SIAM J. SCI. COMPUT.
Vol. 0, No. 0, pp. S146–S172

HYBRID PROJECTION METHODS WITH RECYCLING FOR INVERSE PROBLEMS*

JIAHUA JIANG†, JULIANNE CHUNG†, AND ERIC DE STURLER†

Abstract. Iterative hybrid projection methods have proven to be very effective for solving large linear inverse problems due to their inherent regularizing properties as well as the added flexibility to select regularization parameters adaptively. In this work, we develop Golub–Kahan-based hybrid projection methods that can exploit compression and recycling techniques in order to solve a broad class of inverse problems where memory requirements or high computational cost may otherwise be prohibitive. For problems that have many unknown parameters and require many iterations, hybrid projection methods with recycling can be used to compress and recycle the solution basis vectors to reduce the number of solution basis vectors that must be stored, while obtaining a solution accuracy that is comparable to that of standard methods. If reorthogonalization is required, this may also reduce computational cost substantially. In other scenarios, such as streaming data problems or inverse problems with multiple datasets, hybrid projection methods with recycling can be used to efficiently integrate previously computed information for faster and better reconstruction. Additional benefits of the proposed methods are that various subspace selection and compression techniques can be incorporated, standard techniques for automatic regularization parameter selection can be used, and the methods can be applied multiple times in an iterative fashion. Theoretical results show that, under reasonable conditions, regularized solutions for our proposed recycling hybrid method remain close to regularized solutions for standard hybrid methods and reveal important connections among the resulting projection matrices. Numerical examples from image processing show the potential benefits of combining recycling with hybrid projection methods.

Key words. Golub–Kahan bidiagonalization, hybrid projection methods, recycling, compression, inverse problems, tomography

AMS subject classifications. 15A29, 65F10, 68U10, 65F22

DOI. 10.1137/20M1349515

1. Introduction.

Inverse problems arise in many applications, where the goal is to approximate some unknown parameters of interest from indirect measurements or observations [21, 25, 22, 47, 12]. We consider discretized linear inverse problems of the form

$$\mathbf{b} = \mathbf{A}\mathbf{x}_{\text{true}} + \boldsymbol{\epsilon}, \tag{1.1}$$

where $\mathbf{A} \in \mathbb{R}^{M\times N}$ models the forward process, $\mathbf{b} \in \mathbb{R}^M$ contains observed data, $\mathbf{x}_{\text{true}} \in \mathbb{R}^N$ represents the desired parameters, and $\boldsymbol{\epsilon} \in \mathbb{R}^M$ is assumed to be Gaussian white noise. Given $\mathbf{b}$ and $\mathbf{A}$, the goal is to compute an approximation of $\mathbf{x}_{\text{true}}$. An important problem with discretized linear inverse problems is that there are many small singular values smoothly decaying to zero (though leveling off at machine precision). The noise-free problem typically satisfies the discrete Picard condition, that is, the absolute values of the components of the right-hand side along the left singular vectors decay faster to zero than the singular values. However, in the presence of noise this is no longer true, and naive (regular or least-squares) inversion, $\mathbf{A}^\dagger\mathbf{b} = \mathbf{A}^\dagger\mathbf{A}\mathbf{x}_{\text{true}} + \mathbf{A}^\dagger\boldsymbol{\epsilon}$, results in very large noise components in directions corresponding to the small singular

*Received by the editors July 1, 2020; accepted for publication (in revised form) December 28, 2020; published electronically May 3, 2021.
https://doi.org/10.1137/20M1349515
Funding: This work was supported by NSF grants DMS 1654175 and DMS 1720305.
†Department of Mathematics, Virginia Tech, Blacksburg, VA 24061 USA (jiahua@vt.edu, jmchung@vt.edu, sturler@vt.edu).

values (where $\mathbf{A}^\dagger$ is the pseudoinverse of $\mathbf{A}$). So $\|\mathbf{A}^\dagger\boldsymbol{\epsilon}\|_2 \gg \|\mathbf{A}^\dagger\mathbf{A}\mathbf{x}_{\text{true}}\|_2$, which completely spoils the approximate solution.

In iterative Krylov subspace methods, like LSQR, this problem manifests itself through a phenomenon called *semiconvergence*, where during the early iterations the solution converges to the true solution, but once the Krylov subspace starts to approximate left singular vectors corresponding to the small singular values, the corresponding noise components in the approximate solution are significantly amplified, and the approximate solution diverges from $\mathbf{x}_{\text{true}}$ and converges to the naive solution; see Figure 4.2 for an illustration.

To compute a meaningful solution, a regularized problem is solved instead. A common approach is to solve the Tikhonov regularized problem,

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda^2 \|\mathbf{x}\|_2^2, \tag{1.2}$$

where $\lambda \geqslant 0$ is a (yet-to-be-determined) regularization parameter that balances the data-fit term and the regularization term. We remark that extensions to the general-form Tikhonov problem can be made, which often requires a transformation to standard form [22]. Although the Tikhonov problem has been studied for many years, various computational challenges have motivated the development of hybrid iterative projection methods for computing an approximate solution to (1.2); seminal papers include [36, 5]. In a hybrid projection method, the original problem is projected onto small subspaces of increasing dimension and the projected problem is solved using variational regularization. By regularizing the projected problem, hybrid methods can stabilize the convergence behavior of the method, and the regularization parameter does not need to be known in advance. An additional benefit is that these iterative methods can handle problems where matrices $\mathbf{A}$ and $\mathbf{A}^\top$ are so large that they can not be constructed but can be accessed via function evaluations.

However, one of the main disadvantages of hybrid methods compared to standard iterative methods is the need to store the basis vectors for solution computation, which can present significant computational bottlenecks if many iterations are needed or if there are many unknowns. Furthermore, these methods are typically embedded within a larger problem that needs to be solved, e.g., optimal experimental design or nonlinear frameworks. So, it may be required to solve a sequence of inverse problems, e.g., where the forward model is parameterized such that the change in the model from one problem to the next is relatively small, or to compute and update solutions from streaming data. Rather than start each solution computation from scratch, we assume that a few vectors for the solution subspace can be provided, and our goals are to improve upon the given subspace and to compute a regularized solution efficiently in the improved subspace.

In this paper, we develop *recycling* Golub–Kahan-based hybrid projection methods that combine a recycling Golub–Kahan bidiagonalization (recycling GKB) process with tools from compression to solve a broad class of inverse problems. More specifically, our proposed hybrid projection method uses recycling techniques to improve a given solution subspace and then efficiently computes a regularized solution to the projected problem, with automatic regularization parameter selection. The general approach consists of three steps, which can be used in an iterative fashion. First, we begin with a suitable set of orthonormal basis vectors, denoted $\mathbf{W}_{k-1} \in \mathbb{R}^{N\times(k-1)}$. This may be provided (e.g., from a related problem or from expert knowledge) or may need to be determined (e.g., via compression of previous solutions). With an initial guess of the solution, $\mathbf{x}^{(1)}$, the second step is to use a recycling GKB process

to generate the columns of $\widetilde{\mathbf{V}}_\ell \in \mathbb{R}^{N\times \ell}$ that span a particular Krylov subspace, and extend the solution space to be $\mathcal{R}([\mathbf{W}_{k-1} \ \mathbf{x}^{(1)} \ \widetilde{\mathbf{V}}_\ell])$ where $\mathcal{R}(\cdot)$ denotes the column space of a matrix. The third step is to find a suitable regularization parameter λ and compute a solution to the regularized projected problem,

$$\min_{\mathbf{x}\in\mathcal{R}\left(\begin{bmatrix}\mathbf{W}_{k-1} & \mathbf{x}^{(1)} & \widetilde{\mathbf{V}}_\ell\end{bmatrix}\right)} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 + \lambda^2 \|\mathbf{x}\|_2^2 . \tag{1.3}$$

The main approach (corresponding to steps 2 and 3) is described in subsections 3.1 and 3.2, and some compression approaches that can be used in step 1 are provided in subsection 3.3.

Recycling techniques for iterative methods have been considered for multiple Krylov solvers and a wide range of applications, but mainly for square system matrices and for well-posed problems [38, 45, 49, 30, 1, 44, 28, 29, 13, 32]. Augmented LSQR methods have been described in [3, 2] for well-posed least-squares problems that require many LSQR iterations. By augmenting Krylov subspaces using harmonic Ritz vectors that approximate singular vectors associated with the small singular values, this approach can reduce computational cost by using implicit restarts for improved convergence. However, when applied to ill-posed inverse problems, the augmented LSQR method without an explicit regularization term exhibits semiconvergence behavior. Other approaches for augmenting or enriching Krylov subspaces are described in [23, 6, 27], where Krylov subspaces are combined with vectors containing important information about the desired solution (e.g., a low-dimensional subspace). These methods can improve the solution accuracy by incorporating information about the desired solution into the solution process, but the improvement in accuracy *significantly* depends on the quality of the provided vectors. Modifications to include specified solution vectors have been considered for the conjugate gradient method [6] and for the TSVD approach [27]. A hybrid enriched bidiagonalization (HEB) method that stably and efficiently augments a "well-chosen enrichment subspace" with the standard Krylov basis associated with LSQR is described in [23]. Contrary to the HEB method, our recycling GKB method generates the extension subspace vectors $\widetilde{\mathbf{V}}_\ell$ such that we *improve* on the space, rather than just augment it. Thus, as we will demonstrate in section 4, our approach can handle a wider range of problems and provide more accurate solutions.

The paper is organized as follows. In section 2, we provide a brief overview on hybrid projection methods, where we focus on methods based on the standard GKB process. Then in section 3, we propose new hybrid projection methods that are based on the recycling GKB process and describe techniques for incorporating regularization automatically and efficiently. We also describe some examples of compression methods that can be used in step 1 of the proposed approach and provide theoretical results. In particular, we investigate the impact of compression and recycling on the projected problem and show important results that relate the regularized solution from a recycling approach to that from a standard approach. Numerical results are provided in section 4, and conclusions are provided in section 5.

2. Background on hybrid iterative methods. Hybrid approaches that embed regularization within iterative methods date back to seminal papers by O'Leary and Simmons in 1981 [36] and Björck in 1988 [5], and the number of extensions and developments in the area of hybrid methods continues to grow; see, e.g., [17, 23, 15, 16, 40, 4, 8, 9, 10, 39]. We focus on hybrid methods based on the GKB process, which generate an m-dimensional Krylov subspace using matrix $\mathbf{A}^\top\mathbf{A}$ and vector $\mathbf{A}^\top\mathbf{b}$,

$$\mathcal{K}_m\left(\mathbf{A}^\top\mathbf{A}, \mathbf{A}^\top\mathbf{b}\right) = \operatorname{span}\left\{\mathbf{A}^\top\mathbf{b}, (\mathbf{A}^\top\mathbf{A})\mathbf{A}^\top\mathbf{b}, \ldots, (\mathbf{A}^\top\mathbf{A})^{m-1}\mathbf{A}^\top\mathbf{b}\right\}.$$

The GKB process[1] [19] can be described as follows. Let $\beta_1 = \|\mathbf{b}\|_2$, $\mathbf{u}_1 = \mathbf{b}/\beta_1$, and $\alpha_1\mathbf{v}_1 = \mathbf{A}^\top\mathbf{u}_1$. Then at the jth iteration of the GKB process, we generate vectors $\mathbf{u}_{j+1}$ and $\mathbf{v}_{j+1}$ such that

$$\beta_{j+1}\mathbf{u}_{j+1} = \mathbf{A}\mathbf{v}_j - \alpha_j\mathbf{u}_j \quad \text{and} \quad \alpha_{j+1}\mathbf{v}_{j+1} = \mathbf{A}^\top\mathbf{u}_{j+1} - \beta_{j+1}\mathbf{v}_j, \tag{2.1}$$

and after m iterations we have the relationships

$$\mathbf{A}\mathbf{V}_m = \mathbf{U}_{m+1}\mathbf{B}_m, \tag{2.2}$$

$$\mathbf{A}^\top\mathbf{U}_{m+1} = \mathbf{V}_m\mathbf{B}_m^\top + \alpha_{m+1}\mathbf{v}_{m+1}\mathbf{e}_{m+1}^\top = \mathbf{V}_{m+1}\mathbf{L}_{m+1}^\top, \tag{2.3}$$

where $\mathbf{V}_m = \begin{bmatrix}\mathbf{v}_1 & \dots & \mathbf{v}_m\end{bmatrix} \in \mathbb{R}^{N\times m}$ and $\mathbf{U}_{m+1} = \begin{bmatrix}\mathbf{u}_1 & \dots & \mathbf{u}_{m+1}\end{bmatrix} \in \mathbb{R}^{M\times(m+1)}$ contain orthonormal columns,

$$\mathbf{B}_m = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_m & \alpha_m \\ & & & \beta_{m+1} \end{bmatrix} \in \mathbb{R}^{(m+1)\times m} \tag{2.4}$$

is a bidiagonal matrix, and $\mathbf{L}_{m+1} = \begin{bmatrix}\mathbf{B}_m & \alpha_{m+1}\mathbf{e}_{m+1}\end{bmatrix}$. Given these relations, an approximate least-squares solution can be computed as $\mathbf{x}_m = \mathbf{V}_m\mathbf{y}_m$ where $\mathbf{y}_m$ is the solution to the *projected* least-squares problem,

$$\min_{\mathbf{x}\in\mathcal{R}(\mathbf{V}_m)} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 = \min_{\mathbf{y}} \|\mathbf{B}_m\mathbf{y} - \beta_1\mathbf{e}_1\|_2^2. \tag{2.5}$$

In standard LSQR implementations, the columns of $\mathbf{V}_m$ and $\mathbf{U}_{m+1}$ do not need to be stored and efficient updates can be used to minimize storage requirements. That is, the storage cost is very low (e.g., $M + 2N$ for LSQR) due to a 3-term recurrence property. For such iterative methods, the main computational cost at each iteration is a matrix-vector product with $\mathbf{A}$ and its transpose.

However, when applied to ill-posed inverse problems, standard iterative methods exhibit semiconvergent behavior, whereby solutions improve in early iterations but become contaminated with inverted noise in later iterations [22]. Thus, it is desirable to consider a hybrid iterative projection method that combines iterative regularization with a variational regularization method such as Tikhonov regularization. One approach is to solve the Tikhonov problem (1.2) by applying any iterative least-squares solver (e.g., LSQR) to the equivalent augmented system,

$$\min_{\mathbf{x}} \left\| \begin{bmatrix}\mathbf{A} \\ \lambda\mathbf{I}\end{bmatrix}\mathbf{x} - \begin{bmatrix}\mathbf{b} \\ \mathbf{0}\end{bmatrix} \right\|_2^2. \tag{2.6}$$

The main challenge is that the regularization parameter λ must be selected a priori, which can be difficult especially for large-scale problems. Another hybrid iterative approach is to project the problem onto Krylov subspaces of increasing dimension

[1] We assume no termination of the iteration, and therefore the dimension of $K_m(\mathbf{A}^\top\mathbf{A}, \mathbf{A}^\top\mathbf{b})$ is m.

and to compute the solution at the mth iteration as $\mathbf{x}_m = \mathbf{V}_m\mathbf{y}_m$ where $\mathbf{y}_m$ solves the *projected, regularized* problem,

$$\min_{\mathbf{x}\in\mathcal{R}(\mathbf{V}_m)} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 + \lambda^2\|\mathbf{x}\|_2^2 = \min_{\mathbf{y}} \|\mathbf{B}_m\mathbf{y} - \beta_1\mathbf{e}_1\|_2^2 + \lambda^2\|\mathbf{y}\|_2^2\,. \tag{2.7}$$

One benefit of this approach is that the regularization parameter for the projected problem can be easily and automatically estimated during the iterative process [31, 8, 40]. However, a potential disadvantage is the storage of $\mathbf{V}_m$ which is needed for solution computation. For some problems where the solution can be represented in only a few basis vectors, this additional storage is not a concern. However, for large-scale problems where storage of these vectors becomes too demanding, the proposed hybrid projection methods with recycling and compression that we describe in the next section can be used to reduce this computational cost.

3. Hybrid projection methods with recycling. Using iterative hybrid projection methods to solve large-scale inverse problems can be quite effective. We are interested in scenarios where one has an initial solution subspace (e.g., from a prior reconstruction or from a sequence of reconstructions), and the goal is to incorporate such information to not only augment but also improve or enhance the solution subspace, thereby improving the quality of the subsequent solution approximations. For example, for problems requiring many iterations, the memory required to store the basis vectors for solution computation in canonical hybrid projection methods can exceed capabilities or result in significantly longer computational times. The proposed *hybrid projection methods with recycling* can be used to ameliorate the memory requirements without sacrificing the quality of the solution, where a main ingredient is the recycling GKB process. Here, we modify the classical GKB process to augment and enhance a given orthonormal basis. Then, the recycling GKB process can be combined with a regularization technique to give an efficient hybrid projection method. Finally, by exploiting various compression approaches, compression and recycling can be repeated in an iterative fashion until a desired reconstruction is obtained. An overview of the general approach is provided in Algorithm 3.1.

Algorithm 3.1 Hybrid projection method with recycling and compression.

Require: $\mathbf{A}$, $\mathbf{b}$, $\mathbf{W}_{k-1}$, $\mathbf{x}^{(1)}$
while desired solution not obtained **do**
  $\ell = 1$
  Construct $\mathbf{W}_k$; see subsection 3.1.
  while storage is available and stopping criteria not satisfied **do**
    Use recycling GKB to compute augmented subspace $\widetilde{\mathbf{V}}_\ell$; see subsection 3.1.
    Compute regularization parameter.
    Solve regularized, projected problem; see subsection 3.2.
    $\ell = \ell + 1$
  end while
  Use compression to get $\mathbf{W}_{k-1}$; see subsection 3.3.
end while

Notice that even though a large number of iterations can be performed, the size of the projected problem will never exceed the set storage limit. Furthermore, theoretical results provided in subsection 3.4 show that under reasonable conditions, regularized solutions obtained from the recycling GKB approach remain close to the standard

GKB solution. We will also address a special case where $\mathbf{W}_{k-1}$ and $\mathbf{x}^{(1)}$ come from a standard Krylov approach and TSVD is used for compression.

For all derivations and results in this section, we assume exact arithmetic and no breakdown of the algorithms.

3.1. Recycling Golub–Kahan bidiagonalization. In this section, we assume that an approximate solution (or initial guess) $\mathbf{x}^{(1)}$ and a matrix $\mathbf{W}_{k-1} \in \mathbb{R}^{N\times(k-1)}$ with orthonormal columns are given, and we describe the recycling GKB process that can be used to augment the solution subspace using recycling techniques. First, assuming $\mathbf{x}^{(1)} \notin \mathcal{R}(\mathbf{W}_{k-1})$, we set $\mathbf{W}_k = \begin{bmatrix}\mathbf{W}_{k-1} & \breve{\mathbf{x}}^{(1)}\end{bmatrix} \in \mathbb{R}^{N\times k}$ where $\breve{\mathbf{x}}^{(1)} = \left(\mathbf{x}^{(1)} - \mathbf{W}_{k-1}\mathbf{W}_{k-1}^\top\mathbf{x}^{(1)}\right) / \left\|\mathbf{x}^{(1)} - \mathbf{W}_{k-1}\mathbf{W}_{k-1}^\top\mathbf{x}^{(1)}\right\|_2$. Now, $\mathbf{W}_k$ represents the *recycled* subspace and $\mathbf{W}_k^\top\mathbf{W}_k = \mathbf{I}_k$, and the approximate solution (or initial guess) $\mathbf{x}^{(1)}$ is always in the search space. Thus subsequent (regularized) approximations may preserve this search direction. If $\mathbf{x}^{(1)} \in \mathcal{R}(\mathbf{W}_{k-1})$, then $\mathbf{W}_{k-1}$ can be used as the recycled subspace.

Next, take the skinny QR factorization of $\mathbf{A}\mathbf{W}_k$,

$$\mathbf{A}\mathbf{W}_k = \mathbf{Y}_k\mathbf{R}_k \in \mathbb{R}^{M\times k}, \tag{3.1}$$

compute $\breve{\mathbf{r}}^{(1)} = \mathbf{b} - \mathbf{A}\breve{\mathbf{x}}^{(1)}$, and set

$$\widetilde{\mathbf{b}} = \breve{\mathbf{r}}^{(1)} - \mathbf{Y}_k\boldsymbol{\zeta} \quad \text{where} \quad \boldsymbol{\zeta} = \mathbf{Y}_k^\top\breve{\mathbf{r}}^{(1)}. \tag{3.2}$$

The basic approach is to extend the solution space with an additional ℓ vectors generated by the recycling GKB process. Starting with $\widetilde{\mathbf{u}}_1 = \widetilde{\mathbf{b}}/\widetilde{\beta}_1$ where $\widetilde{\beta}_1 = \|\widetilde{\mathbf{b}}\|_2$ (note $\widetilde{\mathbf{u}}_1 \perp \mathbf{Y}_k$) and $\widetilde{\alpha}_1\widetilde{\mathbf{v}}_1 = \mathbf{A}^\top\widetilde{\mathbf{u}}_1$, at the jth iteration of the recycling GKB process, we generate vectors $\widetilde{\mathbf{u}}_{j+1}$ and $\widetilde{\mathbf{v}}_{j+1}$ as

$$\widetilde{\beta}_{j+1}\widetilde{\mathbf{u}}_{j+1} = \left(\mathbf{I} - \mathbf{Y}_k\mathbf{Y}_k^\top\right)\mathbf{A}\widetilde{\mathbf{v}}_j - \widetilde{\alpha}_j\widetilde{\mathbf{u}}_j, \tag{3.3}$$

$$\widetilde{\alpha}_{j+1}\widetilde{\mathbf{v}}_{j+1} = \mathbf{A}^\top\widetilde{\mathbf{u}}_{j+1} - \widetilde{\beta}_{j+1}\widetilde{\mathbf{v}}_j, \tag{3.4}$$

and after ℓ iterations, we have the following recurrence relation (cf. (2.2)–(2.3)):

$$\left(\mathbf{I} - \mathbf{Y}_k\mathbf{Y}_k^\top\right)\mathbf{A}\widetilde{\mathbf{V}}_\ell = \widetilde{\mathbf{U}}_{\ell+1}\widetilde{\mathbf{B}}_\ell, \tag{3.5}$$

$$\mathbf{A}^\top\widetilde{\mathbf{U}}_{\ell+1} = \widetilde{\mathbf{V}}_\ell\widetilde{\mathbf{B}}_\ell^\top + \widetilde{\alpha}_{\ell+1}\widetilde{\mathbf{v}}_{\ell+1}\mathbf{e}_{\ell+1}^\top, \tag{3.6}$$

where $\widetilde{\mathbf{V}}_\ell = \begin{bmatrix}\widetilde{\mathbf{v}}_1 & \dots & \widetilde{\mathbf{v}}_\ell\end{bmatrix} \in \mathbb{R}^{N\times\ell}$, $\widetilde{\mathbf{U}}_{\ell+1} = \begin{bmatrix}\widetilde{\mathbf{u}}_1 & \dots & \widetilde{\mathbf{u}}_{\ell+1}\end{bmatrix} \in \mathbb{R}^{M\times(\ell+1)}$, and bidiagonal matrix $\widetilde{\mathbf{B}}_\ell \in \mathbb{R}^{(\ell+1)\times\ell}$ is constructed during the iterative process. Notice that by construction $\widetilde{\mathbf{U}}_{\ell+1}^\top\mathbf{Y}_k = \mathbf{O}$, where $\mathbf{O}$ is the zero matrix, and hence $\mathbf{A}^\top\widetilde{\mathbf{u}}_j \perp \mathbf{W}_k$ for $j = 1,\dots,(\ell+1)$, since $\mathbf{W}_k^\top\mathbf{A}^\top\widetilde{\mathbf{U}}_{\ell+1} = \mathbf{R}_k^\top\mathbf{Y}_k^\top\widetilde{\mathbf{U}}_{\ell+1} = \mathbf{O}$. Hence, $\widetilde{\mathbf{v}}_1 \perp \mathbf{W}_k$, and if we assume that $\widetilde{\mathbf{v}}_i \perp \mathbf{W}_k$ for $i = 1,\dots,j$, then by induction we have from (3.4),

$$\widetilde{\alpha}_{j+1}\mathbf{W}_k^\top\widetilde{\mathbf{v}}_{j+1} = \mathbf{W}_k^\top\mathbf{A}^\top\widetilde{\mathbf{u}}_{j+1} - \widetilde{\beta}_{j+1}\mathbf{W}_k^\top\widetilde{\mathbf{v}}_j = \mathbf{0}.$$

Thus, $\mathbf{W}_k \perp \widetilde{\mathbf{V}}_\ell$ in exact arithmetic, without explicit orthogonalization. We notice from (3.5) that $\mathbf{A}\widetilde{\mathbf{V}}_\ell = \mathbf{Y}_k\mathbf{Y}_k^\top\mathbf{A}\widetilde{\mathbf{V}}_\ell + \widetilde{\mathbf{U}}_{\ell+1}\widetilde{\mathbf{B}}_\ell$, so we have the recycling GKB relation,

$$\mathbf{A}\begin{bmatrix}\mathbf{W}_k & \widetilde{\mathbf{V}}_\ell\end{bmatrix} = \begin{bmatrix}\mathbf{Y}_k & \widetilde{\mathbf{U}}_{\ell+1}\end{bmatrix}\begin{bmatrix}\mathbf{R}_k & \mathbf{Y}_k^\top\mathbf{A}\widetilde{\mathbf{V}}_\ell \\ \mathbf{0} & \widetilde{\mathbf{B}}_\ell\end{bmatrix}, \tag{3.7}$$

where $[\mathbf{W}_k \ \ \widetilde{\mathbf{V}}_\ell]$ and $[\mathbf{Y}_k \ \ \widetilde{\mathbf{U}}_{\ell+1}]$ both contain orthonormal columns.

Thus far, we have described a recycling GKB approach that can be used to augment a given solution subspace. A distinguishing factor of this approach compared with existing enhancement methods is that the new, augmented, Krylov subspace depends on the recycled subspace. Indeed, one can characterize the augmented solution subspace as a Krylov subspace of the form

$$\mathcal{R}\left(\widetilde{\mathbf{V}}_\ell\right) = \mathcal{K}_\ell\left(\mathbf{A}^\top\left(\mathbf{I}-\mathbf{Y}_k\mathbf{Y}_k^\top\right)\mathbf{A}, \mathbf{A}^\top\left(\mathbf{I}-\mathbf{Y}_k\mathbf{Y}_k^\top\right)\breve{\mathbf{r}}^{(1)}\right).$$

3.2. Hybrid projection methods using the recycling GKB. Next, we describe how the recycling GKB process can be incorporated within a hybrid projection method for efficient regularized solution computation. Suppose we have performed ℓ iterations of the recycling GKB process, and we are interested in computing approximate Tikhonov solutions in the augmented solution subspace $\mathcal{R}([\mathbf{W}_k \quad \widetilde{\mathbf{V}}_\ell])$, i.e., we are looking for solutions of the form $\mathbf{x}_{k,\ell} = [\mathbf{W}_k \quad \widetilde{\mathbf{V}}_\ell]\mathbf{y}$ where $\mathbf{y} = \begin{bmatrix}\mathbf{c}^\top & \mathbf{d}^\top\end{bmatrix}^\top$ for some vectors $\mathbf{c} \in \mathbb{R}^k$ and $\mathbf{d} \in \mathbb{R}^\ell$. Using the fact that $\widetilde{\mathbf{b}} = \breve{\mathbf{r}}^{(1)} - \mathbf{Y}_k\boldsymbol{\zeta} = \widetilde{\beta}_1\widetilde{\mathbf{U}}_{\ell+1}\mathbf{e}_1$ and $\breve{\mathbf{x}}^{(1)} = \mathbf{W}_k\mathbf{e}_k$, we have

$$\mathbf{b} = \breve{\mathbf{r}}^{(1)} + \mathbf{A}\breve{\mathbf{x}}^{(1)} = \mathbf{Y}_k\zeta + \widetilde{\beta}_1\widetilde{\mathbf{U}}_{\ell+1}\mathbf{e}_1 + \mathbf{A}\mathbf{W}_k\mathbf{e}_k \tag{3.8}$$

$$= \begin{bmatrix} \mathbf{Y}_k & \widetilde{\mathbf{U}}_{\ell+1} \end{bmatrix}\begin{bmatrix} \zeta + \mathbf{R}_k\mathbf{e}_k \\ \widetilde{\beta}_1\mathbf{e}_1 \end{bmatrix}. \tag{3.9}$$

Then, using (3.7), the residual can be written as

$$\mathbf{b} - \mathbf{A}\begin{bmatrix}\mathbf{W}_k & \widetilde{\mathbf{V}}_\ell\end{bmatrix}\begin{bmatrix}\mathbf{c}\\ \mathbf{d}\end{bmatrix} = \begin{bmatrix}\mathbf{Y}_k & \widetilde{\mathbf{U}}_{\ell+1}\end{bmatrix}\left(\begin{bmatrix} \boldsymbol{\zeta} + \mathbf{R}_k\mathbf{e}_k \\ \widetilde{\beta}_1\mathbf{e}_1 \end{bmatrix} - \begin{bmatrix} \mathbf{R}_k & \mathbf{Y}_k^\top\mathbf{A}\widetilde{\mathbf{V}}_\ell \\ \mathbf{O} & \widetilde{\mathbf{B}}_\ell \end{bmatrix}\begin{bmatrix}\mathbf{c}\\ \mathbf{d}\end{bmatrix}\right). \tag{3.10}$$

Thus, the next iterate of the hybrid projection method with recycling is given by

$$\mathbf{x}^{(2)} = \begin{bmatrix} \mathbf{W}_k & \widetilde{\mathbf{V}}_\ell \end{bmatrix}\widetilde{\mathbf{y}}_\lambda, \tag{3.11}$$

where

$$\widetilde{\mathbf{y}}_\lambda = \arg\min_{\mathbf{y}} \left\| \begin{bmatrix} \mathbf{R}_k & \mathbf{Y}_k^\top\mathbf{A}\widetilde{\mathbf{V}}_\ell \\ \mathbf{O} & \widetilde{\mathbf{B}}_\ell \end{bmatrix}\mathbf{y} - \begin{bmatrix} \boldsymbol{\zeta} + \mathbf{R}_k\mathbf{e}_k \\ \widetilde{\beta}_1\mathbf{e}_1 \end{bmatrix}\right\|_2^2 + \lambda^2\left\|\mathbf{y}\right\|_2^2. \tag{3.12}$$

Notice that the coefficient matrix in the projected problem

$$\widehat{\mathbf{B}}_{k,\ell} = \begin{bmatrix} \mathbf{R}_k & \mathbf{Y}_k^\top\mathbf{A}\widetilde{\mathbf{V}}_\ell \\ \mathbf{0} & \widetilde{\mathbf{B}}_\ell \end{bmatrix} \in \mathbb{R}^{(k+\ell+1)\times(k+\ell)} \tag{3.13}$$

is modest in size. Thus, standard regularization parameter selection methods can be used to choose λ for (3.12). Based on the above derivation, we can interpret iterates of the hybrid projection method with recycling as optimal solutions in a $(k+\ell)$ dimensional subspace. That is, for fixed $\lambda \geqslant 0$,

$$\mathbf{x}^{(2)} = \underset{\mathbf{x}\in\mathcal{R}\left(\begin{bmatrix}\mathbf{W}_k & \widetilde{\mathbf{V}}_\ell\end{bmatrix}\right)}{\arg\min} ||\mathbf{A}\mathbf{x}-\mathbf{b}||_2^2 + \lambda^2\left\|\mathbf{x}\right\|_2^2. \tag{3.14}$$

If the solution is not sufficiently accurate, the process can be repeated in an iterative fashion by selecting a new subspace $\mathcal{R}(\mathbf{W}_{k-1}^{(new)}) \subset \mathcal{R}([\mathbf{W}_k \ \widetilde{\mathbf{V}}_\ell])$ (for example, using one of the compression approaches provided in the next section), where $\mathbf{W}_{k-1}^{(new)}$ has orthonormal columns, and set $\mathbf{W}_k^{(new)} = [\mathbf{W}_{k-1}^{(new)} \ \breve{\mathbf{x}}^{(2)}]$ with

$$\breve{\mathbf{x}}^{(2)} = \left(\mathbf{I} - \mathbf{W}_{k-1}^{(new)} \left(\mathbf{W}_{k-1}^{(new)}\right)^\top\right) \mathbf{x}^{(2)} / \left\| \left(\mathbf{I} - \mathbf{W}_{k-1}^{(new)} \left(\mathbf{W}_{k-1}^{(new)}\right)^\top\right) \mathbf{x}^{(2)} \right\|_2 .$$

Note, that $\mathcal{R}(\mathbf{W}_k^{(new)}) \subset \mathcal{R}([\mathbf{W}_k \ \widetilde{\mathbf{V}}_\ell])$. Next, we set $\breve{\mathbf{r}}^{(2)} = \mathbf{b} - \mathbf{A}\breve{\mathbf{x}}^{(2)}$, and repeat the steps above.

We remark on the additional computational cost if full reorthogonalization is desired. In particular, the recursion

$$\widetilde{\alpha}_{j+1}\widetilde{\mathbf{v}}_{j+1} = \left(\mathbf{I} - \mathbf{W}_k\mathbf{W}_k^\top\right)\mathbf{A}^\top\widetilde{\mathbf{u}}_{j+1} - \widetilde{\beta}_{j+1}\widetilde{\mathbf{v}}_j \tag{3.15}$$

can be used in place of (3.4) to ensure that the solution basis vectors $[\mathbf{W}_k \ \widetilde{\mathbf{V}}_\ell]$ are orthogonal in floating point arithmetic. In this case, the additional computational cost is $4kN$ operations for each iteration.

3.3. Compression approaches. One feature of the hybrid projection methods with recycling is the ability to combine compression and extension of the solution space in an iterative manner. That is, compression techniques can reduce the total number of solution vectors that we need to store, which can be followed by enhancement of the space, and this can be done without significantly degrading the accuracy of the resulting reconstruction. More specifically, let $\mathbf{V}^c$ represent the current set of basis vectors, and assume that we can only afford to store m vectors of length N. When the number of columns in $\mathbf{V}^c$ reaches m, we can compress the vectors in $\mathbf{V}^c$ to get $\mathbf{W}_{k-1} \in \mathbb{R}^{N\times(k-1)}$ (see line 10 in Algorithm 3.1). Then, we can construct $\mathbf{W}_k$ using an initial guess or current approximate solution and use the method described in subsection 3.1 to augment the space with $\widetilde{\mathbf{V}}_\ell$, where $\ell = m - k$.

In this section, we focus on four compression strategies for constructing $\mathbf{W}_{k-1}$ that are well-suited for solving inverse problems with the recycling GKB process. These include TSVD, reduced basis decomposition (RBD), solution-oriented compression, and sparsity enforcing compression. The described compression strategies follow two perspectives: (1) decompose $\widehat{\mathbf{B}}_{k,\ell} \in \mathbb{R}^{(m+1)\times m}$ defined in (3.13) and use truncation (e.g., TSVD and RBD) or (2) use components in the solution of the projected problem (3.12) to identify the important columns of $\mathbf{V}^c$ (e.g., sparsity enforcing and solution-oriented compression). Throughout this subsection, we define $1 \leqslant q < m$ as the largest number of length N vectors we wish to keep after compression and $\epsilon_{tol} > 0$ is a tolerance for the compression.

First we describe the TSVD approach for compressing $\mathbf{V}^c$. Let the SVD of $\widehat{\mathbf{B}}_{k,\ell}$ be given as

$$\widehat{\mathbf{B}}_{k,\ell} = \mathbf{\Psi}_{m+1}\mathbf{\Sigma}_m\mathbf{\Phi}_m^\top, \tag{3.16}$$

where $\mathbf{\Psi}_{m+1} \in \mathbb{R}^{(m+1)\times(m+1)}$ and $\mathbf{\Phi}_m \in \mathbb{R}^{m\times m}$ are orthogonal matrices, and $\mathbf{\Sigma}_m \in \mathbb{R}^{(m+1)\times m}$ is a diagonal matrix containing singular values $\sigma_i, i = 1, \ldots, m$. If $\sigma_q < \epsilon_{tol}$, we let $k-1 = i$, where i is the largest index such that $\sigma_i \geqslant \epsilon_{tol}$; otherwise $k-1 = q$. The key point of this compression strategy is that we identify the important columns

of $\mathbf{V}^c$ as those corresponding to the large singular values of $\widehat{\mathbf{B}}_{k,\ell}$. The compressed representation of $\mathbf{V}^c$ is given by

$$\mathbf{W}_{k-1} = \mathbf{V}^c \mathbf{\Phi}_{k-1}, \tag{3.17}$$

where $\mathbf{\Phi}_{k-1}$ contains the first $k-1$ columns of $\mathbf{\Phi}_m$.

Second, we exploit tools from reduced order modeling [7] to compress the solution vectors. For $1 \leqslant i \leqslant q$, we consider the RBD of $\widehat{\mathbf{B}}_{k,\ell}^\top$,

$$\widehat{\mathbf{B}}_{k,\ell}^\top = \mathbf{S}_i \mathbf{T}_i, \tag{3.18}$$

where $\mathbf{S}_i \in \mathbb{R}^{m \times i}$ contains orthonormal columns and transformation matrix $\mathbf{T}_i \in \mathbb{R}^{i \times (m+1)}$. Define $\mathcal{E}_i = \max_{1 \leqslant j \leqslant (m+1)} \|\widehat{\mathbf{B}}_{k,\ell}^\top(:,j) - \mathbf{S}_i \mathbf{T}_i(:,j)\|$. If $\mathcal{E}_q < \epsilon_{tol}$, we let $k-1 = i$, where i is the largest index such that $\mathcal{E}_i \geqslant \epsilon_{tol}$; otherwise $k-1 = q$. We use $\mathbf{S}_{k-1}$ to indicate important columns of $\mathbf{V}^c$, thus the compressed vectors are obtained as $\mathbf{W}_{k-1} = \mathbf{V}^c \mathbf{S}_{k-1}$.

The third compression approach is motivated by the notion that the absolute value of each component of the solution to the projected problem $\widetilde{\mathbf{y}}_\lambda = \begin{bmatrix} \tilde{y}_1, & \ldots, & \tilde{y}_m \end{bmatrix}^\top$ is indicative of the important columns of $\mathbf{V}^c$. We define I_m, J_m as an index set at the mth iteration:

$$I_m = \{i : \quad |\tilde{y}_i| > \epsilon_{tol},\ 1 \leqslant i \leqslant N\}, \tag{3.19}$$

$$J_m = \{i : \quad |\tilde{y}_i| \text{ are the largest } q \text{ components},\ 1 \leqslant i \leqslant N\}. \tag{3.20}$$

For solution-oriented compression, we define $\mathbf{W}_{k-1} = \begin{bmatrix} \mathbf{v}_{m_1} & \ldots & \mathbf{v}_{m_{k-1}} \end{bmatrix}$, where $k - 1 = |I_m \cap J_m|$, $\{m_j\}_{j=1}^{k-1} \subseteq (I_m \cap J_m)$ and $m_1 \leqslant \cdots \leqslant m_{k-1}$.

The fourth compression approach called sparsity-enforcing compression is intuitively similar to the solution-oriented method. The basic idea is to use $\widetilde{\mathbf{y}}_\lambda$ to identify the important vectors in $\mathbf{V}^c$; however, the difference is that we employ a sparsity enforcing regularization term on the projected problem. A standard algorithm such as SpaRSA [50] can be used to solve for $\widetilde{\mathbf{y}}_\lambda$, and then corresponding vectors of $\mathbf{V}^c$ can be extracted similarly to solution-oriented compression. It is worth mentioning that $\widetilde{\mathbf{y}}_\lambda$ is only used for identifying important columns of $\mathbf{V}^c$ (i.e., for compression) and not for the solution computation.

3.4. Theoretical analysis of hybrid projection methods with recycling. In this section, we analyze theoretical properties of regularized solutions and the projected system using compression and recycling, in the important case that we run m steps of standard GKB (see section 2), compress the search space to dimension k, as described in section 3.3 with $\mathcal{R}(\mathbf{W}_k) \subset \mathcal{R}(\mathbf{V}_m)$, and carry out ℓ steps of recycling GKB (see section 3.1) which is incorporated in a hybrid projection method (see section 3.2). This scenario corresponds to the case where we can store a maximum of m vectors of length N, but a hybrid projection method with standard GKB requires more iterations to converge.

First, we analyze the storage requirements. Let j denote the number of iterations for a standard hybrid method. Without full reorthogonalization, we need to save $\mathbf{V}_j \in \mathbb{R}^{N \times j}$, bidiagonal matrix $\mathbf{B}_j \in \mathbb{R}^{(j+1) \times j}$, and $\mathbf{u}_{j+1} \in \mathbb{R}^{M \times 1}$, where the storage cost is dominated by $\mathbf{V}_j$ if N is large. The total storage cost of standard hybrid iterative methods is

$$\mathcal{C}_{\text{HyBR}}(j) := 2j + (N+2)j + M.$$

As j increases, $\mathcal{C}_{\text{HyBR}}(j)$ is dominated by Nj. Thus, for very-large-scale problems, $\mathcal{C}_{\text{HyBR}}(j)$ increases rapidly and can easily exceed the storage limit. For the proposed recycling GKB hybrid method, we need to save $\mathbf{W}_k \in \mathbb{R}^{N\times k}, \mathbf{Y}_k \in \mathbb{R}^{M\times k}, \mathbf{R}_k \in \mathbb{R}^{k\times k}, \mathbf{e}_k \in \mathbb{R}^{k\times 1}, \widetilde{\mathbf{B}}_\ell \in \mathbb{R}^{(\ell+1)\times\ell}, \widetilde{\mathbf{V}}_\ell \in \mathbb{R}^{N\times\ell}, \boldsymbol{\zeta} \in \mathbb{R}^{k\times 1}$, $\mathbf{Y}_k^\top \mathbf{A}\widetilde{\mathbf{V}}_\ell \in \mathbb{R}^{k\times\ell}$, and $\widetilde{\mathbf{u}}_\ell \in \mathbb{R}^{M\times 1}$, where $\mathbf{R}_k$ is an upper triangular matrix and $\widetilde{\mathbf{B}}_\ell$ is a bidiagonal matrix. Since $\ell = m - k$, the storage cost of recycling GKB is

$$\begin{aligned}\mathcal{C}_{\text{HyBR-recycle}} &:= k^2/2 + (N + M + 2)k + 2\ell + (N + 1)\ell + k\ell \\ &= (N + 2)m + Mk + k^2/2 + \ell(k + 1) \\ &< m^2/2 + (N + M + 2)m.\end{aligned}$$

Therefore, the storage requirements of $\mathcal{C}_{\text{HyBR-recycle}}$ do not grow with the number of iterations.

Next, we consider several consequences of compression and augmentation for the projected problem. We are interested in comparing the properties of the GKB matrix $\widehat{\mathbf{B}}_{k,\ell}$ obtained with recycling with properties of the GKB matrix $\mathbf{B}_{m+\ell}$ obtained with $m + \ell$ standard GKB iterations. In addition, we show that under reasonable assumptions and for the same regularization parameter, the regularized solution from the recycling approach is close to the regularized solution from the standard approach. For the particular case of TSVD compression, we give precise and (a posteriori) computable bounds.

We start with a lemma that shows important relations between the generated subspaces and then consider its consequence for relations between $\widehat{\mathbf{B}}_{k,\ell}$ and $\mathbf{B}_{m+\ell}$.

LEMMA 3.1. *Let $\mathbf{V}_{m+\ell}$, $\mathbf{U}_{m+\ell+1}$, and $\mathbf{B}_{m+\ell}$ be the matrices computed after $m+\ell$ iterations of standard GKB, following* (2.1). *Let $\mathbf{x}^{(1)}$ be a (arbitrary) regularized solution computed from $\mathcal{R}(\mathbf{V}_m)$, $\mathcal{R}(\mathbf{W}_{k-1}) \subset \mathcal{R}(\mathbf{V}_m)$ (obtained by any compression method), and $\mathbf{W}_k$ be computed as described at the start of section* 3.1 *with $\mathbf{Y}_k$, $\mathbf{R}_k$ given in* (3.1). *In addition, let $\widetilde{\mathbf{U}}_{\ell+1} = \begin{bmatrix}\widetilde{\mathbf{u}}_1 & \dots & \widetilde{\mathbf{u}}_{\ell+1}\end{bmatrix}$ and $\widetilde{\mathbf{V}}_\ell = \begin{bmatrix}\widetilde{\mathbf{v}}_1 & \dots & \widetilde{\mathbf{v}}_\ell\end{bmatrix}$ be obtained after ℓ iterations of recycling GKB following* (3.3)–(3.4). *Then*

$$\mathcal{R}\left(\widetilde{\mathbf{U}}_{\ell+1}\right) \subset \mathcal{R}(\mathbf{U}_{m+\ell+1}) \quad \text{and} \quad \mathcal{R}\left(\widetilde{\mathbf{V}}_\ell\right) \subset \mathcal{R}(\mathbf{V}_{m+\ell}). \tag{3.21}$$

Proof. We prove the result by induction. In recycling GKB, $\widetilde{\mathbf{u}}_1 = \widetilde{\mathbf{b}}/\|\widetilde{\mathbf{b}}\|_2$ with $\widetilde{\mathbf{b}} = \breve{\mathbf{r}}^{(1)} - \mathbf{Y}_k\mathbf{Y}_k^\top\breve{\mathbf{r}}^{(1)}$ and $\breve{\mathbf{r}}^{(1)} = \mathbf{b} - \mathbf{A}\breve{\mathbf{x}}^{(1)}$. By construction, $\breve{\mathbf{x}}^{(1)} \in \mathcal{R}(\mathbf{V}_m)$, and hence $\mathbf{A}\breve{\mathbf{x}}^{(1)} \in \mathcal{R}(\mathbf{AV}_m)$. We also have $\mathcal{R}(\mathbf{Y}_k) \subset \mathcal{R}(\mathbf{AV}_m)$, and, using (2.2)–(2.3), $\mathcal{R}(\mathbf{AV}_m) \subset \mathcal{R}(\mathbf{U}_{m+1})$. Since $\mathbf{b} = \beta_1\mathbf{u}_1 \in \mathcal{R}(\mathbf{U}_{m+1})$, we have $\widetilde{\mathbf{u}}_1 \in \mathcal{R}(\mathbf{U}_{m+1})$. Therefore, $\mathbf{A}^\top\widetilde{\mathbf{u}}_1 \in \mathcal{R}(\mathbf{A}^\top\mathbf{U}_{m+1}) = \mathcal{R}(\mathbf{V}_{m+1})$, and as $\widetilde{\alpha}_1\widetilde{\mathbf{v}}_1 = \mathbf{A}^\top\widetilde{\mathbf{u}}_1$, $\widetilde{\mathbf{v}}_1 \in \mathcal{R}(\mathbf{V}_{m+1})$. Since $\mathbf{A}\widetilde{\mathbf{v}}_1 \in \mathcal{R}(\mathbf{AV}_{m+1}) \subset \mathcal{R}(\mathbf{U}_{m+2})$ and $\widetilde{\beta}_2\widetilde{\mathbf{u}}_2 = (\mathbf{I} - \mathbf{Y}_k\mathbf{Y}_k^\top)\mathbf{A}\widetilde{\mathbf{v}}_1 - \widetilde{\alpha}_1\widetilde{\mathbf{u}}_1$, it follows that $\widetilde{\mathbf{u}}_2 \in \mathcal{R}(\mathbf{U}_{m+2})$. Now assume that $\widetilde{\mathbf{u}}_{i+1} \in \mathcal{R}(\mathbf{U}_{m+i+1})$ and $\widetilde{\mathbf{v}}_i \in \mathcal{R}(\mathbf{V}_{m+i})$ for $i = 1, \dots, j$. Since $\mathcal{R}(\mathbf{Y}_k) \subset \mathcal{R}(\mathbf{U}_{m+1})$, we get from (3.3)–(3.4) that

$$\widetilde{\mathbf{v}}_{j+1} \in \mathcal{R}(\mathbf{V}_{m+j+1}) \quad \text{and} \quad \widetilde{\mathbf{u}}_{j+1} \in \mathcal{R}(\mathbf{U}_{m+j+1}). \qquad \square$$

The next result is presented without its (straightforward) proof.

LEMMA 3.2. *Let $\mathbf{A} \in \mathbb{R}^{M\times N}$ and $\mathbf{b} \in \mathbb{R}^M$, and let $\mathbf{P} \in \mathbb{R}^{N\times N}$ and $\mathbf{Q} \in \mathbb{R}^{M\times M}$ be orthogonal matrices. For any given λ, the Tikhonov solutions*

$$\begin{aligned}\mathbf{x}_\lambda &= \arg\min_{\mathbf{x}\in\mathbb{R}^N} \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \lambda^2\|\mathbf{x}\|_2^2, \\ \widetilde{\mathbf{x}}_\lambda &= \arg\min_{\widetilde{\mathbf{x}}\in\mathbb{R}^N} \|\mathbf{QAP}^\top\widetilde{\mathbf{x}} - \mathbf{Qb}\|_2^2 + \lambda^2\|\widetilde{\mathbf{x}}\|_2^2\end{aligned}$$

satisfy $\mathbf{x}_\lambda = \mathbf{P}^\top\widetilde{\mathbf{x}}_\lambda$.

Next, we derive the orthogonal transformations that relate the Lanczos bases for the recycling GKB iteration with compression to those of the standard GKB iteration, and the resulting relations between $\widehat{\mathbf{B}}_{k,\ell}$ and $\mathbf{B}_{m+\ell}$. From Lemma 3.1 and the construction of $\mathbf{W}_k$ and $\mathbf{Y}_k$, we see that $\mathcal{R}(\mathbf{W}_k) + \mathcal{R}(\widetilde{\mathbf{V}}_\ell) \subset \mathcal{R}(\mathbf{V}_{m+\ell})$ and $\mathcal{R}(\mathbf{Y}_k)+\mathcal{R}(\widetilde{\mathbf{U}}_{\ell+1}) \subset \mathcal{R}(\mathbf{U}_{m+\ell+1})$. In addition, by construction the matrices $[\mathbf{W}_k \;\; \widetilde{\mathbf{V}}_\ell]$, $[\mathbf{Y}_k \;\; \widetilde{\mathbf{U}}_{\ell+1}]$, $\mathbf{V}_{m+\ell}$, and $\mathbf{U}_{m+\ell+1}$ have orthonormal columns. Hence, there exist orthogonal matrices $\mathbf{T} = \begin{bmatrix}\mathbf{T}_1 & \mathbf{T}_2 & \mathbf{T}_c\end{bmatrix} \in \mathbb{R}^{(m+\ell+1)\times(m+\ell+1)}$ and $\mathbf{Z} = \begin{bmatrix}\mathbf{Z}_1 & \mathbf{Z}_2 & \mathbf{Z}_c\end{bmatrix} \in \mathbb{R}^{(m+\ell)\times(m+\ell)}$ such that $\mathbf{Y}_k = \mathbf{U}_{m+\ell+1}\mathbf{T}_1$, $\widetilde{\mathbf{U}}_{\ell+1} = \mathbf{U}_{m+\ell+1}\mathbf{T}_2$, $\mathbf{W}_k = \mathbf{V}_{m+\ell}\mathbf{Z}_1$, and $\widetilde{\mathbf{V}}_\ell = \mathbf{V}_{m+\ell}\mathbf{Z}_2$. The subspace $\mathcal{R}(\mathbf{V}_{m+\ell}\mathbf{Z}_c)$ is the orthogonal complement of the compressed solution space $\mathcal{R}\left(\mathbf{V}_{m+\ell}\begin{bmatrix}\mathbf{Z}_1 & \mathbf{Z}_2\end{bmatrix}\right)$ with respect to the (full) GKB solution space $\mathcal{R}(\mathbf{V}_{m+\ell})$. An analogous relation holds for $\mathcal{R}(\mathbf{U}_{m+\ell+1}\mathbf{T}_c)$. Substituting these relations in (3.7) and using the fact that $\mathbf{U}_{m+\ell+1}[\mathbf{T}_1 \;\; \mathbf{T}_2]$ has orthonormal columns, we obtain

$$
\begin{aligned}
\mathbf{A}\begin{bmatrix}\mathbf{W}_k & \widetilde{\mathbf{V}}_\ell\end{bmatrix} &= \begin{bmatrix}\mathbf{Y}_k & \widetilde{\mathbf{U}}_{\ell+1}\end{bmatrix}\begin{bmatrix}\mathbf{R}_k & \mathbf{Y}_k^\top\mathbf{A}\widetilde{\mathbf{V}}_\ell \\ \mathbf{0} & \widetilde{\mathbf{B}}_\ell\end{bmatrix} \\
&\Leftrightarrow \mathbf{A}\mathbf{V}_{m+\ell}[\mathbf{Z}_1 \;\; \mathbf{Z}_2] = \mathbf{U}_{m+\ell+1}[\mathbf{T}_1 \;\; \mathbf{T}_2]\,\widehat{\mathbf{B}}_{k,\ell} \\
&\Rightarrow \widehat{\mathbf{B}}_{k,\ell} = \begin{bmatrix}\mathbf{T}_1 & \mathbf{T}_2\end{bmatrix}^\top \mathbf{U}_{m+\ell+1}^\top\mathbf{A}\mathbf{V}_{m+\ell}\begin{bmatrix}\mathbf{Z}_1 & \mathbf{Z}_2\end{bmatrix} \\
&\qquad = \begin{bmatrix}\mathbf{T}_1 & \mathbf{T}_2\end{bmatrix}^\top \mathbf{B}_{m+\ell}\begin{bmatrix}\mathbf{Z}_1 & \mathbf{Z}_2\end{bmatrix}. \qquad (3.22)
\end{aligned}
$$

Blockwise, we have $\mathbf{T}_1^\top\mathbf{B}_{m+\ell}\mathbf{Z}_1 = \mathbf{R}_k$, $\mathbf{T}_1^\top\mathbf{B}_{m+\ell}\mathbf{Z}_2 = \mathbf{Y}_k^\top\mathbf{A}\widetilde{\mathbf{V}}_\ell$, $\mathbf{T}_2^\top\mathbf{B}_{m+\ell}\mathbf{Z}_1 = \mathbf{O}$, and $\mathbf{T}_2^\top\mathbf{B}_{m+\ell}\mathbf{Z}_2 = \widetilde{\mathbf{B}}_\ell$. For the (3,1) block we have

$$
\begin{aligned}
\mathbf{T}_c^\top\mathbf{B}_{m+\ell}\mathbf{Z}_1 &= \mathbf{T}_c^\top\mathbf{U}_{m+\ell+1}^\top\mathbf{A}\mathbf{V}_{m+\ell}\mathbf{Z}_1 = \mathbf{T}_c^\top\mathbf{U}_{m+\ell+1}^\top\mathbf{A}\mathbf{W}_k = \mathbf{T}_c^\top\mathbf{U}_{m+\ell+1}^\top\mathbf{Y}_k\mathbf{R}_k \\
&= \mathbf{T}_c^\top\mathbf{U}_{m+\ell+1}^\top\mathbf{U}_{m+\ell+1}\mathbf{T}_1\mathbf{R}_k = \mathbf{O}.
\end{aligned}
$$

For the (3,2) block we have

$$
\begin{aligned}
\mathbf{T}_c^\top\mathbf{B}_{m+\ell}\mathbf{Z}_2 &= \mathbf{T}_c^\top\mathbf{U}_{m+\ell+1}^\top\mathbf{A}\mathbf{V}_{m+\ell}\mathbf{Z}_2 = \mathbf{T}_c^\top\mathbf{U}_{m+\ell+1}^\top\mathbf{A}\widetilde{\mathbf{V}}_\ell \\
&= \mathbf{T}_c^\top\mathbf{U}_{m+\ell+1}^\top(\mathbf{Y}_k\mathbf{Y}_k^\top\mathbf{A}\widetilde{\mathbf{V}}_\ell + \widetilde{\mathbf{U}}_{\ell+1}\widetilde{\mathbf{B}}_\ell) \\
&= \mathbf{T}_c^\top\mathbf{U}_{m+\ell+1}^\top\mathbf{U}_{m+\ell+1}\begin{bmatrix}\mathbf{T}_1 & \mathbf{T}_2\end{bmatrix}\begin{bmatrix}\mathbf{Y}_k^\top\mathbf{A}\widetilde{\mathbf{V}}_\ell \\ \widetilde{\mathbf{B}}_\ell\end{bmatrix} = \mathbf{O}.
\end{aligned}
$$

This gives the following lemma.

LEMMA 3.3.

$$
\begin{bmatrix}\mathbf{T}_1 & \mathbf{T}_2 & \mathbf{T}_c\end{bmatrix}^\top \mathbf{B}_{m+\ell}\begin{bmatrix}\mathbf{Z}_1 & \mathbf{Z}_2 & \mathbf{Z}_c\end{bmatrix} = \begin{bmatrix}\mathbf{R}_k & \mathbf{Y}_k^\top\mathbf{A}\widetilde{\mathbf{V}}_\ell & \mathbf{T}_1^\top\mathbf{B}_{m+\ell}\mathbf{Z}_c \\ \mathbf{O} & \widetilde{\mathbf{B}}_\ell & \mathbf{T}_2^\top\mathbf{B}_{m+\ell}\mathbf{Z}_c \\ \mathbf{O} & \mathbf{O} & \mathbf{T}_c^\top\mathbf{B}_{m+\ell}\mathbf{Z}_c\end{bmatrix}. \qquad (3.23)
$$

Next we consider the difference between the regularized solution to (3.12) and the regularized solution to the full (transformed) problem with system matrix (3.23). In particular, we analyze the backward error, and then consider bounds on the backward error for the special case of compression based on the TSVD. Let λ be given, typically

an appropriate λ for the regularized problem (3.12), and let $\widetilde{\mathbf{y}}_\lambda$ be given as in (3.12), i.e.,

$$\left(\widehat{\mathbf{B}}_{k,\ell}^\top \widehat{\mathbf{B}}_{k,\ell} + \lambda^2 \mathbf{I}\right) \widetilde{\mathbf{y}}_\lambda = \widehat{\mathbf{B}}_{k,\ell}^\top \begin{bmatrix} \mathbf{R}_k \mathbf{e}_k + \boldsymbol{\zeta} \\ \widetilde{\beta}_1 \mathbf{e}_1 \end{bmatrix}. \tag{3.24}$$

We consider the residual of the approximate solution $\begin{bmatrix} \widetilde{\mathbf{y}}_\lambda^\top & \mathbf{0}^\top \end{bmatrix}^\top$ for the regularized (transformed) full problem. According to (3.23),

$$\begin{aligned}
&\left(\left(\mathbf{T}^\top \mathbf{B}_{m+\ell} \mathbf{Z}\right)^\top \left(\mathbf{T}^\top \mathbf{B}_{m+\ell} \mathbf{Z}\right) + \lambda^2 \mathbf{I}\right) \mathbf{y} \\
&\quad = \left(\mathbf{T}^\top \mathbf{B}_{m+\ell} \mathbf{Z}\right)^\top \mathbf{T}^\top \mathbf{e}_1 \beta_1 = \left(\mathbf{T}^\top \mathbf{B}_{m+\ell} \mathbf{Z}\right)^\top \mathbf{T}^\top \mathbf{U}_{m+\ell+1}^\top \mathbf{b} \\
&\quad = \left(\mathbf{T}^\top \mathbf{B}_{m+\ell} \mathbf{Z}\right)^\top \begin{bmatrix} \mathbf{Y}_k & \widetilde{\mathbf{U}}_{\ell+1} & \mathbf{U}_{m+\ell+1} \mathbf{T}_c \end{bmatrix}^\top \mathbf{b} = \left(\mathbf{T}^\top \mathbf{B}_{m+\ell} \mathbf{Z}\right)^\top \begin{bmatrix} \mathbf{R}_k \mathbf{e}_k + \boldsymbol{\zeta} \\ \widetilde{\beta}_1 \mathbf{e}_1 \\ \mathbf{0} \end{bmatrix}.
\end{aligned}$$

The residual for $\begin{bmatrix} \widetilde{\mathbf{y}}_\lambda^\top & \mathbf{0}^\top \end{bmatrix}^\top$ for the full transformed problem is given by

$$\begin{aligned}
\mathbf{r}_\lambda &= \begin{bmatrix} \widehat{\mathbf{B}}_{k,\ell}^\top & \mathbf{O} \\ \mathbf{Z}_c^\top \mathbf{B}_{m+\ell}^\top \begin{bmatrix} \mathbf{T}_1 & \mathbf{T}_2 \end{bmatrix} & \mathbf{Z}_c^\top \mathbf{B}_{m+\ell}^\top \mathbf{T}_c \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \mathbf{R}_k \mathbf{e}_k + \boldsymbol{\zeta} \\ \widetilde{\beta}_1 \mathbf{e}_1 \end{bmatrix} \\ \mathbf{0} \end{bmatrix} \\
&\quad - \begin{bmatrix} \widehat{\mathbf{B}}_{k,\ell}^\top \widehat{\mathbf{B}}_{k,\ell} + \lambda^2 \mathbf{I} & \widehat{\mathbf{B}}_{k,\ell}^\top \begin{bmatrix} \mathbf{T}_1 & \mathbf{T}_2 \end{bmatrix}^\top \mathbf{B}_{m+\ell} \mathbf{Z}_c \\ \mathbf{Z}_c^\top \mathbf{B}_{m+\ell}^\top \begin{bmatrix} \mathbf{T}_1 & \mathbf{T}_2 \end{bmatrix} \widehat{\mathbf{B}}_{k,\ell} & \mathbf{Z}_c^\top \mathbf{B}_{m+\ell}^\top \mathbf{B}_{m+\ell} \mathbf{Z}_c + \lambda^2 \mathbf{I} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{y}}_\lambda \\ \mathbf{0} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{0} \\ \mathbf{Z}_c^\top \mathbf{B}_{m+\ell}^\top \begin{bmatrix} \mathbf{T}_1 & \mathbf{T}_2 \end{bmatrix} \left(\begin{bmatrix} \mathbf{R}_k \mathbf{e}_k + \boldsymbol{\zeta} \\ \widetilde{\beta}_1 \mathbf{e}_1 \end{bmatrix} - \widehat{\mathbf{B}}_{k,\ell} \widetilde{\mathbf{y}}_\lambda \right) \end{bmatrix}.
\end{aligned} \tag{3.25}$$

Note that

$$\widehat{\mathbf{r}}_\lambda = \begin{bmatrix} \mathbf{R}_k \mathbf{e}_k + \boldsymbol{\zeta} \\ \widetilde{\beta}_1 \mathbf{e}_1 \end{bmatrix} - \widehat{\mathbf{B}}_{k,\ell} \widetilde{\mathbf{y}}_\lambda \tag{3.26}$$

is just the residual for the regularized solution of (3.12) with the chosen λ, and its norm is known. The corresponding residuals for the full system are $[\mathbf{Y}_k \ \widetilde{\mathbf{U}}_{\ell+1}] \widehat{\mathbf{r}}_\lambda$, obtained with compression and recycling, and $[\mathbf{Y}_k \ \widetilde{\mathbf{U}}_{\ell+1} \ (\mathbf{U}_{m+\ell+1} \mathbf{T}_c)] \mathbf{r}_\lambda$, obtained with $m+\ell$ steps of standard GKB, but with the regularization parameter and solution from the compression and recycling approach. This gives the following theorem.

THEOREM 3.4. *Let* $\mathbf{Z}$, $\mathbf{T}$, $\mathbf{B}_{m+\ell}$, $\mathbf{r}_\lambda$, *and* $\widehat{\mathbf{r}}_\lambda$ *be defined as above. Then,*

$$\begin{aligned}
\mathbf{r}_\lambda &= \begin{bmatrix} \mathbf{0} \\ \mathbf{Z}_c^\top \mathbf{B}_{m+\ell}^\top \begin{bmatrix} \mathbf{T}_1 & \mathbf{T}_2 \end{bmatrix} \widehat{\mathbf{r}}_\lambda \end{bmatrix}, \\
\|\mathbf{r}_\lambda\|_2 &= \left\| \mathbf{Z}_c^\top \mathbf{B}_{m+\ell}^\top \begin{bmatrix} \mathbf{T}_1 & \mathbf{T}_2 \end{bmatrix} \widehat{\mathbf{r}}_\lambda \right\|_2.
\end{aligned}$$

Before we analyze $\|\mathbf{r}_\lambda\|_2$ and what it means for the difference between the solutions from the regularized compressed problem with recycling and the regularized full problem for the same regularization parameter, consider the case that this residual of the regularized full problem is (relatively) small. In that case, the backward error is (relatively) small, and for a well-chosen regularization parameter the matrix is

well-conditioned. Hence the difference between the regularized solution for the full problem and the regularized solution for the compressed problem is small. Assuming the regularization parameter is larger than the smallest singular values, the condition number of the regularized matrix depends on the largest singular value and the regularization parameter. In general, $\sigma_{max}(\mathbf{B}_{m+l}) \approx \sigma_{max}(\widehat{\mathbf{B}}_{k,\ell})$, and hence choosing λ such that the compressed Tikhonov problem is well-conditioned implies that the full Tikhonov problem would be well-conditioned for the same λ.

Analysis for TSVD-based compression. Next, we consider a more detailed analysis in the case that compression is done using TSVD. For simplicity, we consider compression after the first m iterations of standard GKB, so (2.2) is satisfied, and ℓ subsequent steps of recycling GKB. We can extend this to an analysis for multiple compression and recycling steps, but this is left for future work.

Let $\mathbf{B}_m = \mathbf{\Psi}_{m+1}\mathbf{\Sigma}_m\mathbf{\Phi}_m^\top$ be the SVD of $\mathbf{B}_m$ with

$$\mathbf{\Sigma}_m = \operatorname{diag}(\sigma_1, \ldots, \sigma_m), \tag{3.27}$$

and let $\mathbf{x}^{(1)}$ be a regularized solution. We take $\mathbf{W}_k = \mathbf{V}_m \begin{bmatrix} \mathbf{\Phi}_{k-1} & \boldsymbol{\xi} \end{bmatrix}$, where, following section 3.1, $\mathbf{w}_k = \breve{\mathbf{x}}^{(1)}$ and so $\boldsymbol{\xi} = \mathbf{V}_m^\top \breve{\mathbf{x}}^{(1)}$. This gives

$$\mathbf{Y}_k\mathbf{R}_k = \mathbf{A}\mathbf{W}_k = \mathbf{U}_{m+1}\mathbf{B}_m \begin{bmatrix} \mathbf{\Phi}_{k-1} & \boldsymbol{\xi} \end{bmatrix} = \mathbf{U}_{m+1} \begin{bmatrix} \mathbf{\Psi}_{k-1}\mathbf{\Sigma}_{k-1} & \widetilde{\boldsymbol{\eta}} \end{bmatrix}$$

with $\widetilde{\boldsymbol{\eta}} = \mathbf{B}_m\boldsymbol{\xi}$. Since $\boldsymbol{\xi} \perp \mathbf{\Phi}_{k-1}$, we have $\widetilde{\boldsymbol{\eta}} \perp \mathbf{\Psi}_{k-1}$, and for the QR decomposition $\mathbf{Y}_k\mathbf{R}_k = \mathbf{A}\mathbf{W}_k$,

$$\mathbf{Y}_k = \mathbf{U}_{m+1} \begin{bmatrix} \mathbf{\Psi}_{k-1} & \boldsymbol{\eta} \end{bmatrix}, \qquad \mathbf{R}_k = \operatorname{diag}(\sigma_1, \ldots, \sigma_{k-1}, r_{kk}) \text{ with } r_{kk} = \|\widetilde{\boldsymbol{\eta}}\|_2 \leqslant \sigma_k. \tag{3.28}$$

Next, we bound $\|\mathbf{r}_\lambda\|_2$ by bounding $\|\mathbf{Z}_c^\top\mathbf{B}_{m+\ell}^\top \begin{bmatrix} \mathbf{T}_1 & \mathbf{T}_2 \end{bmatrix}\|_F$, which is an obvious upper bound for $\|\mathbf{Z}_c^\top\mathbf{B}_{m+\ell}^\top \begin{bmatrix} \mathbf{T}_1 & \mathbf{T}_2 \end{bmatrix}\|_2$. We note that this Frobenius norm bound is computable a posteriori (without extra cost). First, consider $\mathbf{Z}_c^\top\mathbf{B}_{m+\ell}^\top\mathbf{T}_1$. Since $\mathcal{R}(\mathbf{A}^\top\mathbf{Y}_k) \subset \mathcal{R}(\mathbf{V}_{m+1})$ and $\mathbf{Z}_c^\top\mathbf{B}_{m+\ell}^\top\mathbf{T}_1 = \mathbf{Z}_c^\top\mathbf{V}_{m+\ell}^\top\mathbf{A}^\top\mathbf{Y}_k$,

$$\|\mathbf{Z}_c^\top\mathbf{B}_{m+\ell}^\top\mathbf{T}_1\|_F^2 = \|\mathbf{A}^\top\mathbf{Y}_k\|_F^2 - \|\mathbf{Z}_1^\top\mathbf{B}_{m+\ell}^\top\mathbf{T}_1\|_F^2 - \|\mathbf{Z}_2^\top\mathbf{B}_{m+\ell}^\top\mathbf{T}_1\|_F^2.$$

We have

$$\begin{aligned} \mathbf{A}^\top\mathbf{Y}_k = \mathbf{A}^\top\mathbf{U}_{m+1} \begin{bmatrix} \mathbf{\Psi}_{k-1} & \boldsymbol{\eta} \end{bmatrix} &= \mathbf{V}_{m+1} \begin{bmatrix} \mathbf{B}_m^\top \\ \mathbf{e}_{m+1}^\top\alpha_{m+1} \end{bmatrix} \begin{bmatrix} \mathbf{\Psi}_{k-1} & \boldsymbol{\eta} \end{bmatrix} \\ \Rightarrow \|\mathbf{A}^\top\mathbf{Y}_k\|_F^2 &= \|\mathbf{\Sigma}_{k-1}\|_F^2 + \|\mathbf{B}_m^\top\boldsymbol{\eta}\|_2^2 + \alpha_{m+1}^2\|\mathbf{e}_{m+1}^\top \begin{bmatrix} \mathbf{\Psi}_{k-1}\boldsymbol{\eta} \end{bmatrix}\|_2^2 \\ &\leqslant \sigma_1^2 + \cdots + \sigma_{k-1}^2 + \sigma_k^2 + \alpha_{m+1}^2. \end{aligned}$$

Note that $\mathbf{B}_m^\top\boldsymbol{\eta}$ and $\mathbf{e}_{m+1}^\top \begin{bmatrix} \mathbf{\Psi}_{k-1} & \boldsymbol{\eta} \end{bmatrix}$ can be computed at negligible cost during the algorithm. Also, $\mathbf{Z}_1^\top\mathbf{B}_{m+\ell}^\top\mathbf{T}_1 = \mathbf{R}_k^\top$, which implies $\|\mathbf{Z}_1^\top\mathbf{B}_{m+\ell}^\top\mathbf{T}_1\|_F^2 = \sigma_1^2 + \cdots + \sigma_{k-1}^2 + r_{kk}^2$ (with $r_{kk} = \|\mathbf{B}_m\boldsymbol{\xi}\|_2$), and $\|\mathbf{Z}_2^\top\mathbf{B}_{m+\ell}^\top\mathbf{T}_1\|_F^2 = \|\mathbf{Y}_k^\top\mathbf{A}\widetilde{\mathbf{V}}_\ell\|_F^2$, which also can be computed at negligible cost during the algorithm. This gives

$$\begin{aligned} \|\mathbf{Z}_c^\top\mathbf{B}_{m+\ell}^\top\mathbf{T}_1\|_F^2 &= \|\mathbf{B}_m^\top\boldsymbol{\eta}\|_2^2 - \|\mathbf{B}_m\boldsymbol{\xi}\|_2^2 + \alpha_{m+1}^2\|\mathbf{e}_{m+1}^\top \begin{bmatrix} \mathbf{\Psi}_{k-1} & \boldsymbol{\eta} \end{bmatrix}\|_2^2 - \|\mathbf{Y}_k^\top\mathbf{A}\widetilde{\mathbf{V}}_\ell\|_F^2 \qquad (3.29) \\ &\leqslant \sigma_k^2 - r_{kk}^2 + \alpha_{m+1}^2 - \|\mathbf{Y}_k^\top\mathbf{A}\widetilde{\mathbf{V}}_\ell\|_F^2. \qquad (3.30) \end{aligned}$$

Note that $\boldsymbol{\eta} = \mathbf{B}_m\boldsymbol{\xi}/\|\mathbf{B}_m\boldsymbol{\xi}\|_2$, and hence $\|\mathbf{B}_m^\top\boldsymbol{\eta}\|_2^2 - \|\mathbf{B}_m\boldsymbol{\xi}\|_2^2$ tends to be small. For $\mathbf{Z}_c^\top\mathbf{B}_{m+\ell}^\top\mathbf{T}_2$, we have

$$\begin{aligned}\mathbf{Z}_c^\top\mathbf{B}_{m+\ell}^\top\mathbf{T}_2 &= \mathbf{Z}_c^\top\mathbf{V}_{m+\ell}^\top\mathbf{A}^\top\mathbf{U}_{m+\ell+1}\mathbf{T}_2 = \mathbf{Z}_c^\top\mathbf{V}_{m+\ell}^\top\mathbf{A}^\top\widetilde{\mathbf{U}}_{\ell+1}\\ &= \mathbf{Z}_c^\top\mathbf{V}_{m+\ell}^\top\left(\widetilde{\mathbf{V}}_\ell\widetilde{\mathbf{B}}_\ell^\top + \widetilde{\alpha}_{\ell+1}\widetilde{\mathbf{v}}_{\ell+1}\mathbf{e}_{\ell+1}^\top\right) = \widetilde{\alpha}_{\ell+1}\mathbf{Z}_c^\top\mathbf{V}_{m+\ell}^\top\widetilde{\mathbf{v}}_{\ell+1}\mathbf{e}_{\ell+1}^\top,\end{aligned}$$

and hence

$$\|\mathbf{Z}_c^\top\mathbf{B}_{m+\ell}^\top\mathbf{T}_2\|_F = |\widetilde{\alpha}_{\ell+1}|\|\mathbf{Z}_c^\top\mathbf{V}_{m+\ell}^\top\widetilde{\mathbf{v}}_{\ell+1}\|_2 \leqslant |\widetilde{\alpha}_{\ell+1}|. \tag{3.31}$$

This derivation proves the following theorem.

THEOREM 3.5. *Let* $\mathbf{A}$, $\mathbf{b}$, $\mathbf{V}_{m+\ell}$, $\mathbf{U}_{m+\ell+1}$, $\mathbf{B}_{m+\ell}$, *and* $\widehat{\mathbf{B}}_{k,\ell}$ *be defined as above (using TSVD for compression). Then* $\mathbf{r}_\lambda$ *in* (3.25) *satisfies*

$$\|\mathbf{r}_\lambda\|_2 \leqslant \|\widehat{\mathbf{r}}_\lambda\|_2\left(\sigma_k^2 - r_{kk}^2 + \alpha_{m+1}^2 - \|\mathbf{Y}_k^\top\mathbf{A}\widetilde{\mathbf{V}}_\ell\|_F^2 + \widetilde{\alpha}_{\ell+1}^2\right)^{1/2}, \tag{3.32}$$

where $\widehat{\mathbf{r}}_\lambda$ *is given in* (3.26).

Finally, we provide a few more useful properties of the matrix $\mathbf{T}^\top\mathbf{B}_{m+\ell}\mathbf{Z}$. First, we would like a bound on its maximum singular value, so we can estimate the condition number of the regularized matrix in (3.25). Note that its smallest eigenvalue is larger than λ^2, where λ is the regularization parameter. Using $\sigma_{max}(\cdot)$ to denote the largest singular value of a matrix,

$$\sigma_{max}\left(\mathbf{T}^\top\mathbf{B}_{m+\ell}\mathbf{Z}\right) \leqslant \sigma_{max}\left(\mathbf{T}^\top\mathbf{B}_{m+\ell}\begin{bmatrix}\mathbf{Z}_1 & \mathbf{Z}_2\end{bmatrix}\right) + \sigma_{max}(\mathbf{T}^\top\mathbf{B}_{m+\ell}\mathbf{Z}_c) \tag{3.33}$$

$$\leqslant \sigma_{max}\left(\mathbf{T}^\top\mathbf{B}_{m+\ell}\begin{bmatrix}\mathbf{Z}_1 & \mathbf{Z}_2\end{bmatrix}\right) + \|\mathbf{T}^\top\mathbf{B}_{m+\ell}\mathbf{Z}_c\|_F. \tag{3.34}$$

Estimating $\mathbf{T}_c^\top\mathbf{B}_{m+\ell}\mathbf{Z}_c$ is difficult, and we use a conjecture that we test numerically below. Consider the blocks of $\mathbf{B}_{m+\ell}$,

$$\mathbf{B}_{m+\ell} = \begin{bmatrix}\mathbf{B}_m & \begin{bmatrix}\mathbf{0}_{m\times\ell}\\ \alpha_{m+1}\mathbf{e}_1^\top\end{bmatrix}\\ \mathbf{0}_{\ell\times m} & \overline{\mathbf{B}}_\ell\end{bmatrix} \text{ with } \overline{\mathbf{B}}_\ell = \begin{bmatrix}\beta_{m+2} & \alpha_{m+2} & & \\ & \ddots & \ddots & \\ & & \beta_{m+\ell} & \alpha_{m+\ell}\\ & & & \beta_{m+\ell+1}\end{bmatrix} \in \mathbb{R}^{\ell\times\ell}. \tag{3.35}$$

We also define

$$\overline{\overline{\mathbf{B}}}_\ell = \begin{bmatrix}\alpha_{m+1}\mathbf{e}_1^\top\\ \\ \overline{\mathbf{B}}_\ell\end{bmatrix}. \tag{3.36}$$

Since $\|\mathbf{B}_m\|_F^2 + \|\overline{\overline{\mathbf{B}}}_\ell\|_F^2 = \|\mathbf{B}_{m+\ell}\|_F^2 = \|\mathbf{T}^\top\mathbf{B}_{m+\ell}\mathbf{Z}\|_F^2 = \|\mathbf{T}^\top\mathbf{B}_{m+\ell}\begin{bmatrix}\mathbf{Z}_1 & \mathbf{Z}_2\end{bmatrix}\|_F^2 + \|\mathbf{T}^\top\mathbf{B}_{m+\ell}\mathbf{Z}_c\|_F^2$, we get

$$\|\mathbf{T}^\top\mathbf{B}_{m+\ell}\mathbf{Z}_c\|_F^2 = \|\mathbf{B}_m\|_F^2 - \|\mathbf{R}_k\|_F^2 - \|\mathbf{Y}_k^\top\mathbf{A}\widetilde{\mathbf{V}}_\ell\|_F^2 + \left(\|\overline{\overline{\mathbf{B}}}_\ell\|_F^2 - \|\widetilde{\mathbf{B}}_\ell\|_F^2\right). \tag{3.37}$$

Since the sequences of vectors $\widetilde{\mathbf{v}}_1, \widetilde{\mathbf{v}}_2, \ldots$ and $\mathbf{v}_{m+1}, \mathbf{v}_{m+2}, \ldots$ both extend the Krylov space beyond $\mathcal{R}(\mathbf{V}_m)$, and both sequences are orthogonal to the approximate dominant singular vectors $\mathbf{W}_k$, we conjecture as follows.

CONJECTURE 1. $\|\overline{\overline{\mathbf{B}}}_\ell\|_F^2 \approx \|\widetilde{\mathbf{B}}_\ell\|_F^2$.

We provide numerical confirmation of this conjecture at the end of this section. Using Conjecture 1, (3.37), (3.27), and (3.28), we obtain the following (approximate) bound:

(3.38)
$$\|\mathbf{T}^\top \mathbf{B}_{m+\ell}\mathbf{Z}_c\|_F^2 = \sigma_k^2 - r_{kk}^2 + \sigma_{k+1}^2 + \dots \sigma_m^2 - \|\mathbf{Y}_k^\top \mathbf{A}\widetilde{\mathbf{V}}_\ell\|_F^2 + \left(\|\overline{\overline{\mathbf{B}}}_\ell\|_F^2 - \|\widetilde{\mathbf{B}}_\ell\|_F^2\right)$$

(3.39)
$$\lessapprox \sigma_k^2 - r_{kk}^2 + \sigma_{k+1}^2 + \cdots + \sigma_m^2.$$

This gives the approximate bound,

$$\sigma_{max}(\mathbf{B}_{m+\ell}) = \sigma_{max}(\mathbf{T}^\top \mathbf{B}_{m+\ell}\mathbf{Z})$$

(3.40)
$$\lessapprox \left(\sigma_{max}{}^2\left(\begin{bmatrix} \mathbf{R}_k & \mathbf{Y}_k^\top \mathbf{A}\widetilde{\mathbf{V}}_\ell \\ \mathbf{0} & \widetilde{\mathbf{B}}_\ell \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\right) + \sigma_k^2 - r_{kk}^2 + \sigma_{k+1}^2 + \ldots + \sigma_m^2\right)^{1/2},$$

which in general is only modestly larger than σ_1. Note that the first term on the right-hand side, which equals $\sigma_{max}{}^2(\widehat{\mathbf{B}}_{k,\ell})$, is easily computable.

Using (3.38), (3.29)–(3.31), and $\sigma_k \geqslant \|\mathbf{B}_m^\top \boldsymbol{\eta}\|_2 \geqslant r_{kk}$, we also obtain an approximate bound for the bottom right block of $\mathbf{T}^\top \mathbf{B}_{m+\ell}\mathbf{Z}$,

(3.41)
$$\begin{aligned}\|\mathbf{T}_c^\top \mathbf{B}_{m+\ell}\mathbf{Z}_c\|_F^2 &= \sigma_k^2 - r_{kk}^2 + \sigma_{k+1}^2 + \cdots + \sigma_m^2 - \|\mathbf{Y}_k^\top \mathbf{A}\widetilde{\mathbf{V}}_\ell\|_F^2 \\ &\quad + \left(\|\overline{\overline{\mathbf{B}}}_\ell\|_F^2 - \|\widetilde{\mathbf{B}}_\ell\|_F^2\right) - \|\mathbf{B}_m^\top \boldsymbol{\eta}\|_2^2 \\ &\quad + r_{kk}^2 - \alpha_{m+1}^2 \|\mathbf{e}_{m+1}^\top \begin{bmatrix} \boldsymbol{\Psi}_{k-1} & \boldsymbol{\eta} \end{bmatrix}\|_2^2 \\ &\quad + \|\mathbf{Y}_k^\top \mathbf{A}\widetilde{\mathbf{V}}_\ell\|_F^2 - \widetilde{\alpha}_{\ell+1}^2 \|\mathbf{Z}_c^\top \mathbf{V}_{m+\ell}^\top \widetilde{\mathbf{v}}_{\ell+1}\|_2^2 \\ &\lessapprox \sigma_k^2 - r_{kk}^2 + \sigma_{k+1}^2 + \cdots + \sigma_m^2.\end{aligned}$$

Finally, we note that, in general, as the dominant singular vectors are captured relatively quickly and progressively better in the Krylov spaces, the coefficients $|\alpha_j|$ and $|\beta_{j+1}|$ have a decreasing trend. Hence, $|\alpha_{m+1}|$ and $|\widetilde{\alpha}_{\ell+1}|$ tend to be small compared with σ_k. In Figure 3.1, we numerically verify this trend for α_j for the example in subsection 4.1.

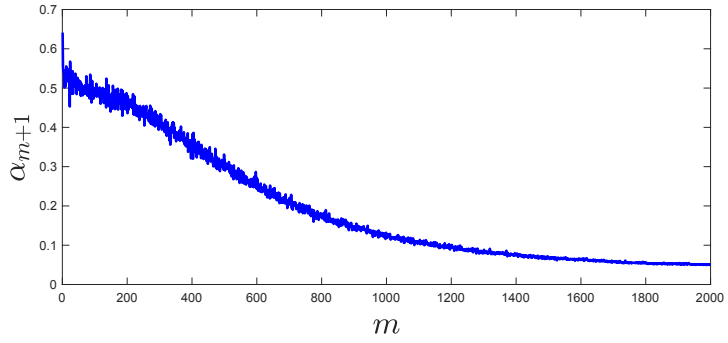


FIG. 3.1. *Values of α_{m+1} for the example in subsection* 4.1.

TABLE 3.1

Differences as defined in (3.42) *for various image deblurring examples with the blur function being a Gaussian with a missing piece. The noise level is* 0.2%.

Image	d	$\left\|\overline{\overline{\mathbf{B}}}_\ell\right\|_F^2$	$\left\|\widetilde{\mathbf{B}}_\ell\right\|_F^2$	σ_{30}
grain	0.0593	9.3606	9.3013	0.6280
plane	0.1345	9.4065	9.2720	0.6332
peppers	0.1548	9.4280	9.2732	0.6286
cameraman	0.1261	9.4353	9.3092	0.6283

TABLE 3.2

Differences as defined in (3.42) *for the grain example with different blur functions. The noise level is* 0.2%.

Blurring information	d	$\left\|\overline{\overline{\mathbf{B}}}_\ell\right\|_F^2$	$\left\|\widetilde{\mathbf{B}}_\ell\right\|_F^2$	σ_{30}
Gaussian with a missing piece	0.0593	9.3606	9.3013	0.6280
Nonsymmetric Gaussian blur with parameters $[2, 3, 0]$	0.1094	9.4610	9.3515	0.6153
Nonsymmetric out of focus blur with radius 9	0.1134	9.4804	9.3670	0.6160
Box car blur (11×11 blur)	0.1440	9.4977	9.3538	0.6155

TABLE 3.3

Differences as defined in (3.42) *for different noise levels for the grain example with the blur function being a Gaussian with a missing piece.*

Noise level	d	$\left\|\overline{\overline{\mathbf{B}}}_\ell\right\|_F^2$	$\left\|\widetilde{\mathbf{B}}_\ell\right\|_F^2$	σ_{30}
0.005	0.0586	9.3599	9.3013	0.6280
0.01	0.0580	9.3593	9.3013	0.6279
0.05	0.0592	9.3605	9.3013	0.6275
0.1	0.0605	9.3618	9.3013	0.6271

To get a better understanding of the final term in the bound, we provide the difference

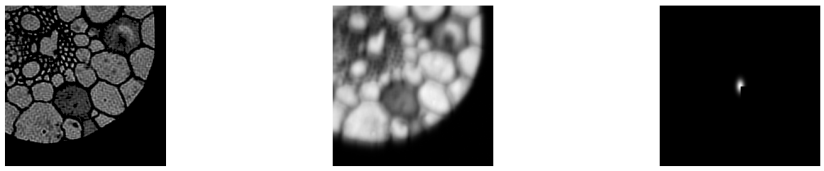
$$d = \left| \left\|\overline{\overline{\mathbf{B}}}_\ell\right\|_F^2 - \left\|\widetilde{\mathbf{B}}_\ell\right\|_F^2 \right| \tag{3.42}$$

for some commonly used numerical examples from RestoreTools [34]. For all examples, we set $m = 50$, $k = q = 30$, and $\ell = 20$. All images are 256×256. In Table 3.1 we consider different images, in Table 3.2 we consider different types of blur, and in Table 3.3 we consider different noise levels. We observe that d is much smaller than σ_k in regular scenarios, so σ_k likely dominates the upper bound of $\|\mathbf{T}_c^\top \mathbf{B}_{m+\ell} \mathbf{Z}_c\|_F^2$ in (3.41).

4. Numerical results. In this section, we compare the performance of the proposed hybrid projection methods with recycling to that of the conventional hybrid methods using examples from image processing. We consider various scenarios where the recycling hybrid projection methods can alleviate storage requirements and improve reconstructions when solving inverse problems. In subsection 4.1 we consider a linear image deblurring problem where standard hybrid methods may be limited by the storage of many vectors in the solution space. We investigate the performance of various compression methods and parameter selection methods. Then in subsection 4.2, we consider two tomographic reconstruction examples, one for a streaming data problem and another for a problem with modified projection angles using real data. MATLAB codes will be provided at https://github.com/juliannechung/HyBRrecycle.

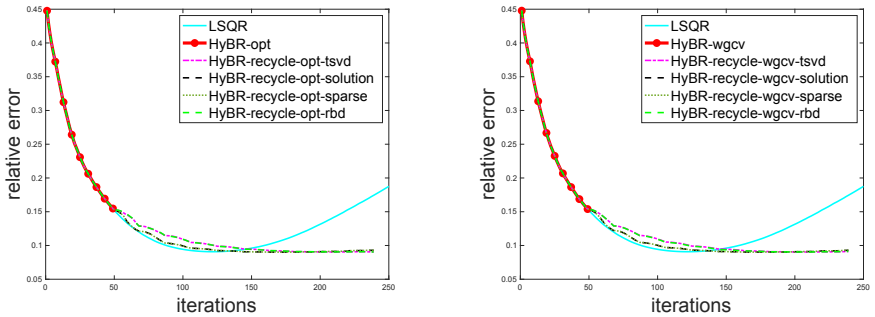
4.1. Image deblurring example. This example is an image deblurring problem from RestoreTools [34], where the goal is to reconstruct a true image of a grain, which has 256×256 pixels, from an observed blurred image that contains Gaussian white noise at a noise level of 0.2%, i.e., $\frac{\|\boldsymbol{\epsilon}\|_2}{\|\mathbf{A}\mathbf{x}_{\text{true}}\|_2} = 0.002$. The true image, blurred and noisy image, and point spread function (PSF) for the grain example are shown in Figure 4.1. An image deblurring problem with a smaller noise level usually requires more iterations to converge, which for standard hybrid methods means that we need to store more solution vectors. For this example, assume that we can store at most 50 solution basis vectors, each of size 65536×1. We will show that the proposed hybrid projection method with recycling and compression, henceforth denoted HyBR-recycle, can handle this scenario. We will investigate various compression techniques from subsection 3.3, where the stopping criteria are (1) the maximum number of basis vectors saved after compression is $q = 30$ and (2) the compression tolerance is $\varepsilon_{tol} = 10^{-6}$.

In Figure 4.2, we provide the relative reconstruction error norms per iteration of HyBR-recycle with various compression strategies, where for comparison we include the relative error norms for LSQR (with no additional regularization) and for a standard hybrid method denoted by HyBR. In the left plot, we use the optimal regularization parameter at each iteration, which is not available in practice, and in the right plot, we use the weighted GCV (WGCV) [8] method for regularization



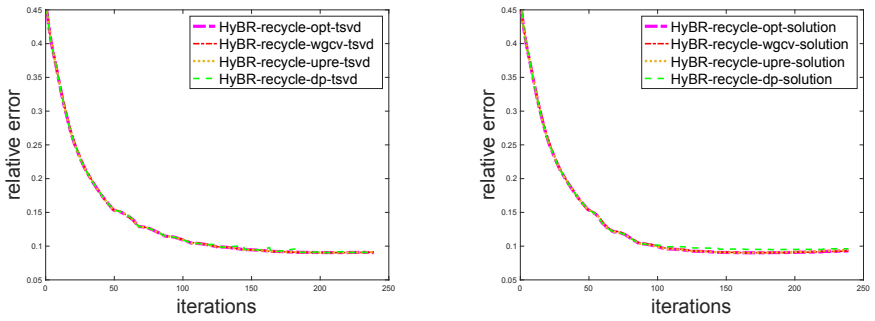
(a) True image (b) Noisy blurred image (c) PSF

FIG. 4.1. *Image deblurring example.*



(a) Optimal (b) WGCV

FIG. 4.2. *Relative reconstruction error norms for hybrid projection methods with recycling using different compression strategies. The left plot corresponds to selecting the optimal regularization parameter at each iteration, and the right plot corresponds to selecting the regularization parameter using WGCV. For all methods, we assume that storage of solution basis vectors is limited to* 50.



(a) TSVD (b) Solution-oriented method

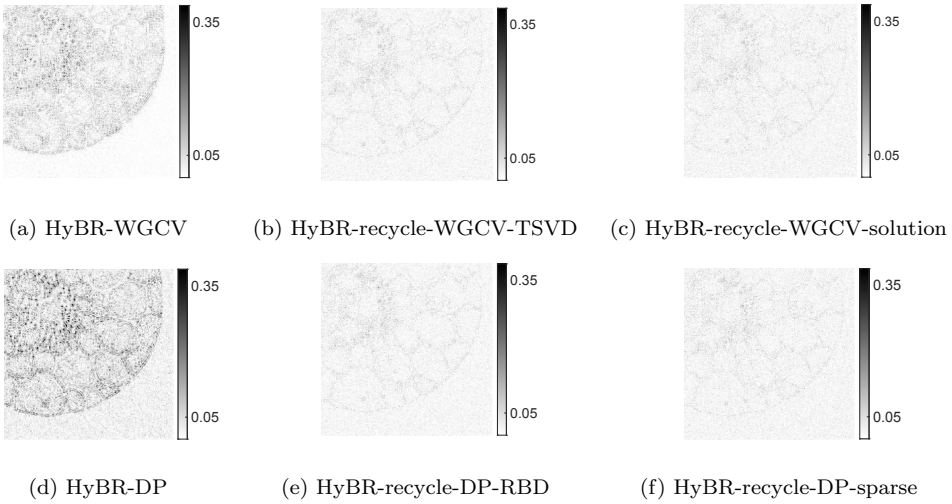
FIG. 4.3. *Relative reconstruction errors for hybrid projection methods with recycling and compression for the grain deblurring example with different regularization parameter choice methods.*

parameter selection. We observe that LSQR exhibits semiconvergence, and HyBR-opt is not able to achieve high accuracy due to the fact that the storage limit was capped at 50 solution vectors. These results demonstrate the competitiveness of hybrid projection methods with recycling in limited storage situations. Moreover, we notice that for both regularization parameter choice methods, solution-oriented compression and sparsity-enforcing compression resulted in slightly smaller relative error norms than TSVD and RBD.

Next, in Figure 4.3 we compare various methods for selecting regularization parameters in hybrid projection methods with recycling. We consider two compression techniques (TSVD and solution-oriented), and we provide relative reconstruction error norms for parameter choice methods: the WGCV, the unbiased predictive risk estimator (UPRE) [48, 46], and the discrepancy principle (DP) [33]. These parameter choice methods are among many that have been investigated in the context of hybrid iterative methods; e.g., see [18] and references therein. For these experiments, we use the true noise level for both UPRE and DP, but estimates of the noise level can be obtained in practice [11, 35, 26]. For comparison, we provide results for the optimal regularization parameter. We observe that all of the considered regularization parameter selection methods result in relative reconstruction error norms that are close to those for the optimal regularization parameter.

The absolute error images (in inverted colormap) corresponding to reconstructions of the grain image are provided in Figure 4.4. The top row corresponds to reconstructions obtained using WGCV, and the bottom row corresponds to reconstructions obtained using DP. For HyBR-recycle-WGCV, we provide results for TSVD and solution-oriented compression. For HyBR-recycle-DP, we provide results for RBD and sparsity enforcing compression. The HyBR-recycle results correspond to 239 iterations. For comparison, we provide absolute error images for the standard hybrid methods after 50 iterations. Notice that due to the forced storage limit, absolute errors for HyBR-WGCV and HyBR-DP reconstructions are large (corresponding to darker regions in Figures 4.4(a) and (d), respectively). These observations are consistent with the relative error norms provided in Figure 4.2.

Finally, we use this example to numerically demonstrate the bound derived in Theorem 3.5. In Figure 4.5, we provide the norm of the residual for the transformed problem and the derived upper bound from (3.32) for one cycle of HyBR-recycle after compression with TSVD. At each iteration, the same regularization parameter was



(a) HyBR-WGCV (b) HyBR-recycle-WGCV-TSVD (c) HyBR-recycle-WGCV-solution

(d) HyBR-DP (e) HyBR-recycle-DP-RBD (f) HyBR-recycle-DP-sparse

FIG. 4.4. *Absolute error images (in inverted colormap) for the grain image for WGCV and DP.*

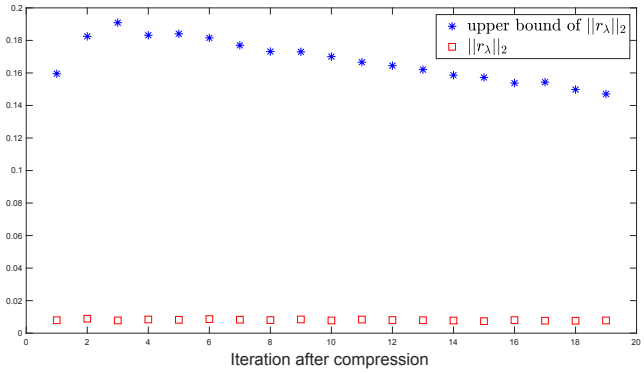


FIG. 4.5. *Illustration of bound on the residual norm derived in Theorem* 3.5.

used to compute the residual from HyBR-recycle (i.e., $\|\hat{\mathbf{r}}_\lambda\|_2$) and the residual from the regularized full GKB for the HyBR-recycle solution (i.e., $\|\mathbf{r}_\lambda\|_2$). Although the bound is an overestimate, the result shows that we do not expect the solution of HyBR-recycle after compression to be far from the solution to the regularized Tikhonov problem when using the full GKB.

4.2. Tomography reconstruction examples. Next, we investigate various scenarios in tomographic reconstruction where multiple reconstruction problems must be solved, and the hybrid projection methods with recycling can be used to incorporate information (e.g., basis vectors) from previous reconstructions to solve the current reconstruction problem. We consider two scenarios.

1. In the case of dynamic or streaming data inverse problems, reconstructions must be updated as data are being collected. This may arise in applications such as microCT, where immediate reconstructions are used as feedback to inform the data acquisition process [37].

2. Oftentimes, we must solve several reconstruction problems where the projection angles are slightly modified. This might arise in an optimal experimental design framework where the goal is to determine the optimal angles for image formation [42] or in a sampling framework [43].

Before describing the details of the experiments, we describe four general approaches. Assume that we have r reconstruction problems,

$$\min_{\mathbf{x}} \|\mathbf{A}_1\mathbf{x} - \mathbf{b}_1\|_2^2 + \lambda_1^2 \|\mathbf{x}\|_2^2, \tag{4.1}$$

$$\vdots$$

$$\min_{\mathbf{x}} \|\mathbf{A}_i\mathbf{x} - \mathbf{b}_i\|_2^2 + \lambda_i^2 \|\mathbf{x}\|_2^2, \tag{4.2}$$

$$\vdots$$

$$\min_{\mathbf{x}} \|\mathbf{A}_r\mathbf{x} - \mathbf{b}_r\|_2^2 + \lambda_r^2 \|\mathbf{x}\|_2^2. \tag{4.3}$$

Depending on the problem setup and noise level, the regularization parameter for each problem $\lambda_1, \ldots, \lambda_r$ may be different. Thus, in all of our approaches, we select these regularization parameters automatically in a hybrid framework.

1. Using $\mathbf{A}_1$ and $\mathbf{b}_1$, we run m iterations of the standard GKB on (4.1), compress the computed solution vectors into $k_1 - 1$ orthonormal vectors $\mathbf{W}_{k_1-1} \in \mathbb{R}^{N\times(k_1-1)}$, and save matrix $\mathbf{W}_{k_1} = \begin{bmatrix}\mathbf{W}_{k_1-1} & \breve{\mathbf{x}}^{(1)}\end{bmatrix}$, where

$$\breve{\mathbf{x}}^{(1)} = (\mathbf{x}^{(1)} - \mathbf{W}_{k_1-1}\mathbf{W}_{k_1-1}^\top\mathbf{x}^{(1)}) / \left\|\mathbf{x}^{(1)} - \mathbf{W}_{k_1-1}\mathbf{W}_{k_1-1}^\top\mathbf{x}^{(1)}\right\|_2$$

and $\mathbf{x}^{(1)}$ is the corresponding solution of (4.1). Then for a subsequent problem with $\mathbf{A}_i$ and $\mathbf{b}_i$ $(1 < i < r)$, we use $\mathbf{W}_{k_i-1}^{(new)}$ obtained from the previous problem and run HyBR-recycle on (4.2) saving matrix $\mathbf{W}_{k_i}^{(new)}$. Finally we solve (4.3) using HyBR-recycle starting with $\mathbf{W}_{k_r-1}^{(new)}$.
2. We run a standard HyBR method with automatic regularization parameter selection on any of the r reconstruction problems (e.g., the last one).
3. For comparison, we provide the results for HyBR with automatic regularization parameter selection on the entire problem,

$$\min_{\mathbf{x}} \left\| \begin{bmatrix}\mathbf{A}_1\\ \vdots \\ \mathbf{A}_r\end{bmatrix}\mathbf{x} - \begin{bmatrix}\mathbf{b}_1\\ \vdots \\ \mathbf{b}_r\end{bmatrix} \right\|_2^2 + \lambda^2 \|\mathbf{x}\|_2^2. \tag{4.4}$$

We remark that in streaming scenarios, this can be considered as the ideal case and should produce the solution with overall smallest relative error. However, we assume that this cannot be computed in practice and use it merely as a comparison.
4. We take an average of solutions computed from (4.1) to (4.3) independently; this is common in tomography.

4.2.1. Streaming data. For the first experiment, we use the parallel tomography example from IRTools [14, 24], where the true image is a 1024×1024 Shepp–Logan phantom so $\mathbf{x}_{\text{true}} \in \mathbb{R}^{1024^2}$. The true image can be found in Figure 4.6(a). We test two cases for this example, where the first case has two reconstruction problems ($r = 2$)

and the second case has four reconstruction problems ($r = 4$). The DP is used for regularization parameter selection in all the results presented in this subsection.

Case 1. We assume that data is being streamed such that the first reconstruction problem corresponds to 90 equally spaced projection angles between 0° and 89°, and the second problem corresponds to 90 equally spaced projection angles between 90° and 179°. In terms of dimensions, $\mathbf{A}_1, \mathbf{A}_2 \in \mathbb{R}^{90\cdot 1448 \times 1024^2}$ and $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^{90\cdot 1448}$. The noise level for each observed image is 0.02, which means that $\frac{\|\boldsymbol{\epsilon}_i\|_2}{\|\mathbf{A}_i \mathbf{x}_{\text{true}}\|_2} = 0.02$ for $i = 1, 2$. The observations are provided in Figure 4.6. The limit of the storage of solution basis vectors (each $1{,}048{,}576 \times 1$) is assumed to be 50. The stopping criteria for compression are defined by the maximum number of the basis vectors we want to keep after each compression, which we assume here to be 10, and a tolerance, which we assume to be $\varepsilon_{tol} = 10^{-6}$.

A plot of the relative reconstruction error norms per iteration for the four approaches described above is provided in Figure 4.7. The average of images is not an iterative process, but the relative error norm corresponding to the average solution is denoted with a dotted line, for comparison. We see that HyBR-recycle produces reconstructions with relative reconstruction error norms that are smaller than both HyBR with the second dataset and the average of images, demonstrating that the inclusion

(a) True image (b) $\mathbf{b}_1$: $0° - 89°$ (c) $\mathbf{b}_2$: $90° - 179°$

FIG. 4.6. *Streaming tomography example, Case* 1. *The true image is provided in* (a)*, along with two observed sinograms* $\mathbf{b}_1, \mathbf{b}_2$ *corresponding to projections taken at* 1° *intervals from* 0° *to* 89° *and* 90° *to* 179°*, respectively.*

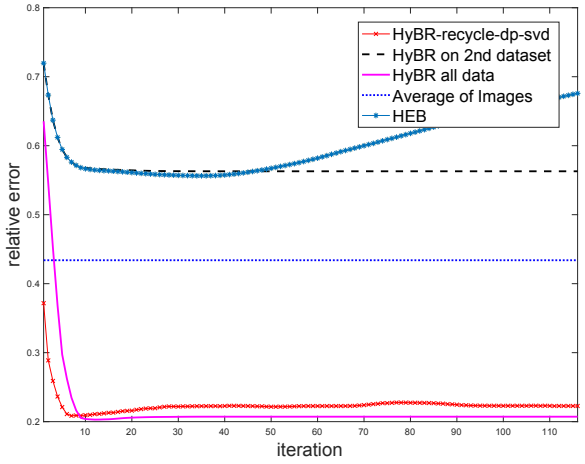


FIG. 4.7. *Streaming tomography example, Case* 1*: Relative reconstruction error norms.*

of the 11 basis images in the HyBR-recycle framework was beneficial. Notice that HyBR with all of the data produces reconstructions with smallest reconstruction error norm, as expected. We also compare to the HEB approach without regularization. The main point of this comparison is to demonstrate that HEB is not as accurate as HyBR-recycle since the generated basis is not improved. Image reconstructions with corresponding absolute error images in inverted colormap are provided in Figure 4.8. In terms of CPU time, HyBR-recycle-dp-svd took 79.85 sec, HyBR with the second dataset took 98.21 sec and HyBR with the entire dataset took 181.64 sec.

In general, we found that our methods are stable with respect to the choice of q as long as we don't choose extreme values (i.e., q should not be too small or too close to the storage limit). For this example, we provide in Figure 4.9 the relative reconstruction error norms per iteration for various choices of q, where we observe that good results can be obtained even when q is quite small. However, this is not always the case (for the example in section 4.2.2, selecting q too small resulted in poor reconstructions because too many important vectors were removed).

Case 2. Next we assume that data is being streamed such that we have four reconstruction problems corresponding to 45 equally spaced projection angles in 0°–44°, 45°–90°, 91°–135°, and 136°–179° respectively. In terms of dimensions, $\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_4 \in \mathbb{R}^{45\cdot 1448 \times 1024^2}$ and $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4 \in \mathbb{R}^{45\cdot 1448}$. The noise level for each observed image is 0.02, which means that $\frac{\|\boldsymbol{\epsilon}_i\|_2}{\|\mathbf{A}_i \mathbf{x}_{\text{true}}\|_2} = 0.02$ for $i = 1, 2, 3, 4$. The observed sinograms are provided in Figure 4.10.

This time, we limit the storage of the solution basis vectors (each still $1{,}048{,}576 \times 1$) to be 15. For the stopping criteria for compression, we set the maximum number

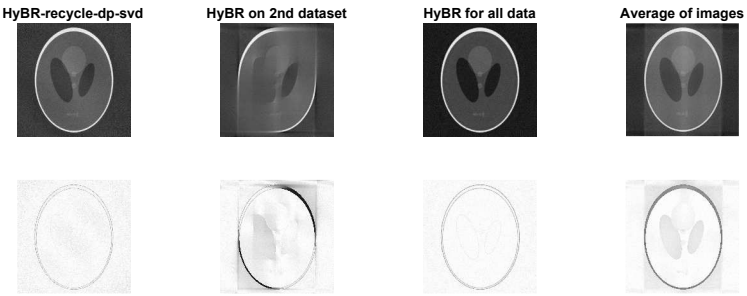


FIG. 4.8. *Streaming tomography example, Case 1: Reconstructions and error images (in inverted colormap).*

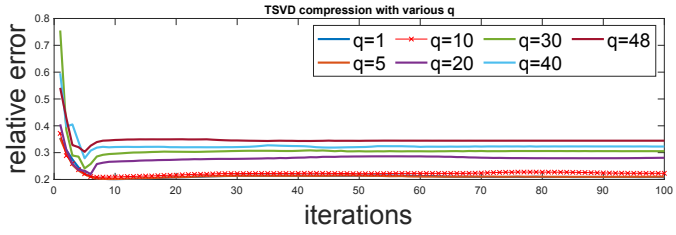


FIG. 4.9. *Streaming tomography example, Case 1: Investigating different values of q.*

(a) $\mathbf{b}_1$: $0^\circ - 44^\circ$ (b) $\mathbf{b}_2$: $45^\circ - 89^\circ$ (c) $\mathbf{b}_3$: $90^\circ - 134^\circ$ (d) $\mathbf{b}_4$: $135^\circ - 179^\circ$

FIG. 4.10. *Streaming tomography example, Case* 2. *The true image is provided in Figure* 4.6(a), *and the four observed sinogram images are provided here.*

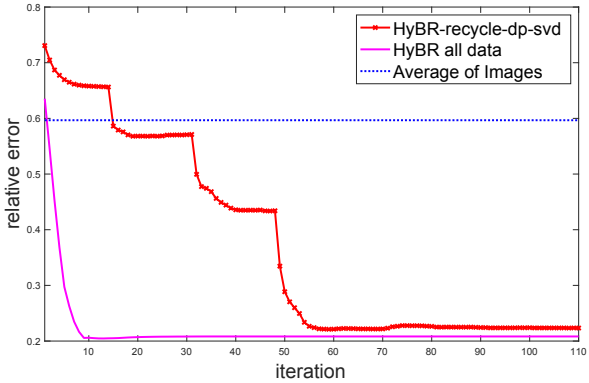


FIG. 4.11. *Streaming tomography example, Case* 2*: Relative reconstruction error norms.*

of saved vectors after each compression to be 5 and the compression tolerance to be $\varepsilon_{tol} = 10^{-6}$. For the first reconstruction problem, we run standard HyBR for $m = 15$ iterations.

A plot of the relative reconstruction error norms per iteration is provided in Figure 4.11. For the HyBR-recycle-dp-svd method, we provide relative reconstruction error norms from the second to the last reconstruction problem (that is, the 1st to 14th iterations correspond to the first problem with standard HyBR, the 15th to 31st iterations correspond to the second problem, the 32nd to 48th iterations correspond to the third problem, and the 49th to 110th iterations correspond to the fourth problem). Image reconstructions with absolute error images are provided in Figure 4.12, and reconstructions from HyBR-recycle-dp-svd at various iterations are provided in Figure 4.13. In terms of overall CPU time, HyBR-recycle took 43.10 sec, HyBR with one dataset took 96.72 sec (i.e., the time to compute an average of solutions), and HyBR with the entire dataset took 226.70 sec.

We observe that HyBR-recycle produces reconstructions with relative reconstruction error norms that are much smaller than the average of images. HyBR with all of the data is the most accurate approach, as expected, but it is more costly. Furthermore, for large-scale *sequential* problems where it is not desirable to wait until all data have been collected to perform reconstruction, HyBR-recycle provides an efficient approach to compute regularized solutions with comparable accuracy.

4.2.2. Tomographic reconstruction of a walnut. To test the practicality of the hybrid projection methods with recycling, we present reconstruction results from actual tomographic x-ray projection data from a walnut [20]. This example consists of four reconstruction problems, where the projection angles for each problem

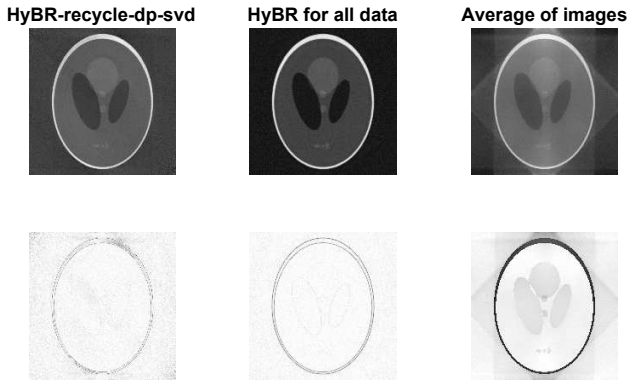


FIG. 4.12. *Streaming tomography example, Case* 2*: Reconstructions and error images (in inverted colormap).*

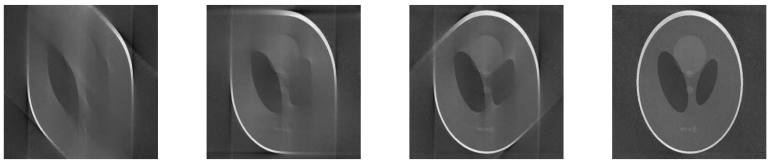


FIG. 4.13. *Streaming tomography example, Case* 2*: Reconstructions from the HyBR-recycle-dp-svd method at iterations* 14*,* 31*,* 48*, and* 110*, respectively.*

are slightly modified. The need to solve multiple problems with modified projection angles arises in various scenarios including optimal experimental design frameworks [42] and optimization to correct for uncertain angles [41]. We investigate the use of hybrid projection methods with recycling to reuse the solution and solution basis vectors acquired from one reconstruction to efficiently solve another reconstruction problem with modified angles. Since the data for this example are taken from real experiments, the true solution is not available.

We are given a set of 120 fan-beam projections taken at an angular step of three degrees. The number of rays per projection is 328. The first system corresponds to 30 equally spaced projection angles between 3° and 351°, which gives $\mathbf{A}_1 \in \mathbb{R}^{30\cdot 328\times 328^2}$ and $\mathbf{b}_1 \in \mathbb{R}^{30\cdot 328}$. The second system is generated using 30 equally spaced projection angles between 6° and 354°, which gives $\mathbf{A}_2 \in \mathbb{R}^{30\cdot 328\times 328^2}$ and $\mathbf{b}_2 \in \mathbb{R}^{30\cdot 328}$. The third system is generated using 30 equally spaced projection angles between 9° and 357°, which gives $\mathbf{A}_3 \in \mathbb{R}^{30\cdot 328\times 328^2}$ and $\mathbf{b}_3 \in \mathbb{R}^{30\cdot 328}$. The fourth system is generated using 30 equally spaced projection angles between 12° and 360°, which gives $\mathbf{A}_4 \in \mathbb{R}^{30\cdot 328\times 328^2}$ and $\mathbf{b}_4 \in \mathbb{R}^{30\cdot 328}$.

For HyBR-recycle, we initialize with $m = 100$ iterations of the standard GKB with $\mathbf{A}_1$ and $\mathbf{b}_1$ to get $\mathbf{x}^{(1)}$, and we compress the basis vectors $\mathbf{V}_{100}$ to get $\mathbf{W}_{90}$. Then $\mathbf{W}_{91} = \begin{bmatrix}\mathbf{W}_{90} & \breve{\mathbf{x}}^{(1)}\end{bmatrix}$, where $\breve{\mathbf{x}}^{(1)} = (\mathbf{x}^{(1)} - \mathbf{W}_{90}\mathbf{W}_{90}^\top\mathbf{x}^{(1)})/\left\|\mathbf{x}^{(1)} - \mathbf{W}_{90}\mathbf{W}_{90}^\top\mathbf{x}^{(1)}\right\|_2$. Given the initial set of basis vectors in $\mathbf{W}_{91}$, we use HyBR-recycle with the four different compression techniques described in subsection 3.3. For all of the considered methods for this problem, we allow storage for a maximum of 100 solution vectors.

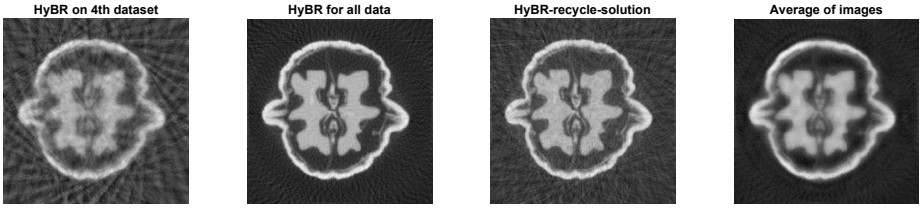


FIG. 4.14. *Tomography walnut example: Image reconstructions.*

TABLE 4.1

Tomography walnut example: Relative differences computed as $\|\mathbf{x} - \mathbf{x}_{all}\|_2 / \|\mathbf{x}_{all}\|_2$ *where* $\mathbf{x}_{all}$ *represents the numerical solution computed by HyBR for all data. For solution-oriented compression, we provide results for various choices of* q*. All results use GCV for selecting the regularization parameter.*

HyBR on 4th dataset	0.3102	Average of images	0.2679
HyBR-recycle-tsvd	0.1814	HyBR-recycle-solution (q=90)	0.1841
HyBR-recycle-rbd	0.1732	HyBR-recycle-solution (q=20)	0.2271
HyBR-recycle-sparse	0.1832	HyBR-recycle-solution (q=1)	0.9008

For HyBR-recycle, the maximum number of vectors to save at compression is 90 with the compression tolerance being $\varepsilon_{tol} = 10^{-6}$. We allow two cycles of HyBR-recycle for each dataset.

We compare these results to HyBR on the fourth dataset, HyBR for all data, and the average of images obtained from four separate HyBR reconstructions, one for each dataset. The reconstructions from the standard approaches are obtained after 100 iterations. For all of these experiments, GCV is used to select the regularization parameter. From the image reconstructions provided in Figure 4.14, we observe that the lack of data for HyBR on the fourth dataset results in artifacts, whereas the average of images is quite blurry. We only provide the HyBR-recycle reconstruction using solution-oriented compression, but we remark that similar results were observed for all compression approaches. Let $\mathbf{x}_{all}$ denote the solution from HyBR for all data. For various reconstructions $\mathbf{x}$, we provide in Table 4.1 relative differences computed as $\|\mathbf{x} - \mathbf{x}_{all}\|_2 / \|\mathbf{x}_{all}\|_2$. We also provide relative differences for different choices of q to emphasize that our methods are not very sensitive to the choice of q but that it should not be selected too small. In summary, the HyBR-recycle reconstructions contain some noise, but are overall sharper than taking an average of image reconstructions (this is clear especially around the edges), and do not suffer from artifacts from limited data.

5. Conclusions. In this paper, we have described Golub–Kahan-based hybrid projection methods with recycling that use compression and recycling to overcome potential memory limitations. We described a variety of problems that can be solved using these methods. For example, we can solve very large problems where the number of basis vectors becomes too large for memory storage. These methods can be used to efficiently solve a sequence of regularized problems (e.g., changing regularization terms or nonlinear solvers) and problems with streaming data. We emphasize that the general approach can also be used in an iterative fashion to improve on existing solutions. The main computational benefits include improved regularized solutions, reduced memory requirements, and automatic selection of the regularization

parameter. Theoretical results show connections between projected problems and relationships between regularized solutions. Numerical results demonstrate that our approach can efficiently and accurately solve various inverse problems from image processing.

REFERENCES

[1] K. Ahuja, P. Benner, E. de Sturler, and L. Feng, *Recycling BiCGSTAB with an application to parametric model order reduction*, SIAM J. Sci. Comput., 37 (2015), pp. S429–S446, https://doi.org/10.1137/140972433.

[2] J. Baglama and L. Reichel, *Augmented implicitly restarted Lanczos bidiagonalization methods*, SIAM J. Sci. Comput., 27 (2005), pp. 19–42.

[3] J. Baglama, L. Reichel, and D. Richmond, *An augmented LSQR method*, Numer. Algorithms, 64 (2013), pp. 263–293.

[4] F. S. V. Bazán and L. S. Borges, *GKB-FP: An algorithm for large-scale discrete ill-posed problems*, BIT, 50 (2010), pp. 481–507.

[5] A. Björck, *A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations*, BIT, 28 (1988), pp. 659–670.

[6] D. Calvetti, L. Reichel, and A. Shuibi, *Enriched Krylov subspace methods for ill-posed problems*, Linear Algebra Appl., 362 (2003), pp. 257–273.

[7] Y. Chen, *Reduced basis decomposition: A certified and fast lossy data compression algorithm*, Comput. & Math. Appl., 70 (2015), pp. 2566–2574.

[8] J. Chung, J. G. Nagy, and D. P. O'Leary, *A weighted GCV method for Lanczos hybrid regularization*, Electron. Trans. Numer. Anal., 28 (2008), pp. 149–167.

[9] J. Chung and K. Palmer, *A hybrid LSMR algorithm for large-scale Tikhonov regularization*, SIAM J. Sci. Comput., 37 (2015), pp. S562–S580.

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

[11] D. L. Donoho, *De-noising by soft-thresholding*, IEEE Trans. Inform. Theory, 41 (1995), pp. 613–627.

[12] C. L. Epstein, *Introduction to the Mathematics of Medical Imaging*, 2nd ed., SIAM, Philadelphia, 2008.

[13] L. Feng, P. Benner, and J. G. Korvink, *Parametric model order reduction accelerated by subspace recycling*, in Proceedings of the 48h IEEE Conference on Decision and Control and 28th Chinese Control Conference, IEEE, 2009, pp. 4328–4333.

[14] S. Gazzola, P. C. Hansen, and J. G. Nagy, *IR tools: A MATLAB package of iterative regularization methods and large-scale test problems*, Numer. Algorithms, 81 (2019), pp. 773–811.

[15] S. Gazzola and J. G. Nagy, *Generalized Arnoldi–Tikhonov method for sparse reconstruction*, SIAM J. Sci. Comput., 36 (2014), pp. B225–B247.

[16] S. Gazzola, P. Novati, and M. R. Russo, *Embedded techniques for choosing the parameter in Tikhonov regularization*, Numer. Linear Algebra Appl., 21 (2014), pp. 796–812.

[17] S. Gazzola, P. Novati, and M. R. Russo, *On Krylov projection methods and Tikhonov regularization*, Electron. Trans. Numer. Anal., 44 (2015), pp. 83–123.

[18] S. Gazzola and M. Sabaté Landman, *Krylov methods for inverse problems: Surveying classical, and introducing new, algorithmic approaches*, GAMM-Mitt., 43 (2020), e202000017.

[19] G. Golub and W. Kahan, *Calculating the singular values and pseudoinverse of a matrix*, SIAM J. Numer. Anal., 2 (1965), pp. 205–224.

[20] K. Hämäläinen, L. Harhanen, A. Kallonen, A. Kujanpää, E. Niemi, and S. Siltanen, *Tomographic X-ray Data of a Walnut*, Math. Model. Comput. 4, preprint arXiv:1502.04064, 2015.

[21] P. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems*, Fundam. Algorithms 7, SIAM, Philadelphia, 1997.

[22] P. C. Hansen, *Discrete Inverse Problems: Insight and Algorithms*, SIAM, Philadelphia, 2010.

[23] P. C. Hansen, Y. Dong, and K. Abe, *Hybrid enriched bidiagonalization for discrete ill-posed problems*, Numer. Linear Algebra Appl., 26 (2019), e2230.

[24] P. C. Hansen and J. S. Jorgensen, *AIR Tools* II*: Algebraic iterative reconstruction methods, improved implementation*, Numer. Algorithms, 79 (2018), pp. 107–137.

[25] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring Images: Matrices, Spectra and Filtering*, Fundam. Algorithms 3, SIAM, Philadelphia, 2006.

[26] I. Hnětynková, M. Plešinger, and Z. Strakoš, *The regularizing effect of the Golub-Kahan iterative bidiagonalization and revealing the noise level in the data*, BIT, 49 (2009), pp. 669–696.
[27] M. E. Hochstenbach and L. Reichel, *Subspace-restricted singular value decompositions for linear discrete ill-posed problems*, J. Comput. Appl. Math., 235 (2010), pp. 1053–1064.
[28] C. Jin, X.-C. Cai, and C. Li, *Parallel domain decomposition methods for stochastic elliptic equations*, SIAM J. Sci. Comput., 29 (2007), pp. 2096–2114.
[29] S. Keuchel, J. Biermann, and O. von Estorff, *A combination of the fast multipole boundary element method and Krylov subspace recycling solvers*, Eng. Anal. Bound. Elem., 65 (2016), pp. 136–146.
[30] M. Kilmer and E. de Sturler, *Recycling subspace information for diffuse optical tomography*, SIAM J. Sci. Comput., 27 (2006), pp. 2140–2166, https://doi.org/10.1137/040610271.
[31] 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.
[32] L. A. M. Mello, E. de Sturler, G. H. Paulino, and E. C. N. Silva, *Recycling Krylov subspaces for efficient large-scale electrical impedance tomography*, Comput. Methods Appl. Mech. Engrg., 199 (2010), pp. 3101–3110, https://doi.org/10.1016/j.cma.2010.06.001.
[33] V. A. Morozov, *On the solution of functional equations by the method of regularization*, Dokl. Akad. Nauk, 167 (1966), pp. 510–512.
[34] J. G. Nagy, K. Palmer, and L. Perrone, *Iterative methods for image deblurring: A MATLAB object-oriented approach*, Numer. Algorithms, 36 (2004), pp. 73–93.
[35] R. Neelamani, H. Choi, and R. Baraniuk, *ForWaRD: Fourier-wavelet regularized deconvolution for ill-conditioned systems*, IEEE Trans. Signal Process., 52 (2004), pp. 418–433.
[36] D. O'Leary and J. Simmons, *A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems*, SIAM J. Sci. Stat. Comput., 2 (1981), pp. 474–489.
[37] D. Y. Parkinson, D. M. Pelt, T. Perciano, D. Ushizima, H. Krishnan, H. S. Barnard, A. A. MacDowell, and J. Sethian, *Machine learning for micro-tomography*, in Developments in X-Ray Tomography XI, International Society for Optics and Photonics, 2017, 103910J.
[38] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti, *Recycling Krylov subspaces for sequences of linear systems*, SIAM J. Sci. Comput., 28 (2006), pp. 1651–1674.
[39] L. Reichel, F. Sgallari, and Q. Ye, *Tikhonov regularization based on generalized Krylov subspace methods*, Appl. Numer. Math., 62 (2012), pp. 1215–1228.
[40] R. A. Renaut, I. Hnětynková, and J. Mead, *Regularization parameter estimation for large-scale Tikhonov regularization using a priori information*, Comput. Statist. Data Anal., 54 (2010), pp. 3430–3445.
[41] N. A. B. Riis and Y. Dong, *A new iterative method for CT reconstruction with uncertain view angles*, in International Conference on Scale Space and Variational Methods in Computer Vision, Springer, New York, 2019, pp. 156–167.
[42] L. Ruthotto, J. Chung, and M. Chung, *Optimal experimental design for inverse problems with state constraints*, SIAM J. Sci. Comput., 40 (2018), pp. B1080–B1100.
[43] J. T. Slagel, J. Chung, M. Chung, D. Kozak, and L. Tenorio, *Sampled Tikhonov regularization for large linear inverse problems*, Inverse Problems, 35 (2019), 114008.
[44] K. M. Soodhalter, *Block Krylov subspace recycling for shifted systems with unrelated right-hand sides*, SIAM J. Sci. Comput., 38 (2016), pp. A302–A324, https://doi.org/10.1137/140998214.
[45] K. M. Soodhalter, D. B. Szyld, and F. Xue, *Krylov subspace recycling for sequences of shifted linear systems*, Appl. Numer. Math., 81 (2014), pp. 105–118.
[46] C. M. Stein, *Estimation of the mean of a multivariate normal distribution*, Ann. Statist., 9 (1981), pp. 1135–1151.
[47] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*, SIAM, Philadelphia, 2005.
[48] C. R. Vogel, *Computational Methods for Inverse Problems*, Frontiers in Appl. Math. 23, SIAM, Philadelphia, 2002.
[49] S. Wang, E. de Sturler, and G. H. Paulino, *Large-scale topology optimization using preconditioned Krylov subspace methods with recycling*, Internat. J. Numer. Methods Engrg., 69 (2007), pp. 2441–2468, https://doi.org/10.1002/nme.1798.
[50] S. J. Wright, R. D. Nowak, and M. A. Figueiredo, *Sparse reconstruction by separable approximation*, IEEE Trans. Signal Process., 57 (2009), pp. 2479–2493.