A JOINT BIDIAGONALIZATION BASED ALGORITHM FOR LARGE SCALE LINEAR DISCRETE ILL-POSED PROBLEMS IN GENERAL-FORM REGULARIZATION∗

ZHONGXIAO JIA† AND YANFEI YANG‡

Abstract. Based on the joint bidiagonalization process of a large matrix pair \(\{A, L\}\), we propose and develop an iterative regularization algorithm for the large scale linear discrete ill-posed problem in general-form regularization: \(\min \|Lx\|\) subject to \(x \in \mathcal{S} = \{x| \|Ax - b\| \leq \tau \|e\|\}\) with \(e\) a white noise and \(\tau > 1\) slightly, where \(L\) is a regularization matrix. Our algorithm has been shown to have the desired semi-convergence property that any single regularization method must possess and is different from the hybrid one proposed by Kilmer et al., which is based on the same process but solves the general-form Tikhonov regularization problem: \(\min \{\|Ax - b\|^2 + \lambda^2 \|Lx\|^2\}\). We prove that the iterates take the form of attractive filtered generalized singular value decomposition (GSVD) expansions, which get insight into the regularizing effects of the proposed method. We use the L-curve criterion or the discrepancy principle to determine \(k^*\). The algorithm is simple and effective, and numerical experiments illustrate that it often computes more accurate regularized solutions than the hybrid one.

Key words. Linear discrete ill-posed, general-form regularization, joint bidiagonalization, GSVD, filtered GSVD expansion, semi-convergence, LSQR, hybrid, discrepancy principle

AMS subject classifications. 65F22, 65F10, 65F35, 65F50, 65J20

1. Introduction. Consider the solution of the large scale linear discrete ill-posed problem

(1.1) \(\min_{x \in \mathbb{R}^n} \|Ax - b\| \quad \text{or} \quad Ax = b, \quad A \in \mathbb{R}^{m \times n}, \quad b \in \mathbb{R}^m\),

where the norm \(\|\cdot\|\) is the 2-norm of a vector or matrix, the matrix \(A\) is ill conditioned with its singular values decaying to zero with no obvious gap between consecutive ones, and the right-hand side \(b = b_{true} + e\) is noisy and assumed to be contaminated by a white noise \(e\), where \(b_{true}\) is the noise-free right-hand side and \(\|e\| < \|b_{true}\|\). Such kind of problem arises in a variety of applications such as computerized tomography, image deblurring, signal processing, geophysics, heat propagation, biomedical and optical imaging, groundwater modeling, and many others; see, e.g., [1, 2, 6, 11, 17, 21, 22]. Since \(b\) contains the noise \(e\) and \(A\) is extremely ill conditioned, the naive solution \(x_{naive} = A^+b\) is unbounded in norm and is a meaningless approximation to the true solution \(x_{true} = A^+b_{true}\), where \(A^+\) denotes the Moore-Penrose inverse of a matrix. Therefore, one has to use regularization to obtain a best possible approximation to \(x_{true}\) [10, 12].

Assume that \(Ax_{true} = b_{true}\) and \(m \geq n\). Then two essentially equivalent dominating regularization approaches are the following general-form regularization

(1.2) \(\min_{x \in \mathbb{R}^n} \|Lx\| \quad \text{subject to} \quad x \in \mathcal{S} = \{x| \|Ax - b\| \leq \tau \|e\|\}\)

with some \(\tau > 1\) and the general-form Tikhonov regularization

(1.3) \(\min_x \{\|Ax - b\|^2 + \lambda^2 \|Lx\|^2\}\),

∗This work was supported in part by the National Science Foundation of China (No. 11771249).
†Department of Mathematical Sciences, Tsinghua University, 100084 Beijing, China (jiazx@tsinghua.edu.cn).
‡Department of Mathematical Sciences, Tsinghua University, 100084 Beijing, China (yangyf14@mails.tsinghua.edu.cn).
where $L \in \mathbb{R}^{p \times n}$ is a regularization matrix and $\lambda > 0$ is the regularization parameter. If $L = I_n$, the $n \times n$ identity matrix, then (1.2) and (1.3) are called standard-form regularization problems. If $A$ and $L$ satisfy

\begin{equation}
\mathcal{N}(A) \cap \mathcal{N}(L) = \emptyset, \quad \text{i.e.,} \quad \operatorname{rank}
\begin{pmatrix}
A \\
L
\end{pmatrix}
= n,
\end{equation}

the solution to (1.3) is unique. In practical applications, $L$ is typically chosen as a scaled approximation of the first or second derivative operator, respectively [10, 12].

Unlike the case $L = I_n$, there has been no so much work on Krylov subspace algorithms for (1.2) or (1.3). We refer the reader to, e.g., [4, 5, 8] for some available algorithms and codes. In practical applications, it may be impractical to solve them by direct solvers, and instead only iterative solvers are computationally viable. This is often the case for two or three dimensional case, e.g., where $L$ is the sum of Kronecker products. Recently, some randomized algorithms have been proposed and developed [16] without requiring direct (generalized) inversion of $L$, which appear to work efficiently for the problems that the singular values of $A$ decay fast, where a large scale least squares problem is iteratively solved at each step.

Zha [27] proposes a joint bidiagonalization process that successively reduces the matrix pair \{$A, L$\} to upper bidiagonal forms. Based on the process, Zha proposes a joint bidiagonalization method for computing a few largest or smallest generalized singular values and the associated singular vectors of a large matrix pair \{$A, L$\}.

Kilmer et al. [18] adapt Zha’s joint bidiagonalization process to linear discrete ill-posed problems in general-form Tikhonov regularization and develop a joint bidiagonalization process that successively reduces the matrix pair \{$A, L$\} to lower and upper bidiagonal forms. Based on the process, they propose a hybrid projection method for (1.3). If the global optimal regularization parameter $\lambda_{\text{opt}}$ can be found, the proposed hybrid method is reasonable since the iterates produced turn out to lie in legitimate subspaces, which is critical for the possible success of an iterative Krylov regularization method. Their hybrid algorithm is an inner-outer iterative one: first, each outer iteration exploits the joint bidiagonalization process to project the matrix pair \{$A, L$\} onto a pair of low dimensional subspaces, which involves the solution of a large scale linear least squares problem with the coefficient matrix $(A^T, L^T)^T$ that is supposed to be solved iteratively, called inner iterations. At each outer iteration, one solves a small projected general-form Tikhonov regularization problem. Finally, one solves a large scale least squares problem with the coefficient matrix $(A^T, L^T)^T$ to form a regularized solution. The outer iteration proceeds until the regularized solutions stabilize, that is, their accuracy cannot be improved. Within the hybrid framework, the iteration number do not play the role of the regularization parameter any longer; instead, one needs to determine an optimal Tikhonov regularization parameter for a projected problem in general-form Tikhonov regularization generated at each outer iteration. Kilmer et al. mention that all the large scale least squares problems should be solved by some iterative solver, e.g., the LSQR algorithm, but they, in fact, solved them by QR factorizations in the experiments on a two dimensional image deblurring problem of $m = n = 4,096$.

The algorithm in [18] may encounter some essential difficulties. It is well known from, e.g., [10, 12], that any regularization is based on an underlying requirement that the discrete Picard condition for (1.3) is satisfied. This requirement is absolutely necessary, only under which can one compute a useful regularized solution with some accuracy. Notice that a small projected general-form Tikhonov regularization
A JOINT BIDIAGONALIZATION BASED ALGORITHM

problem is solved at each outer iteration [18], which needs to determine an optimal regularization parameter for the small problem itself. One fundamental fact that is crucial but has received very little attention until the work [9, 20] is: with $L = I_n$, that (1.3) satisfies the discrete Picard condition does not mean that the projected problems fulfill discrete Picard conditions too, and known sufficient conditions for the projected problems to satisfy the discrete Picard conditions require that the singular values of the matrices involved in the projected problems approximate the large singular values of $A$ in natural order. For (1.3), an adaption of this result to $L \neq I_n$ says that the projected problems are guaranteed to inherit the discrete Picard conditions only when the generalized singular values of the projected matrix pairs approximate the large generalized singular values of the matrix pair $\{A, L\}$ in natural order. However, we must point out that such sufficient conditions are quite stringent and the approximations in natural order can only be guaranteed for severely and moderately ill-posed problems [13, 14]. For the definition of severely and moderately ill-posed problems, see [10, 12].

One important point deserving enough attention is that if the above sufficient conditions are met then the iterative algorithm used resembles the truncated GSVD (TGSVD) method [10] until the occurrence of semi-convergence. At this time, a best regularized solution has been already found and is as accurate as the best TGSVD solution. If it is the case, any complicated hybrid projection method is not needed, and what we need is to determine the semi-convergence point by some parameter-choice methods, e.g., the L-curve criterion and the discrepancy principle. We should remind that a severe difficulty for hybrid projection methods is that, whenever the discrete Picard conditions fail for the projected problems, optimal regularization parameters for them are poorly defined. As a consequence, the regularized solutions may exhibit irregular behavior. As a matter of fact, when developing a hybrid LSQR variant, by requiring that the singular values of the projected matrices approximate the large singular values of $A$ in natural order, Renaut et al. [20] prove that an optimal regularization parameter for each projected problem can be reliably determined by a weighted GCV (WGCV) parameter-choice method and it converges to the global optimal regularization parameter for (1.3) as the outer iteration proceeds.

In this paper, based on the joint bidiagonalization process [18], we will propose a novel iterative regularization algorithm for solving (1.2) other than (1.3). First, we make use of the joint bidiagonalization process to project (1.2) onto a sequence of pairs of low dimension subspaces and obtain a sequence of projected problems, which involve matrix pairs of small size. Then at each outer iteration we solve a projected problem. Remarkably, we find that the solution of each of them reduces to an ordinary small least squares problem with the coefficient matrices being single lower bidiagonal other than the matrix pair, so that it is very simple and cheap to solve all the projected problems by QR factorizations. Theoretically, we rigorously prove that our algorithm must have a typical semi-convergence property and the iteration number plays the role of the regularization parameter: as the joint bidiagonalization process proceeds, more and more dominant generalized singular components of $\{A, L\}$ are captured, and the regularized solutions converge to the true solution $x_{true}$ of (1.1) until some iteration, after which the noise $e$ starts to deteriorate the regularized solutions, which instead converge to $x_{noise}$. Importantly, we prove that the iterates obtained by our algorithm take the form of filtered GSVD expansions, a desired and insightful property, which indicates that the iterates not only lie in legitimate subspaces but also get more insight into the regularizing effects of the algorithm.
Since the residual norms monotonically decrease and the semi-norms of solutions monotonically increase, we can use the L-curve criterion and the discrepancy principle to estimate the optimal regularization parameter $k^*$, at which the semi-convergence occurs. We make an analysis on the discrepancy principle and explain how to design a reliable stopping criterion and compute a meaningful regularized solution.

We will numerically compare our algorithm with the hybrid algorithm in [18], in which we make use of the GCV and WGCV parameter choice methods to determine an optimal regularization parameter for each small projected problem. We are mainly concerned with the accuracy of the best regularized solutions by our algorithm and the hybrid one. The experiments on several real-world problems will illustrate the superiority of our algorithm.

The paper is organized as follows. In Section 2, we overview GSVD, the general-form Tikhonov regularization method and the TGSVD method, and present the joint bidiagonalization process of $\{A, L\}$. In Section 3, we describe the hybrid method in [18]. In Section 4, we propose our joint bidiagonalization based method and make an analysis on it. In Section 5, we consider the practical determination of the optimal regularization parameter. Numerical experiments are presented in Section 6. Finally, we conclude the paper with further remarks in Section 7.

2. GSVD, regularization methods and joint bidiagonalization. In this section, we provide some necessary background. We describe GSVD, the TGSVD method, the filtered GSVD method, and the joint bidiagonalization process proposed in [27] and developed in [18].

Consider the compact QR factorization

\[
\begin{pmatrix}
A \\
L
\end{pmatrix} = QR,
\]

where $Q = \begin{pmatrix} Q_A \\ Q_L \end{pmatrix} \in \mathbb{R}^{(m+p) \times n}$ is column orthonormal with $Q_A \in \mathbb{R}^{m \times n}$, $Q_L \in \mathbb{R}^{p \times n}$, and $R \in \mathbb{R}^{n \times n}$ is upper triangular and nonsingular because of the assumption (1.4). We have $A = Q_AR$, $L = Q_LR$, and $Q_A^T Q_A + Q_L^T Q_L = I_n$.

Let the CS decomposition of the matrix pair $\{Q_A, Q_L\}$ be

\[
Q_A = P_A CW^T, \quad Q_L = P_LS W^T,
\]

where $P_A \in \mathbb{R}^{m \times m}$, $P_L \in \mathbb{R}^{p \times p}$, and $W \in \mathbb{R}^{n \times n}$ are orthogonal, and $C \in \mathbb{R}^{m \times n}$ and $S \in \mathbb{R}^{p \times n}$ are diagonal matrices satisfying $C^T C + S^T S = I_n$; see [3, Section 4.2]. Then the GSVD of $\{A, L\}$ is

\[
A = P_A CG^{-1}, \quad L = P_LS G^{-1}
\]

with $G = (g_1, g_2, \ldots, g_n) = R^{-1}W \in \mathbb{R}^{n \times n}$, and the vectors $g_i$ are the right singular vectors of $\{A, L\}$. Following the unconventional but more convenient way [18], we order the entries of the diagonal matrices $C$ and $S$ so that

\[
1 \geq c_1 \geq \cdots \geq c_{\min\{n,p\}} \geq 0, c_{\min\{n,p\}+1} = \cdots = c_n = 1,
\]

\[
0 \leq s_1 \leq \cdots \leq s_{\min\{n,p\}} \leq 1.
\]
By the GSVD (2.3), the general-form Tikhonov solution $x_\lambda$ to (1.3) takes a filtered GSVD expansion:

$$x_\lambda = (A^T A + \lambda^2 L^T L)^{-1} A^T b = G (C^T C + \lambda^2 S^T S)^{-1} C^T P^T_A b$$

(2.6)  

$$= \sum_{i=1}^{\min\{n, p\}} \frac{c_i^2}{c_i^2 + \lambda^2 s_i^2} p_{i,A}^T b g_i + \sum_{i=\min\{n, p\}+1}^{n} p_{i,A}^T b g_i,$$

where $P_A = (p_{1,A}, p_{2,A}, \ldots, p_{m,A})$, $f_i = \frac{c_i^2}{c_i^2 + \lambda^2 s_i^2}$ are filters, and the second term lies in the null space $\mathcal{N}(L)$ of $L$, which is spanned by the vectors $g_{\min\{n, p\}+1}, \ldots, g_n$. We address that the regularization does not affect the second term. This is simply the filtered GSVD method for solving (1.3).

The discrete Picard condition [10] states that the Fourier coefficients $|p_{i,A}^T b|$ must, on average, decay faster than the $c_i$. Hence the $|p_{i,A}^T b|$ decay until the noise $\epsilon$ dominates the $|p_{i,A}^T b|$, that is, the $|p_{i,A}^T b| \approx |p_{i,A}^T b_i|$ after $i > k_0$, while $|p_{i,A}^T b| \approx |p_{i,A}^T b_{true}|$, $i = 1, 2, \ldots, k_0$, where $k_0$ is called the transition or cutting-off point. Therefore, a good regularized solution $x_\lambda$ must capture the $k_0$ dominant GSVD components of $\{A, L\}$ and meanwhile dampen those for $i > k_0$ as much as possible. An optimal regularization parameter $\lambda_{opt}$ can be determined by some parameter-choice methods, e.g., the discrepancy principle, the $L$-curve criterion, and the generalized cross validation (GCV) or weighted GCV (WGCV) method; see [10, 12] and also [2, 11].

Alternatively, making use of the GSVD of $\{A, L\}$, one can develop the TGSVD method and computes the TGSVD solution

$$x^\text{tgsvd}_k = \sum_{i=1}^{k} \frac{p_{i,A}^T b}{c_i} g_i + \sum_{i=\min\{n, p\}+1}^{n} p_{i,A}^T b g_i, \quad k = 1, 2, \ldots, \min\{n, p\},$$

(2.7)

where the first term consists of the first $k$ dominant GSVD components of $\{A, L\}$. The TGSVD solution takes a special filtered GSVD expansion, where the filters $f_i = 1$ for $i = 1, 2, \ldots, k$ and $f_i = 0$ for $i = k+1, \ldots, \min\{n, p\}$. Under the discrete Picard condition, the TGSVD method exhibits semi-convergence: $x^\text{tgsvd}_k$ converge to $x^\text{true}_k$ and $Lx^\text{true}_k$ for $k \leq k_0$, afterwards they diverge and instead converge to $x^\text{naive}_k$ and $Lx^\text{naive}_k$, respectively. A best possible TGSVD solution $x^\text{tgsvd}_{k_0}$ is thus obtained for $k = k_0$.

We notice that the second terms in (2.6) and (2.7) are the same and they disappear when $p \geq n$. For later use, we write them as

$$g_\perp = \sum_{i=\min\{n, p\}+1}^{n} p_{i,A}^T b g_i \in \mathcal{N}(L).$$

(2.8)

Now we review a procedure that jointly diagonalizes the matrix pair $\{A, L\}$ to lower and upper bidiagonal forms. Applying the BIDIAG-1 algorithm and BIDIAG-2 algorithm in [24] to $Q_A$ and $Q_L$, respectively, which are the lower and upper Lanczos bidiagonalization processes, we can reduce $Q_A$ and $Q_L$ to lower and upper bidiagonal forms, respectively. The two processes can be written in matrix form:

$$Q_AV_k = U_{k+1}B_k, \quad Q_A^T U_{k+1} = V_k B_k^T + \alpha_{k+1}v_{k+1}^T c_{k+1}^T,$$

(2.9)

$$Q_L^T \tilde{V}_k = \tilde{U}_k \tilde{B}_k, \quad Q_L^T \tilde{U}_k = \tilde{V}_k B_k^T + \beta_k v_{k+1}^T c_k^T,$$

(2.10)
where \(e_{k+1}\) and \(e_k\) are the \((k+1)\)th and \(k\)th canonical vectors of dimensions \(k+1\) and \(k\), respectively,

\[
U_{k+1} = (u_1, \ldots, u_{k+1}) \in \mathbb{R}^{m \times (k+1)}, \quad \hat{U}_k = (\tilde{u}_1, \ldots, \tilde{u}_k) \in \mathbb{R}^{p \times k},
\]

and

\[
V_k = (v_1, \ldots, v_k) \in \mathbb{R}^{n \times k}, \quad \hat{V}_k = (\tilde{v}_1, \ldots, \tilde{v}_k) \in \mathbb{R}^{n \times k}
\]

are column orthonormal, and

\[
B_k = \begin{pmatrix}
\alpha_1 & \beta_2 & \alpha_2 \\
\beta_3 & \ddots & \ddots \\
& \ddots & \ddots \\
& & \beta_{k+1}
\end{pmatrix} \in \mathbb{R}^{(k+1) \times k}, \quad \hat{B}_k = \begin{pmatrix}
\hat{\alpha}_1 & \hat{\beta}_1 \\
& \ddots & \ddots \\
& & \hat{\beta}_{k-1} \\
& & & \hat{\alpha}_k
\end{pmatrix} \in \mathbb{R}^{k \times k}
\]

are lower bidiagonal and upper bidiagonal, respectively. Zha [27] and Kilmer et al. [18] have investigated the relationships between \(V_k\) and \(\hat{V}_k\) defined in (2.12) and between \(B_k\) and \(\hat{B}_k\) defined in (2.13), respectively, and they have established the following result.

**Theorem 2.1.** If \(v_1 = \hat{v}_1\) in (2.12), then

\[
\hat{v}_{j+1} = (-1)^j v_{j+1}, \quad \hat{\alpha}_j \hat{\beta}_j = \alpha_{j+1} \beta_{j+1}, \quad j = 1, \ldots, k.
\]

A combination of (2.9)–(2.13), Theorem 2.1 and the QR factorization (2.1) shows that \(A\) and \(L\) can be jointly bidiagonalized [18, 27], as summarized below.

**Theorem 2.2.** Assume that \(A \in \mathbb{R}^{m \times n}\) and \(L \in \mathbb{R}^{p \times n}\) with \(m \geq n\). Then there exist orthogonal matrices \(U \in \mathbb{R}^{m \times m}\), \(\hat{U} \in \mathbb{R}^{p \times p}\) and \(V \in \mathbb{R}^{n \times n}\), and a lower bidiagonal \(B \in \mathbb{R}^{m \times n}\), an upper bidiagonal \(\hat{B} \in \mathbb{R}^{p \times n}\), and an invertible \(Z\) such that

\[
A = U B Z^{-1},
\]

\[
L = \hat{U} \hat{B} Z^{-1},
\]

where \(Z = R^{-1} V\) and \(\hat{B} = \hat{B} D\) with \(D = \text{diag}(1, -1, 1, -1, \ldots)\), and the remaining matrices are obtained by running joint bidiagonalization to completion. In particular, when \(p < n\), the columns \(p+1, \ldots, n\) of \(\hat{B}\) contain only zero entries.

From (2.15) and (2.16), we obtain \(k\)-step joint bidiagonalization relations

\[
AZ_k = U_{k+1} B_k,
\]

\[
LZ_k = \hat{U}_k \hat{B}_k,
\]

where \(Z_k \in \mathbb{R}^{n \times k}\) is the first \(k\) columns of \(Z\), and \(B_k\) and \(\hat{B}_k\) are the first \((k+1) \times k\) and \(k \times k\) submatrices of \(B\) and \(\hat{B}\), respectively.

For \(A\) and \(L\) large, the computation of \(Q_A\) and \(Q_L\) is impractical. In order to avoid explicitly computing \(Q_A\) and \(Q_L\), inspired by Zha’s work [27], Kilmer et al. [18] develop a joint bidiagonalization (JBD) process, denoted by Algorithm 1, to compute the matrices in (2.11)–(2.13), in which \(0_p\) denotes the zero vector of dimension \(p\).
Algorithm 1  \textit{k-step joint bidiagonalization (JBD) process.}

1: $\beta_1 u_1 = b$, $\beta_1 = \|b\|$. \\
2: $\alpha_1 \tilde{v}_1 = QQ^T \begin{pmatrix} u_1 \\ 0_p \end{pmatrix}$. \\
3: $\alpha_1 \tilde{u}_1 = \tilde{v}_1 (m + 1 : m + p)$. \\
4: \textbf{for} $i = 1, 2, \ldots, k$ \textbf{do} \\
5: \hspace{1em} $\beta_{i+1} u_{i+1} = \tilde{v}_i (1 : m) - \alpha_i u_i$. \\
6: \hspace{1em} $\alpha_{i+1} \tilde{v}_{i+1} = QQ^T \begin{pmatrix} u_{i+1} \\ 0_p \end{pmatrix} - \beta_{i+1} \tilde{v}_i$. \\
7: \hspace{1em} $\hat{\beta}_i = (\alpha_{i+1} \beta_{i+1}) / \hat{\alpha}_i$. \\
8: \hspace{1em} $\hat{\alpha}_{i+1} \tilde{u}_{i+1} = (-1)^i \tilde{v}_{i+1} (m + 1 : m + p) - \hat{\beta}_i \tilde{u}_i$. \\
9: \textbf{end for}

Let $\tilde{u}_i = \begin{pmatrix} u_i \\ 0_p \end{pmatrix}$. At each iteration $i = 1, 2, \ldots, k + 1$, Algorithm 1 needs to compute $QQ^T \tilde{u}_i$, which is not accessible since $Q$ is not available. Notice that $QQ^T \tilde{u}_i$ is nothing but the orthogonal projection of $\tilde{u}_i$ onto the column space of $\begin{pmatrix} A \\ L \end{pmatrix}$, which means that $QQ^T \tilde{u}_i = \begin{pmatrix} A \\ L \end{pmatrix} \tilde{x}_i$. Clearly, $\tilde{x}_i$ is the solution to the least squares problem:

\begin{equation}
\tilde{x}_i = \arg \min_{\tilde{x} \in \mathbb{R}^n} \left\| \begin{pmatrix} A \\ L \end{pmatrix} \tilde{x} - \tilde{u}_i \right\|.
\end{equation}

Since the least squares problem is large scale, it is generally only feasible to solve it by an iterative solver, e.g., the most commonly used LSQR algorithm [24]. Here we have two remarks.

**Remark 2.1.** Theoretically, at outer iteration $k$, the inner least square problem \[(2.19)\] is solved accurately in order to guarantee that \[(2.17)\] and \[(2.18)\] hold exactly. It is unknown whether or not the solution accuracy can be relaxed by allowing possibly large inexactness in the algorithm [18] and ours to be presented later. This issue is certainly complicated. We do not investigate it in the current paper. In finite precision arithmetic, we suppose that \[(2.19)\] is solved by the Matlab function \texttt{lsqr.m} with the default stopping criterion $10^{-6}$.

**Remark 2.2.** To ensure the numerical orthogonality of the computed $U_{k+1}$, $\hat{U}_k$ and $V_k$, we use one step reorthogonalization during the process in implementations.

3. The hybrid projection based method in [18]. Algorithm 1 takes

\begin{equation}
U_{k+1} (\beta_1 e_1) = b, \quad \beta_1 = \|b\|,
\end{equation}

where $e_1$ is the first canonical vector of dimension $k + 1$. For a given regularization parameter $\lambda$, the hybrid projection based method in [18] seeks the solution $x_k^\lambda \in \text{span} \{Z_k\}$ such that

\begin{equation}
\min_{x \in \text{span} \{Z_k\}} \left\{ \|Ax - b\|^2 + \lambda^2 \|Lx\|^2 \right\} = \|Ax_k^\lambda - b\|^2 + \lambda^2 \|Lx_k^\lambda\|^2.
\end{equation}

Exploit \[(3.1)\], \[(2.17)\] and \[(2.18)\], and write

\begin{equation}
x_k^\lambda = Z_k y_k^\lambda.
\end{equation}
It is direct to justify that
\[
\|Ax_k^λ - b\|^2 + \lambda^2\|Lx_k^λ\|^2 = \|B_ky_k^λ - \beta_1e_1\|^2 + \lambda^2\|\tilde{B}_ky_k^λ\|^2
\]
\[
= \min_y\{\|B_ky - \beta_1e_1\|^2 + \lambda^2\|\tilde{B}_ky\|^2\}.
\]

Therefore, at iteration \(k\) the hybrid projection based method in [18] solves a projected general-form Tikhonov regularization problem
\[
(3.3) \quad \min_y\{\|B_ky - \beta_1e_1\|^2 + \mu_k^2\|\tilde{B}_ky\|^2\},
\]
where the new notation \(\mu_k > 0\) is introduced to specialize the regularization parameter for the projected problem at iteration \(k\). The key point is the determination of an optimal regularization \(\mu_{k_{\text{opt}}}\) for (3.3). Adapted the results of Renault et al. [20] with \(L = I_n\) to (3.3), the optimal \(\mu_{k_{\text{opt}}}\) determined by the GCV or WGCV method converges to the global optimal regularization parameter \(\lambda_{\text{opt}}\) for (1.3) as \(k\) increases under the assumption that the generalized singular values of \(\{B_k, \tilde{B}_k\}\) approximate the large singular values of \(\{A, L\}\) in natural order; if the assumption is not fulfilled, such convergence may fail, implying that the regularized solution \(x_{k_{\text{opt}}}^\mu\) may behave irregular and does not stabilize for \(k\) sufficiently large. As a consequence, it may be hard to stop the hybrid algorithm properly, and even for \(k\) sufficiently large the regularized solution \(x_{k_{\text{opt}}}^\mu\) may not be as accurate as \(x_{\lambda_{\text{opt}}}\), the best regularized solution to (1.3) associated with \(\lambda = \lambda_{\text{opt}}\).

Now we show how to compute \(x_{k_{\text{opt}}}^\lambda\) when \(y_k^\lambda\) is known. Let \(\tilde{V}_k = (\tilde{v}_1, \ldots, \tilde{v}_k) \in \mathbb{R}^{(m+p) \times k}\) be generated by Algorithm 1. Then
\[
(3.4) \quad \tilde{V}_k = QV_k = QR(R^{-1}V_k) = \begin{pmatrix} A \\ L \end{pmatrix} Z_k,
\]
from which, (2.1) and (3.2) it follows that
\[
(3.5) \quad \begin{pmatrix} A \\ L \end{pmatrix} x_k^\lambda = \tilde{V}_k y_k^\lambda.
\]

Kilmer et al. [18] show that one only needs to form \(x_k^\lambda\) explicitly when it is accepted as the final regularized solution.

Regarding the determination of \(\lambda_{\text{opt}}\), other than determining \(\mu_{k_{\text{opt}}}\) for each projected problem (3.3), Kilmer et al. [18] use the L-curve criterion to tentatively estimate \(\lambda_{\text{opt}}\): they assume that a set of \(\lambda\)-values is prescribed, and derive some update formulas for all the quantities, including regularized solutions, residual norms, and the semi-norms \(\|Lx_k^\lambda\|\), which can be efficiently computed for the a-prior set of \(\lambda\)-values. For sufficiently large \(k\) at which all the needed \(k_0\) dominant GSVD components of \(\{A, L\}\) are thought to have been captured, drawing the picture of \((\log \|Ax_k^\lambda - b\|, \log \|Lx_k^\lambda\|)\) for the given set of \(\lambda\)-values, they attempt to obtain a L-curve and pick up the \(\lambda\)-value at the corner as an approximation \(\lambda_{\text{opt}}\).

Their approach to determining \(\lambda_{\text{opt}}\) faces two challenging issues: the first is how to effectively determine a sufficiently large \(k\), and the second is how to choose a good a-prior set of \(\lambda\)-values which include the optimal regularization parameter \(\lambda_{\text{opt}}\) or its good approximation. As a matter of fact, the first issue is common in any hybrid LSQR algorithm, and there has been no reliable approach to resolve it. In our implementation, we will take a regular manner, as done in, e.g., [10, 20], and determine \(\mu_{k_{\text{opt}}}\) for each (3.3) by the GCV code [11] and the WGCV code adapted from [2], both of which need to compute the GSVD of \(\{B_k, \tilde{B}_k\}\) at cost of \(O(k^3)\) flops.
4. Our joint bidiagonalization based algorithm. Instead of solving (1.3), we now present a joint bidiagonalization based algorithm for solving (1.2), which is an iterative regularization method and shown to have the attractive semi-convergence, at which the best regularized solution is found. In the algorithm, the iteration number \( k \) plays the role of the regularization parameter.

Still, we seek \( x_k \in \text{span}\{Z_k\} \) and write it in the form

\[
x_k = Z_k y_k.
\]

Replace \( A \) and \( L \) by \( AZ_k \) and \( LZ_k \) in (1.2). We solve the reduced general-form regularization problem

\[
\min \| LZ_k y \| \quad \text{subject to} \quad y \in S = \{ y : \| AZ_k y - b \| = \min \}
\]

starting with \( k = 1 \) onwards. Make use of (3.1), (2.17) and (2.18). Then (4.2) becomes

\[
\min \| \bar{B}_k y \| \quad \text{subject to} \quad y \in S = \{ y : \| B_k y - \beta_1 e_1 \| = \min \}
\]

starting with \( k = 1 \) onwards. After the solution \( y_k \) for (4.3) is computed, in terms of (3.4), (2.1) and \( x_k = Z_k y_k \), we then solve

\[
\begin{pmatrix} \tilde{A} \\ \tilde{L} \end{pmatrix} x_k = \tilde{V}_k y_k
\]

for \( x_k \).

Now let us investigate the solution of the constrained problem (4.3). As it will turn out, (4.3) amounts to an ordinary unconstrained linear least squares problem. We have the following results.

**Theorem 4.1.** Assume that Algorithm 1 does not break down at iteration \( k \leq \min\{n, p\} \). Then the solution \( y_k \) to (4.3) is

\[
y_k = \arg \min_y \| B_k y - \beta_1 e_1 \| = \beta_1 B_k^\dagger e_1.
\]

**Proof.** Let \( \bar{y} = \bar{B}_k y \). Then under the assumption on Algorithm 1, \( \bar{B}_k \) is nonsingular. Therefore, (4.3) is equivalent to

\[
\min \| \bar{y} \| \quad \text{subject to} \quad \bar{y} \in S = \{ \bar{y} : \| (B_k \bar{B}_k^{-1}) \bar{y} - \beta_1 e_1 \| = \min \}.
\]

Notice that \( B_k \) is of column full rank, so is \( B_k \bar{B}_k^{-1} \). As a result, we have

\[
\bar{y}_k = \beta_1 (B_k \bar{B}_k^{-1})^\dagger e_1
\]

\[
= \beta_1 (\bar{B}_k B_k^\dagger) e_1
\]

with the second equation holding because \( B_k \) is of column full rank and \( \bar{B}_k \) is nonsingular. Then the solution \( y_k \) to (4.3) is

\[
y_k = \bar{B}_k^{-1} \bar{y}_k = \beta_1 \bar{B}_k^{-1} \left((\bar{B}_k B_k^\dagger) e_1\right) = \beta_1 B_k^\dagger e_1.
\]

(4.5) indicates that \( y_k \) is simply the solution to the ordinary least squares problem \( \min_y \| B_k y - \beta_1 e_1 \| \) and \( \bar{B}_k \) is not invoked. Let

\[
B_k = Q_k R_k
\]
be the compact QR factorization of $B_k$, which can be computed by exploiting Givens rotations at cost of $O(k)$ flops. From (4.5), we obtain

\[(4.7) \quad y_k = \beta_1 R_k^{-1} Q^T_k e_1\]

at cost of $O(k)$ flops; see [24] for details.

Next, we consider the efficient computation of the residual norm $\|Ax_k - b\|$ and the semi-norm $\|Lx_k\|$.

**Theorem 4.2.** Let the matrices $U_{k+1}, \hat{U}_k, B_k$ and $\bar{B}_k$ be defined in (2.17) and (2.18). Then

\[(4.8) \quad \|Ax_k - b\| = \|B_k y_k - \beta_1 e_1\|,\]
\[(4.9) \quad \|Lx_k\| = \|\bar{B}_k y_k\|.

**Proof.** Notice $x_k = Z_k y_k$, and exploit (3.1) and (2.17). We obtain

\[Ax_k = AZ_k y_k = U_{k+1} B_k y_k.\]

Since $U_{k+1}$ is column orthonormal, it is direct to derive (4.8) by the orthogonal invariance of the 2-norm. Similarly, we have

\[(4.10) \quad Lx_k = LZ_k y_k = \hat{U}_k \bar{B}_k y_k.\]

Since $\hat{U}_k$ is column orthonormal, we have (4.9).

This theorem shows that, by making use of structures of $B_k$ and $\bar{B}_k$, both $\|Ax_k - b\|$ and $\|Lx_k\|$ can be computed very efficiently at cost of $O(k)$ flops without forming $x_k$ explicitly.

Now we analyze our algorithm, establish some important results and get insight into its regularizing effects. Let $\tilde{w} = Rx$. Then by (2.1), we have

\[(4.11) \quad \min_x \|Ax - b\| = \min_{\tilde{w}} \|Q_A \tilde{w} - b\|.\]

First, it is direct to establish the following result, similar to Theorem 4.3 in [18].

**Lemma 4.3.** Let $x_k$ be the regularized solution obtained by our algorithm. Then

\[(4.12) \quad x_k = R^{-1} \tilde{w}_k, \quad \tilde{w}_k = \arg \min_{\tilde{w} \in \mathcal{K}_k} \|Q_A \tilde{w} - b\|,\]

where $\mathcal{K}_k$ is the $k$ dimensional Krylov subspace

\[(4.13) \quad \mathcal{K}_k = \text{span}\{Q^T_A b, (Q^T_A Q_A)Q^T_A b, \ldots, (Q^T_A Q_A)^{k-1} Q^T_A b\}.

**Proof.** Write $\tilde{w} = V_k y$, where $V_k$ is generated by (2.9) and $\text{span}\{V_k\} = \mathcal{K}_k$. Then from (2.9) we obtain

\[\min_{\tilde{w} \in \mathcal{K}_k} \|Q_A \tilde{w} - b\| = \min_y \|B_k y - \beta_1 e_1\|.\]

Let $y_k = \arg \min_y \|B_k y - \beta_1 e_1\|$. By the definition (3.4) of $Z_k$, we have $x_k = Z_k y_k = R^{-1} V_k y_k = R^{-1} \tilde{w}_k$. \qed
To present our main theoretical result and make an insightful analysis on the regularizing effects of the proposed algorithm, we need to make some necessary preparations and notation changes. Firstly, for the regularization matrix $L \in \mathbb{R}^{p \times n}$ of rank $\min\{n, p\}$, from the SVD (2.2) of $Q_A$ and $Q_L$ and the labeling orders (2.4) and (2.5), it is obvious that the singular values $c_i$ and $s_i$ must satisfy

$$0 < c_1, s_i < 1, \ i = 1, 2, \ldots, \min\{n, p\}.$$

If $p \geq n$, we retain the notation (2.2) and (2.4), and have $1 > c_1 \geq c_2 \geq \cdots \geq c_n > 0$. If $p < n$, different from (2.4), we relabel the $c_i$ and use the new notation

$$1 = c_1 > c_2 \geq c_3 \geq \cdots \geq c_{p+1}, \tag{4.14}$$

where $c_1 = 1$ is the largest singular value of $Q_A$ with the multiplicity $n-p$. That is, we reassign the indices $i$ of $c_i$ defined by (2.4) to $i+1$, $i = 1, 2, \ldots, p$, and shift the largest singular value one of $Q_A$ with multiplicity $n-p$ in (2.4) to $c_1 = 1$. Correspondingly, we permute the columns of $P_A, W$ in $Q_A$ defined by (2.2), and $G$ defined by (2.3) by moving their respective last $n-p$ columns to the first ones and renaming

$$P_A := (P_{1,A}, P_{2,A}, \ldots, P_{n-n+p+1,A}),$$

$$W := (W_1, w_2, \ldots, w_{p+1}),$$

$$G := (G_1, g_2, \ldots, g_{p+1}).$$

With the new notation, we have the range $\mathcal{R}(G_1) = \mathcal{N}(L)$, i.e., $LG_1 = 0$. Keep in mind that if $p \geq n$ then $P_A, W$ and $G$ remain the same as in (2.2) and (2.3), and $\mathcal{N}(L) = \emptyset$.

Secondly, it is well known [3] that the Lanczos bidiagonalization method for computing the singular values $c_i$ of $Q_A$ with the starting vector $b/\|b\|$ mathematically amounts to the symmetric Lanczos method for computing the eigenvalues $c_i^2$ of $Q_A^T Q_A$ with the starting vector $Q_A^T b/\|Q_A^T b\|$. It is remarkable that the symmetric Lanczos method works on $Q_A^T Q_A$ as if $Q_A^T Q_A$ has only simple eigenvalues $c_i^2$ [25]. As a consequence, the Lanczos bidiagonalization method works on $Q_A$ as if the singular values $c_i$ of $Q_A$ are all simple. Notice that the singular values $\tilde{c}_i$, called the Ritz values, of the projected matrix $B_k$ are always simple provided that the Lanczos bidiagonalization process does not break down at step $k$. The Lanczos bidiagonalization method uses the $\tilde{c}_i$ as approximations to the $k$ distinct singular values $c_i$ of $Q_A$. For a rigorous and complete derivation and many details, we refer to [15].

Next we establish an attractive and desired property that the regularized solution $x_k$ has a filtered GSVD expansion.

**Theorem 4.4.** Assume that the $c_i$ are labeled as (4.14) and simple for $p < n$, the matrices $P_A, W$ and $G$ defined as above, and $g_\perp$ defined by (2.8). Then

$$x_k = f_1^{(k)} g_\perp + \sum_{i=2}^{p+1} f_i^{(k)} \frac{P_A^T b}{c_i} g_i, \ k = 1, 2, \ldots, n, \tag{4.15}$$

where the filters

$$f_i^{(k)} = 1 - \prod_{j=1}^{k} \frac{c_j^2 - c_i^2}{c_j^2}, \ i = 1, 2, \ldots, p+1. \tag{4.16}$$
Notice that the LSQR algorithm starting with \( u_1 = b/\|b\| \) applied to (4.12) is mathematically equivalent to the conjugate gradient (CG) method applied to the normal equation \( Q_A^T Q_A w = Q_A^T b \) of (4.12) with the starting vector \( w_0 = 0 \). Let \( w_{ls} = Q_A^T b \) be the solution to \( \min_{\tilde{w}} \| Q_A \tilde{w} - b \| \). Then by the SVD (2.2) of \( Q_A \) and the notation (4.14) we obtain

\[
w_{ls} = W_1 P_{1,A}^T b + \sum_{i=2}^{p+1} \frac{p_{i,A}^T b}{c_i} w_i,
\]

where the first term is the sum of the \( n - p \) SVD components of \( Q_A \) corresponding to the largest singular value \( c_1 = 1 \) with the multiplicity \( n - p \).

With our notation and (4.17), keep in mind a well-known result (cf., e.g., [26, Property 2.8]) on the CG iterates that states

\[
\tilde{w}_k = (I - q_k(Q_A^T Q_A)) w_{ls},
\]

where \( q_k(\mu) \) is the \( k \)-th residual polynomial of CG at iteration \( k \) and \( q_k(0) = 1 \), whose roots are the Ritz values \( \tilde{c}_j^2 \) of \( Q_A^T Q_A \) with respect to \( K_k \) defined by (4.13), i.e.,

\[
q_k(c_i^2) = \prod_{j=1}^k \frac{\tilde{c}_j^2 - c_i^2}{\tilde{c}_j^2}, \quad i = 1, 2, \ldots, p + 1.
\]

Substituting (4.17) into (4.18) yields

\[
\tilde{w}_k = f_1^{(k)} W_1 P_{1,A}^T b + \sum_{i=2}^{p+1} f_i^{(k)} \frac{p_{i,A}^T b}{c_i} w_i, \quad k = 1, 2, \ldots, p + 1
\]

with \( f_i^{(k)} \) defined by (4.16).

Recall from (2.1)–(2.3) that \( G = R^{-1} W \). It then follows that \( G_1 = R^{-1} W_1 \) and \( g_k = R^{-1} w_k \). From (4.12), since \( x_k = R^{-1} \tilde{w}_k \), premultiplying (4.19) by \( R^{-1} \) establishes (4.15) by noticing that \( G_1 P_{1,A}^T b = g_\perp \) in our new notation. \( \Box \)

If \( p \geq n \), then \( g_\perp = 0 \) in (2.8), and the first term is zero in (4.15) and the second term becomes

\[
x_k = \sum_{i=1}^n f_i^{(k)} \frac{p_{i,A}^T b}{c_i} g_i, \quad k = 1, 2, \ldots, n.
\]

In this case, (4.15) is a filtered GSVD expansion similar to (2.6). If \( p < n \), the first term

\[
f_1^{(k)} g_\perp \in N(L)
\]

in (4.15), which resembles the term \( g_\perp \) in (2.6) and (2.7). On the other hand,

\[
\sum_{i=2}^{p+1} f_i^{(k)} \frac{p_{i,A}^T b}{c_i} g_i
\]

in (4.15) corresponds to the first term in (2.6) by noticing that in our notation the indices \( i + 1 \) in the sum correspond to the \( i \) in (2.6). A difference is that the general-form Tikhonov regularization solution (2.6) and TGSVD solution (2.7) do not affect
g⊥, while our algorithm multiplies it by a factor \( f_1^{(k)} \). Nonetheless, \( f_1^{(k)} \rightarrow 1 \) as \( c_1 \) converges to \( c_1 \); since \( g⊥ \) and \( f_1^{(k)} g⊥ \) lie in \( \mathcal{N}(L) \), they have no effect on \( Lx_\lambda, Lx_\text{tgsvd} \) and \( Lx_k \).

It is known from [10, Theorem 2.1.1, p.23] that the \( c_i \) decay like the singular values \( \sigma_i \) of \( A \) when the matrix \((A^T, L^T)^T\) is well conditioned, which is true provided that \( L \) is well conditioned, as is usually the case in practical applications. In the meantime, notice from (2.2) and (2.3) that \( Q_A \) and \( A \) share the same \( P_A \) and the problems \( \min_x \|Ax - b\| \) and \( \min_w \|Q_Aw - b\| \) have the same right-hand side \( b \). Therefore, the two problems satisfy the same discrete Picard condition, and it can be justified that the TGSVD method applied to \( \min_x \|Ax - b\| \) and TSVD method applied to \( \min_w \|Q_Aw - b\| \) compute the respective best regularized solution at the same optimal regularization parameter \( k_0 \).

Furthermore, as has been proved in [14], since the \( c_i \) decay and are clustered at zero, the singular values of \( B_k \) converge to the large singular values \( c_i \) of \( Q_A \) in natural order for severely and moderately ill-posed problems until the occurrence of semi-convergence of LSQR for solving \( \min_w \|Q_Aw - b\| \). From (4.15) and (4.16), it is easily justified that \( f_i^{(k)} \approx 1 \) for \( i = 1, 2, \ldots, k \) and \( f_i^{(k)} \approx 0 \) for \( i = k + 1, \ldots, p + 1 \) when the \( k \) Ritz values \( \tilde{c}_i \) approximate the large singular values of \( Q_A \) in natural order; we refer the reader to [10, pp. 146-148] for more details. This means that \( x_k \) mainly contains the first \( k \) dominant GSVD components of \( \{A, L\} \) and filters the others corresponding to the small generalized singular values before the semi-convergence. Based on the filtered SVD expansions of LSQR iterates, the first author of this paper has given more general and detailed elaborations on the regularizing effects of LSQR [14]. With the equivalence (4.11), adapted the results of [14] to our current context, this theorem shows that the proposed joint bidiagonalization based method exhibit typical semi-convergence at some iteration \( k^* \): \( x_k \) and \( Lx_k \) converge to \( x_\text{true} \) and \( Lx_\text{true} \) for \( k \leq k^* \) and afterwards they are deteriorated by the noise \( e \) and diverge for \( k > k^* \). Therefore, the iteration number \( k \) plays the role of the regularization parameter. Precisely, the semi-convergence of the joint bidiagonalization based algorithm occurs at iteration \( k^* \), which is such that \( \|L(x_k - x_\text{true})\| \) is minimal over all \( k = 1, 2, \ldots, \min\{n, p\} \).

**Remark 4.1.** Kübler et al. [18] have shown that the iterates \( x_k^1 \) by the hybrid projection based method lie in the following legitimate subspaces:

\[
\text{span}(G(C^TC)^iC^TP_k^Tb)_{i=0}^{k-1},
\]

to which our iterates \( x_k \) also belong, but they have not derived any filtered GSVD expansions of \( x_k^1 \) in the basis \( \{g_i\}_{i=1}^n \).

5. The determination of the optimal regularization parameter \( k^* \). For our joint bidiagonalization based algorithm, since the residual norm \( \|Ax_k - b\| = \|B_ky_k - \beta_1c_1\| \) monotonically decreases and the semi-norm \( \|Lx_k\| = \|B_ky_k\| \) monotonically increases practically with respect to \( k \), the L-curve criterion and the discrepancy principle suit well for a practical determination of \( k^* \). We plot the curve

\[
(\log(\|B_ky_k - \beta_1c_1\|), \log(\|B_ky_k\|))
\]

and then determine \( k \) at its overall corner as an estimate of \( k^* \). This is routine, and we do not repeat the determination procedure; see [10, 11, 12].

If \( \|e\| \) or its accurate estimate is known in advance, the discrepancy principle is the simplest and a reliable choice. Notice that \( b = b_\text{true} + \epsilon \). We have

\[
(5.1) \quad \|\epsilon\| - \|Ax_k - b_\text{true}\| \leq \|Ax_k - b\| \leq \|Ax_k - b_\text{true}\| + \|\epsilon\|.
\]
Therefore, we should make \( \|Ax_k - b\| \) as small as possible, so that the above lower and upper bounds are close. We should remind that it is impossible to obtain \( x_k = x_{true} \) in the presence of \( e \) since there must be some loss of accuracy in a best regularized solution for any regularization method \([7]\). To this end, we should stop the algorithm at the first iteration \( k \) satisfying

\[
(5.2) \quad \|Ax_k - b\| = \|B_k y_k - \beta_1 e_1\| \leq \tau \|e\|
\]

with \( \tau > 1 \) slightly, e.g., \( \tau = 1.1 \) or smaller. We then use such \( k \) as an estimate of the optimal regularization parameter \( k^* \). We must point out that a \( \tau > 1 \) considerably, e.g., \( \tau = 2 \), is generally unsafe and may underestimate \( k^* \) substantially since the lower and upper bounds in (5.1) will be far away, which implies a risk, as argued now: suppose \( \|Ax_k - b\| = \tau \|e\| \). Then it follows from (5.1) that

\[
\|Ax_k - b_{true}\| \geq (\tau - 1)\|e\| = \|e\|
\]

for \( \tau = 2 \), meaning that the problem (1.1) is over-regularized and the noise-free equation \( Ax_{true} = b_{true} \) is solved too inaccurately. As a result, it is a \( \tau > 1 \) slightly that may make \( \|Ax_k - b_{true}\| < \|e\| \), so that a good regularized solution \( x_k \) is obtained.

Embedded with the above parameter-choice methods, we can now present our joint bidiagonalization based algorithm, called JBDQR and named Algorithm 2.

Algorithm 2 (JBDQR) Given \( A \in \mathbb{R}^{m \times n} \) and \( L \in \mathbb{R}^{p \times n} \), solve \((1.2)\) and compute the regularized solution \( x_{k^*} \) at semi-convergence.

1: Starting with \( k = 1 \), run Algorithm 1, and obtain the small projected problem (4.5).
2: Compute the minimum 2-norm solution \( y_k \) to (4.5).
3: Compute \( \|Ax_k - b\| \) and \( \|Lx_k\| \) by the formulas (4.8) and (4.9).
4: Determine the optimal regularization parameter \( k^* \) by the L-curve criterion or check if the discrepancy principle (5.2) is satisfied. If \( k^* \) is not found, set \( k = k+1 \), and update Algorithm 1. Then go to Step 2.
5: After \( k^* \) is determined, form the regularized solution \( x_{k^*} \) by solving (4.4).

6. Numerical examples. In this section, we report numerical experiments to demonstrate that our JBDQR algorithm works well and the best regularized solutions obtained by it are at least as accurate as those obtained by the hybrid one proposed by Kilmer et al. \([18]\) and can be substantially more accurate than the latter ones. We also compare the optimal regularization parameters determined by the L-curve criterion and the discrepancy principle with the true optimal \( k^* \).

We choose some one dimensional examples from the regularization toolbox \([11]\) and some two dimensional problems from the Matlab Image Processing Toolbox and \([2, 23]\); see Table 1, where the two dimensional image deblurring problems \( rice \) and \( mri \) are from the Matlab Image Processing Toolbox. We denote the relative noise level

\[
\epsilon = \frac{\|e\|}{\|b_{true}\|}.
\]

For the noise-free problems \( Ax_{true} = b_{true} \) in Table 1, we add a white noise \( e \) with zero mean and a prescribed noise level \( \epsilon \) to \( b_{true} \) and form the noisy \( b = b_{true} + e \). To simulate exact arithmetic, the complete reorthogonalization is used in Algorithm 1.
A JOINT BIDIAGONALIZATION BASED ALGORITHM

Table 1
The description of test problems.

| Problem        | Description                                    | Ill-posedness |
|----------------|------------------------------------------------|---------------|
| shaw           | One-dimensional image restoration model [11]   | severe        |
| baart          | First kind Fredholm integral equation [11]    | severe        |
| heat           | Inverse heat equation [11]                     | moderate      |
| deriv2         | Computation of second derivative [11]         | mild          |
| rice           | Two dimensional image deblurring               | unknown       |
| mri            | Two dimensional image deblurring               | unknown       |
| AtmosphericBlur30 | Two dimensional image deblurring [2, 23] | unknown       |
| GaussianBlur422 | Two dimensional image deblurring [2, 23] | unknown       |

We abbreviate Algorithm 2 as JBDQR, the hybrid one in [18] using the GCV and WGCV parameter-choice methods as JBDGCV and JBDWGCV, respectively. Let $x_k^{reg}$ denote the regularized solutions obtained by each of the algorithms. We use the relative error

\[ \frac{\|L(x_k^{reg} - x_{true})\|}{\|Lx_{true}\|} \]

(6.1)

to plot the convergence curve of each algorithm with respect to $k$. In the tables to be presented, we will list the smallest relative errors and iteration steps used by JBDGCV and JBDWGCV in parentheses, the optimal iteration steps $k^*$ at which the semi-convergence of JBDQR occurs in the parentheses and the estimated ones for $k^*$ determined by the L-curve criterion and the discrepancy principle (5.2) as well as the corresponding relative errors in the parentheses. We use the Matlab function lsqr.m to solve (2.19), (3.5) and (4.4) with the default stopping tolerance $tol = 10^{-6}$.

All the computations are carried out in Matlab R2015b 64-bit on Intel Core i3-2120 CPU 3.30GHz processor and 4 GB RAM with the machine precision $\epsilon_{\text{mach}} = 2.22 \times 10^{-16}$ under the Microsoft Windows 7 64-bit system.

6.1. One dimensional case. The four test problems shaw, baart, heat and deriv2 are severely, moderately and mildly ill-posed, respectively. For each of them we use the code of [11] to generate $A$, $x_{true}$ and $b_{true}$. We mention that deriv2 has three kinds of right-hand sides, distinguished by the parameter "example = 1, 2, 3". we only report the results on the parameter "example = 2" since we have obtained very similar results on the problem with "example = 1, 3". Purely for test purposes, for severely ill-posed problems shaw and baart, we take $m = n = 1,024$, and for moderately and mildly ill-posed problems heat and deriv2, we take $m = n = 3,000$. Purely for test purposes, we choose

\[ L = L_1 = \begin{pmatrix}
1 & -1 \\
1 & -1 \\
& & \ddots & \ddots \\
& & & 1 & -1
\end{pmatrix} \in \mathbb{R}^{(n-1) \times n}, \]

(6.2)

which is a scaled discrete approximation of the first derivative operator in one dimensional case. For the scaled discrete approximation of the second derivative operator, we have similar findings and have observed very similar phenomena. Hence we only report the results on $L = L_1$. 
Table 2
The relative errors and estimates for the optimal regularization parameters $k^*$ by the L-curve criterion for the test problems with $L = L_1$.

| $\varepsilon = 10^{-2}$ | $\varepsilon = 10^{-3}$ | $\varepsilon = 10^{-4}$ |
|-------------------------|-------------------------|-------------------------|
|                         |                         |                         |
| JBDGCV                  | JBDWGCV                  | JBDQR ($k^*$)           | estimates for $k^*$ |
| shaw                    | 0.5398(18)               | 0.2094(3)               | 2(0.2126)           |
| baart                   | 0.5574(7)                | 0.5405(2)               | 3(0.5625)           |
| heat                    | 0.3758(60)               | 0.2186(13)              | 5(0.3284)           |
| deriv2                  | 0.4270(60)               | 0.3363(4)               | 2(0.3853)           |
|                         |                         |                         |                       |
| JBDGCV                  | JBDWGCV                  | JBDQR ($k^*$)           | estimates for $k^*$ |
| shaw                    | 0.1930(13)               | 0.1732(5)               | 2(0.1918)           |
| baart                   | 0.5442(9)                | 0.5038(4)               | 2(0.5376)           |
| heat                    | 0.1794(100)              | 0.1456(25)              | 23(0.1485)          |
| deriv2                  | 0.3884(60)               | 0.2635(10)              | 8(0.3161)           |
|                         |                         |                         |                       |
| JBDGCV                  | JBDWGCV                  | JBDQR ($k^*$)           | estimates for $k^*$ |
| shaw                    | 0.1664(14)               | 0.1378(8)               | 8(0.1378)           |
| baart                   | 0.5346(9)                | 0.4136(5)               | 3(0.5354)           |
| heat                    | 0.1360(100)              | 0.1275(37)              | 35(0.1283)          |
| deriv2                  | 0.2916(60)               | 0.2452(15)              | 12(0.2606)          |

In Table 2, we display the relative errors of the best regularized solutions by JBDQR, JBDGCV and JBDWGCV with $L = L_1$ and $\varepsilon = 10^{-2}$, $10^{-3}$, $10^{-4}$, respectively. As we can see from the table, the best regularized solutions by JBDQR are at least as accurate as and can be more accurate than those by JBDGCV and JBDWGCV for all the test problems; see, e.g., shaw, heat and deriv2 for $\varepsilon = 10^{-2}$, and deriv2 for $\varepsilon = 10^{-3}$. We observe from the table that for each test problem the best regularized solution by JBDQR is correspondingly more accurate and requires a bigger $k^*$ for a smaller $\varepsilon$. All these are expected and justify that the smaller $\varepsilon$ is, the better regularized solution is extracted, that is, the more GSVD dominant components of $\{A, L\}$ are needed to form it. Finally, for JBDQR, we see that for each problem and given $\varepsilon$, almost all the regularization parameters $k^*$ determined by the L-curve criterion are quite reliable and close to the true $k^*$ except shaw for $\varepsilon = 10^{-3}$. But we also find that the L-curve criterion underestimates the true $k^*$ more or less, that is, the estimates for $k^*$ by the L-curve criterion oversmoothes the ill-posed problems.

Figure 1 depicts the convergence processes of JBDQR, JBDGCV and JBDWGCV for $L = L_1$ and $\varepsilon = 10^{-3}$. We observe from the figure and Table 2 that, in the most cases, the best regularized solutions by JBDQR are more accurate and can be considerably more accurate than those by JBDGCV and JBDWGCV. In addition, for the severely ill-posed shaw and baart we find that JBDGCV and JBDWGCV behave very similarly and the convergence processes are almost indistinguishable. Remarkably, we see that the regularized solutions obtained by them converge first, then stabilize for a while, and finally diverge dramatically, while, for heat and deriv2, they start to stabilize after $k$ becomes large. We have also observed that the smaller $\varepsilon$ is, the later they start to stabilize, though we do not draw all the corresponding figures. The phenomena
Table 3
The relative errors and estimates for the optimal regularization parameters $k^*$ by the discrepancy principle for the test problems with $L = L_1$.

| $\varepsilon = 10^{-2}$ | $\tau = 1.005$ | $\tau = 1.1$ | $\tau = 1.2$ | $\tau = 2.0$ |
|--------------------------|----------------|--------------|--------------|--------------|
| shaw                     | 0.3031(1)      | 0.3031(1)    | 0.3031(1)    | 0.3031(1)    |
| baart                    | 0.5421(1)      | 0.5421(1)    | 0.5421(1)    | 0.5421(1)    |
| heat                     | 0.3152(6)      | 0.3757(3)    | 0.4629(2)    | 0.5410(1)    |
| deriv2                   | 0.3853(2)      | 0.4187(1)    | 0.4187(1)    | 0.4187(1)    |

| $\varepsilon = 10^{-3}$ | $\tau = 1.005$ | $\tau = 1.1$ | $\tau = 1.2$ | $\tau = 2.0$ |
|--------------------------|----------------|--------------|--------------|--------------|
| shaw                     | 0.1888(2)      | 0.1888(2)    | 0.1888(2)    | 0.2338(1)    |
| baart                    | 0.5376(2)      | 0.5422(1)    | 0.5422(1)    | 0.5422(1)    |
| heat                     | 0.1669(20)     | 0.2196(11)   | 0.2377(10)   | 0.3258(5)    |
| deriv2                   | 0.3398(6)      | 0.4291(2)    | 0.4291(2)    | 0.4651(1)    |

| $\varepsilon = 10^{-4}$ | $\tau = 1.005$ | $\tau = 1.1$ | $\tau = 1.2$ | $\tau = 2.0$ |
|--------------------------|----------------|--------------|--------------|--------------|
| shaw                     | 0.1632(5)      | 0.1882(3)    | 0.1882(3)    | 0.1906(2)    |
| baart                    | 0.5354(3)      | 0.5354(3)    | 0.5400(2)    | 0.5437(1)    |
| heat                     | 0.1356(28)     | 0.1443(25)   | 0.1473(24)   | 0.1745(19)   |
| deriv2                   | 0.2606(12)     | 0.3019(9)    | 0.3400(7)    | 0.3813(4)    |

for shaw and baart do not comply with the expectation that the regularized solutions ultimately stabilize as the subspace is expanded sufficiently large. The reason is due to the fact that the discrete Picard conditions for the projected problems are satisfied poorly as $k$ increases, as we have argued in the introduction. In contrast, JBDQR has always exhibited the typical semi-convergence for all the problems, which justifies our theory.

Figure 2 depicts the L-curves given by JBDQR with $L = L_1$ and $\varepsilon = 10^{-3}$. We use the function $L_{\text{corner}}$ in [11] to determine the overall corner and give an estimate $k^*$. We see that for the moderately and mildly ill-posed problems heat and deriv2 there are much better "L" shape curves, which enable us to determine the optimal $k^*$ more reliably and accurately than those for the severely ill-posed problems shaw and baart. This is because JBDQR converges very fast and uses very few iterations to achieve the semi-convergence for shaw and baart. Indeed, the L-curve criterion does not work well for shaw and baart with $\varepsilon = 10^{-3}$.

Since $\|e\|$ is known for the above test problems, we can use the discrepancy principle (5.2) to estimate the optimal $k^*$. Table 3 reports the results obtained, in which we have taken the four $\tau = 1.005$, 1.1, 1.2 and 2.0. Compared with the $k^*$ in Table 2, we have found that the discrepancy principle always underestimate $k^*$ and the problems are over-regularized. We have observed that the reliable determination of $k^*$ critically depend on $\tau$, and the closer $\tau$ is to one, the more reliable the estimates are. Particularly, except for shaw and baart with $\varepsilon = 10^{-2}$, the choice $\tau = 2$ is obviously very bad, and it produces very poor estimates for $k^*$ and leads to considerably less accurate regularized solutions than $\tau = 1.005$ does.
Fig. 1. The relative error of JBDQR, JBDGCV and JBDWGCV with $L = L_1$ and $\epsilon = 10^{-3}$: (a) shaw; (b) baart; (c) heat; (d) deriv2.

Fig. 2. The determination of $k^*$ by the L-curve criterion by JBDQR with $L = L_1$, $\epsilon = 10^{-3}$: (a) shaw; (b) baart; (c) heat; (d) deriv2.
6.2. Two dimensional case. In this section, we test some two dimensional image deblurring problems listed in Table 1. The goal is to restore an image $x_{true}$ from a blurred and noisy image $b = b_{true} + e$.

We consider the problemsrice and mri from the Matlab Image Processing Toolbox. The exact image $x_{true}$ of rice is an $N \times N$ subimage and that of mri is the 15th slice of the three dimensional MRI image dataset which has $N \times N$ pixels. The blurred operator $A$ is a symmetric doubly Toeplitz PSF matrix and is of Kroneck product form $A = (2\pi\sigma^2)^{-1}T \otimes T \in \mathbb{R}^{N^2 \times N^2}$, where $T \in \mathbb{R}^{N \times N}$ is a symmetric banded Toeplitz matrix with half-bandwidth $b$ and $\sigma$ controls the width of Gaussian PSF. In what follows, we use band = 16, $\sigma = 2$ and $N = 128$. The size of rice and mri is $m = n = 128^2 = 16,384$.

We also consider the problems AtmosphericBlur30 and GaussianBlur422 of $m = n = 256^2 = 65,536$ from [23]. The blurring of AtmosphericBlur30 is caused by atmospheric turbulence, and GaussianBlur422 is spatially invariant Gaussian blur. The exact images are generated by the input command “load AtmosphericBlur30” and “load GaussianBlur422”, and the blurring operators are generated by the codes psf-Matrix(PSF, center, ‘zero’) and psfMatrix(P0) from [23], respectively. We abbreviate AtmosphericBlur30 and GaussianBlur422 as blur30 and blur422, respectively.

For the experimental purpose, we choose the regularization matrix

$$L = \begin{pmatrix} I_N \otimes L_1 \\ L_1 \otimes I_N \end{pmatrix} \in \mathbb{R}^{N(N-1) \times N^2}$$

with $L_1$ defined in (6.2), which is the scaled discrete approximation of the first derivative operator in two dimensional case, and $I_N$ the identity matrix of order $N$. The white noise $e$ with zero mean are generated so that the relative noise levels $\varepsilon = 5 \cdot 10^{-2}$, $10^{-2}$ and $10^{-3}$, respectively.

Besides the smallest relative errors defined by (6.1), Table 4 also lists the relative errors of the corresponding best regularized solutions obtained by JBDGCV, JBDWGCV and JBDQR, which are defined by

$$\frac{\|x_{true} - x_{reg}\|}{\|x_{true}\|}$$

and marked "no $L$" in the parentheses that follow the matrix names. We can see that for these four problems the solution accuracy of JBDQR is considerably higher than that of JBDGCV and JBDWGCV, no matter which relative error is used. From the table, it is clear that the estimates for $k^*$ by the L-curve criterion are quite rough and considerable underestimates except for blur30 with $\varepsilon = 10^{-3}$. This indicates that the L-curve criterion does not work well for determining $k^*$ for difficult two dimensional problems. The fundamental cause is that $\|\hat{B}_k y_k\|$ still increases slowly even after $k > k^*$, such that the curve of $(\log(\|\hat{B}_k y_k - \beta_1 e_1\|), \log(\|\hat{B}_k y_k\|))$ does not form a good L-shape.

Since $\|e\|$ is known for the above test problems, we also use the discrepancy principle criterion (5.2) to estimate the optimal $k^*$. We report the results obtained when $\tau = 1.005, 1.1, 1.2$ and $2.0$ in Table 5. We can see that, for the four $\tau > 1$'s, the regularization parameters determined by the discrepancy principle have big differences for both the solution accuracy and the estimates for $k^*$. It is obvious that the estimates are much better for $\tau = 1.005$ than those when $\tau = 2$. Again, this indicates that $\tau = 2$ is definitively a very bad choice.
Table 4

The relative errors and estimates for $k^*$ by the L-curve criterion.

|          | JBDGCV | JBDWGCV | JBDQR ($k^*$) | estimates for $k^*$ |
|----------|--------|---------|--------------|---------------------|
| $\varepsilon = 5 \cdot 10^{-2}$ |        |         |              |                     |
| rice     | 0.8778(3) | 0.8736(3) | 0.8397(5)    | 4(0.8411)           |
| rice(no L) | 0.1175(3) | 0.1142(3) | 0.0950(4)    |                     |
| mri      | 0.9602(3) | 0.9498(4) | 0.8848(13)   | 6(0.9007)           |
| mri(no L) | 0.3066(3) | 0.2932(3) | 0.2324(13)   |                     |
| blur30   | 0.9835(3) | 0.9827(3) | 0.9124(16)   | 5(0.9521)           |
| blur30(no L) | 0.5056(3) | 0.4999(3) | 0.3036(16)   |                     |
| blur422  | 0.9459(9) | 0.9443(10)| 0.9109(62)   | 24(0.9203)          |
| blur422(no L) | 0.2843(9) | 0.2823(10)| 0.2522(58)   |                     |
| $\varepsilon = 10^{-2}$ |        |         |              |                     |
| rice     | 0.8372(7) | 0.8363(7) | 0.7774(23)   | 11(0.7951)          |
| rice(no L) | 0.0931(7) | 0.0927(7) | 0.0764(22)   |                     |
| mri      | 0.8923(13)| 0.8782(19)| 0.8421(51)   | 25(0.8514)          |
| mri(no L) | 0.2367(13)| 0.2258(19)| 0.2024(50)   |                     |
| blur30   | 0.9697(6) | 0.9603(9) | 0.7975(65)   | 40(0.8243)          |
| blur30(no L) | 0.4238(5) | 0.3900(9) | 0.2095(65)   |                     |
| blur422  | 0.9459(9) | 0.9443(10)| 0.9109(62)   | 24(0.9203)          |
| blur422(no L) | 0.2843(9) | 0.2823(10)| 0.2522(58)   |                     |
| $\varepsilon = 10^{-3}$ |        |         |              |                     |
| rice     | 0.7638(38)| 0.7539(52)| 0.7136(166)  | 145(0.7140)         |
| rice(no L) | 0.0726(38)| 0.0705(52)| 0.0626(163)  |                     |
| mri      | 0.8305(101)| 0.8225(151)| 0.7949(451)  | 293(0.7988)         |
| mri(no L) | 0.1957(101)| 0.1917(151)| 0.1799(447)  |                     |
| blur30   | 0.9628(9) | 0.7984(75)| 0.5670(433)  | 577(0.5907)         |
| blur30(no L) | 0.3988(9) | 0.2060(75)| 0.1110(438)  |                     |
| blur422  | 0.9137(59)| 0.9046(103)| 0.8736(542)  | 284(0.8794)         |
| blur422(no L) | 0.2536(59)| 0.2471(103)| 0.2285(549)  |                     |

Figure 3 draws the convergence processes of JBDQR, JBDGCV and JBDWGCV for $\varepsilon = 10^{-2}$. We can see that the best regularized solutions by JBDQR are more accurate than the counterparts by JBDGCV and JBDWGCV; the convergence curves of JBDGCV and JBDWGCV first decrease with $k$, then increase for a while and finally stabilize, but JBDQR has typical semi-convergence phenomenons for all the problems.

A final note on Table 4 and Figure 3 is that the best regularized solutions by JBDWGCV are slightly more accurate than those by JBDGCV, which are different from the previous results in one dimensional case.

Figure 4 draws the exact images and the reconstructed images for the four test problems with $\varepsilon = 10^{-2}$. Clearly, the reconstructed images by JBDQR are at least as sharp as those by JBDGCV and JBDWGCV, and some of the former ones can be much sharper than the latter, e.g., blur30.
Fig. 3. The relative errors by JBDQR, JBDWGCV and JBDGCV with $\varepsilon = 10^{-2}$: (a) rice; (b) mri; (c) blur30; (d) blur422.

Fig. 4. The exact images and the reconstructed images for the four two dimensional test problems with $\varepsilon = 10^{-2}$: (a) rice; (b) mri; (c) blur30; (d) blur422.
Table 5

The relative errors and estimates for $k^*$ by the discrepancy principle.

|       | $\varepsilon = 5 \cdot 10^{-2}$ | $\varepsilon = 10^{-2}$ | $\varepsilon = 10^{-3}$ |
|-------|-------------------------------|--------------------------|----------------------------|
|       | $\tau = 1.005$ | $\tau = 1.1$ | $\tau = 1.2$ | $\tau = 2.0$ | $\tau = 1.005$ | $\tau = 1.1$ | $\tau = 1.2$ | $\tau = 2.0$ | $\tau = 1.005$ | $\tau = 1.1$ | $\tau = 1.2$ | $\tau = 2.0$ |
| rice  | 0.8462(3)        | 0.8556(2)       | 0.8556(2)       | 0.8791(1)     | 0.7989(10) | 0.8198(6) | 0.8267(5) | 0.8541(2) | 0.7288(62) | 0.7369(46) | 0.7430(38) | 0.7709(18) |
| mri   | 0.9007(6)        | 0.9156(4)       | 0.9267(3)       | 0.9653(1)     | 0.8564(21) | 0.8652(15) | 0.8729(12) | 0.8985(6) | 0.8121(141) | 0.8179(105) | 0.8223(85) | 0.8378(41) |
| blur30| 0.9181(12)       | 0.9575(5)       | 0.9700(3)       | 0.9849(1)     | 0.8019(56) | 0.8285(38) | 0.8564(27) | 0.9312(9) | 0.6083(241) | 0.6183(216) | 0.6286(194) | 0.6889(110) |
| blur422| 0.9444(6)       | 0.9549(3)       | 0.9608(2)       | 0.9707(1)     | 0.9230(21) | 0.9308(13) | 0.9367(9)  | 0.9501(4)  | 0.8861(182) | 0.8931(119) | 0.8972(92) | 0.9118(39) |

For $\varepsilon = 5 \cdot 10^{-2}$ and $10^{-3}$, we have similar findings to those by Figures 3–4.

7. Conclusions. In this paper, we have proposed a joint bidiagonalization based algorithm for solving large scale linear discrete ill-posed problems in general-form regularization. This algorithm is different from the hybrid projection based method proposed in [18], which exploits the same joint bidiagonalization process and explicitly regularizes each projected problem generated at every iteration.

We have analyzed the proposed algorithm and established a number of theoretical results. Particularly, we have proved that the iterates take the desired and attractive form of filtered GSVD expansions. These results rigorously show that the algorithm must possess the semi-convergence property and get more insight into the regularizing effects of the algorithm. In the meantime, we have considered efficient implementations of the algorithm and paid some attention to the discrepancy principle by highlighting a suitable choice of stopping criteria. Our algorithm is simpler and easier to implement than the hybrid one, and it is also more reliable than the latter.

We have made numerical experiments on a number of problems to justify numerous aspects of the proposed algorithm, e.g., solution accuracy and reliability. The results have illustrated that our algorithm often computes more accurate regularized solutions than the hybrid algorithm.

There are some important unsolved problems. As we have seen, a bottleneck of our algorithm and the hybrid one is solve a large scale least squares problem at each outer iteration, which is generally costly, especially when the solution accuracy of these problems is high. It is unclear if the solution accuracy can be relaxed substantially, at least at some outer iterations, similar to the randomized SVD algorithms proposed
in [16] that solve the general-form regularization problem (1.2). If they can be solved with considerably relaxed accuracy, we will gain much, and the overall efficiency of the algorithm can be improved substantially. The solution accuracy requirement on the inner least squares problems will constitute our forthcoming work.

REFERENCES

[1] R. C. Aster, B. Borchers, and C. H. Thurber, *Parameter Estimation and Inverse Problems*, Second Edition, Elsevier, New York, 2013.

[2] S. Berisha and J. G. Nagy, *Restore Tools: Iterative methods for image restoration*, 2012. Available from http://www.mathcs.emory.edu/~nagy/RestoreTools.

[3] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, PA, 1996.

[4] J. Chung and A. K. Saibaba, *Generalized hybrid iterative methods for large-scale Bayesian inverse problems*, SIAM J. Sci. Comput., 39 (2017), pp. S24–S46.

[5] J. Chung, A. K. Saibaba, M. Brown and E. Westman, *Efficient generalized Golub-Kahan based methods for dynamic inverse problems*, Inverse Probl., 34 (2018), 024005.

[6] H. W. Engl, *Regularization methods for the stable solution of inverse problems*, Surveys Math. Indust., 3 (1993), pp. 71–143.

[7] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, Kluwer Academic Publishers, 2000.

[8] S. Gazzola, P. C. Hansen and J. G. Nagy, *IR Tools: A Matlab package of iterative regularization methods and large-scale test problems*, arXiv:1712.05602v1 [math.NA], 2017.

[9] S. Gazzola and P. Novati, *Inheritance of the discrete Picard condition in Krylov subspace methods*, BIT Numer. Math., 56 (2016), pp. 893–918.

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

[11] Hansen. *Regularization tools version 4.0 for Matlab 7.3*, Numer. Algor., 46 (2007), pp. 189–194.

[12] Discrete Inverse Problems: Insight and Algorithms, SIAM, Philadelphia, PA, 2010.

[13] Z. Jia, *Approximation accuracy of the Krylov subspaces for linear discrete ill-posed problems*, arXiv:math.NA/1805.10132v1, 2018.

[14] Z. Jia and Y. Yang, *Modified Truncated randomized singular value decomposition (MTRSVD) algorithms for large scale discrete ill-posed problems with general-form regularization*, Inverse Probl., 34 (2018), 055031.

[15] *The low rank approximations and Ritz values in LSQR for linear discrete ill-posed problems*, manuscript, 2018.

[16] J. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, Applied Mathematical Sciences 160, Springer, 2005.

[17] M. E. Kilmer, P. C. Hansen, and M. I. Espanol, *A projection-based approach to general-form Tikhonov regularization*, SIAM J. Sci. Comput., 29 (2007), pp. 315–330.

[18] M. E. Kilmer and D. P. O’Leary, *Choosing regularization parameters in iterative methods for ill-posed problems*, SIAM J. Matrix Anal.Appl., 22 (2001), pp. 1204–1221.

[19] R. A. Renaut, S. Vatankhah, and V. E. Ardesta, *Hybrid and iteratively reweighted regularization by unbiased predictive risk and weighted GCV for projected systems*, SIAM J. Sci. Comput., 39 (2017), pp. B221–B243.

[20] K. Miller, *Least squares methods for ill-posed problems with a prescribed bound*, SIAM J. Math. Anal., 1 (1970), pp. 52–74.

[21] E. Natterer, *The Mathematics of Computerized Tomography*, John Wiley, New York, 1986.

[22] J. G. Nagy, K. Palmer, and L. Perrone, *Iterative methods for image deblurring: A Matlab object-oriented approach*, Numer. Algor., 36 (2004), pp. 73–93.

[23] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Maths. Soft., 8 (1982), pp. 43–71.

[24] B. N. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, PA, 1998.

[25] A. van der Sluis and H. A. van der Vorst, *The rate of convergence of conjugate gradients*, Numer. Math., 48 (1986), pp. 543–560.

[26] H. Zha, *Computing the generalized singular values/vectors of large sparse or structured matrix pairs*, Numer. Math., 72 (1996), pp. 391–417.