shaw** test problem [8]; it is smooth with two humps. Then we generate the exact and noisy right-hand sides by

$$(5.1) \quad \bar{b} = A \bar{x}, \quad b = \bar{b} + e, \quad \|e\| / \|\bar{b}\| = 0.05,$$

where the random elements of e are Gaussian; all elements are from $\mathcal{N}(0, \sigma_b^2)$ where the variance scales the noise as desired.

We applied both the BA ITERATION and the SHIFTED BA ITERATION with $B = \hat{A}^T$ to these problems. For the BA ITERATION we use the relaxation parameter $\omega = 1.9 / \|\hat{A}^T A\|$ and for the SHIFTED BA ITERATION we use ω equal to the upper bound in (3.4) with the factor 2 replaced by 1.9. For both systems we used **eig** to compute the eigenvalues; the spectral radius is $\rho(BA) = 1.00$ by construction, and the leftmost eigenvalues are

$$\begin{aligned} \lambda_{\text{lm}}(\hat{A}_{\text{well}}^T A_{\text{well}}) &= 4.78 \cdot 10^{-5}, \\ \lambda_{\text{lm}}(\hat{A}_{\text{ill}}^T A_{\text{ill}}) &= -2.02 \cdot 10^{-5} \pm i 3.83 \cdot 10^{-6}. \end{aligned}$$

Note that for A_{ill} the leftmost eigenvalue is a complex conjugate pair with a negative real part. Hence we expect the BA ITERATION to exhibit nonconvergence for the problems with A_{ill} . To ensure convergence of the SHIFTED BA ITERATION we choose

$$\alpha = 2 |\operatorname{Re}(\lambda_{\text{lm}}(\hat{A}^T A))|.$$

The convergence histories for the noisy data ($e \neq 0$) are shown in Figure 1; the similar plots for the noise-free data ($e = 0$) are almost identical. We make the following observations:

- The left plot confirms that the BA ITERATION converges only for the well-conditioned matrix for which all eigenvalues have a positive real part and that it converges to $x^* = A_{\mathcal{N}(B)}^\dagger b = B(AB)^{-1}b$.
- The right plot confirms that the SHIFTED BA ITERATION converges for both matrices and that it converges to $x_\alpha^* = B(AB + \alpha I)^{-1}b$. For the

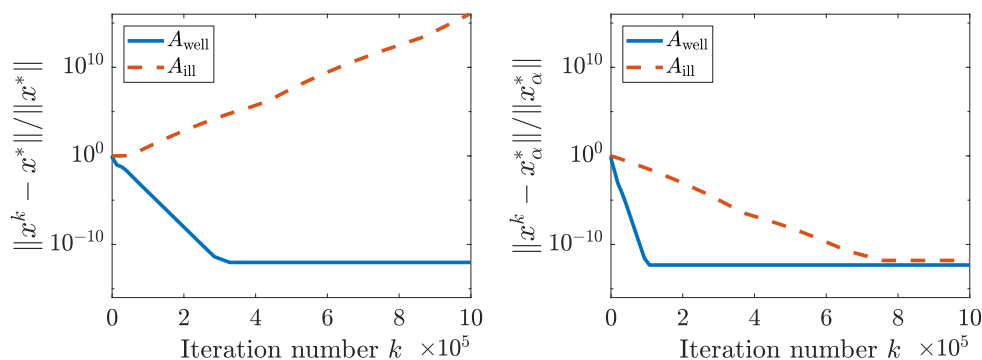


FIG. 1. The convergence histories for the small test problems with noisy data ($e \neq 0$); the similar plots for noise-free data ($e = 0$) are almost identical. Left: results for the BA ITERATION which exhibits nonconvergence for the ill-conditioned matrix. Right: results for the SHIFTED BA ITERATION with $\alpha = 2 |\operatorname{Re}(\lambda_{\text{lm}}(\hat{A}^T A))|$; this method converges for both matrices.

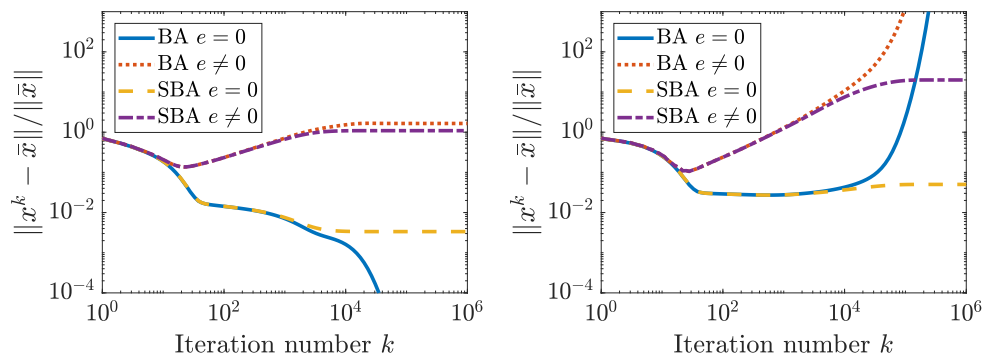


FIG. 2. The semiconvergence histories of the two iterative methods for A_{well} (left) and A_{ill} (right). For the problems with noisy data, we obtain a reasonably accurate solution at the point of semiconvergence. See the text for a detailed discussion.

well-conditioned system the convergence is much faster compared to the ill-conditioned system. Also, the SHIFTED BA ITERATION converges faster than the BA ITERATION for the well-conditioned system.

Having confirmed the convergence of the methods, it is also relevant to study how well the methods are able to approximate the exact solution \bar{x} . To illustrate this, Figure 2 shows plots of the reconstruction errors $\|x^k - \bar{x}\| / \|\bar{x}\|$ versus iteration number k . We make several observations:

- For the well-conditioned matrix A_{well} and noise-free data ($e = 0$) the BA ITERATION converges to the exact solution \bar{x} as predicted by Theorem 2.2 with square and full-rank A and B .
- For the ill-conditioned matrix A_{ill} the BA ITERATION always diverges.
- For noise-free data ($e = 0$) the SHIFTED BA ITERATION converges to a slightly perturbed solution \bar{x}_α^* with

$$\|\bar{x}_\alpha^* - \bar{x}\| / \|\bar{x}\| = 3.36 \cdot 10^{-3} \quad \text{for } A_{\text{well}},$$

$$\|\bar{x}_\alpha^* - \bar{x}\| / \|\bar{x}\| = 5.01 \cdot 10^{-2} \quad \text{for } A_{\text{ill}}.$$

- For noisy data ($e \neq 0$) the BA ITERATION for A_{well} as well as the SHIFTED BA ITERATION for both A_{well} and A_{ill} converge to a fixed point that is quite far from the exact solution. Exactly the same behavior occurs for iterations that use a matching transpose. The main point is that for noisy data the iterations exhibit *semiconvergence*, where the iterates produce a good approximation to \bar{x} during the initial iterations.

In conclusion, these experiments verify the benefit of using the SHIFTED BA ITERATION, namely, guaranteed convergence while retaining the semiconvergence that all algebraic iterative methods rely on for noisy data.

5.2. Test problem from the ASTRA toolbox. The second test problem comes from X-ray CT using a parallel-beam geometry with 90 projections in the angular range 0° – 180° , and a detector with 80 pixels and length equal to the image size. The exact solution \bar{x} represents a 128×128 discretization of the Shepp–Logan phantom. Hence the problem size is $m = 7200$ and $n = 16384$, and the problem is underdetermined. For such underdetermined CT problems the algebraic iterative

methods can give much better results than the “standard” methods based on filtered backprojection [17]. As before, the exact and noisy data are generated according to (5.1).

The matrices A and \hat{A}^T that represent the forward projections and backprojections come from the ASTRA toolbox [22] used in conjunction with “spot operators” [23]. Specifically, we use the ASTRA function `opTomo` to compute the matrix-vector multiplications with these matrices. For the forward projection the GPU-version of ASTRA uses the interpolation model, also known as Joseph’s method [12], while the backprojection uses the line model with linear interpolation between detector pixels (as done, e.g., in `iradon` in MATLAB). The matrices are not stored; if we store them, then the sparsity of A and \hat{A}^T is 1.32% and 2.35%, respectively. Measures of nonsymmetry and nonnormality of BA are

$$\frac{\|\frac{1}{2}((BA) - (BA)^T)\|_F}{\|BA\|_F} = 0.125, \quad \frac{\|(BA)(BA)^T - (BA)^T(BA)\|_F}{\|BA\|_F^2} = 0.0235.$$

The spectral radius is $\rho(BA) \approx 1.76 \cdot 10^4$.

For this test problem we use the algorithms from section 4 to estimate the leftmost eigenvalue of BA , and we compare the performance of the following strategies:

- `eigs` from MATLAB with options `maxit` = 1500, `tol` = 10^{-2} , `SIGMA` = ‘`sr`’;
- `ks` is the Krylov–Schur method (Algorithm 1) with options `maxit` = 1500, absolute tolerance `tol` = 10^{-2} , `mindim` = 30, `maxdim` = 60, `target` = ‘`-inf`’ (for the leftmost eigenvalue);
- `jd` is the Jacobi–Davidson method with options `maxit` = 1500, `tol` = 10^{-2} , `mindim` = 30, `maxdim` = 60, `target` = ‘`-inf`’;
- `fovN` is the field of values based method (Algorithm 2) with options `maxit` = `N`, `mindim` = 30, `maxdim` = 60.

Table 1 shows results for two cases:

1. All matrix-vector multiplications are performed on the GPU, using the ASTRA function `opTomo`. These multiplications are performed in *single precision*. The Jacobi–Davidson method `jd` did not converge, and this may be due to the single-precision computations on the GPU.
2. The two matrices A and \hat{A}^T are explicitly computed and stored as sparse matrices. Like all the other computations, these computations are performed on the CPU in *double precision*.

We see that for these CT problems the Krylov–Schur method uses the least amount of computations, corresponding to the work in performing about 500 iterations of the SHIFTED BA ITERATION method. When solving several CT problems with the same geometry, and hence the same matrices, this is acceptable. Even when an unmatched pair is used only once, this may be an acceptable overhead to ensure convergence and trust in the computed solution.

Figure 3 reports the work involved in the eigenvalue estimation with the Krylov–Schur method, as measured by matrix-vector multiplications for different numbers of image pixels $n = 64^2, 128^2, 256^2, 512^2, 1024^2, 2048^2$. The number of rows is $m \approx 0.45n$. We see that for larger problems the number of matrix-vector multiplications seems to be proportional to $n^{1/3}$.

Figure 4 shows the convergence histories when applying the two iterative algorithms to this problem, with α and ω chosen as before. We observe the same behavior as before: the BA ITERATION diverges, while the SHIFTED BA ITERATION converges to a fixed point. Moreover, the SHIFTED BA ITERATION exhibits semiconvergence as

TABLE 1

Estimation of the leftmost eigenvalue λ_{lm} of BA with the methods discussed in section 4. We use the ASTRA test problem mentioned in the text. The methods stop when we reach convergence with absolute tolerance 10^{-2} or after the fixed number of iterations used in the field of values based method Algorithm 2 fovN for $N = 10, 15$, and 20 iterations, respectively. We show the mean number of matrix-vector multiplications (MVMs) as well as the mean and standard deviation of the estimated λ_{lm} . The top half of the table shows results for the case when the matrix-vector multiplications are done by the ASTRA function `opTomo`, while the bottom half is for the case when we explicitly store the matrices. The difference is due to the difference in precision (single versus double).

	25 trials with <code>opTomo</code> that utilizes the GPU	
	Mean MVM	Mean λ_{lm} (st. dev.)
fov10	660	-0.8921 ($1.167 \cdot 10^{-1}$)
fov15	960	-0.9323 ($7.478 \cdot 10^{-3}$)
fov20	1260	-0.9354 ($3.579 \cdot 10^{-3}$)
ks	1041	-0.9281 ($2.510 \cdot 10^{-5}$)
eigs	1278	-0.9281 ($4.103 \cdot 10^{-5}$)
	25 trials with A and \hat{A}^T explicitly stored	
	Mean MVM	Mean λ_{lm} (st. dev.)
fov10	660	-0.8920 ($1.171 \cdot 10^{-1}$)
fov15	960	-0.9322 ($7.647 \cdot 10^{-3}$)
fov20	1260	-0.9354 ($3.581 \cdot 10^{-3}$)
ks	1039	-0.9281 ($2.334 \cdot 10^{-5}$)
eigs	1261	-0.9281 ($3.893 \cdot 10^{-5}$)
jd	1404	-0.9281 ($4.624 \cdot 10^{-5}$)

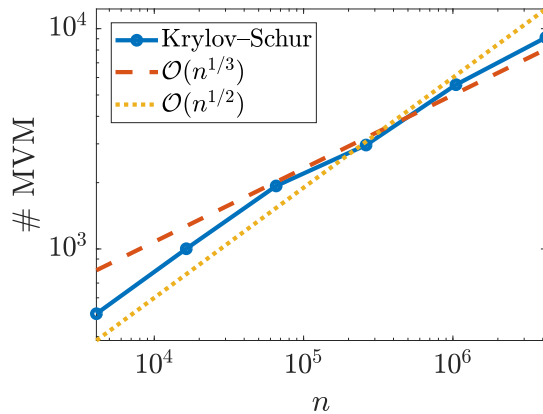


FIG. 3. The number of matrix-vector multiplications (MVMs) needed to estimate the leftmost eigenvalue with the Krylov-Schur method, as a function of the number of image pixels n .

expected, and the minimum reconstruction error—at the point of semiconvergence—does not deteriorate when using the shifted method.

6. Conclusions. We have considered algebraic iterative reconstruction methods with an unmatched backprojector, i.e., the backprojector is not the exact adjoint or transpose of the forward projector. In particular we are concerned with the common situation where the iterative method does not converge, due to the nonsymmetry of the iteration matrix. We propose a modified algorithm that uses a small shift parameter, we define the conditions that guarantee convergence to a fixed point of a slightly perturbed problem, and we give perturbation bounds for this fixed point. We also discuss how to efficiently estimate the leftmost eigenvalue of a certain matrix, which

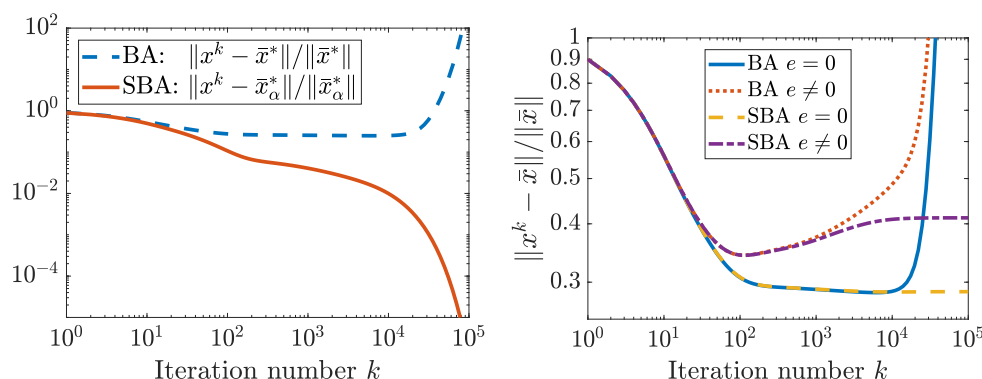


FIG. 4. Left: the convergence of the BA ITERATION and the SHIFTED BA ITERATION for noise-free data ($e = 0$); as expected the former does not converge. Right: semiconvergence histories of the two iterative methods for noise-free as well as noisy data.

is needed to compute the shift parameter in the modified algorithm. Numerical experiments with artificial test problems as well as a test problem from computed tomography illustrate the use of the new algorithm. Our MATLAB code is available on request.

Acknowledgments. We thank Willem Jan Palenstijn (CWI) for sharing his insight into the ASTRA package. We acknowledge the inspiration from Tommy Elfving and Martin S. Andersen who, independently, suggested the shift as a remedy for the nonconvergence. Finally, we thank two anonymous referees for many valuable comments that helped to improve the paper.

REFERENCES

- [1] A. BIGURI, M. DOSANJH, S. HANCOCK, AND M. SOLEIMANI, *TIGRE: A MATLAB-GPU toolbox for CBCT image reconstruction*, Biomed. Phys. Eng. Express, 2 (2016), 055010; the software available from <https://github.com/CERN/TIGRE>.
- [2] T. ELFVING AND P. C. HANSEN, *Unmatched projector/backprojector pairs: Perturbation and convergence analysis*, SIAM J. Sci. Comput., 40 (2018), pp. A573–A591.
- [3] T. ELFVING, P. C. HANSEN, AND T. NIKAZAD, *Semiconvergence and relaxation parameters for projected SIRT algorithms*, SIAM J. Sci. Comput., 34 (2012), pp. A2000–A2017.
- [4] W. HACKBUSCH, *Iterative Solution of Large Sparse Systems of Equations*, Springer, New York, 2016.
- [5] P. C. HANSEN, *The discrete Picard condition for discrete ill-posed problems*, BIT, 5 (1990), pp. 658–672.
- [6] P. C. HANSEN, *Test matrices for regularization methods*, SIAM J. Sci. Comput., 16 (1995), pp. 506–512.
- [7] P. C. HANSEN, *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM, Philadelphia, 1998.
- [8] P. C. HANSEN, *Regularization Tools Version 4.0 for MATLAB 7.3*, Numer. Algorithms, 46 (2007), pp. 189–194.
- [9] P. C. HANSEN, *Oblique projections and standard-form transformations for discrete inverse problems*, Numer. Linear Algebra Appl., 20 (2013), pp. 250–258.
- [10] P. C. HANSEN AND J. S. JØRGENSEN, *AIR Tools II: Algebraic iterative reconstruction methods, improved implementation*, Numer. Algorithms, 79 (2018), pp. 107–137.
- [11] R. HORN AND C. R. JOHNSON, *Matrix Analysis*, 2nd ed., Cambridge University Press, Cambridge, UK, 2013.
- [12] P. M. JOSEPH, *An improved algorithm for reprojecting rays through pixel images*, IEEE Trans. Medical Imag., 3 (1982), pp. 192–196.
- [13] D. A. LORENZ, S. ROSE, AND F. SCHÖPFER, *The randomized Kaczmarz method with mismatched adjoint*, BIT, 58 (2018), pp. 1079–1098.

- [14] K. MEERBERGEN AND D. ROOSE, *Matrix transformations for computing rightmost eigenvalues of large sparse non-symmetric eigenvalue problems*, IMA J. Numer. Anal., 16 (1996), pp. 297–346.
- [15] K. MEERBERGEN AND M. SADKANE, *Using Krylov approximations to the matrix exponential operator in Davidson's method*, Appl. Numer. Math., 31 (1999), pp. 331–351.
- [16] F. NATTERER, *The Mathematics of Computerized Tomography*, Classics in Appl. Math. 32, SIAM, Philadelphia, 2001.
- [17] R. C. NELSON, S. FEUERLEIN, AND D. T. BOLL, *New iterative reconstruction techniques for cardiovascular computed tomography: How do they work, and what are the advantages and disadvantages?*, J. Cardiovascular Computed Tomography, 5 (2011), pp. 286–292.
- [18] G. L. G. SLEIJPEN AND H. A. VAN DER VORST, *A Jacobi–Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 401–425.
- [19] D. C. SORESENSEN, *Implicit application of polynomial filters in a k -step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.
- [20] G. W. STEWART, *A Krylov–Schur algorithm for large eigenproblems*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 601–614.
- [21] G. W. STEWART AND J. G. SUN, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.
- [22] W. VAN AARLE, W. J. PALENSTIJN, J. CANT, E. JANSSENS, R. BLEICHRODT, A. DABRAVOLSKI, J. DE BEENHOUWER, K. J. BATENBURG, AND J. SIJBERS, *Fast and flexible X-ray tomography using the ASTRA toolbox*, Opt. Express, 24 (2016), pp. 25129–25147; the software available from <http://www.astra-toolbox.com>.
- [23] E. VAN DEN BERG AND M. P. FRIEDLANDER, *Spot—A Linear-Operator Toolbox*, <http://www.cs.ubc.ca/labs/scl/spot/>.
- [24] S. VAN DER MAAR, K. J. BATENBURG, AND J. SIJBERS, *Experiences with Cell-BE and GPU for tomography*; in *Embedded Computer Systems: Architectures, Modeling, and Simulation, Proceedings 9th International Workshop SAMOS*, K. Bertels, N. Dimopoulos, C. Silvano, and S. Wong, eds., Springer, Berlin, 2009, pp. 298–307.
- [25] G. L. ZENG AND G. T. GULLBERG, *Unmatched projection/backprojection pairs in iterative reconstruction algorithms*, IEEE Trans. Med. Imaging, 19 (2000), pp. 548–555.