Applied Numerical Mathematics 157 (2020) 159–177

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

A joint bidiagonalization based iterative algorithm for large scale general-form Tikhonov regularization ☆

Zhongxiao Jia *, Yanfei Yang

Department of Mathematical Sciences, Tsinghua University, 100084 Beijing, China

ARTICLE INFO

Article history:
Received 10 September 2019
Received in revised form 11 April 2020
Accepted 2 June 2020
Available online 15 June 2020

Keywords:
Linear discrete ill-posed
General-form regularization
Joint bidiagonalization
GSVD
Filtered GSVD expansion
Semi-convergence
L-curve criterion
Discrepancy principle

ABSTRACT

Based on the joint bidiagonalization (JBD) process of the matrix pair $\{A, L\}$, an iterative regularization algorithm, called JBDQR, is proposed and developed for large scale linear discrete ill-posed problems in general-form Tikhonov regularization. It is proved that the JBDQR iterates take the form of attractive filtered generalized singular value decomposition (GSVD) expansions, where the filters are given explicitly and insightful. This result and a detailed analysis on it show that JBDQR must have the desired semi-convergence property, where the iteration number k plays the role of the regularization parameter. Embedded with the L-curve criterion or the discrepancy principle that is used to estimate the optimal k^* at which the semi-convergence occurs, JBDQR can compute a satisfying good regularized solution. JBDQR is theoretically solid and effective, and it is simple to implement. Numerical experiments confirm our results and the robustness of JBDQR.

1. Introduction

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

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

where $\|\cdot\|$ 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 many applications such as computerized tomography, image deblurring, signal processing, geophysics, heat propagation, biomedical and optical imaging, groundwater modeling, etc.; see, e.g., [1,2,6,11,19,22–24]. Since b contains the noise e and A is extremely ill conditioned, the naive solution $x_{naive} = A^{\dagger}b$ is very large in norm and is a meaningless approximation to the true solution $x_{true} = A^{\dagger}b_{true}$, where † denotes the Moore-Penrose inverse of a matrix. Therefore, one has to use regularization to obtain a good approximation to x_{true} [10,12].

Assume that $Ax_{true} = b_{true}$ with $m \geq n$ and e is white noise. Then two essentially equivalent dominating regularization approaches are the following discrepancy principle based general-form regularization

☆ This work was supported in part by the National Science Foundation of China (No. 11771249).
* Corresponding author.
E-mail addresses: jiazx@tsinghua.edu.cn (Z. Jia), yanfeiyang2018@gmail.com (Y. Yang).

https://doi.org/10.1016/j.apnum.2020.06.001
0168-9274/

$$\min \|Lx\| \quad \text{subject to} \quad x \in \mathcal{S} = \{x \mid \|Ax - b\| \le \tau \|e\|\} \tag{1.2}$$

with some $\tau \approx 1$ and the general-form Tikhonov regularization

$$\min_x \left\{ \|Ax - b\|^2 + \lambda^2 \|Lx\|^2 \right\}, \tag{1.3}$$

where $L \in \mathbb{R}^{p \times n}$ is a regularization matrix and $\lambda > 0$ is the regularization parameter; see [10, p. 11, 85, 105, 179] and [12, p. 63-4, 172, 181-2] for the equivalence of the above two formulations. We mention that the Tikhonov regularization (1.3) must require that e is white noise [10, p. 100], while the discrepancy principle based regularization formulation (1.2) is general and does not have such requirement. If $(A^T, L^T)^T$ has full column rank n, i.e.,

$$\mathcal{N}(A) \cap \mathcal{N}(L) = \{0\}, \tag{1.4}$$

the solution to (1.3) is unique for each λ. The equivalence of (1.2) and (1.3) does not require this condition as, for a given vector x, a common vector z in $\mathcal{N}(A)$ and $\mathcal{N}(L)$ does not affect the (semi-)norm $\|L(x+z)\| = \|Lx\|$ and the residual norm $\|A(x+z) - b\| = \|Ax - b\|$. In practical applications, L is typically chosen as a scaled approximation of the first or second derivative operator [10,12]. Particularly, if $L = I_n$, the $n \times n$ identity matrix, then (1.2) and (1.3) reduce to standard-form regularization problems.

For $L \ne I_n$, (1.2) and (1.3) can be mathematically transformed to the corresponding standard forms with $L = I_n$ and A replaced by $AL_A^\dagger$, where

$$L_A^\dagger = (I - (A(I - L^\dagger L))^\dagger A)L^\dagger \tag{1.5}$$

is *A-weighted pseudoinverse of* L and $L_A^\dagger = L^\dagger$ when $p \ge n$; see [10] for details. This is computationally viable and attractive if not much effort is needed by applying $L_A^\dagger$, e.g., when L is banded with small bandwidth and has a known null space; we refer the reader to, e.g., [4,5,8] for some available algorithms and codes. In many applications, however, such transformation is computationally unfeasible. In this paper, we assume that such transformation is not computationally viable and focus on solving the problems (1.2) and (1.3) directly.

Before proceeding, it is important to keep in mind two basic ingredients for solving (1.2) and (1.3) successfully; see, e.g., [10,12,20]: (i) A good regularized solution must capture the dominant generalized singular value decomposition (GSVD) components of the matrix pair $\{A, L\}$ and meanwhile suppress those corresponding to small generalized singular values. (ii) It is the generalized right singular vectors of the matrix pair $\{A, L\}$ *other than* the right singular vectors of A itself that form a more suitable basis to express a regularized solution [12, p. 179-181], as will also be reviewed in Section 2.

Exploiting the generalized Arnoldi process proposed by Li and Ye [21] for the solution of quadratic eigenvalue problems, Reichel *et al.* [29] present a hybrid iterative algorithm for solving (1.3). The generalized Arnoldi process successively reduces the square matrix pair $\{A, L\}$ to a sequence of small matrix pairs $\{H_A, H_L\}$, where the projection matrix H_L quickly becomes full as the number of iterations k increases. The reduction exploits only A and L but does not use their transposes, so the information on A^T and L^T is lacking. This may make the underlying subspace mix GSVD components at each iteration and thus unable to capture a dominant generalized right singular subspace of $\{A, L\}$. In fact, for $L = I_n$, Hansen in his book [12, p. 126] points out that Arnoldi process based methods, such as RRGMRES, mix the SVD components in each iteration, their success is highly problem dependent, and they can be successful when the mixing of the SVD components is weak, e.g., A is (nearly) symmetric. It follows from the above that generalized Arnoldi methods have similar limitations. Indeed, we have observed from [29] that the regularized solutions behaved quite irregularly when A is nonsymmetric; see Example 5.1 there. This means that the algorithm is not only hard to stop properly but also fail to solve the problem.

Hochstenbach *et al.* [13] propose an extended Golub-Kahan bidiagonalization process to reduce $\{A, L\}$ to a sequence of small matrix pairs $\{H_{k+1,k}, K_{k,k}\}$ with $H_{k+1,k}$ and $K_{k,k}$ being upper Hessenberg and triangular, respectively. The process degenerates to the standard Golub-Kahan bidiagonalization process when $L = I_n$. They then develop a hybrid projection algorithm for solving (1.3), which is projected onto a sequence of generalized Krylov subspaces generated by $A^T b$ and the matrices $A^T A$ and $L^T L$ simultaneously. The underlying solution subspace contains the standard Krylov subspace generated by $A^T b$ and $A^T A$ and that generated by $A^T b$ and $L^T L$. Each iteration computes the matrix-vector products with A^T, A, L^T and L and uses longer recurrences during orthonormalization of basis vectors as the iteration proceeds; see [13,31]. The Krylov subspace generated by $A^T b$ and $A^T A$ favors dominant right singular vectors of the single matrix A, but it generally bears no relation to dominant generalized right singular vectors of the matrix pair $\{A, L\}$.

Zha [32] proposes a joint bidiagonalization (JBD) process that successively reduces the matrix pair $\{A, L\}$ to upper bidiagonal forms. Based on this process, Zha proposes a JBD 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.* [20] adapt Zha's JBD process to discrete linear ill-posed problems in general-form Tikhonov regularization and develop a JBD process that successively reduces the matrix pair $\{A, L\}$ to lower and upper bidiagonal forms, respectively. Based on this process, they propose a hybrid projection method for solving (1.3). It is argued in [20] that the underlying solution subspaces are *legitimate* since they appear to be related to the generalized right singular vectors of $\{A, L\}$ more directly. At each iteration, the method needs 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 iteration. Therefore, the JBD process forms an inner-outer iterative process. Fortunately, $(A^T, L^T)^T$ is generally well conditioned as L is typically so in applications [10,12]. In these cases, the LSQR algorithm [26] can solve the mentioned least squares problems efficiently. At each iteration of the hybrid projection method [20], 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 when projection subspaces become sufficiently large, that is, the accuracy of regularized solutions cannot be improved. In the hybrid method, one needs to determine an optimal or suitable regularization parameter for each projected general-form Tikhonov regularization problem, and, in the meantime, has to judge when the projection subspace is large enough. Unfortunately, the meaning of 'sufficiently large' or 'large enough' has been vague, and no a deterministic definition has been given without ambiguity up to now.

It is well known from, e.g., [10,12], that any regularization is based on an underlying requirement that the discrete Picard condition for a given problem is satisfied, only under which can one compute a useful regularized solution. One fundamental fact that is crucial but has received little attention until the work [9,28] is: With $L = I_n$, the requirement that (1.3) satisfies the discrete Picard condition does not mean that the projected problems fulfill discrete Picard conditions too. Under the sufficient conditions that, at iteration k, the k singular values of the small matrix involved in the resulting projected problem approximate the largest singular values of A in natural order, that is, they interlace the first $k+1$ large singular values of A, the projected problems are proved to satisfy the discrete Picard conditions. Moreover, only under such conditions, it has been proved in [9,28] that the optimal regularization parameters $\mu_{opt}^{(k)}$ of projected problems converge to the optimal λ_{opt} of the original (1.3) as k increases [28]. These results carry over to the case that $L \neq I_n$. However, it has been proved in [14,15] that, for $L = I_n$ and LSQR, the approximations in natural order are guaranteed only for severely and moderately ill-posed problems and such property generally does not hold when the singular values of A do not decay so fast, e.g., those matrices in mildly ill-posed problems. For the definition of severely, moderately and mildly ill-posed problems, see [10,12] and the complement [14]. Importantly, these results imply that for a given hybrid projection based method, whenever the mentioned sufficient conditions fail to be fulfilled, a hybrid projection method may behave irregularly and the regularized solutions may not stabilize even when the projection subspaces have been sufficiently large, causing that the regularized solutions may fail to converge to the best regularized solution of (1.3) corresponding to the optimal λ_{opt}.

In this paper, instead of developing any hybrid projection based algorithm for solving (1.3), based on the JBD process [20], we will alternatively propose a projection iterative algorithm for solving the equivalent (1.2), where the iteration number k will play the role of the regularization parameter. First, we exploit the JBD process to project (1.2) onto a sequence of low dimensional subspaces and obtain a sequence of small projected problems, which involve matrix pairs of small size. Then at each outer iteration we solve a projected problem. We will prove that the solution of each projected problem reduces to that of an *ordinary* small least squares problem with the coefficient matrix being lower bidiagonal, so that it is very cheap and reliable to solve it by using the QR factorization at $\mathcal{O}(k)$ flops. Thus, we abbreviate the algorithm as JBDQR. One of the benefits is that we no longer determine an optimal or suitable regularization parameter for each projected general-form Tikhonov problem, which itself, unlike the optimal λ_{opt} of the original problem (1.3), may be hard to define because the projected problem may fail to satisfy the discrete Picard condition, so that regularized solutions may behave irregularly and may be hard to stabilize even though projection subspaces are expanded sufficiently large, as opposed to one's expectation. The other benefit is that we avoid being obsessed with the unclear definition of 'a projection subspace sufficiently large' in the hybrid projection method [20].

We will provide strong theoretical supports for JBDQR by establishing a number of results. Most importantly, we prove that the JBDQR iterates take the form of filtered GSVD expansions and are expressed *explicitly* in the generalized right singular vector basis of $\{A, L\}$, a desired and insightful property. Then we analyze the filters in the expansions, shed light on the regularizing effects of JBDQR, and prove that the iterates capture more and more dominant GSVD components of $\{A, L\}$ as the solution subspace is expanded. Precisely, we shall prove that JBDQR must have the desired semi-convergence property: As the solution subspace dimension k increases, more and more dominant generalized singular components of $\{A, L\}$ are captured, and the regularized solutions become increasingly better approximations to the true solution x_{true} of (1.1) until some iteration, afterwards the regularized solutions start to be deteriorated by the noise e and instead converge to the native solution $x_{naive} = A^{\dagger}b$. This is exactly the semi-convergence phenomenon, a desired property that a regularization method must have. These results rigorously show that the iteration number k plays the role of the regularization parameter. In computations, once the iteration k^*, at which the semi-convergence of JBDQR occurs and the best possible regularized solution is found, is determined or estimated, we accept the corresponding regularized solution as a practically best possible regularized solution obtained by JBDQR.

As it will turn out, since the residual norms monotonically decrease and the semi-norms of solutions practically increase monotonically, the L-curve criterion fits nicely into *approximately* determining the optimal regularization parameter k^*. If the noise level $\|e\|$ or its accurate estimate is known, the discrepancy principle is also a natural choice of estimating k^*. We will numerically justify the effectiveness and robustness of JBDQR and also compare JBDQR with the hybrid projection algorithm in [20], in which we make use of the GCV and WGCV parameter-choice methods [2,10,12,28] to approximately determine a suitable regularization parameter for each projected problem and find their best possible regularized solutions whenever they become (hopefully) stabilized for some iterations.

The paper is organized as follows. In Section 2, we overview GSVD, the general-form Tikhonov regularization method, the Truncated GSVD (TGSVD) method, and the JBD process of $\{A, L\}$. In Section 3, we propose the JBDQR algorithm for solving (1.2). In Section 4, we make a theoretical 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 in Section 7.

2. GSVD, general-form Tikhonov regularization and the JBD process

In this section we provide some necessary background. We describe GSVD, the general-form Tikhonov regularization, and the JBD process initially proposed in [32] for the computation of GSVD and adapted in [20] to the solution of (1.3).

Consider the compact QR factorization

$$\begin{pmatrix} A \\ L \end{pmatrix} = Q\,R, \tag{2.1}$$

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). From (2.1) we have $A = Q_A R$, $L = Q_L R$, 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

$$Q_A = P_A C W^T, \quad Q_L = P_L S W^T, \tag{2.2}$$

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

$$A = P_A C G^{-1}, \quad L = P_L S G^{-1} \tag{2.3}$$

with $G = (g_1, g_2, \ldots, g_n) = R^{-1}W \in \mathbb{R}^{n\times n}$, and the g_i are the generalized right singular vectors of $\{A, L\}$. We order the entries of the diagonal matrices C and S as

$$1 \ge c_1 \ge \cdots \ge c_{\min\{n,p\}} \ge 0, \; c_{\min\{n,p\}+1} = \cdots = c_n = 1, \tag{2.4}$$

$$0 \le s_1 \le \cdots \le s_{\min\{n,p\}} \le 1. \tag{2.5}$$

The numbers c_i/s_i, $i = 1, 2, \ldots, \min\{n, p\}$, are the *nontrivial* generalized singular values of $\{A, L\}$, and the infinitely large $+\infty$ corresponding to $c_i = 1$, $i = \min\{n, p\} + 1, \ldots, n$ is an $n - \min\{n, p\}$ multiple *trivial* generalized singular values. Note that if $\min\{n, p\} = n$ then $\{A, L\}$ has only nontrivial generalized singular values though some of them can also be $+\infty$, which corresponds to the case that $c_i = 1$, $s_i = 0$.

By the GSVD (2.3), the general-form Tikhonov regularized solution x_λ to (1.3) takes a filtered GSVD expansion

$$\begin{aligned} x_\lambda &= (A^T A + \lambda^2 L^T L)^{-1} A^T b = G(C^T C + \lambda^2 S^T S)^{-1} C^T P_A^T b \\ &= \sum_{i=1}^{\min\{n,p\}} \frac{c_i^2}{c_i^2 + \lambda^2 s_i^2} \frac{p_{i,A}^T b}{c_i} g_i + \sum_{i=\min\{n,p\}+1}^{n} p_{i,A}^T b\, g_i, \end{aligned} \tag{2.6}$$

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 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 a filtered GSVD form of x_λ, and a suitable λ must be such that x_λ captures certain dominant GSVD components and meanwhile suppresses those corresponding to small generalize singular values; see [10,12] for details. For our later use, write

$$g_\perp = \sum_{i=\min\{n,p\}+1}^{n} p_{i,A}^T b\, g_i \in \mathcal{N}(L). \tag{2.7}$$

The discrete Picard condition [10] states that the 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 Gaussian white noise e dominates the $|p_{i,A}^T b|$, that is, the $|p_{i,A}^T b| \approx |p_{i,A}^T e|$ stagnates and is dominated by e after $i > k_0$ for some k_0, while $|p_{i,A}^T b| \approx |p_{i,A}^T b_{true}| > |p_{i,A}^T e|$ is dominated by b_{true}, $i = 1, 2, \ldots, k_0$, where k_0 is called the transition point. Therefore, a good regularized solution x_λ must capture the k_0 dominant GSVD components of $\{A, L\}$ and meanwhile dampen those for $i > k_0$ as much as possible. A nearly optimal regularization parameter λ_{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 compute a sequence of the TGSVD solutions

$$x_k^{tgsvd} = \sum_{i=1}^{k} \frac{p_{i,A}^T b}{c_i} g_i + \sum_{i=\min\{n,p\}+1}^{n} p_{i,A}^T b g_i, \quad k = 1, 2, \ldots, \min\{n, p\}, \tag{2.8}$$

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_k^{tgsvd} and Lx_k^{tgsvd} converge to x_{true} and Lx_{true} for $k \le k_0$, afterwards they diverge and instead converge to x_{naive} and Lx_{naive}, respectively. A best TGSVD solution is $x_{k_0}^{tgsvd}$.

We notice that the second terms in (2.6) and (2.8) are the same and they disappear when $p \ge n$.

For A and L large, the computation of the GSVD of $\{A, L\}$ is infeasible, and the above TGSVD method and the Tikhonov regularization method are computationally impractical due to the excessive storage requirement and too high computational cost.

Now we review a procedure that jointly diagonalizes the matrix pair $\{A, L\}$ to lower and upper bidiagonal forms. Applying the lower and upper Lanczos bidiagonalization processes in [26] to Q_A and Q_L, respectively, one can reduce Q_A and Q_L to lower and upper bidiagonal forms:

$$Q_A V_k = U_{k+1} B_k, \quad Q_A^T U_{k+1} = V_k B_k^T + \alpha_{k+1} v_{k+1} e_{k+1}^T, \tag{2.9}$$

$$Q_L \widehat{V}_k = \widehat{U}_k \widehat{B}_k, \quad Q_L^T \widehat{U}_k = \widehat{V}_k \widehat{B}_k^T + \widehat{\beta}_k \widehat{v}_{k+1} e_k^T, \tag{2.10}$$

where e_{k+1} and e_k are the $(k+1)$th and kth canonical vectors of dimensions $k+1$ and k, respectively, the column orthonormal matrices

$$U_{k+1} = (u_1, \ldots, u_{k+1}) \in \mathbb{R}^{m\times(k+1)}, \quad \widehat{U}_k = (\widehat{u}_1, \ldots \widehat{u}_k) \in \mathbb{R}^{p\times k}, \tag{2.11}$$

$$V_k = (v_1, \ldots, v_k) \in \mathbb{R}^{n\times k}, \quad \widehat{V}_k = (\widehat{v}_1, \ldots, \widehat{v}_k) \in \mathbb{R}^{n\times k}, \tag{2.12}$$

and the lower and upper bidiagonal matrices

$$B_k = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix} \in \mathbb{R}^{(k+1)\times k}, \quad \widehat{B}_k = \begin{pmatrix} \widehat{\alpha}_1 & \widehat{\beta}_1 & & \\ & \widehat{\alpha}_2 & \ddots & \\ & & \ddots & \widehat{\beta}_{k-1} \\ & & & \widehat{\alpha}_k \end{pmatrix} \in \mathbb{R}^{k\times k}. \tag{2.13}$$

Zha [32] and Kilmer *et al.* [20] have investigated intimate relationships between the above two processes and the simultaneous reduction of $\{A, L\}$. For $v_1 = \hat{v}_1$, they have established the following results (cf. [20, Theorems 2.1-2]).

Assume that $A \in \mathbb{R}^{m\times n}$ and $L \in \mathbb{R}^{p\times n}$ with $m \ge n$. Then there exist orthogonal matrices $U \in \mathbb{R}^{m\times m}$, $\widehat{U} \in \mathbb{R}^{p\times p}$ and $V_n \in \mathbb{R}^{n\times n}$, and a lower bidiagonal $B_n \in \mathbb{R}^{m\times n}$, an upper bidiagonal $\bar{B}_n \in \mathbb{R}^{p\times n}$, and an invertible Z_n such that

$$A = U B_n Z_n^{-1}, \tag{2.14}$$

$$L = \widehat{U} \bar{B}_n Z_n^{-1}, \tag{2.15}$$

where $Z_n = R^{-1} V_n$ and $\bar{B}_n = \widehat{B}_n D$ with $D = \mathrm{diag}(1, -1, 1, -1, \ldots)$, and the remaining matrices are obtained by running (2.9) and (2.10) to completion. In particular, when $p < n$, the columns $p+1, \ldots, n$ of $\bar{B}_n$ contain only zero entries.

From (2.14) and (2.15), we obtain k-step JBD relations

$$A Z_k = U_{k+1} B_k, \tag{2.16}$$

$$L Z_k = \widehat{U}_k \bar{B}_k, \tag{2.17}$$

where $Z_k = R^{-1} V_k \in \mathbb{R}^{n\times k}$, and B_k and $\bar{B}_k$ are the first $(k+1)\times k$ and $k\times k$ submatrices of B_n and $\bar{B}_n$, respectively.

For A and L large, the QR factorization (2.1) is impractical. In order to avoid explicitly computing Q_A and Q_L, inspired by Zha's work [32], Kilmer *et al.* [20] develop a 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.

Let $\widetilde{u}_i = \begin{pmatrix} u_i \\ 0_p \end{pmatrix}$. For $i = 1, 2, \ldots, k+1$, Algorithm 1 needs to compute $Q\,Q^T \widetilde{u}_i$. Notice that $Q\,Q^T \widetilde{u}_i$ is nothing but the orthogonal projection of $\widetilde{u}_i$ onto the column space of $\begin{pmatrix} A \\ L \end{pmatrix}$, which means that $Q\,Q^T \widetilde{u}_i = \begin{pmatrix} A \\ L \end{pmatrix} \widetilde{x}_i$ with $\widetilde{x}_i$ being the solution to the least squares problem

$$\widetilde{x}_i = \arg\min_{\widetilde{x}\in\mathbb{R}^n} \left\| \begin{pmatrix} A \\ L \end{pmatrix} \widetilde{x} - \widetilde{u}_i \right\|. \tag{2.18}$$

Algorithm 1 k-step joint bidiagonalization (JBD) process.

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

Since this inner 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 [26], called inner iteration.

Remark 2.1. Relations (2.16) and (2.17) hold when, at outer iteration i, the inner least squares problem (2.18) is solved accurately. It is unknown if the solution accuracy can be relaxed by allowing possibly large inexactness in the algorithm [20] and ours to be presented later. We do not investigate it in the current paper. In computations, we solve (2.18) by using the Matlab function lsqr with the default stopping criterion 10^{-6}.

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

3. The JBDQR algorithm

Instead of solving (1.3) by the JBD based hybrid projection method in [20], we shall propose a JBD process based projection algorithm for solving the equivalent (1.2). The method is simpler and easier to implement than the hybrid one, and at each outer iteration it solves a small projected problem of (1.2), which is different from the projected general-form Tikhonov regularization problem resulting from (1.3), whose discrete Picard condition may fail to be fulfilled so that an inner optimal regularization parameter may be problematic for the projected general-form Tikhonov regularization problem. We will establish a number of theoretical results and get insight into the regularizing effects of the method. Particularly, we prove that the method must have the desired semi-convergence property and the iteration number k will play the role of the regularization parameter in the new method.

In Algorithm 1, we have

$$U_{k+1}(\beta_1 e_1) = b, \quad \beta_1 = \|b\|, \tag{3.1}$$

where e_1 is the first canonical vector of dimension $k+1$. Let $\widetilde{V}_k = (\widetilde{v}_1, \ldots, \widetilde{v}_k) \in \mathbb{R}^{(m+p)\times k}$ be generated by Algorithm 1. Then

$$\widetilde{V}_k = Q\,V_k = Q\,R(R^{-1}V_k) = \begin{pmatrix} A \\ L \end{pmatrix} Z_k, \tag{3.2}$$

We now project the large scale regularization problem (1.2) onto a sequence of low dimensional subspaces $\mathrm{span}\{Z_k\}$ and compute regularized solutions $x_k \in \mathrm{span}\{Z_k\}$, which can be written as

$$x_k = Z_k y_k. \tag{3.3}$$

From (3.3), the above projection is equivalent to replacing A and L by AZ_k and LZ_k in (1.2) and solving the *reduced* general-form regularization problem

$$\min \|LZ_k y\| \quad \text{subject to} \quad y \in \{y \mid \|AZ_k y - b\| = \min\} \tag{3.4}$$

for y_k, starting with $k = 1$ until $\|AZ_k y - b\| \le \tau\|e\|$ with $\tau \approx 1$. Make use of (3.1), (2.16) and (2.17). Then the problem (3.4) becomes the *reduced* general-form regularization problem

$$\min \|\bar{B}_k y\| \quad \text{subject to} \quad y \in \{y \mid \|B_k y - \beta_1 e_1\| = \min\} \tag{3.5}$$

starting with $k = 1$ onwards. After the solution y_k to problem (3.5) is computed, in terms of (3.2) and (3.3), we compute x_k by solving

$$\begin{pmatrix} A \\ L \end{pmatrix} x_k = \widetilde{V}_k y_k. \tag{3.6}$$

Now let us investigate the solution of problem (3.5).

Theorem 3.1. *Assume that Algorithm 1 does not break down at iteration $k \leq \min\{n, p\}$. Then the solution y_k to (3.5) is*

$$y_k = \arg\min_{y\in\mathbb{R}^k} ||B_k y - \beta_1 e_1|| = \beta_1 B_k^{\dagger} e_1. \tag{3.7}$$

Proof. We give two proofs. The first one is as follows. Let $\widetilde{y} = \bar{B}_k y$. Then under the assumption on Algorithm 1, $\bar{B}_k$ is nonsingular. Therefore, (3.5) is equivalent to

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

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

$$\begin{aligned}\widetilde{y}_k &= \beta_1 (B_k \bar{B}_k^{-1})^{\dagger} e_1 \\ &= \beta_1 \bar{B}_k B_k^{\dagger} e_1\end{aligned}$$

with the second equality holding because B_k is of column full rank and $\bar{B}_k$ is nonsingular. Then the solution y_k to (3.5) is

$$y_k = \bar{B}_k^{-1}\widetilde{y}_k = \beta_1 \bar{B}_k^{-1}\bar{B}_k B_k^{\dagger} e_1 = \beta_1 B_k^{\dagger} e_1.$$

The second proof is as follows. We investigate the set of solution candidates such that $\|B_k y - \beta_1 e_1\| = \min$ in (3.5). Note that B_k is of full column rank. Therefore, the null space $\mathcal{N}(B) = \{0\}$, and the solution y_k to $\min_{y\in\mathbb{R}^k} \|B_k y - \beta_1 e_1\|$ is uniquely given by $y_k = \beta_1 B_k^{\dagger} e_1$. As a result, such y_k solves (3.5). □

Relation (3.7) indicates that the solution y_k to (3.5) is simply the solution to the ordinary least squares problem $\min_{y\in\mathbb{R}^k} ||B_k y - \beta_1 e_1||$ and $\bar{B}_k$ is not invoked. Let

$$B_k = Q_k R_k \tag{3.8}$$

be the compact QR factorization of B_k, which can be efficiently computed by exploiting Givens rotations at cost of $\mathcal{O}(k)$ flops. From (3.7), we obtain

$$y_k = \beta_1 R_k^{-1} Q_k^T e_1 \tag{3.9}$$

at cost of $\mathcal{O}(k)$ flops; see [26] for details. The above JBD process based procedure for computing x_k is called JBDQR.

Next, we consider the efficient computation of the residual norm $\|Ax_k - b\|$ and the semi-norm $\|Lx_k\|$ without explicitly forming x_k itself. Similar to Theorem 3.1 in [20], it is straightforward to establish the following results.

Theorem 3.2. *Let the matrices B_k and $\bar{B}_k$ be defined in (2.16) and (2.17). Then*

$$\|Ax_k - b\| = \|B_k y_k - \beta_1 e_1\|, \tag{3.10}$$

$$\|Lx_k\| = \|\bar{B}_k y_k\|. \tag{3.11}$$

Proof. Notice $x_k = Z_k y_k$. Exploiting (3.1) and (2.16) yields

$$Ax_k = AZ_k y_k = U_{k+1} B_k y_k.$$

Since U_{k+1} is column orthonormal, it is straightforward to obtain (3.10) by the orthogonal invariance of the 2-norm. Similarly, we have

$$Lx_k = LZ_k y_k = \widehat{U}_k \bar{B}_k y_k. \tag{3.12}$$

Since $\widehat{U}_k$ is column orthonormal, (3.11) holds. □

This theorem shows that, by exploiting the structures of B_k and $\bar{B}_k$, both $\|Ax_k - b\|$ and $\|Lx_k\|$ can be computed at cost of $\mathcal{O}(k)$ flops without forming x_k explicitly. We only need to compute x_k by solving (3.6) when x_k is accepted as a regularized solution. We sketch the above JBQR as Algorithm 2 and postpone some details to Section 5.

Algorithm 2 JBDQR for solving (1.2).

1: Starting with $k = 1$, run Algorithm 1, and obtain the small projected problem (3.5).
2: Use the QR factorization to compute the solution y_k, defined by (3.9), to (3.5).
3: Compute $\|Ax_k - b\|$ and $\|Lx_k\|$ by the formulas (3.10) and (3.11).
4: If x_k is accepted as the final regularized solution, then compute it by solving (3.6).

4. Regularization properties of JBDQR

In this section we establish some important theoretical results on JBDQR and get insight into its regularizing effects. Let $\widetilde{w} = Rx$. Then by (2.1), we have

$$\min_{x\in\mathbb{R}^n} \|Ax - b\| = \min_{\widetilde{w}\in\mathbb{R}^n} \|Q_A\widetilde{w} - b\|. \tag{4.1}$$

First, similar to Theorem 4.3 and (3.4) in [20], it is straightforward to derive the following result.

lemma 4.1. *Let x_k be the regularized solution obtained by our algorithm. Then*

$$x_k = R^{-1}\widetilde{w}_k,\ \ \widetilde{w}_k = \arg\min_{\widetilde{w}\in\mathcal{K}_k} \|Q_A\widetilde{w} - b\|, \tag{4.2}$$

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

$$\mathcal{K}_k = \mathrm{span}\{Q_A^T b, (Q_A^T Q_A)Q_A^T b, \ldots, (Q_A^T Q_A)^{k-1}Q_A^T b\}, \tag{4.3}$$

and the solution subspace

$$\mathrm{span}\{Z_k\} = R^{-1}\mathcal{K}_k = \mathrm{span}\{G(C^TC)^i C^T P_A^T b\}_{i=0}^{k-1}, \tag{4.4}$$

where the matrices P_A and C are defined by (2.2).

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

$$\min_{\widetilde{w}\in\mathcal{K}_k} \|Q_A\widetilde{w} - b\| = \min_{y\in\mathbb{R}^k} \|B_k y - \beta_1 e_1\|.$$

Let $y_k = \arg\min_y \|B_k y - \beta_1 e_1\|$. By the definition (3.2) of Z_k, we have $x_k = Z_k y_k = R^{-1}V_k y_k = R^{-1}\widetilde{w}_k$. By (2.2) and (2.3), a direct justification shows (4.4). □

To present our main theoretical result and make an insightful analysis on the regularizing effects of JBDQR, we need to make some necessary preparations and notation changes. We consider two cases that $p \geq n$ and $p < n$, respectively.

For the case that $p \geq n$, if the rank of $L \in \mathbb{R}^{p\times n}$ is equal to n, 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 c_i and s_i must satisfy

$$0 < c_i,\ s_i < 1,\ i = 1, 2, \ldots, n.$$

Therefore, all the generalized singular values c_i/s_i are finite. On the other hand, if the rank of L is smaller than n, then some $c_i = 1$ and $s_i = 0$, and $\{A, L\}$ has some nontrivial *infinitely* large generalized singular value(s). However, whether or not the rank of L is equal to n makes no difference on our later results and analysis in the solution of (1.2), though it may cause some difficulties when numerically computing several GSVD components of $\{A, L\}$ corresponding to the largest generalized singular values. In this case, we retain the notations (2.3), (2.4) and (2.5).

For the case that $p < n$, differently from (2.4), we relabel the c_i and s_i, and use the new notation

$$1 = c_1 \geq c_2 \geq c_3 \geq \cdots \geq c_{p+1}, \tag{4.5}$$

where $c_1 = 1$ is the largest singular value of Q_A with the multiplicity $n - p$, which corresponds to the trivial infinitely large generalized singular value of $\{A, L\}$. That is, we reassign the indices i of c_i defined by (2.4) to $i + 1$, and shift the largest singular value of Q_A to $c_1 = 1$ with the multiplicity $n - p$ in (2.4). Correspondingly, we permute the columns of P_A and W in Q_A defined by (2.2), and G defined by (2.3) by moving their respective last $n - p$ columns to the front and naming them as $P_{1,A}, W_1$ and G_1, and then rewrite

$$P_A := (P_{1,A}, p_{2,A}, \ldots, p_{m-n+p+1,A}), \tag{4.6}$$

$$W := (W_1, w_2, \ldots, w_{p+1}), \tag{4.7}$$

$$G := (G_1, g_2, \ldots, g_{p+1}). \tag{4.8}$$

With the new notation, 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 are the same as in (2.2) and (2.3), and $\mathcal{N}(L) = \{\mathbf{0}\}$.

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 $Q_A^T Q_A$ independently of the multiplicities of the c_i^2, that is, the method cannot determine true multiplicities of c_i^2 without restarts and works as if $Q_A^T Q_A$ has only simple eigenvalues c_i^2 [27]. 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 $\widetilde{c}_i$, 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. These properties illustrate that the k-step Lanczos bidiagonalization method uses the $\widetilde{c}_i$ as approximations to the k *distinct* singular values c_i of Q_A. For a rigorous and complete derivation and many details, we refer the reader to [17].

Analogously to the filtered SVD expansion [10, p. 146] of the LSQR iterates for solving (1.2) with $L = I_n$, we next establish an attractive and desired property that the regularized solution x_k has a filtered GSVD expansion which is *explicitly* expressed in the generalized right singular vector basis $\{g_i\}_{i=1}^n$ of $\{A, L\}$. We then make a detailed analysis on the result and get insight into the regularizing effects of JBDQR.

Theorem 4.1. *Assume that the c_i, $i = 2, 3, \ldots, p+1$ are labeled as (4.5) for $p < n$, the matrices P_A, W and G defined as above, and $g_\perp$ defined by (2.7). Then*

$$x_k = f_1^{(k)} g_\perp + \sum_{i=2}^{p+1} f_i^{(k)} \frac{p_{i,A}^T b}{c_i} g_i, \ k = 1, 2, \ldots, n, \tag{4.9}$$

where the filters

$$f_i^{(k)} = 1 - \prod_{j=1}^{k} \frac{\widetilde{c}_j^2 - c_i^2}{\widetilde{c}_j^2}, \ i = 1, 2, \ldots, p+1. \tag{4.10}$$

For $p \geq n$, we have

$$x_k = \sum_{i=1}^{n} f_i^{(k)} \frac{p_{i,A}^T b}{c_i} g_i, \ k = 1, 2, \ldots, n, \tag{4.11}$$

where the filters

$$f_i^{(k)} = 1 - \prod_{j=1}^{k} \frac{\widetilde{c}_j^2 - c_i^2}{\widetilde{c}_j^2}, \ i = 1, 2, \ldots, n. \tag{4.12}$$

Proof. Notice that the LSQR algorithm with the starting vector $u_1 = b/\|b\|$ applied to (4.2) 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 (4.2) with the starting vector $w_0 = 0$. Let $w_{ls} = Q_A^\dagger b$ be the solution to $\min_{\widetilde{w} \in \mathbb{R}^n} \|Q_A \widetilde{w} - b\|$. Then exploiting the SVD (2.2) of Q_A and (4.5)–(4.7), we obtain

$$w_{ls} = W_1 P_{1,A}^T b + \sum_{i=2}^{p+1} \frac{p_{i,A}^T b}{c_i} w_i, \tag{4.13}$$

where the first term is the SVD component of w_{ls} in the right singular subspace of Q_A corresponding to the largest singular value $c_1 = 1$.

Keep in mind a well-known result (cf., e.g., [30, Property 2.8]) on the CG iterates that, in our case, states

$$\widetilde{w}_k = (I - q_k(Q_A^T Q_A)) w_{ls}, \tag{4.14}$$

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 $\widetilde{c}_j^2$ of $Q_A^T Q_A$ with respect to $\mathcal{K}_k$ defined by (4.3), i.e.,

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

Substituting (4.13) into (4.14) yields

$$\widetilde{w}_k = f_1^{(k)} W_1 P_{1,A}^T b + \sum_{i=2}^{p+1} f_i^{(k)} \frac{p_{i,A}^T b}{c_i} w_i,\ k = 1, 2, \ldots, p+1 \tag{4.15}$$

with $f_i^{(k)}$ defined by (4.10).

Recall from (2.1)–(2.3) that $G = R^{-1}W$. It then follows from (4.7) and (4.8) that $G_1 = R^{-1}W_1$ and $g_i = R^{-1}w_i$, $i = 2, 3, \ldots, p+1$. From (4.2), since $x_k = R^{-1}\widetilde{w}_k$, premultiplying (4.15) by R^{-1} establishes (4.9) by noticing that $G_1 P_{1,A}^T b = g_\perp$ in our new notation.

If $p \geq n$, then $g_\perp = 0$ in (2.7), the first term in (4.9) is zero, and the second term becomes

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

which proves (4.11) with the filters $f_i^{(k)}$ defined by (4.12). □

Clearly, for $p \geq n$, (4.11) is a filtered GSVD expansion similar to (2.6). If $p < n$, the first term

$$f_1^{(k)} g_\perp \in \mathcal{N}(L)$$

in (4.9), which resembles the term $g_\perp$ in (2.6). On the other hand, the second term

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

in (4.9) corresponds to the first term in (2.6) by noticing that in our notation the indices $i+1$ in the sum correspond to the indices i in (2.6). A difference is that the general-form Tikhonov regularization solution (2.6) does not affect $g_\perp$, while our algorithm multiplies it by a factor $f_1^{(k)}$. Nonetheless, $f_1^{(k)} \to 1$ as $\widetilde{c}_1$ converges to c_1. Since $g_\perp$ and $f_1^{(k)} g_\perp$ lie in $\mathcal{N}(L)$, they have no effect on Lx_λ and Lx_k.

It is known from [10, Theorem 2.1.1, p. 23] that the c_i decay like the singular values σ_i of A when the matrix $(A^T, L^T)^T$ is well conditioned, which is guaranteed since, in applications, L is typically well conditioned. In the meantime, notice from (2.2) and (2.3) that Q_A and A share the same P_A and the problems $\min_{x\in\mathbb{R}^n} \|Ax - b\|$ and $\min_{\widetilde{w}\in\mathbb{R}^n} \|Q_A w - b\|$ have the same right-hand side b. Therefore, these two problems satisfy the same discrete Picard condition, and JBDQR for the former and LSQR for the latter exhibit the same regularizing effects.

Moreover, as has been proved in [15], 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 $\min_{\widetilde{w}\in\mathbb{R}^n} \|Q_A \widetilde{w} - b\|$. From (4.9) and (4.10), 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 $\widetilde{c}_j$ approximate the large singular values of Q_A in natural order. If the singular values of B_k do not approximate the large singular values of Q_A in natural order for some k, it has been shown in [14,15] that the regularized solutions $\widetilde{w}_k$ in (4.14) converge to w_{ls} until some iteration and afterwards they diverge. Therefore, in any event, the semi-convergence must occur at some iteration. We refer the reader to [10, pp. 146-148] and [14,15] for more details. Similar results have been established for the Conjugate Gradient Minimal Error (CGME) method, a regularizing Krylov solver, in [16]. If Q_A has multiple singular values, we refer the reader to [17].

Due to the equivalence (4.1) and $x_k = R^{-1}\widetilde{w}_k$, the results of [14,15] can be naturally adapted to our current context. Theorem 4.1 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 until the semi-convergence of JBDQR. Precisely, JBDQR exhibits typical semi-convergence at some iteration k^*: x_k (resp. Lx_k) converges to x_{true} (resp. Lx_{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, and the semi-convergence of JBDQR occurs at iteration k^*.

We should point out that the size of k^* may differ when determining the semi-convergence in terms of different errors $\|x_k - x_{true}\|$ and $\|L(x_k - x_{true})\|$. In either case, the optimal k^* should be such that either $\|x_k - x_{true}\|$ or $\|L(x_k - x_{true})\|$ attains its minimum at k^* over all $k = 1, 2, \ldots, n$. It is worthwhile to notice that, according to [10, Theorems 4.5.1-2], for general-form Tikhonov regularization, theoretically it may be more meaningful to check the semi-convergence by $\|L(x_k - x_{true})\|$. This measure is implicitly reflected and supported by some parameter-choice methods, e.g., the L-curve criterion, where we draw the logarithm of the semi-norm $\|Lx_k\|$ other than that of the standard 2-norm $\|x_k\|$ and the overall corner of the L-curve (approximately) attains at the point $(\log\|Ax_{k^*} - b\|, \log\|Lx_{k^*}\|)$ on the curve; see [10] for more details. In the sequel, when speaking of semi-convergence, we mean the measure $\|L(x_k - x_{true})\|$.

5. The determination of an optimal regularization parameter k^*

For JBDQR, since the residual norm $\|Ax_k - b\| = \|B_k y_k - \beta_1 e_1\|$ monotonically decreases and the semi-norm $\|Lx_k\| = \|\bar{B}_k y_k\|$ monotonically increases practically with respect to k, the L-curve criterion suits well for a practical determination of k^* independently of whether or not the noise level $\|e\|$ is known. We plot the curve

$$\big(\log(\|B_k y_k - \beta_1 e_1\|), \log(\|\bar{B}_k y_k\|)\big)$$

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 [10–12].

If $\|e\|$ or its accurate estimate is known in advance, the discrepancy principle [7,10,12] is the simplest parameter-choice method. We stop the algorithm at the first iteration k satisfying

$$\|Ax_k - b\| = \|B_k y_k - \beta_1 e_1\| \le \tau \|e\| \tag{5.1}$$

with $\tau \approx 1$, e.g., $\tau = 1.1$ or smaller. We then use such k as an estimate of the optimal regularization parameter k^*. We mention that a $\tau > 1$ considerably, e.g., $\tau = 2$, is generally unsafe and may underestimate k^* substantially.

Embedded with the above parameter-choice methods, we can now present our complete JBDQR algorithm, named Algorithm 3.

Algorithm 3 (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 (3.5).
2: Use the QR factorization to compute the solution y_k to (3.5).
3: Compute $\|Ax_k - b\|$ and $\|Lx_k\|$ by the formulas (3.10) and (3.11).
4: Determine the optimal regularization parameter k^* by the L-curve criterion or check if the discrepancy principle (5.1) 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 (3.6).

Finally, we should say something on the efficiency comparison of Algorithm 3 and the hybrid projection algorithm [20]. It is obvious that at outer iteration k the two algorithms cost completely the same since they use the same JBD process to obtain the same solution subspace and solve the same inner least squares problem (2.18). This means that, at each outer iteration and for the same prescribed stopping criterion for inner iterations, LSQR uses the same iterations, i.e., the same numbers of matrix-vector product evaluations required with A, A^T, L, L^T to terminate inner iterations in Algorithm 3 and the hybrid projection algorithm [20]. The cost gain of JBDQR consists in the high efficiency of computing y_k in (3.9), which costs only $\mathcal{O}(k)$ flops, cheaper than the hybrid algorithm in [20], which needs to solve the $(2k+1)\times k$ least squares problems with an optimal or suitable inner regularization parameter to be determined and costs $\mathcal{O}(k^3)$ flops and at least $\mathcal{O}(k^2)$ flops, no matter which parameter-choice method is exploited to compute the inner regularization parameter for each projected problem. Clearly, compared to the cost of one step JBD process, solving inner least squares problems are insignificant. These show that if the same iterations are performed then Algorithm 3 is a little cheaper than the hybrid projection algorithm [20]. Therefore, the key of efficiency comparison relies on how many outer iterations each algorithm uses to solve a given problem successfully, and the fewer outer iterations are used, the more efficient an algorithm is.

6. Numerical experiments

In this section, we report numerical experiments to demonstrate that the proposed JBDQR method embedded with the L-curve criterion and the discrepancy principle works well and the regularized solutions obtained by it are at least as accurate as and, in fact, are often considerably more accurate than those obtained by the hybrid method in [20]. We also compare the regularization parameters estimated by the L-curve criterion and the discrepancy principle with the 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 [2,25]; 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

$$\varepsilon = \frac{\|e\|}{\|b_{true}\|}. \tag{6.1}$$

For the noise-free problems $Ax_{true} = b_{true}$ in Table 1, we add a Gaussian white noise e with zero mean and a prescribed relative noise level ε to b_{true} and form the noisy $b = b_{true} + e$. To simulate exact arithmetic, complete reorthogonalization is used in Algorithm 1.

We abbreviate Algorithm 3 as JBDQR, the hybrid one in [20] using the GCV and Weighted GCV (WGCV) parameter-choice methods [2,10,12,28] as JBDGCV and JBDWGCV, respectively. Let x_k^{reg} denote the regularized solutions obtained by each of the three algorithms. As we have addressed previously, in terms of [10, Theorems 4.5.1-2], in the general-form regularization context it is more instructive and suitable to use the relative error

$$\frac{\|L(x_k^{reg} - x_{true})\|}{\|Lx_{true}\|} \tag{6.2}$$

other than the standard relative error of x_k^{reg} to plot the regular 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

Table 1
The description of test problems.

Problem	Description
shaw	One-dimensional image restoration model [11]
baart	First kind Fredholm integral equation [11]
heat	Inverse heat equation [11]
deriv2	Computation of second derivative [11]
rice	Two dimensional image deblurring
mri	Two dimensional image deblurring
AtmosphericBlur30	Two dimensional image deblurring [2,25]
GaussianBlur422	Two dimensional image deblurring [2,25]

Table 2
The relative errors defined by (6.2) and the estimates for the optimal regularization parameters k^* by the L-curve criterion for the first four test problems in Table 1 with $L = L_1$.

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

in parentheses, the minimal relative errors by JBDQR at the semi-convergence and the optimal k^* in the parentheses, and the estimated ones for k^* determined by the L-curve criterion and the discrepancy principle (5.1) and the corresponding relative errors in the parentheses. We use the Matlab function lsqr to solve (2.18) and (3.6) with the default stopping tolerance $tol = 10^{-6}$.

All the computations were preformed using Matlab R2015b 64-bit on Intel Core i3-2120 CPU 3.30 GHz 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

In Table 1, the test problems shaw and baart are severely ill-posed, and heat and deriv2 are moderately ill-posed. 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 "$example = 1, 3$". In the experiments, for shaw and baart, we take $m = n = 1024$, and heat and deriv2, we take $m = n = 3000$. Purely for test purposes, we choose

$$L = L_1 = \begin{pmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{pmatrix} \in \mathbb{R}^{(n-1)\times n}, \tag{6.3}$$

which is a scaled discrete approximation of the first derivative operator in the one dimensional case. We comment that for the scaled discrete approximation of the second derivative operator, we have observed very similar phenomena. Hence we only report the results on $L = L_1$. Here we remind an obvious fact that the three algorithms JBDQR, JBDGCV and JBDWGCV are all based on the JBD process and never explicitly form or compute the A-weighted pseudoinverses, defined by (1.5), of such L and later ones (cf. (6.4)) in the two dimensional case.

In Table 2, we display the relative errors of the best regularized solutions by JBDQR with the L-curve criterion, JBDGCV and JBDWGCV with $L = L_1$ and $\varepsilon = 10^{-3},\ 10^{-4}$, respectively, where ε is defined by (6.1). As we can see from the table, the regularized solutions by JBDQR are at least as accurate as and can be considerably more accurate than those by JBDGCV and JBDWGCV for all the test problems; see, e.g., the results on deriv2. Here the regularized solutions by JBDQR correspond to the last (fifth) column of the table. We observe from the table that for each test problem the best regularized solution by JBDQR at semi-convergence is correspondingly more accurate and requires a bigger k^* for a smaller ε. All these are expected and justify that the smaller ε 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 ε, almost all the

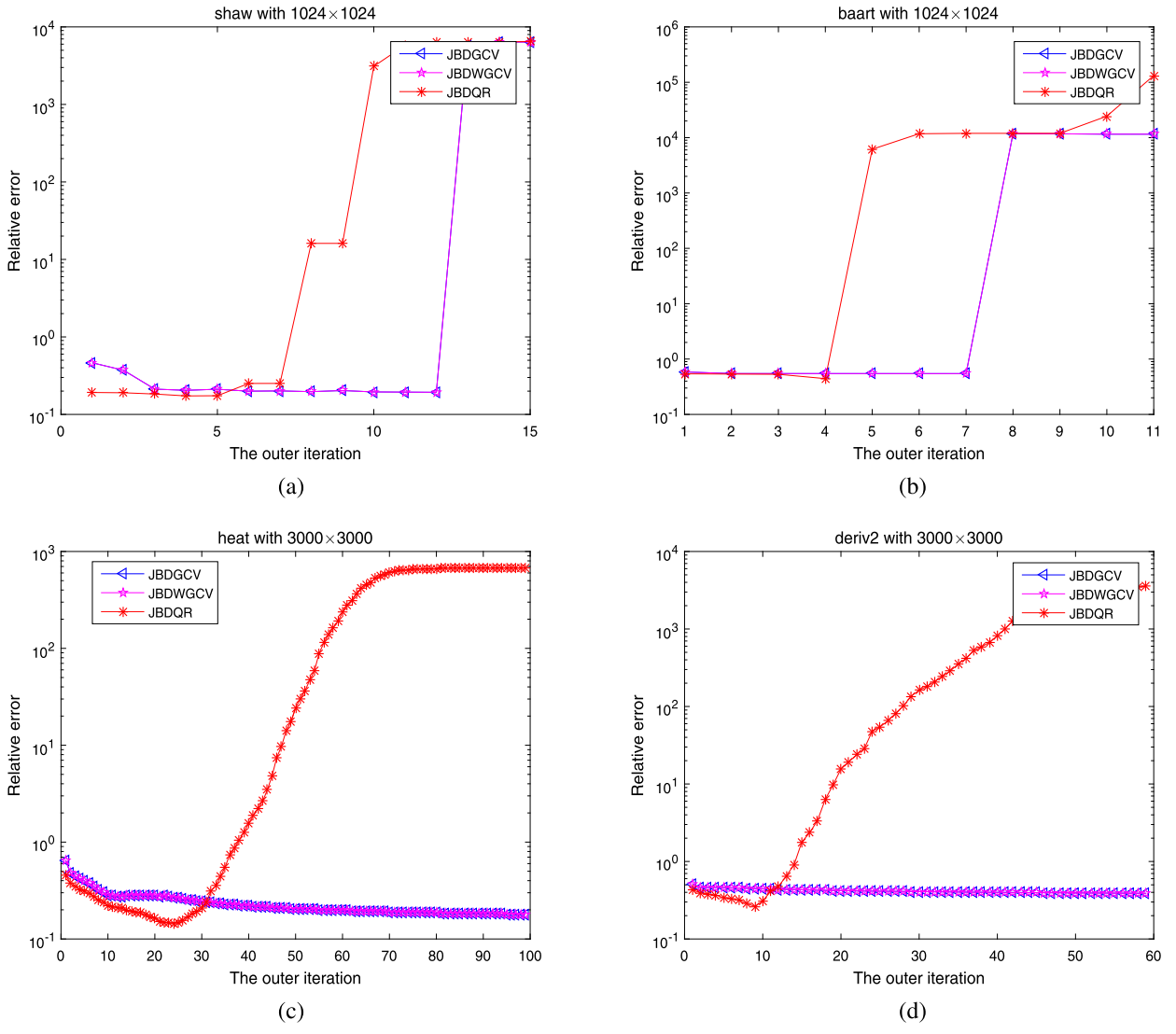

Fig. 1. The relative errors, defined by (6.2), of JBDQR, JBDGCV and JBDWGCV with $L = L_1$ and $\varepsilon = 10^{-3}$: (a) shaw; (b) baart; (c) heat; (d) deriv2.

estimates for the optimal 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 oversmooths the regularized solutions. Even so, by comparing the fourth and fifth columns of the table, we see that the regularized solutions obtained by JBDQR associated with estimated k^*'s are almost as accurate as the best regularized solutions obtained by JBDQR associated with the true k^*'s. From these results, as far as a parameter-choice method is concerned, we should stress that the important is the accuracy of the regularized solutions other than the regularization parameters themselves. It is often the case that the regularized solutions may have almost the same accuracy for quite different regularization parameters, and this is particularly true when the overall corner of a L-curve is not so sharp.

Recall the comments at the end of Section 5. At the same iteration, JBDQR is a little cheaper than JBDGCV and JBDWGCV. It is seen from Table 2 that, for each problem and given ε, (i) JBDQR was faster to reach the semi-convergence than JBDGCV and JBDWGCV became stabilized and obtained their respective best possible solutions and (ii) the estimates for k^* were (considerably) smaller than the iterations at which JBDGCV and JBDWGCV found their respective best possible regularized solutions. That is, the former used fewer iterations to successfully solve the problems than the latter two algorithms. Therefore, JBDQR is more efficient than JBDGCV and JBDWGCV on the test problems; in particular, for heat and deriv2, JBDQR used much fewer iterations and outperformed JBDGCV and JBDWGCV very substantially.

Fig. 1 depicts the convergence processes of JBDQR, JBDGCV and JBDWGCV for $L = L_1$ and $\varepsilon = 10^{-3}$. We observe from the figure that the best regularized solutions by JBDQR at semi-convergence are 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

Table 3
The relative errors defined by (6.2) and the estimates for the optimal regularization parameters k^* by the discrepancy principle for the first four test problems in Table 1 with $L = L_1$.

$\varepsilon = 10^{-3}$				
	$\tau = 1.005$	$\tau = 1.1$	$\tau = 1.2$	$\tau = 2.0$
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)

and the convergence processes are almost indistinguishable. Importantly, we see that the regularized solutions obtained by JBDGCV and JBDWGCV converge first, then stabilize for a while, and finally diverge dramatically, while, for heat and deriv2, they start to stabilize after k becomes large. This means that, as opposed to one's common expectation, the hybrid JBDGCV and JBDWGCV do not necessarily stabilize as subspaces are expanded sufficiently large and thus do not work reliably.

By summarizing the above, for JBDGCV and JBDWGCV, the phenomena 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 generalized singular values of some projected problems do not approximate the largest ones of $\{A, L\}$ in natural order as k increases, as we have stated in the introduction. In contrast, JBDQR has always exhibited the typical semi-convergence for all the problems: the iterates Lx_k^{reg} monotonically converge to Lx_{true} until $k = k^*$, after $k > k^*$ they diverge. This justifies our result and analysis.

Since $\|e\|$ is known for the above test problems, we can use the discrepancy principle (5.1) to estimate the optimal k^* and compute the corresponding regularizes solutions. 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 underestimate k^* more or less. We have observed that the reliable determination of k^* critically depend on τ, and the closer τ is to one, the more reliable the estimates are. Particularly, 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. Moreover, we have found that for all a given ε, JBDQR with $\tau = 1.005$ was significantly faster and used considerably fewer iterations to solve each problem than JBDGCV and JBDWGCV did correspondingly. Therefore, JBDQR is (much) more efficient than the other two algorithms.

Finally, for JBDQR we note from Tables 2–3 that the regularized solution obtained by using the L-curve criterion for each problem is a little more accurate than that obtained by using the discrepancy principle. It is interesting to mention that, as we have found that, when a smaller τ was taken, e.g., $\tau = 0.5$, JBDQR with two parameter-choice methods computed regularized solutions with similar accuracy for each problem. These results indicate that the common suggestion that $\tau > 1$ slightly is not necessarily a very best choice. However, if no more information on A and b_{true} is available in practice, one does not know how to choose τ more accurately. On the other hand, compared with Table 3, for the given four τ, the results in Table 2 imply that JBDQR with the L-curve criterion indeed solved (1.2) for given A and L, that is, the regularized solutions obtained by JBDQR with the L-curve criterion naturally satisfy the constraints in (1.2) since the regularized solutions are more accurate than those obtained by JBDQR with the discrepancy principle for given τ. We have numerically verified this assertion, and the details are omitted.

6.2. Two dimensional case

We test the two dimensional image deblurring problems listed in Table 1. The goal is to restore an image x_{true} from a blurred and noisy image $b = b_{true} + e$.

For the problems rice and mri, 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 band and σ 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 $m = n = 128^2 = 16,284$.

For the problems AtmosphericBlur30 and GaussianBlur422 of $m = n = 256^2 = 65,536$ from [25], the blurring of AtmosphericBlur30 is caused by atmospheric turbulence, and GaussianBlur422 is spatially invariant Gaussian blur. And the blurring operators are generated by the codes psfMatrix(PSF, center, 'zero') and psfMatrix(PSF) from [25], respectively. We abbreviate AtmosphericBlur30 and GaussianBlur422 as blur30 and blur422, respectively.

Table 4
The relative errors defined by (6.2) and the estimates for k^* by the L-curve criterion for the latter four test problems in Table 1 with L defined by (6.4).

$\varepsilon = 20\%$				
	JBDGCV	JBDWGCV	JBDQR (k^*)	estimates for k^*
rice	0.9531(1)	0.9547(3)	0.8859(1)	2(0.9477)
mri	0.9894(2)	0.9889(2)	0.9385(2)	3(0.9396)
blur30	0.9937(2)	0.9935(2)	0.9627(4)	2(0.9705)
blur422	0.9862(6)	0.9857(4)	0.9598(2)	2(0.9598)
$\varepsilon = 10\%$				
	JBDGCV	JBDWGCV	JBDQR (k^*)	estimates for k^*
rice	0.9114(3)	0.9064(2)	0.8620(2)	2(0.8620)
mri	0.9834(2)	0.9763(1)	0.9111(6)	3(0.9204)
blur30	0.9893(2)	0.9890(2)	0.9419(8)	2(0.9701)
blur422	0.9772(2)	0.9759(1)	0.9494(5)	3(0.9522)
$\varepsilon = 5 \times 10^{-2}$				
	JBDGCV	JBDWGCV	JBDQR (k^*)	estimates for k^*
rice	0.8778(3)	0.8736(3)	0.8397(5)	4(0.8411)
mri	0.9602(3)	0.9498(4)	0.8848(13)	6(0.9007)
blur30	0.9835(3)	0.9827(3)	0.9124(16)	5(0.9521)
blur422	0.9459(9)	0.9443(10)	0.9109(62)	24(0.9203)
$\varepsilon = 10^{-2}$				
	JBDGCV	JBDWGCV	JBDQR (k^*)	estimates for k^*
rice	0.8372(7)	0.8363(7)	0.7774(23)	11(0.7951)
mri	0.8923(13)	0.8782(19)	0.8421(51)	25(0.8514)
blur30	0.9697(6)	0.9603(9)	0.7975(65)	40(0.8243)
blur422	0.9459(9)	0.9443(10)	0.9109(62)	24(0.9203)
$\varepsilon = 10^{-3}$				
	JBDGCV	JBDWGCV	JBDQR (k^*)	estimates for k^*
rice	0.7638(38)	0.7539(52)	0.7136(166)	145(0.7140)
mri	0.8305(101)	0.8225(151)	0.7949(451)	293(0.7988)
blur30	0.9628(9)	0.7984(75)	0.5670(433)	577(0.5907)
blur422	0.9137(59)	0.9046(103)	0.8736(542)	284(0.8794)

We choose the regularization matrix

$$L = \begin{pmatrix} I_N \otimes L_1 \\ L_1 \otimes I_N \end{pmatrix} \in \mathbb{R}^{2N(N-1)\times N^2} \tag{6.4}$$

with L_1 defined in (6.3) and I_N the identity matrix of order N, which is the scaled discrete approximation of the first derivative operator in the two dimensional case incorporating no assumptions on boundary conditions; see [12, Chapter 8.1-2]. Keep in mind definition (6.1) of ε. The Gaussian white noises e with zero mean are generated so that the relative noise levels $\varepsilon = 20\%, 10\%, 5 \times 10^{-2}$, 10^{-2} and 10^{-3}, respectively. Table 4 reports the results obtained, where the L-curve criterion is used to estimate the k^* and all the notations and meanings are the same as those in Table 2.

By comparing the results in the fifth column of Table 4 with those in the second and third columns, we can see that for these four problems the solution accuracy of JBDQR with the L-curve criterion is considerably higher than that of JBDGCV and JBDWGCV. We find 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 determine k^* accurately for these two dimensional problems. The fundamental cause is that $\|\bar{B}_k y_k\|$ still increases slowly even after $k > k^*$, causing that the curve of $\big(\log(\|B_k y_k - \beta_1 e_1\|), \log(\|\bar{B}_k y_k\|)\big)$ does not form a sharp L-shape, that is, the points on the overall corner have small curvatures; see [12, p. 72-3]. However, though the L-curve criterion determined k^* only roughly, it is remarkable to notice that the corresponding regularized solutions are almost as accurate as the counterparts associated with the optimal k^*. This is again due to the mentioned feature of overall corner of the L-curve, which makes the points around the corner insensitive to a wide range of k so that the Lx_k^{reg} are close to Lx_{k^*}.

Fig. 2 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, as expected, JBDQR has typical semi-convergence phenomena for all the problems and is more robust and reliable than JBDGCV and JBDWGCV. We have also observed from Table 4 and Fig. 2 that the JBDWGCV works at least as well as JBDGCV and could obtain more accurate regularized solutions for some problems, e.g., blur30.

Insightfully, we observe that JBDGCV and JBDWGCV stabilize earlier than the semi-convergence of JBDQR for all the four test problems. These indicate that the best regularized solutions obtained by the former two methods did not capture

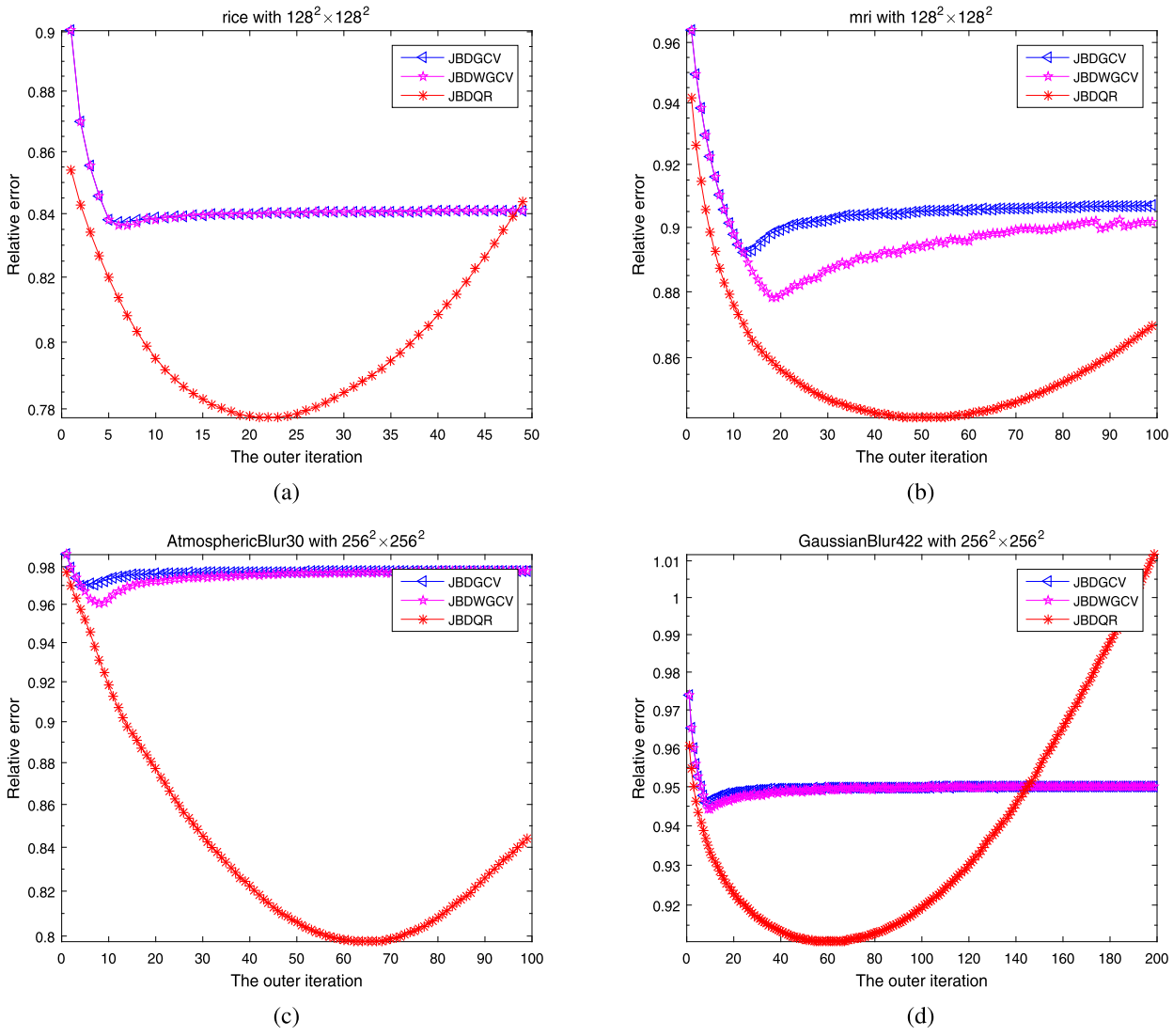

Fig. 2. The relative errors, defined by (6.2), of JBDQR, JBDWGCV and JBDGCV with $\varepsilon = 10^{-2}$: (a) rice; (b) mri; (c) blur30; (d) blur422.

all the needed GSVD components when they became stabilized, so that they are definitely less accurate than those by JBDQR. This is possibly due to the fact that the generalized singular values of some projected problems fail to approximate the largest ones of $\{A, L\}$, leading to the non-convergence of inner regularization parameters for projected problems to the optimal regularization parameter λ_{opt} of the original (1.3).

Since $\|e\|$ is known for the above test problems, we can also use the discrepancy principle criterion (5.1) to estimate the optimal k^* and compute corresponding regularized solutions. In Table 5, we report the results obtained when $\tau = 1.005,\ 1.1,\ 1.2$ and 2.0. We observe that, for the chosen four τ's, the regularization parameters determined by the discrepancy principle lead to substantial differences for both the solution accuracy and the estimates for k^*. Obviously, the estimates are better for $\tau = 1.005$ than those for the other τ, especially much better than those for $\tau = 2$. Comparing the results in Table 4 with those in Table 5, similarly to our findings in the one dimensional case, we have seen that JBDQR with the L-curve criterion computed more accurate regularized solutions than JBDQR with the discrepancy principle when $\tau > 1$.

For these four two dimensional test problems, unlike the one dimensional case, it is hard or meaningless to compare the efficiency of JBDQR with JBDGCV and JBDWGCV. The reason is that although the latter two algorithms often used fewer iterations to stabilize than JBDQR, especially for smaller ε, they computed considerably less accurate regularized solutions than JBDQR did at semi-convergence and JBDQR with the L-curve criterion and the discrepancy principle. In other words, JBDGCV and JBDWGCV did not solve the problems reliably and successfully, and they solved the problems far less satisfactorily than JBDQR did.

Table 5
The relative errors defined by (6.2) and the estimates for k^* by the discrepancy principle for the latter four test problems in Table 1 with L defined by (6.4).

$\varepsilon = 20\%$				
	$\tau = 1.005$	$\tau = 1.1$	$\tau = 1.2$	$\tau = 2.0$
rice	0.8855(1)	0.8855(1)	0.8855(1)	0.8855(1)
mri	0.9385(2)	0.9653(1)	0.9653(1)	0.9653(1)
blur30	0.9627(4)	0.9849(1)	0.9849(1)	0.9849(1)
blur422	0.9598(2)	0.9711(1)	0.9711(1)	0.9711(1)
$\varepsilon = 10\%$				
	$\tau = 1.005$	$\tau = 1.1$	$\tau = 1.2$	$\tau = 2.0$
rice	0.8620(2)	0.8805(1)	0.8805(1)	0.8805(1)
mri	0.9198(4)	0.9433(2)	0.9433(2)	0.9657(1)
blur30	0.9678(7)	0.9848(1)	0.9848(1)	0.9848(1)
blur422	0.9522(3)	0.9611(2)	0.9708(1)	0.9708(1)
$\varepsilon = 5 \times 10^{-2}$				
	$\tau = 1.005$	$\tau = 1.1$	$\tau = 1.2$	$\tau = 2.0$
rice	0.8462(3)	0.8556(2)	0.8556(2)	0.8791(1)
mri	0.9007(6)	0.9156(4)	0.9267(3)	0.9653(1)
blur30	0.9181(12)	0.9575(5)	0.9700(3)	0.9849(1)
blur422	0.9444(6)	0.9549(3)	0.9608(2)	0.9707(1)
$\varepsilon = 10^{-2}$				
	$\tau = 1.005$	$\tau = 1.1$	$\tau = 1.2$	$\tau = 2.0$
rice	0.7989(10)	0.8198(6)	0.8267(5)	0.8541(2)
mri	0.8564(21)	0.8652(15)	0.8729(12)	0.8985(6)
blur30	0.8019(56)	0.8285(38)	0.8564(27)	0.9312(9)
blur422	0.9230(21)	0.9308(13)	0.9367(9)	0.9501(4)
$\varepsilon = 10^{-3}$				
	$\tau = 1.005$	$\tau = 1.1$	$\tau = 1.2$	$\tau = 2.0$
rice	0.7288(62)	0.7369(46)	0.7430(38)	0.7709(18)
mri	0.8121(141)	0.8179(105)	0.8223(85)	0.8378(41)
blur30	0.6083(241)	0.6183(216)	0.6286(194)	0.6889(110)
blur422	0.8861(182)	0.8931(119)	0.8972(92)	0.9118(39)

Fig. 3 draws the exact images and the reconstructed images for the four test problems with $\varepsilon = 10^{-2}$. Clearly, the reconstructed images by JBDQR with the L-curve criterion 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.

For the other ε tested, we have similar findings to those in Figs. 2–3. We have tested the algorithms for numerous ε are very typical. We have observed that the algorithms exhibit very similar phenomena for these ε to the ones for the ε reported in the paper, and the smaller ε is, the more accurate regularized solutions are and the more iterations are needed.

7. Conclusions

We have proposed a JBDQR algorithm for solving (1.2). This algorithm is different from the hybrid projection based method [20] for solving (1.3), which exploits the same JBD process but explicitly regularizes each projected problem generated at every iteration. The mechanism and features of the two algorithms are different: JBDQR has the desired semi-convergence, and the hybrid method is expected to ultimately stabilize when the projection subspace is large enough.

We have established a number of theoretical results on JBDQR and made a detailed analysis on them. The results have shown that JBDQR must possess the semi-convergence property, which gets insight into its regularizing effects. Numerical experiments on a number of problems have justified numerous aspects of JBDQR, e.g., solution accuracy, robustness and implementation simplicity. It turns out that JBDQR should at least be a highly competitive alternative of the relatively complicated hybrid method.

There are some important problems that deserve further considerations. As we have seen, JBDQR and the hybrid one need to solve a large scale least squares problem at each outer iteration, which may be costly. It is unclear whether or not the solution accuracy can be relaxed substantially, at least at some outer iterations, similar to the randomized SVD algorithms proposed in [18] that solve the general-form regularization problem (1.2). If they could be solved with considerably relaxed accuracy, we would gain much, and the overall efficiency of the algorithm could be improved substantially. The solution accuracy requirement on the inner least squares problems will constitute our forthcoming work.

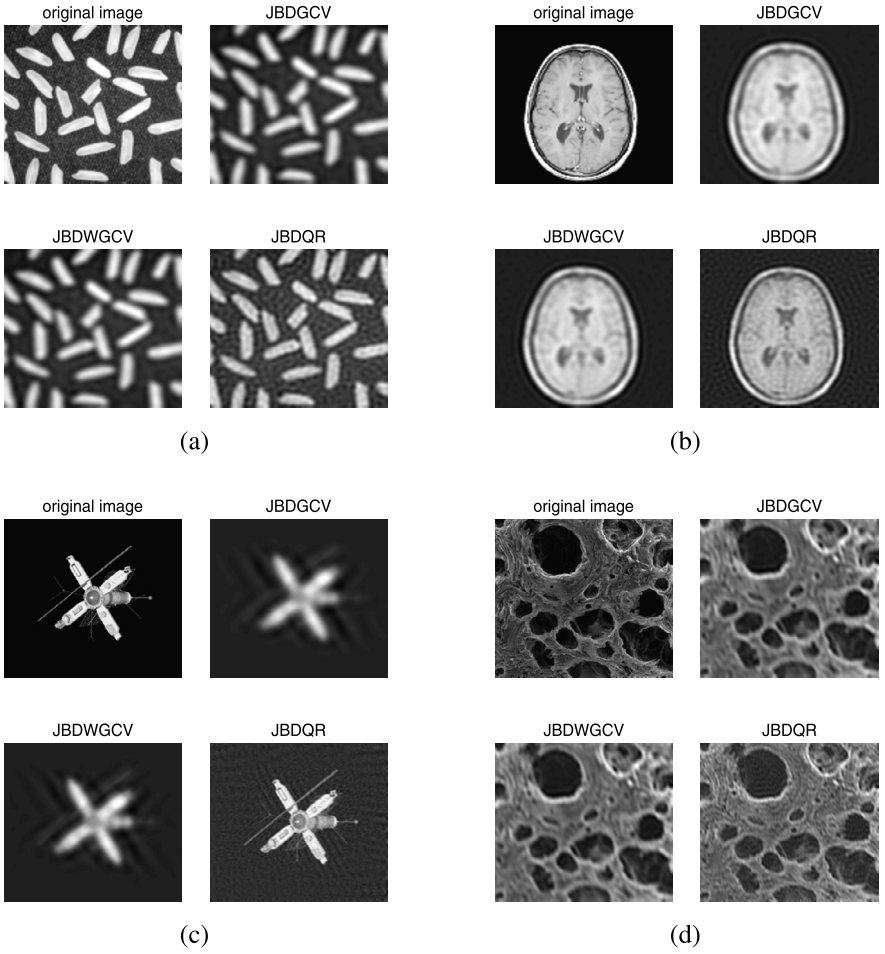

Fig. 3. The exact images and the reconstructed images for the four two dimensional test problems with $\varepsilon = 10^{-2}$: (a) rice; (b) mri; (c) blur30; (d) blur422.

Acknowledgements

We would like to thank the two referees very much for their suggestions and comments, which made us to improve the presentation.

References

[1] R.C. Aster, B. Borchers, C.H. Thurber, Parameter Estimation and Inverse Problems, second edition, Elsevier, New York, 2013.
[2] S. Berisha, J.G. Nagy, Restore tools: iterative methods for image restoration, available from, http://www.mathcs.emory.edu/~nagy/RestoreTools, 2012.
[3] Å. Björck, Numerical Methods for Least Squares Problems, SIAM, Philadelphia, PA, 1996.
[4] J. Chung, A.K. Saibaba, Generalized hybrid iterative methods for large-scale Bayesian inverse problems, SIAM J. Sci. Comput. 39 (2017) S24–S46.
[5] J. Chung, A.K. Saibaba, M. Brown, 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, Surv. Math. Ind. 3 (1993) 71–143.
[7] H.W. Engl, M. Hanke, A. Neubauer, Regularization of Inverse Problems, Kluwer Academic Publishers, 2000.
[8] S. Gazzola, P.C. Hansen, 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, P. Novati, Inheritance of the discrete Picard condition in Krylov subspace methods, BIT Numer. Math. 56 (2016) 893–918.
[10] P.C. Hansen, Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion, SIAM, Philadelphia, PA, 1998.
[11] P.C. Hansen, Regularization tools version 4.0 for Matlab 7.3, Numer. Algorithms 46 (2007) 189–194.
[12] P.C. Hansen, Discrete Inverse Problems: Insight and Algorithms, SIAM, Philadelphia, PA, 2010.
[13] M.E. Hochstenbach, L. Reichel, X. Yu, A Golub-Kahan-type reduction method for matrix pairs, J. Sci. Comput. 65 (2015) 767–789.
[14] Z. Jia, Approximation accuracy of the Krylov subspaces for linear discrete ill-posed problems, J. Comput. Appl. Math. 374 (2020) 112786.
[15] Z. Jia, The low rank approximations and Ritz values in LSQR for linear discrete ill-posed problems, Inverse Probl. 36 (2020) 045013.
[16] Z. Jia, Regularization properties of the Krylov iterative solvers CGME and LSMR for linear discrete ill-posed problems with an application to truncated randomized SVDs, Numer. Algorithms (2020), https://doi.org/10.1007/s11075-019-00865-w.
[17] Z. Jia, Regularization properties of LSQR for linear discrete ill-posed problems in the multiple singular value case and best, near best and general low rank approximations, Inverse Probl., https://doi.org/10.1088/1361-6420/ab9c45.

[18] Z. Jia, 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.
[19] J. Kaipio, E. Somersalo, Statistical and Computational Inverse Problems, Applied Mathematical Sciences, vol. 160, Springer, 2005.
[20] M.E. Kilmer, P.C. Hansen, M.I. Espanol, A projection-based approach to general-form Tikhonov regularization, SIAM J. Sci. Comput. 29 (2007) 315–330.
[21] R. Li, Q. Ye, A Krylov subspace method for quadratic matrix polynomitals with application to constrained least squares problems, SIAM J. Matrix Anal. 25 (2003) 405–428.
[22] J. Liu, M. Yan, T. Teng, Surface-aware blind image deblurring, IEEE Trans. Pattern Anal. Mach. Intell. (2019).
[23] K. Miller, Least squares methods for ill-posed problems with a prescribed bound, SIAM J. Math. Anal. 1 (1970) 52–74.
[24] E. Natterer, The Mathematics of Computerized Tomography, John Wiley, New York, 1986.
[25] J.G. Nagy, K. Palmer, L. Perrone, Iterative methods for image deblurring: a Matlab object-oriented approach, Numer. Algorithms 36 (2004) 73–93.
[26] C.C. Paige, M.A. Saunders, LSQR: an algorithm for sparse linear equations and sparse least squares, ACM Trans. Math. Softw. 8 (1982) 43–71.
[27] B.N. Parlett, The Symmetric Eigenvalue Problem, SIAM, Philadelphia, PA, 1998.
[28] R.A. Renaut, S. Vatankhah, V.E. Ardesta, Hybrid and iteratively reweighted regularization by unbiased predictive risk and weighted GCV for projected systems, SIAM J. Sci. Comput. 39 (2017) B221–B243.
[29] L. Reichel, F. Sgallari, Q. Ye, Tikhonov regularization based on generalized Krylov subspace methods, Appl. Numer. Math. 62 (2012) 1215–1228.
[30] A. van der Sluis, H.A. van der Vorst, The rate of convergence of conjugate gradients, Numer. Math. 48 (1986) 543–560.
[31] I.N. Zwaan, M. Hochstenbach, Multidirectional subspace expansion for one-parameter and multiparameter Tikhonov regularization, J. Sci. Comput. 70 (2017) 990–1007.
[32] H. Zha, Computing the generalized singular values/vectors of large sparse or structured matrix pairs, Numer. Math. 72 (1996) 391–417.