COMPUTATIONAL METHODS FOR LARGE-SCALE INVERSE PROBLEMS: A SURVEY ON HYBRID PROJECTION METHODS*

JULIANNE CHUNG† AND SILVIA GAZZOLA‡

Abstract. This paper surveys an important class of methods that combine iterative projection methods and variational regularization methods for large-scale inverse problems. Iterative methods such as Krylov subspace methods are invaluable in the numerical linear algebra community and have proved important in solving inverse problems due to their inherent regularizing properties and their ability to handle large-scale problems. Variational regularization describes a broad and important class of methods that are used to obtain reliable solutions to inverse problems, whereby one solves a modified problem that incorporates prior knowledge. *Hybrid projection methods* combine iterative projection methods with variational regularization techniques in a synergistic way, providing researchers with a powerful computational framework for solving very large inverse problems. Although the idea of a hybrid Krylov method for linear inverse problems goes back to the 1980s, several recent advances on new regularization frameworks and methodologies have made this field ripe for extensions, further analyses, and new applications. In this paper, we provide a practical and accessible introduction to hybrid projection methods in the context of solving large (linear) inverse problems.

Keywords: inverse problems, projection methods, regularization, Krylov methods, Tikhonov regularization, variational regularization, image deconvolution, computed tomography

1. Introduction.

We provide a gentle introduction to hybrid projection methods for regularizing inverse problems by answering three essential questions: (1) what is an inverse problem?, (2) what is regularization and why do we need it?, and (3) why should we use hybrid projection methods?

1.1. What is an inverse problem?.

Inverse problems arise in various scientific applications, including astronomy, geoscience, biomedical sciences, mining engineering, and medicine (see, e.g., [4, 23, 24, 32, 55, 75, 108, 111, 132, 136, 147, 235]). In these and other applications, one formulates an inverse problem for the purpose of recovering, from measured noisy data, the hidden object or phenomenon that gave rise to such data. In this survey, we are mainly interested in large, linear inverse problems of the form

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

where a certain unknown quantity of interest is stored in $\mathbf{x}_{\text{true}} \in \mathbb{R}^n$, the observed measurements are collected in $\mathbf{b} \in \mathbb{R}^m$, the matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ represents the forward data acquisition process, and $\mathbf{e} \in \mathbb{R}^m$ represents inevitable errors (noise) that arise from measurement, discretization, or floating point arithmetic. Given measured data, $\mathbf{b}$, and knowledge of the forward model, $\mathbf{A}$, the goal is to compute an approximation of $\mathbf{x}_{\text{true}}$, i.e., a solution to the inverse problem.

It is worth mentioning that, in stating (1.1), we have made three simplifying assumptions. First, we assume that the matrix $\mathbf{A}$ is known exactly while, in practice, one should take into account the error between the assumed mathematical model and the underlying physical model. Second, many inverse problems have an underlying mathematical model that is not linear, i.e., $\mathbf{b} = F(\mathbf{x}_{\text{true}}) + \mathbf{e}$, where $F(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$

*Current version: February 11, 2023.

†Department of Mathematics, Emory University, Atlanta, GA, USA (jmchung@emory.edu).

‡Department of Mathematical Sciences, University of Bath, United Kingdom (S.Gazzola@bath.ac.uk).

is a nonlinear operator that is sometimes referred to as the 'parameter-to-observation' map. For these cases, more sophisticated methods should be used to solve the nonlinear inverse problem; see, e.g., [126, 171, 234]. However, many of these methods require solving a subproblem with an approximate linear model, and the methods described in this survey paper have been successfully used for this purpose; indeed, these simplifying assumptions will be dropped in section 4. Finally, we only consider an additive noise model, although other realistic assumptions (e.g., multiplicative noise or mixed noise models) may need to be incorporated in the model.

Two model applications. In this paper, we focus on two model inverse problems from image processing: image deconvolution and tomographic reconstruction. These two problems will be used throughout the paper for illustrative purposes, so we briefly describe them here. Although the sizes of the problem considered here are typically too small to naturally appear in real-world applications, they are adopted for pedagogical reasons. All of the test problems presented in this survey can be generated using the MATLAB toolbox IR Tools [82][1]; see also section 5.

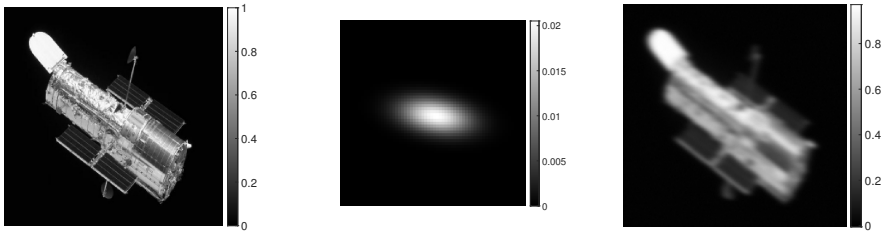
Image deconvolution (or deblurring) problems are very popular in the image and signal processing literature, and are of core importance in fields such as astronomy, biology, and medicine. More precisely, the mathematical model of this problem can be expressed in the continuous setting as an integral equation

$$b(\mathbf{s}) = \int a(\mathbf{s},\mathbf{t})x(\mathbf{t})d\mathbf{s} \quad + e\,, \tag{1.2}$$

where $\mathbf{s},\mathbf{t} \in \mathbb{R}^2$ represent spatial locations. The kernel or point spread function (PSF) $a(\mathbf{s},\mathbf{t})$ defines the blur, and, if the kernel has the property that $a(\mathbf{s},\mathbf{t}) = a(\mathbf{s}-\mathbf{t})$, then the blur is spatially invariant and the integration in (1.2) is a convolution. From equation (1.2) it can be observed that the blurred image is formed by integrating the PSF with the true image (represented as a scalar-valued function of the spacial location $\mathbf{t}$), which is further corrupted by adding a random perturbation e.

In a realistic setting, images are collected only at discrete points (pixels), and are only available in a finite region (i.e., in a viewable region). Thus, the basic image deconvolution problem is of the form (1.1) where $\mathbf{x}_{\text{true}}$ represents the vectorized sharp image, $\mathbf{A}$ represents the blurring process which specifies how the points in the image are distorted, and $\mathbf{b}$ contains the observed, vectorized, blurred and noisy image. Here we assume that the true and corrupted images have the same size, so that $\mathbf{A} \in \mathbb{R}^{n\times n}$. The kernel $a(\mathbf{s},\mathbf{t})$ has small support, and so pixels in the center of the viewable region are well defined. This results in a sparse, structured matrix $\mathbf{A}$. However, pixels of the blurred image near the boundary of the viewable region are affected by information outside the viewable region. Therefore, in constructing the matrix $\mathbf{A}$, one needs to incorporate boundary conditions to model how the image scene extends beyond the boundaries of the viewable region. Typical boundary conditions include zero, periodic, reflective, and antireflective. We highlight that the discrete problem associated to (1.2) often gives rise to matrices $\mathbf{A}$ with a well-defined structure: for instance, if the blur is spatially invariant and periodic boundary conditions are assumed, then $\mathbf{A}$ is a block circulant matrix with circulant blocks and efficient implementations of any image deconvolution algorithm can be obtained by exploiting structure of the

[1]The MATLAB programs used to produce the illustrations and experiments reported herein are available at the website: https://github.com/juliannechung/surveyhybridprojection. The commented lines in the MATLAB files are designed to make the various commands comprehensible to readers with basic programming experience.



(a) true image **(b)** PSF **(c)** observed image

FIG. 1.1. *Image deblurring test problem.* ***(a)*** *Sharp image of* 256×256 *pixels.* ***(b)*** *Zoomed image (400%) of an anisotropic Gaussian PSF.* ***(c)*** *Blurred and noisy image.*

matrix $\mathbf{A}$. We refer to [4, 17, 136, 200] for a discussion of the many fundamental modeling aspects of the image deconvolution problem and relevant features of the discrete problem.

In Figure 1.1 we display the test data that will be used through this paper, and that can be generated running the MATLAB script `generate_blur`. The size of the images are 256×256 pixels, so $n = 256^2 = 65,536$. The matrix $\mathbf{A}$ models a spatially invariant blur, with reflective boundary conditions and an anisotropic Gaussian PSF associated to the kernel

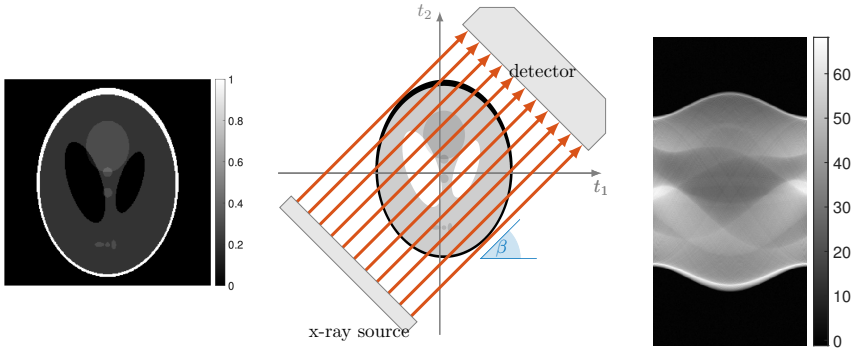
$$a(\mathbf{s}, \mathbf{t}) = \exp\left(-\frac{1}{2}(\mathbf{s}-\mathbf{t})^\top \mathbf{K}^{-1}(\mathbf{s}-\mathbf{t})\right), \quad \text{where} \quad \mathbf{K} = \begin{bmatrix} \sigma_1^2 & \rho^2 \\ \rho^2 & \sigma_2^2 \end{bmatrix}$$

and $\sigma_1^2\sigma_2^2 - \rho^4 > 0$.

Tomographic reconstruction (or computed tomography, CT) is a critical tool in many applications such as nondestructive evaluation and electron microscopy; the advent of newer technologies (e.g., spectral CT and photoacoustic tomography) prompts more efficient and more accurate reconstruction methods. CT consists in computing reconstructions of (parameters of) objects from projections, i.e., data obtained by integration along rays (typically straight lines, also called X-rays) that penetrate a domain. More precisely, this problem can be modelled by the Beer-Lambert law as follows: assuming that the initial intensity I_ℓ^0 of the X-ray ℓ is known, the intensity I_ℓ^1 recorded after the X-ray ℓ has penetrated the object (characterized by the attenuation coefficient $x(\mathbf{t})$, $\mathbf{t} \in \mathbb{R}^2$) is $I_\ell^1 = I_\ell^0 \exp(-\int_\ell x(\mathbf{t}(\ell))d\ell) + e_\ell$, where $d\ell$ is the Lebesgue measure along the X-ray ℓ and e_ℓ is the noise in the measurements. As a consequence, the data collected for the X-ray ℓ can be represented by the integral

$$-\log\left(\frac{I_\ell^1}{I_\ell^0}\right) = \int_\ell x(\mathbf{t}(\ell))d\ell - \log\left(\frac{e_\ell}{I_\ell^0}\right). \tag{1.3}$$

If we consider the so-called parallel-beam geometry, X-rays are typically parametrized by the angle with respect to the horizontal axis and the distance from the origin, and the integral appearing in (1.3) is the Radon transform of $x(\mathbf{t})$; note that other geometries are possible. We assume that m measurements of the form (1.3) are available, collected along m different X-rays at different distance from the origin and rotated by different angles. Employing a classical discretization scheme, which subdivides the object into an array of pixels and assumes that the function $x(\mathbf{t})$ is constant within each pixel, the above integral can be expressed as a discrete sum and the following expression for the (i,j)th entry a_{ij} of the sparse matrix $\mathbf{A}$ can be readily



(a) true image **(b)** tomography setup **(c)** observations

FIG. 1.2. *Parallel X-ray CT test problem.* ***(a)*** *Phantom of size* 256×256 *pixels.* ***(b)*** *Illustration of 2D X-ray parallel-beam tomography setup, modified from [202].* ***(c)*** *Observed sinogram containing data.*

derived as

$$a_{ij} = \begin{cases} L_{ij} & \text{if } j \in \mathcal{S}_i \\ 0 & \text{otherwise} \end{cases},$$

where $\mathcal{S}_i$ is the set of indices of those pixels that are penetrated by the ith X-ray, L_{ij} is the length of the ith X-ray through the jth pixel. Thus, the basic CT problem (1.3) is of the form (1.1), where $\mathbf{x}_{\text{true}}$ represents the unknown material attenuation coefficients, $\mathbf{A}$ represents a physical attenuation process, and $\mathbf{b}$ is the vectorized version of the so-called sinogram, i.e., a 2D array displaying the data measured by a detector, with the projection angles on the horizontal axis and the distance from the origin on the vertical axis. In most cases of tomography, $\mathbf{A}$ is a rectangular matrix, where $m < n$ if fewer measurements are collected than the number of unknowns, or $m > n$ if many projections can be obtained, or parameterization reduces the number of unknowns. Note also that, differently from the image deconvolution example where the forward operator maps images to images, in CT the forward operator maps images to sinogram space: this fundamentally affects the theory and solvers for the two applications. We refer to [32, 133, 174] for more details on tomography reconstruction problems.

In Figure 1.2 we display the test data for tomography that will be used through this paper, and that can be generated running the MATLAB script `generate_tomo`: this program still draws on IR TOOLS, which itself calls functions available within the MATLAB package AIR TOOLS II [132]. The object we wish to recover is the Shepp-Logan 'medical' phantom of size 256×256 pixels, so that $n = 65,536$. The system matrix is modeled after a parallel tomography process, where p parallel rays are generated from a source ideally placed infinitely far from a flat detector; moreover, the source-detector pair is rotated around the object, and measurements are recorded for N angles. Hence the number of observations is $m = pN$. One can vary the parameters that define the measurement geometry, such as the number of angles or the number of rays. For this example, we take measurements from 0 to 179 degrees in intervals of 1 degree, resulting in a set of data with $m = 64798$ measurements, so that $m < n$.

Now that we have addressed what is an inverse problem and looked at some examples of inverse problems, our focus turns to computing solutions to inverse problems, and for this we need regularization.

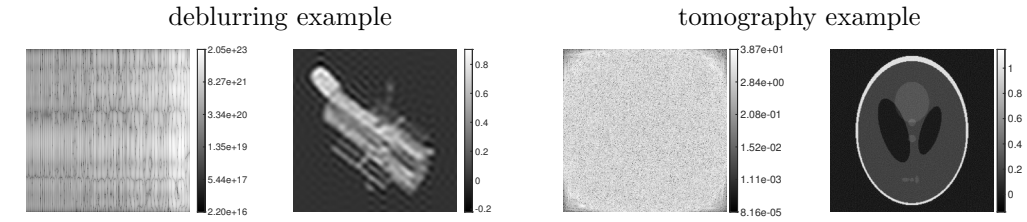


FIG. 1.3. *For each of the image deblurring and tomography test problems, we provide the computed unregularized* $\mathbf{A}^{-1}\mathbf{b}$ *or* $\mathbf{A}^{\dagger}\mathbf{b}$ *in logarithmic scale on the left, and a regularized solution on the right. Clearly the unregularized solution is heavily corrupted with noise and errors, while the regularized solution can provide a good approximation to the true object.*

1.2. What is regularization and why do we need it?. Solving inverse problems is notoriously difficult due to *ill-posedness*[2], whereby data acquisition noise and computational errors can lead to large changes in the computed solution. For instance, if $\mathbf{A}$ is invertible, the solution $\mathbf{A}^{-1}\mathbf{b}$ is dominated by noise and is useless for all practical purposes. The same happens for a more general $\mathbf{A}$ and for the minimal norm least squares solution expressed as

$$\mathbf{x} = \mathbf{A}^{\dagger}\mathbf{b}, \tag{1.4}$$

where $\mathbf{A}^{\dagger}$ is the generalized inverse of $\mathbf{A}$; see [105]. We illustrate this phenomenon for the described test problems, where a tiny amount of noise (here, $\|\mathbf{e}\|_2/\|\mathbf{A}\mathbf{x}_{\text{true}}\|_2 = 10^{-4}$) enters the data collection process. 'Naive' solutions for the image deblurring and tomography test problems are displayed in the first and third frames of Figure 1.3, respectively. It is clear that these solutions are unacceptable approximations to $\mathbf{x}_{\text{true}}$ (so much that, for image deblurring, the available corrupted image $\mathbf{b}$ even looks better!).

Regularization can be applied to inverse problems in order to overcome the inherent issues described above and compute a meaningful solution. The basic idea of regularization is to augment the model in (1.1) with additional information. If proper regularization is applied, a regularized solution (e.g., the ones shown in the second and fourth frame of Figure 1.3) should be close to the true solution. Regularization techniques come in many forms, and more details will be unfolded in the coming sections. However, for now, we consider two types of methods: variational regularization methods and iterative regularization methods.

Variational regularization methods for (1.1) involve solving optimization problems of the form,

$$\min_{\mathbf{x}\in\mathcal{C}} \mathcal{J}(\mathbf{b} - \mathbf{A}\mathbf{x}) + \lambda\mathcal{R}(\mathbf{x}), \tag{1.5}$$

where $\mathcal{J}$ is a loss (or fit-to-data) function, $\mathcal{R}$ is a regularization functional[3] $\lambda > 0$ is a regularization parameter that controls the amount of regularization, thereby determining how faithful the modified problem is to the original problem, and $\mathcal{C}$

[2]Here we consider the discretized problem, but point the reader to [73, 126] for discussions on the full implications of ill-posedness in continuous formulations.

[3]Commonly-used regularization functionals are defined in terms of operators that enforce some smoothness assumptions on the solution (see subsection 4.1) or Gaussian priors (see subsection 4.4).

denotes the set of feasible solutions (e.g., those that satisfy some constraints). The main advantage of formulation (1.5) is that general constraints and priors can be easily incorporated in the problem. Furthermore, well-known optimization methods can be used to solve (1.5). For the particular case where $\mathcal{C} = \mathbb{R}^n$ and $\mathcal{J}(\cdot) = \mathcal{R}(\cdot) = \|\cdot\|_2^2$, direct factorization methods can, in principle, be used to compute a solution (see subsection 2.1), although this is often computationally infeasible for large-scale problems. In this setting, the main disadvantage is the need to select the regularization parameter λ, often prior to solution computation.

On the other hand, *iterative regularization methods* for (1.1) typically consist in applying an iterative solver to

$$\min_{\mathbf{x}} \mathcal{J}(\mathbf{b} - \mathbf{A}\mathbf{x}), \tag{1.6}$$

and computing a regularized solution by early termination of the iterations. In practice, when $\mathcal{J}$ is expressed in the 2-norm, the iterative methods of choice are often subspace projection methods, whereby the problem is projected onto increasing (but relatively small) subspaces, and a projected subproblem is solved at each iteration by imposing certain optimality conditions on the approximated solution. Although an explicit choice of the regularization parameter λ is no longer required (contrary to (1.5)), the stopping criterion essentially serves as a regularization parameter, as it balances how 'faithful' the projected problem is to the original problem, $\mathbf{A}\mathbf{x} = \mathbf{b}$. In this setting, the main disadvantage is that general constraints cannot be easily incorporated, and, even when they can, they can only be handled through complicated nested iterative schemes.

Hybrid regularization methods combine variational and iterative regularization methods, leveraging the best features of each class, to provide a powerful computational framework for solving large-scale inverse problems. In this paper, we focus specifically on *hybrid projection methods* that start off as iterative regularization methods, i.e., the original problem (1.1) is projected onto a subspace of increasing dimension, and then the projected subproblem is solved using a variational regularization method.

1.3. Why should we use hybrid projection methods?. The main motivations for using hybrid projection methods to solve large inverse problems can be grouped as follows.

- For many multidimensional inverse problems (such as the model ones described in subsection 1.1), the matrix $\mathbf{A}$ is very large (to the point that it cannot be explicitly stored). In this setting, only matrix-vector multiplications with $\mathbf{A}$ (and possibly $\mathbf{A}^\top$) can be performed, most often taking advantage of high-performance computing or tools such as GPUs. Therefore, methods that compute a factorization of $\mathbf{A}$ (such as the singular value decomposition) or require access to its entries are ruled out. Hybrid regularization methods can be implemented without explicitly constructing and storing the matrix $\mathbf{A}$, by treating $\mathbf{A}$ and $\mathbf{A}^\top$ as linear operators.
- Even when adopting a variational regularization method (1.5), one may not know a good regularization parameter λ in advance, and standard methods to estimate λ can be expensive: indeed, these typically require first approximating problem (1.1), or solving many instances of (1.1) with different regularization parameters. Hybrid projection methods provide a natural setting to estimate an appropriate value of λ during the solution computation process: often in a heuristic way, but in some cases with theoretical guarantees.

- As mentioned above, many iterative solvers for (1.6), such as projection methods onto Krylov subspaces, have inherent regularizing properties and avoid data overfitting by early termination of the iterations. By using hybrid methods, it is possible to stabilize and enhance the regularized solutions computed by these methods. Moreover, hybrid methods enjoy nice theoretical properties. For instance we know that, for many hybrid frameworks, iterates obtained by first projecting the problem and then regularizing are mathematically equivalent to iterates obtained by first regularizing and then projecting the regularized problem.
- Hybrid methods provide a convenient framework for the development of extensions to general models and regularization terms. For instance, one could incorporate a variety of regularization terms of the form $\mathcal{R}(\mathbf{x}) = \|\mathbf{L}\mathbf{x}\|_2^2$, where $\mathbf{L} \in \mathbb{R}^{p\times n}$, or also consider functions $\mathcal{R}(\mathbf{x})$ that are not expressed in the 2-norm. As we will see in section 4, this amounts to modifying the projection subspace that can be used within hybrid methods. By doing so, other popular (but expensive) iterative regularization schemes (which are often based on nested cycles of iterations) can be avoided.

1.4. Outline of the paper. The remaining part of this survey is organized as follows. In section 2, we describe the general problem setup to provide some background for hybrid methods, including an overview of the most relevant direct and iterative regularization techniques. Then, in section 3, we discuss hybrid projection methods. After providing a brief summary of the historical developments of hybrid methods, we address the two main building blocks of any hybrid method: namely, generating a subspace for the solution (subsection 3.2.1) and solving the projected problem (subsection 3.2.2). Important numerical and theoretical aspects will also be covered: these include strategies to efficiently set the Tikhonov regularization parameter and stop the iterations (subsection 3.3) and available convergence proofs and approximation properties (subsection 3.4). In recent years, we have witnessed the extension of hybrid methods to solve a larger scope of problems and to cover broader scientific applications. In section 4, we provide an overview of some of these extensions:

- Beyond standard-form Tikhonov: Hybrid projection methods for general-form Tikhonov; see subsection 4.1.
- Beyond standard projection subspaces: Enrichment, augmentation, and recycling; see subsection 4.2.
- Beyond the 2-norm: Sparsity-enforcing hybrid projection methods for ℓ_p regularization; see subsection 4.3.
- Beyond deterministic inversion: Hybrid projection methods in a Bayesian setting; see subsection 4.4.
- Beyond linear forward models: Hybrid projection methods for nonlinear inverse problems; see subsection 4.5.

Pointers to relevant software packages are provided in section 5. Conclusions and outlook on future directions are provided in section 6.

Notations. Boldface lower-case letters denote vectors: e.g., $\mathbf{c} \in \mathbb{R}^n$; c_i, $1 \le i \le n$ denotes the ith entry of the vector $\mathbf{c}$. Boldface upper-case letters denote matrices: e.g., $\mathbf{C} \in \mathbb{R}^{m\times n}$; unless otherwise stated, $c_{i,j}$, $1 \le i \le m$, $1 \le j \le n$, denotes the (i,j)th entry of the matrix $\mathbf{C}$. Indexed matching boldface lower-case letters denote matrix columns: e.g., $\mathbf{c}_j \in \mathbb{R}^m$, $j = 1, \ldots, n$ denotes the jth column of the matrix $\mathbf{C}$. $\mathbf{I}_p$ denotes the identity matrix of size p, whose columns $\mathbf{e}_j$, $j = 1, \ldots, p$

are the canonical basis vectors of $\mathbb{R}^p$. The column space of a matrix $\mathbf{C}$ is denoted by $\text{ran}(\mathbf{C})$. Lower-case Greek letters, sometimes indexed, are often used to denote scalars. In the following, when there is no ambiguity (essentially until section 4), the shorthand notation $\|\cdot\| = \|\cdot\|_2$ will be used.

2. Background on direct and iterative regularization. For the simplified case where $\mathcal{J}(\mathbf{b} - \mathbf{A}\mathbf{x}) = \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2$, $\mathcal{R}(\mathbf{x}) = \|\mathbf{x}\|^2$, and $\mathcal{C} = \mathbb{R}^n$ in (1.5), we get the so-called standard-form Tikhonov problem,

$$\min_{\mathbf{x}} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2 + \lambda \|\mathbf{x}\|^2 . \tag{2.1}$$

We stress that the standard-form Tikhonov problem has been widely studied in both the mathematics and statistics communities and has been used in many scientific applications. However, computing Tikhonov-regularized solutions can still be challenging if the size of $\mathbf{x}$ is very large or if λ is not known a priori. In fact, being able to efficiently compute solutions to (2.1) was a main motivation for much of the early works on hybrid projection methods. For a discussion on extensions of hybrid methods to solve more general problems including the general-form Tikhonov problem, see section 4.

2.1. SVD-based direct regularization methods. In this section, we begin with the standard-form Tikhonov problem (2.1), and describe a direct approach based on the singular value decomposition (SVD) of $\mathbf{A}$ to compute a solution. For the problems of interest, direct application of this approach is not computationally feasible. However, the formulations briefly explored here will be useful for analysis and interpretation later in this paper. Furthermore, in the hybrid framework, the direct methods described here can be used to solve the projected problem. We point the interested reader to [126] for a more thorough exposition of direct methods.

Let us assume that $\mathbf{A} \in \mathbb{R}^{m\times n}$ with $\text{rank}(\mathbf{A}) = n \leq m$[4]. The SVD of $\mathbf{A}$ is defined as

$$\mathbf{A} = \mathbf{U}^{\text{A}}\mathbf{\Sigma}^{\text{A}}(\mathbf{V}^{\text{A}})^\top, \tag{2.2}$$

where $\mathbf{U}^{\text{A}} = \begin{bmatrix}\mathbf{u}_1^{\text{A}} & \dots & \mathbf{u}_m^{\text{A}}\end{bmatrix} \in \mathbb{R}^{m\times m}$ and $\mathbf{V}^{\text{A}} = \begin{bmatrix}\mathbf{v}_1^{\text{A}} & \dots & \mathbf{v}_n^{\text{A}}\end{bmatrix} \in \mathbb{R}^{n\times n}$ are orthogonal and $\mathbf{\Sigma}^{\text{A}} = \text{diag}(\sigma_1^{\text{A}}, \dots, \sigma_n^{\text{A}}) \in \mathbb{R}^{m\times n}$ contains the singular values, $\sigma_1^{\text{A}} \geq \sigma_2^{\text{A}} \geq \cdots \geq \sigma_n^{\text{A}} > 0$.

The solution to (2.1) can be written as

$$\mathbf{x}(\lambda) = \sum_{i=1}^{n} \phi_i(\lambda) \frac{(\mathbf{u}_i^{\text{A}})^\top \mathbf{b}}{\sigma_i^{\text{A}}} \mathbf{v}_i^{\text{A}}, \quad \text{where} \quad \phi_i(\lambda) = \frac{(\sigma_i^{\text{A}})^2}{(\sigma_i^{\text{A}})^2 + \lambda} \tag{2.3}$$

are the Tikhonov filter factors. Tikhonov regularization is one example from a wider class of spectral (or SVD) filtering methods that compute regularized solutions by imposing suitable filtering on the SVD components of the solution. The problems we are considering have some very small singular values, and the filter factors ϕ_i should be close to 1 for small i, and should approach 0 for large i. The Tikhonov filter factors $\phi_i(\lambda)$ have this property, with the amount of filtering prescribed by the regularization parameter λ. If $\lambda = 0$, then the filter factors are all equal to 1, and (2.3) is the unregularized solution (1.4). In this case, for small σ_i, components $\mathbf{u}_i^\top \mathbf{e}$ are magnified and overwhelm the solution (see Figure 1.3). For nonzero λ, ϕ_i approaches 1 as σ_i

[4]SVD-based methods can be defined more generally, see e.g., [126].

increases, so if σ_i is large with respect to λ, the i-th contribution to the solution is close to what it is in (1.4), while for small σ_i, the contribution is reduced by the filter factor. We refer the reader to the images provided in Figure 2.3 for an illustration of the impact of the regularization parameter λ on the solution.

Also the truncated SVD (TSVD) method, which regularizes (1.1) by computing

$$\mathbf{x}(k) = \sum_{i=1}^{k} \frac{(\mathbf{u}_i^{\mathrm{A}})^\top \mathbf{b}}{\sigma_i^{\mathrm{A}}} \mathbf{v}_i^{\mathrm{A}} = \sum_{i=1}^{n} \phi_i(k) \frac{(\mathbf{u}_i^{\mathrm{A}})^\top \mathbf{b}}{\sigma_i^{\mathrm{A}}} \mathbf{v}_i^{\mathrm{A}},$$

is a filtering method, with filter factors obtained by first setting a truncation index $k \in \{1, \dots, n\}$, and by then considering $\phi_i(k) = 1$, if $i \leq k$, and $\phi_i(k) = 0$ otherwise. Note that choosing $k = n$ returns the unregularized solution of (1.1). Therefore, k plays the role of regularization parameter.

Every spectral filtering method requires the selection of a regularization parameter. Common strategies to do so include the discrepancy principle [170], the generalized cross-validation (GCV) method [98], the unbiased predictive risk estimation (UPRE) method [227], the L-curve [165], and the normalized cumulative periodogram (NCP) [135]. Since there is not one method that will work for all problems, it is usually a good idea to try a variety of methods. For problems where the SVD of $\mathbf{A}$ is available, one can efficiently apply these regularization parameter strategies. For problems where computing the SVD is not feasible, parameter choice strategies are limited. However, we will see in subsection 3.3 that many of these existing regularization parameter selection techniques are not only feasible but also can be successfully integrated within hybrid projection methods.

The SVD of $\mathbf{A}$ (2.2), besides being an essential building block of SVD-filtering methods, is a pivotal tool for the analysis of discrete inverse problems [73, 125, 126]. For instance, looking at the decay of the singular values, one may infer different degrees of ill-posedness: typically, following [142], a polynomial decay of the form $\sigma_i^{\mathrm{A}} = ci^{-\alpha}$, $c > 0$, is classified as mild (if $0 < \alpha \leq 1$) or moderate (if $\alpha > 1$) ill-posedness, while an exponential decay of the form $\sigma_i^{\mathrm{A}} = \exp(-\alpha i)$, $\alpha > 0$, is classified as severe ill-posedness. Referring to the model applications presented in subsection 1.1, image deblurring is severely ill-posed, while computed tomography is only mildly ill-posed; see, e.g., [136, Chapter 4] and [133, Chapter 7] for a justification, respectively. The Picard condition, which in the continuous setting provides necessary and sufficient conditions for the existence of a solution of the form (1.4) [73], can be adopted in a discrete setting. Indeed, the so-called 'discrete Picard condition' is satisfied when the magnitude of $(\mathbf{u}_i^{\mathrm{A}})^\top \mathbf{b}$, on average, decays to zero faster than the singular values σ_i^{A}. Since these quantities appear in the expression of the (filtered) solutions, assessing if the discrete Picard condition holds can inform us on the existence of a meaningful solution to (1.1) as well as the effectiveness of any (filtering) regularization method applied to (1.1); see [123] for additional details. Tools such as the so-called 'Picard plot' [126] can be used to visually assess the validity of the discrete Picard condition. In Figure 2.1 we provide an illustration of two instances of the Picard plot for the image deblurring test problem introduced in subsection 1.1, one for uncorrupted data and one for data affected by Gaussian white noise: we can clearly see that the discrete Picard condition holds in the first instance (provided that numerically zero quantities are excluded because of the effects of rounding errors), but not in the second instance.

2.2. Iterative regularization methods. Next we describe iterative regularization, where an iterative method is used to approximate the solution of the *unreg-*

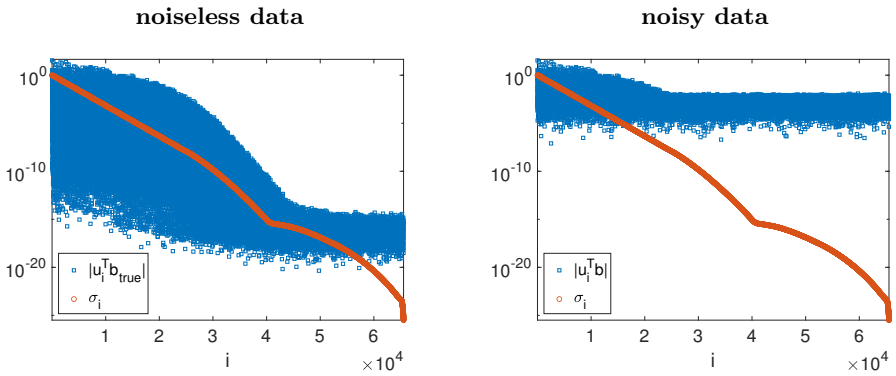


FIG. 2.1. *Picard plots for the image deblurring test problem. On the left frame we consider noiseless data, i.e., the inverse problem reads* $\mathbf{A}\mathbf{x}_{\text{true}} = \mathbf{b} =: \mathbf{b}_{\text{true}}$*: in this case, the discrete Picard condition holds and it is possible to compute (an approximation of)* $\mathbf{x}_{\text{true}}$ *without resorting to regularization. On the right frame we consider data* $\mathbf{b}$ *corrupted by Gaussian white noise* $\mathbf{e}$*, such that* $\|\mathbf{e}\|/\|\mathbf{b}_{\text{true}}\| = 10^{-2}$*: in this case, the discrete Picard condition does not hold, and a meaningful approximation of* $\mathbf{x}_{\text{true}}$ *can be computed only employing regularization. We can clearly see that the singular values for this test problem decay exponentially.*

ularized least-squares problem,

$$\min_{\mathbf{x}} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2 , \tag{2.4}$$

and early termination of the iterative process results in a regularized solution. We focus on projection methods where the underlying concept is to constrain the solution at the kth iteration to lie in a k-dimensional subspace spanned by the (typically orthonormal) columns of some matrix $\mathbf{V}_k = \begin{bmatrix}\mathbf{v}_1 & \cdots & \mathbf{v}_k\end{bmatrix}$, where $\mathbf{v}_i \in \mathbb{R}^n$. That is, the regularized solution is given as

$$\mathbf{x}_k = \mathbf{V}_k \mathbf{y}_k , \quad \text{where} \quad \mathbf{y}_k = \underset{\mathbf{y}\in\mathbb{R}^k}{\arg\min} \|\mathbf{b} - \mathbf{A}\mathbf{V}_k\mathbf{y}\|^2 . \tag{2.5}$$

We are interested in projection methods using Krylov subspaces [205]. Given $\mathbf{C} \in \mathbb{R}^{n\times n}$ and $\mathbf{d} \in \mathbb{R}^n$, a Krylov subspace is defined as

$$\mathcal{K}_k(\mathbf{C}, \mathbf{d}) = \text{span}\{\mathbf{d}, \mathbf{C}\mathbf{d}, \mathbf{C}^2\mathbf{d}, \ldots, \mathbf{C}^{(k-1)}\mathbf{d}\} .$$

Here and in the following, we assume that the dimension of $\mathcal{K}_k(\mathbf{C}, \mathbf{d})$ is k (i.e., no degeneracy). When Krylov methods are employed to solve problem (2.4), $\mathbf{C}$ and $\mathbf{d}$ are defined in terms of $\mathbf{A}$ and $\mathbf{b}$, respectively. In iterative regularization methods based on the Arnoldi process (such as GMRES), the columns of $\mathbf{V}_k$ form an orthonormal basis for $\mathcal{K}_k(\mathbf{A}, \mathbf{b})$, where $\mathbf{A}$ must be square. However, for problems where $\mathbf{A}$ is not square (e.g. in tomography), iterative regularization methods based on the Lanczos or Golub-Kahan bidiagonalization process (such as LSQR [185, 186] or LSMR [78]) are used, and the columns of $\mathbf{V}_k$ form an orthonormal basis for $\mathcal{K}_k(\mathbf{A}^\top\mathbf{A}, \mathbf{A}^\top\mathbf{b})$. Note that LSQR is mathematically equivalent to CGLS, i.e., the conjugate gradient (CG) method applied to the normal equations. All the Krylov methods mentioned so far are minimum residual methods, so that the kth iterate can be written as $\mathbf{x}_k = \mathbf{V}_k\mathbf{y}_k$, where $\mathbf{y}_k$ solves (2.5). More details regarding the generation of these basis vectors will be addressed in subsection 3.2.1.

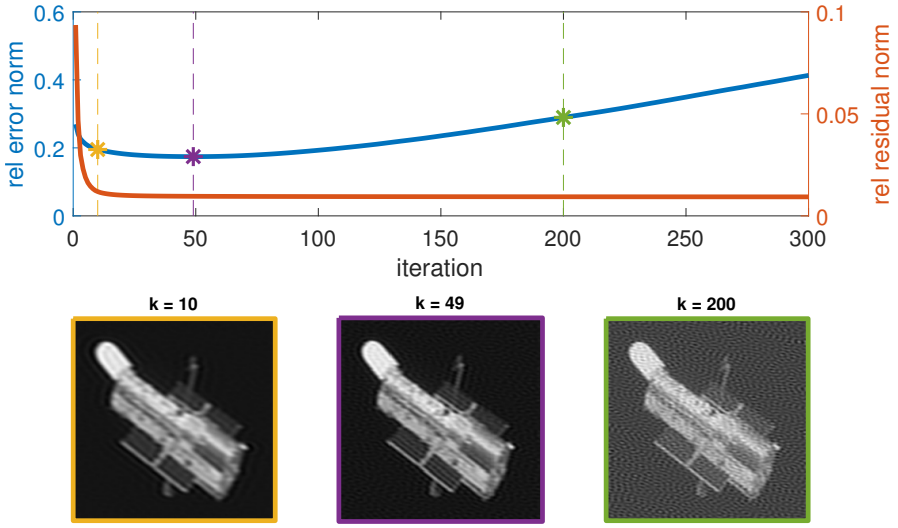


FIG. 2.2. *Illustration of the semi-convergence phenomenon for the image deblurring test problem using the CGLS method to solve (2.4). In the plot we provide the relative reconstruction error norms* $\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|/\|\mathbf{x}_{\text{true}}\|$ *per iteration* k *and the relative residual norms* $\|\mathbf{b} - \mathbf{A}\mathbf{x}_k\|/\|\mathbf{b}\|$ *per iteration* k*. Image reconstructions at iterations* 10*,* 49 *and* 200 *are provided and correspond to the stars in the plot.*

A crucial feature of classical Krylov methods when applied to ill-posed inverse problems is that, oftentimes, these iterative methods exhibit a regularizing effect in that the projection subspace in early iterations provides a good basis for the solution. One of the first Krylov methods that was proven to have regularizing properties is CGLS [119, 175]. Indeed, the CGLS iterates can be expressed as filtered SVD solutions [126]. Since these iterative methods converge to the least-squares solution, we get a phenomenon commonly referred to as 'semi-convergence' (after [174]), whereby the relative reconstruction errors decrease at early iterations but increase at later iterations due to noise manifestation and amplification. See Figure 2.2 for an illustration. Because of this, for iterative regularization, the stopping iteration plays the role of the regularization parameter. There have been many investigations into developing stopping criteria for iterative methods for inverse problems. If a good estimate of the amount of noise is available, the most widely used and intuitive approach is to stop iterating as soon as the value of the objective function in (2.5) reaches the magnitude of the noise: this is the so-called discrepancy principle. GCV and modifications of GCV have also been used for stopping criteria [20, 22, 59]. We again refer to subsection 3.3 for more details about these strategies. The main challenge is that for some problems, the reconstruction can be very sensitive to the choice of the stopping iteration. Thus, if the method stops a little too late, the reconstruction is already contaminated by noise (i.e., under-regularized). On the contrary, if the iterations are stopped too early (i.e., over-regularization), a potentially better reconstruction is precluded. We remark that it has been observed that semi-convergence may appear somewhat 'prematurely' and that it is sometimes important to have a larger approximation subspace (which would otherwise be beneficial for the solution; see [144, 182]).

In some circumstances, even if direct regularization methods are feasible, one may prefer to adopt iterative regularization methods. Some reasons are highlighted in [120], where the following heuristic motivation is given: contrary to TSVD, Krylov

methods (such as LSQR) generate an approximation subspace that is tailored to the current right-hand-side vector. Therefore, the basis vectors for the solution may be better 'adapted' to the given problem than the right singular vectors.

In many fields from numerical linear algebra to differential equations, iterative methods and, in particular, preconditioned Krylov methods, have been immensely successful in solving large, sparse systems of equations efficiently [205]. Iterative methods have also gained widespread use in the inverse problems community. One of the main reasons for this is that neither the matrix $\mathbf{A}$ nor its factorization need to be constructed, and thus, these methods are ideal for large-scale problems. Furthermore, it has been observed many times that the generated Krylov subspaces are rich in information for representing the solution (i.e., corresponding to the large singular vectors). Thus, a reasonable solution can be obtained in only a few iterations. The greatest caveat for inverse problems is semi-convergence, so great care must be taken to find good stopping criteria. In subsection 3.4 we will dwell more on the properties of the approximation subspaces generated by different projection methods.

2.3. Iterative methods for solving the Tikhonov problem. As described in subsection 2.1, the Tikhonov problem (2.1) can be easily solved if the SVD of $\mathbf{A}$ is available and, in this case, many well-regarded parameter choice strategies can be applied to compute a suitable value of the regularization parameter λ. Nonetheless, for large-scale problems where $\mathbf{A}$ cannot be constructed but matrix-vector products with $\mathbf{A}$ and $\mathbf{A}^\top$ can be computed efficiently, iterative projection methods [185, 186] can be used to solve the equivalent Tikhonov problem,

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

However, this approach may not be convenient if a suitable value of λ is not known a priori. In this case, often one must solve problem (2.6) from scratch for many different values of λ and this eventually results in an expensive approach (note that for specific iterative solvers, one may adopt smart strategies to reduce computations; see [81]).

Similar to the discussion in subsection 2.1, the value of λ can have a considerable impact on the quality of the reconstructed solution. Moreover, when using iterative methods to solve the Tikhonov problem, one can to some extent leverage the number of iterations to enforce additional regularization. For illustration, we use CGLS to solve (2.6) for various choices of λ for the image deblurring example and provide relative reconstruction error norms per iteration in Figure 2.3. Notice that if λ is chosen too small, severe semi-convergence appears and a good stopping iteration is as crucial as for the 'purely' iterative (i.e., $\lambda = 0$) methods introduced in subsection 2.2; on the contrary, if λ is chosen too large, the solution is over-regularized and additional iterations cannot mitigate this phenomenon.

We can draw the following conclusions about the successful working of a standard iterative method to solve a Tikhonov regularized problem. First, the user must be confident in the choice of the regularization parameters. Second, the user must be cognizant about the interplay of λ and k for (2.6), as both of these contribute to a suitable regularized solution. As we will see in the next section, hybrid projection methods provide an alternative approach of combining an iterative method and Tikhonov regularization, whereby a regularization parameter can be automatically and adaptively tuned during the iterative process.

3. Hybrid projection methods. A hybrid projection method is an iterative strategy that regularizes problem (2.4) by projecting it onto subspaces of increasing

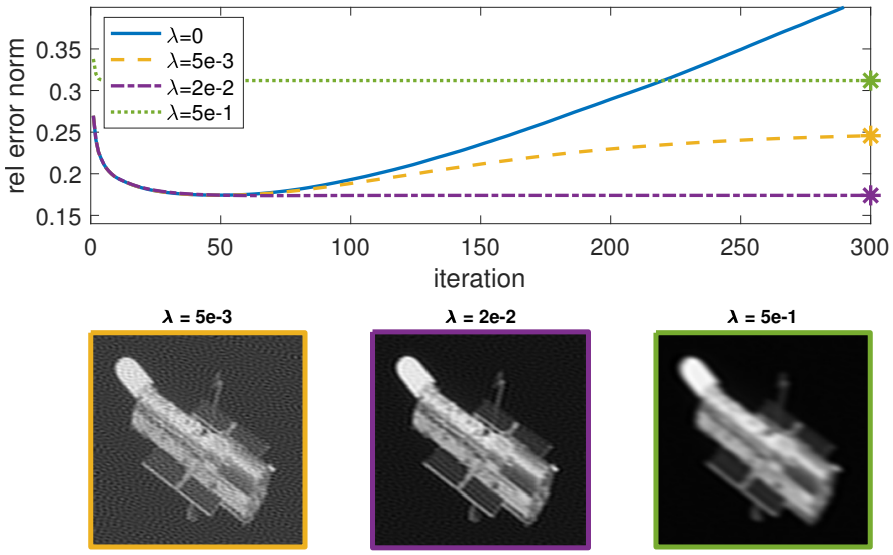


FIG. 2.3. *Illustration of the interplay of the regularization parameter and the number of iterations when using CGLS to solve the Tikhonov problem with a* fixed *regularization parameter. We provide the history of the relative reconstruction error norms for (2.6) versus the number of CGLS iterations, for different values of the regularization parameter λ. Image reconstructions at iteration* 300 *are provided for different choices of λ (these correspond to the stars on the error plots). Notice that for too small values of λ, semi-convergence can be still detected (i.e., the behavior is similar to the 'unregularized' $\lambda = 0$ case, also displayed in Figure 2.2). For too large values of λ, the iterative method can only produce smooth reconstructions that never achieve high accuracy.*

dimension and by solving the projected problem using a variational regularization method. Formally, recalling the framework for iterative projection methods unfolded in (2.5), the solution at the kth iteration of a hybrid projection method for standard Tikhonov can be represented as

$$\mathbf{x}_k(\lambda_k) = \mathbf{V}_k \mathbf{y}_k(\lambda_k)\,, \quad \text{where} \quad \mathbf{y}_k(\lambda_k) = \underset{\mathbf{y}\in\mathbb{R}^k}{\arg\min}\, \mathcal{J}(\mathbf{b} - \mathbf{A}\mathbf{V}_k\mathbf{y}) + \lambda_k \|\mathbf{y}\|^2\,. \tag{3.1}$$

Notice that the solution incorporates both regularization from the projection subspace (determined by the choice of the projection method, i.e., $\mathbf{V}_k$ and k) and a potentially changing variational regularization term (defined by λ_k). Again, for simplicity and for historical reasons, in this section we focus on the standard-form Tikhonov problem (2.1); extensions will be considered in section 4.

An illustration. Before we get into the details of hybrid projection methods, we begin by illustrating the benefits of allowing *adaptive* choices for the regularization parameter, where a different regularization parameter can be used at each iteration, i.e., $\lambda = \lambda_k$. In Figure 3.1 for the image deblurring problem, we provide relative reconstruction error norms per iteration in the left panel and computed regularization parameters per iteration in the right panel. Here, 'opt' refers to selecting at each iteration the regularization parameter that delivers the smallest relative reconstruction error. Parameter selection methods 'DP' and 'wGCV' and others will be discussed in subsection 3.3. Notice that relative reconstruction errors using (appropriately selected) adaptive regularization parameters can overcome semi-convergence behavior (see Figure 2.2) and result in reconstruction errors that are close to a pre-selected optimal regularization parameter. For early iterations, a small λ_k (corresponding to

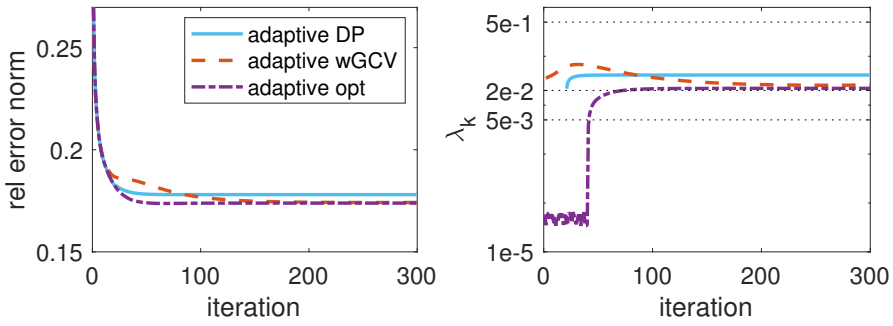


FIG. 3.1. *Illustration of a hybrid projection method where an iterative method is run with an* adaptive *regularization parameter selection method, applied to the image deblurring test problem. In the left panel, we provide the history of the relative reconstruction error norms per iteration k, for $\lambda = \lambda_k$ set adaptively at each iteration using the discrepancy principle (DP) and weighted GCV (wGCV); see subsection 3.3 for more details. The optimal (opt) parameter corresponds to selecting the λ_k that minimizes the relative error at each iteration k. In the right panel, we provide the computed regularization parameters, where the three horizontal lines correspond to $\lambda = 5 \cdot 10^{-3}, 2 \cdot 10^{-2}$ and $5 \cdot 10^{-1}$.*

little variational regularization) is sufficient, but as iterations progress, the adaptive methods should select regularization parameters close to the optimal one.

An algorithm. Now that we have seen the potential benefits of a hybrid projection method, let us discuss the general structure of such algorithms. A sketch of a hybrid projection method for standard Tikhonov regularization is provided in Algorithm 3.1, with links to the appropriate sections of the paper for more details. Notice that each iteration requires the expansion of the projection subspace and the solution of a projected regularized problem where a regularization parameter can be selected at each iteration.

Algorithm 3.1 Hybrid projection method for standard Tikhonov

Input: $\mathbf{A}$, $\mathbf{b}$, $k = 1$, projection method, regularization parameter selection method

1: **while** stopping criterion not satisfied **do**
2: Expand the projection subspace ran($\mathbf{V}_k$); see subsection 3.2.1
3: Select a regularization parameter λ_k; see subsection 3.3
4: Solve the projected regularized problem (3.1); see subsection 3.2.2
5: $k = k + 1$
6: **end while**

Output: $\mathbf{x}_k(\lambda_k)$

An outline of the remaining part of this section is as follows. We begin in subsection 3.1 with a brief historical overview of hybrid projection methods. Details and derivations for hybrid projection methods can be found in subsection 3.2, where we describe some techniques for steps 2 and 4 of Algorithm 3.1: namely, expanding the projection subspace (e.g., via the Arnoldi or Golub-Kahan process) and solving the projected regularized problem. One of the main advantages and features of a hybrid projection method is the ability to select regularization parameters automatically and adaptively (i.e., during the iterative process as in step 3 of Algorithm 3.1). Thus, we dedicate subsection 3.3 to providing an overview of regularization parameter selection methods, with a particular emphasis on methods that have been critical for the success of hybrid projection methods. Although we still do not have a complete analysis of

the regularizing properties of every hybrid projection method, important theoretical results and properties have nonetheless been established for specific methods, and we highlight some of these in subsection 3.4.

3.1. Historical development of hybrid methods. Since the seminal publication by O'Leary and Simmons in 1981 [182], there have been four decades of progress and developments in the field of hybrid projection methods. A quick google scholar search shows that the number of citations for this paper alone nearly doubles in each subsequent decade. In this section, we provide a brief overview of the main contributions and highlights by decade. This is by no means an exhaustive list of publications, and we realize that there may be bias in the selections, but our goal is to provide the reader with some historical context. For a novice reader, this section can be skipped upon first reading.

1981-1990. To the best of our knowledge, the first journal publication to introduce a hybrid projection method was by O'Leary and Simmons in 1981 [182], in which they describe a 'projection-regularization' method that combines the Golub-Kahan bidiagonalization for projection and TSVD for regularization. In the paper, the authors mention that a different hybrid algorithm was proposed independently, with a reference to a technical report by Björck in 1980 [19]. Björck's work appeared in BIT in 1988 [20], where a key difference was that the right hand side vector $\mathbf{b}$ was used as the starting vector, since it was noted that allowing the algorithms to start with an arbitrary vector did not always perform well. Other algorithmic contributions included the use of cross-validation to determine stopping criteria and transformations to standard form based on Eldén's earlier work [71]. Example problems from time series deconvolution were used in [182], but the computational technology at the time was still quite limited [214].

1991-2000. During this period, hybrid projection methods gained significant traction in the numerical linear algebra community as well as practical utility in various seismic imaging applications [206, 239]. As described in Björck's book [21], various researchers were interested in characterizing the regularizing properties of Krylov methods (e.g., see Nemirovskii [175] and various works by Hanke and Hansen [121]), where the main motivation was to determine appropriate stopping criteria for iterative methods when applied to ill-posed problems. Advancements in hybrid methods included investigations into stable computations (e.g., reorthogonalization) by Björck, Grimme and Van Dooren [22] as well as new methods for selecting regularization parameters. For example, Calvetti, Golub and Reichel [34] proposed a hybrid approach where a so-called 'L-ribbon' was computed using a partial Lanczos bidiagonalization. The first application of hybrid methods for large, nonlinear inversion was described in Haber and Oldenburg [114], and hybrid approaches based on projections with GMRES or Arnoldi were described by Calvetti, Morigi, Reichel, and Sgallari in [41]. In terms of software, Hansen laid some groundwork for hybrid methods in the Regularization Tools package [124], where routines for computing the lower bidiagonal matrix and the SVD of a bidiagonal matrix were provided.

Not all of the work on hybridized methods during this time followed the project-then-regularize framework. Related work on estimating the regularization parameter for large-scale Tikhonov problems by exploiting relationships with CG also appeared during this time. For example, Golub and Von Matt [107] estimated the GCV parameter for large problems using relations between Gaussian quadrature rules and Golub-Kahan bidiagonalization, and Frommer and Maass [81] exploited the shift structure of CG methods to solve Tikhonov problems with multiple regularization parameters

simultaneously.

2001-2010. With significant improvements in computational resources and capabilities, researchers became interested in how to realize the benefits of hybrid projection methods in practice. In this decade, we saw a surge in the development of regularization parameter selection methods for hybrid projection methods [35, 59, 160, 168, 231] and various extensions of hybrid methods to include general regularization terms (e.g., total variation [39] and general-form Tikhonov [140, 159]). Still, the main focus was on TSVD and Tikhonov for regularization. New interpretations of hybrid methods based on Lanczos and TSVD were described in [120], and noise level estimation from the Golub-Kahan bidiagonalization process, which can be used for determining the stopping iteration, were described in [139]. A fully automatic MATLAB routine called 'HyBR' was provided in [59], where a Golub-Kahan hybrid projection method with a weighted GCV parameter choice rule was implemented for standard Tikhonov regularization. Hybrid methods were implemented on distributed computing architectures [63] and were used for many new applications such as cryoelectron microscopy and electrocardiography [153]. An overview can be found in [57].

2011-present. At present, hybrid projection methods have gained significant interest in many research fields, and contributions range from new methodologies and advanced theories to innovations in scientific applications. Many of the recent developments in hybrid methods will be elaborated on in section 4, so here we just provide some highlights. In particular, there have been many papers on the Arnoldi-Tikhonov method [91, 93]. Flexible and generalized hybrid methods based on state-of-the-art Krylov subspace methods have been introduced for including more general regularization terms and constraints [54, 61, 84, 88, 162, 192]. There have been connections made to the field of computational uncertainty quantification [209] and new insights in the regularization parameter selection strategies [198], and regularization by approximate matrix functions [25, 64, 180]. There exists a plethora of papers in application and imaging journals, and a new software package called IR Tools [82]. We also refer interested readers to an older comprehensive survey paper on Krylov projection methods and Tikhonov regularization [94]. Other survey papers focusing on specific aspects of hybrid methods, such as their use in image restoration problems [16] or strategies for computing regularization parameters [85] appeared recently.

3.2. Algorithmic approaches to hybrid projection methods. Following derivations of iterative methods for sparse rectangular systems, in subsection 3.2.1 we first describe common techniques to iteratively build a solution subspace and to efficiently project problem (2.4) onto subspaces of increasing dimension. Then, in subsection 3.2.2, we describe techniques to solve the projected regularized problem.

3.2.1. Projecting onto subspaces of increasing dimension. In this section, we address the first building block of a hybrid projection method, which is to use an iterative method to build a sequence of solution subspaces of small but increasing dimension, and to efficiently project the problem onto them. The first important question to address is: in practice, what makes a good solution subspace? Indeed, a clever choice of basis vectors should satisfy multiple requirements. Since the number of iterations required for solution computation should be small, a good basis should be able to accurately capture the important information about the solution in a few vectors. In this sense, the dominant SVD basis vectors (i.e, the first columns of $\mathbf{V}^{\mathrm{A}}$ in (2.2)) would be ideal (recall discussion about filtered SVD in subsection 2.1). Unfortunately, computing the SVD can be infeasible for very large problems. Thus, we

seek a subspace sequence that exhibits a rapid enough decrease in the residual norm $\|\mathbf{r}\|^2 = \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2$ in (2.4) so that early termination provides good approximations [105]. Another desirable property is that the solution subspace can be generated efficiently. Here we mean that the main computational cost per iteration is manageable, i.e, requires one matrix-vector multiplication with $\mathbf{A}$ and possibly one with $\mathbf{A}^\top$. Lastly, for numerical stability it may be desirable to have an orthogonal basis for the solution subspace [181]. Although a variety of projection methods and solution subspaces could be used, here we focus on two of the most common projection methods on Krylov subspaces, namely the Arnoldi process and the Golub-Kahan bidiagonalization process. We will argue that these algorithms compute good solution spaces, according to the criteria listed above.

Arnoldi process. The Arnoldi process generates an orthonormal basis for $\mathcal{K}_k(\mathbf{A}, \mathbf{b})$ at the kth iteration [205]. Given matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ and vector $\mathbf{b} \in \mathbb{R}^n$, with initialization $\mathbf{v}_1 = \mathbf{b}/\beta_1$ where $\beta_1 = \|\mathbf{b}\|$, the kth iteration of the Arnoldi process generates vector $\mathbf{v}_{k+1}$ and scalars $h_{i,k}$ for $i = 1, \ldots, (k+1)$ such that, in matrix form, the following Arnoldi relationship holds,

$$\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\mathbf{H}_k, \tag{3.2}$$

where

$$\mathbf{H}_k := \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,k} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,k} \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & h_{k,k} \\ 0 & \cdots & 0 & h_{k+1,k} \end{bmatrix} \in \mathbb{R}^{(k+1)\times k}, \mathbf{V}_{k+1} := [\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}] \in \mathbb{R}^{n\times(k+1)},$$

and where, in exact arithmetic, $\mathbf{V}_{k+1}^\top \mathbf{V}_{k+1} = \mathbf{I}_{k+1}$. Notice that $\mathbf{V}_{k+1}$ contains an orthonormal basis for $\mathcal{K}_{k+1}(\mathbf{A}, \mathbf{b})$ and $\mathbf{H}_k$ is an upper Hessenberg matrix (i.e., zero below the first subdiagonal). The Arnoldi process is summarized in Algorithm 3.2.

Algorithm 3.2 Arnoldi process

Input: $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{b} \in \mathbb{R}^n$, k

1: $\mathbf{v}_1 = \mathbf{b}/\|\mathbf{b}\|$
2: **for** $j = 1, \ldots, k$ **do**
3: $\quad \mathbf{v} = \mathbf{A}\mathbf{v}_j$
4: $\quad$ **for** $i = 1, \ldots, j$ **do** $h_{i,j} = \mathbf{v}^\top \mathbf{v}_i$ **end for**
5: $\quad \mathbf{v} = \mathbf{v} - \sum_{i=1}^{j} h_{i,j}\mathbf{v}_i$
6: $\quad \mathbf{v}_{j+1} = \mathbf{v}/h_{j+1,j}$ where $h_{j+1,j} = \|\mathbf{v}\|$
7: **end for**

Output: $\mathbf{V}_k$, $\mathbf{H}_k$

Notice that the main computational cost of each iteration of the Arnoldi process is one matrix-vector multiplication with $\mathbf{A}$. Breakdown of the iterative process may happen (when $h_{j+1,j} = 0$), but we are typically not concerned about this because these methods are meaningful only when the number of iterations is not too high. We further note that there is a mathematically equivalent but numerically more stable implementation of the Arnoldi process based on Householder transformations.

One caveat, especially for inverse problems such as tomography, is that the Arnoldi method relies on the fact that $\mathbf{A}$ is square. For rectangular problems, one

could apply the Arnoldi process to the normal equations $\mathbf{A}^\top\mathbf{A}\mathbf{x} = \mathbf{A}^\top\mathbf{b}$, in which case the Arnoldi algorithm simplifies to the symmetric Lanczos tridiagonalization process where, thanks to symmetry, $\mathbf{H}_k$ is tridiagonal rather than upper Hessenberg. However, a numerically favorable approach would be to avoid the normal equations and to work directly with $\mathbf{A}$ and $\mathbf{A}^\top$ separately. This is achieved by adopting the Golub-Kahan bidiagonalization process, which we describe next; we will also state more precisely the links between the Golub-Kahan bidiagonalization and the symmetric Lanczos process.

Golub-Kahan bidiagonalization process. Contrary to the Arnoldi process that generates only one orthonormal basis (i.e, for $\mathcal{K}_k(\mathbf{A}, \mathbf{b})$), the Golub-Kahan bidiagonalization (GKB) process generates two sets of orthonormal vectors that span Krylov subspaces $\mathcal{K}_k(\mathbf{A}^\top\mathbf{A}, \mathbf{A}^\top\mathbf{b})$ and $\mathcal{K}_k(\mathbf{A}\mathbf{A}^\top, \mathbf{b})$ [99]. Given a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ and a vector $\mathbf{b} \in \mathbb{R}^m$, with initialization $\mathbf{u}_1 = \mathbf{b}/\beta_1$ where $\beta_1 = \|\mathbf{b}\|$ and $\mathbf{v}_1 = \mathbf{A}^\top\mathbf{u}_1/\alpha_1$ where $\alpha_1 = \left\|\mathbf{A}^\top\mathbf{u}_1\right\|$, at the kth iteration of the GKB process, we generate vectors $\mathbf{u}_{k+1}$, $\mathbf{v}_{k+1}$, and scalars α_{k+1} and β_{k+1} such that in matrix form, the following relationships hold,

$$\begin{aligned} \mathbf{A}\mathbf{V}_k &= \mathbf{U}_{k+1}\mathbf{B}_k \\ &= \mathbf{U}_k\mathbf{B}_{k,k} + \beta_{k+1}\mathbf{u}_{k+1}\mathbf{e}_k^\top \end{aligned}, \qquad \begin{aligned} \mathbf{A}^\top\mathbf{U}_{k+1} &= \mathbf{V}_k\mathbf{B}_k^\top + \alpha_{k+1}\mathbf{v}_{k+1}\mathbf{e}_{k+1}^\top \\ &= \mathbf{V}_{k+1}\mathbf{B}_{k+1,k+1}^\top \end{aligned}, \tag{3.3}$$

where

$$\mathbf{B}_k := \begin{bmatrix} \alpha_1 & 0 & \cdots & 0 \\ \beta_2 & \alpha_2 & \ddots & \vdots \\ 0 & \beta_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \alpha_k \\ 0 & \cdots & 0 & \beta_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{k,k} \\ \beta_{k+1}\mathbf{e}_k^\top \end{bmatrix} \in \mathbb{R}^{(k+1)\times k},$$

is bidiagonal, $\mathbf{U}_{k+1} := [\mathbf{u}_1, \ldots, \mathbf{u}_{k+1}] \in \mathbb{R}^{m\times(k+1)}$ and $\mathbf{V}_k := [\mathbf{v}_1, \ldots, \mathbf{v}_k] \in \mathbb{R}^{n\times k}$. In exact arithmetic $\mathbf{U}_{k+1}$ has orthonormal columns that span $\mathcal{K}_{k+1}(\mathbf{A}\mathbf{A}^\top, \mathbf{b})$ and $\mathbf{V}_k$ has orthonormal columns that span $\mathcal{K}_k(\mathbf{A}^\top\mathbf{A}, \mathbf{A}^\top\mathbf{b})$. The GKB process is summarized in Algorithm 3.3.

Algorithm 3.3 Golub-Kahan bidiagonlization (GKB) process

Input: $\mathbf{A}$, $\mathbf{b}$, k

1: $\beta_1\mathbf{u}_1 = \mathbf{b}$, where $\beta_1 = \|\mathbf{b}\|$
2: $\alpha_1\mathbf{v}_1 = \mathbf{A}^\top\mathbf{u}_1$, where $\alpha_1 = \left\|\mathbf{A}^\top\mathbf{u}_1\right\|$
3: **for** $j = 1, \ldots, k$ **do**
4: $\quad\beta_{j+1}\mathbf{u}_{j+1} = \mathbf{A}\mathbf{v}_j - \alpha_j\mathbf{u}_j$, where $\beta_{j+1} = \|\mathbf{A}\mathbf{v}_j - \alpha_j\mathbf{u}_j\|$
5: $\quad\alpha_{j+1}\mathbf{v}_{j+1} = \mathbf{A}^\top\mathbf{u}_{j+1} - \beta_{j+1}\mathbf{v}_j$, where $\alpha_{j+1} = \left\|\mathbf{A}^\top\mathbf{u}_{j+1} - \beta_{j+1}\mathbf{v}_j\right\|$
6: **end for**

Output: $\mathbf{U}_{k+1}$, $\mathbf{V}_{k+1}$, $\mathbf{B}_k$

The main computational cost of each iteration of the GKB process is one matrix-vector multiplication with $\mathbf{A}$ and one with $\mathbf{A}^\top$. Breakdown in the GKB algorithm would occur when $\beta_{j+1} = 0$ or $\alpha_{j+1} = 0$ but, similarly to the Arnoldi process, we will assume that breakdowns do not happen. A potential concern with the GKB process is loss of orthogonality in the vectors of $\mathbf{U}_{k+1}$ and $\mathbf{V}_k$, for which it may be necessary to use a reorthogonalization strategy to preserve convergence of the singular

values. Indeed, it has been shown that reorthogonalization of only one set of vectors is necessary [11, 215]; more details are provided in subsection 3.4.

As mentioned above, GKB (3.3) is related to the symmetric Lanczos decomposition. Notice that, multiplying the first equation in (3.3) from the left by $\mathbf{A}^\top$, and using the second equation in (3.3), we obtain

$$\mathbf{A}^\top\mathbf{A}\mathbf{V}_k = \mathbf{A}^\top\mathbf{U}_{k+1}\mathbf{B}_k = \mathbf{V}_{k+1}\mathbf{B}_{k+1,k+1}^\top\mathbf{B}_k = \mathbf{V}_k \underbrace{\mathbf{B}_k^\top\mathbf{B}_k}_{=:\hat{\mathbf{T}}_{k,k}} + \alpha_{k+1}\beta_{k+1}\mathbf{v}_{k+1}\mathbf{e}_k^\top.$$

The above chain of equalities is the symmetric Lanczos algorithm applied to $\mathbf{A}^\top\mathbf{A}$ and $\mathbf{A}^\top\mathbf{b}$, generating an orthonormal basis for $\mathcal{K}_{k+1}(\mathbf{A}^\top\mathbf{A}, \mathbf{A}^\top\mathbf{b})$. Similarly, multiplying the second equation in (3.3) (written in terms of $\mathbf{U}_k$) on the left by $\mathbf{A}$, and using the first equation in (3.3), we obtain

$$\mathbf{A}\mathbf{A}^\top\mathbf{U}_k = \mathbf{A}\mathbf{V}_k\mathbf{B}_{k,k}^\top = \mathbf{U}_k \underbrace{\mathbf{B}_{k,k}\mathbf{B}_{k,k}^\top}_{=:\mathbf{T}_{k,k}} + \alpha_k\beta_{k+1}\mathbf{u}_{k+1}\mathbf{e}_k^\top. \tag{3.4}$$

The above chain of equalities is the symmetric Lanczos algorithm applied to $\mathbf{A}\mathbf{A}^\top$ and $\mathbf{b}$, generating an orthonormal basis for $\mathcal{K}_{k+1}(\mathbf{A}\mathbf{A}^\top, \mathbf{b})$.

Comparison of Solution Subspaces. Although the choice of a projection method relies heavily on the problem being solved, there have been some investigations into which projection methods might be better suited for certain types of problems, especially in light of some recent work on flexible methods, general-form Tikhonov regularization, and matrix transpose approximation. We remark that the discussion here also relates to the material in subsection 3.4.

For Arnoldi based methods, which build the approximation subspace starting from $\mathbf{b}$, a potential concern is the explicit presence of the (rescaled) noise vector among the basis vectors [148]. For problems with very low noise level, this is not a significant drawback [38]. However, various methods have been developed to remedy this. For example, the symmetric range-restricted minimum residual method MR-II [118] and the range-restricted GMRES (RRGMRES) method [37] discard $\mathbf{b}$ and seek solutions in the Krylov subspace $\mathcal{K}_k(\mathbf{A}, \mathbf{A}\mathbf{b})$. For various projection methods, Hansen and Jensen [131] studied the propagation of noise in both the solution subspace and the reconstruction, noting the manifestation of band-pass filtered white noise as 'freckles' in the reconstructions.

For both the Arnoldi and the GKB process, notice that each additional vector in the Krylov solution subspace is generated by matrix-vector multiplication with $\mathbf{A}$ or $\mathbf{A}^\top\mathbf{A}$ respectively. Therefore the vectors defining the Krylov solution space are iteration vectors of the power method for computing the largest eigenpair of a matrix, and hence they become increasingly richer in the direction of the dominant eigenvector of $\mathbf{A}$ or $\mathbf{A}^\top\mathbf{A}$. For many inverse problems (and due to the discrete Picard condition), the orthonormal basis vectors generated in early iterations tend to carry important information about the low-frequency components (i.e., the right singular vectors that correspond to the large singular values) [126]. Therefore, the Krylov subspaces considered in this section can be appropriate choices for use in hybrid projection methods.

It is important to note that the choice of Krylov subspace may be problem dependent. For some image deblurring problems, it has been observed that Arnoldi-based methods may not be suitable since multiplication with $\mathbf{A}$ corresponds to recursive blurring of the observation, resulting in a poor solution subspace. For tomography

applications where the coefficient matrix is often rectangular, it is natural to consider Golub-Kahan based methods; even if the matrix $\mathbf{A}$ modeling a tomographic acquisition problem happens to be square (because of a specific scanning geometry), it should be stressed that standard Arnoldi methods would not generate a meaningful approximation subspace for the solution. This is because multiplication with $\mathbf{A}$ maps images to sinograms, and subsequent mappings to sinograms are neither physically meaningful nor fit for image reconstruction.

Various modifications based on 'preconditioning' of the Krylov methods introduced so far have been proposed: namely, a 'preconditioner' $\widetilde{\mathbf{A}}$ is introduced and standard iterative methods are applied to left- or right-preconditioned problems,

$$\widetilde{\mathbf{A}}\mathbf{A}\mathbf{x} = \widetilde{\mathbf{A}}\mathbf{b} \quad \text{or} \quad \mathbf{A}\widetilde{\mathbf{A}}\bar{\mathbf{y}} = \mathbf{b} \quad \text{with} \quad \mathbf{x} = \widetilde{\mathbf{A}}\bar{\mathbf{y}}\,, \tag{3.5}$$

respectively. Notice that left preconditioning is less commonly used when solving inverse problems, as it also acts on $\mathbf{b}$ and may change the noise statistics of the problem; both left and right preconditioning change the Krylov approximation subspace for the solution and smart, problem-specific choices of $\widetilde{\mathbf{A}}$ can result in efficient methods that perform much better than their unpreconditioned counterparts. For example, for image deblurring problems with nonsymmetric blurs and anti-reflective boundary conditions, a computationally efficient preconditioner $\widetilde{\mathbf{A}} \approx \mathbf{A}^\top$ representing a low-pass filter and enforcing a more symmetric blur, was used to improve the quality of the computed solution when using Arnoldi-Tikhonov methods [68]. In tomography, in order to perform faster computations, one may naturally consider systems of the kind (3.5), where $\mathbf{A}$ and $\widetilde{\mathbf{A}} \approx \mathbf{A}^\top$ constitute an unmatched pair of forward and back projector; see [129], where formulations (3.5) are analysed in the framework of the so-called BA- and AB-GMRES methods.

It should be noted that preconditioning ill-posed problems can be a tricky business [122]. Indeed, besides improving the solution subspace, preconditioning can also be cautiously used with the classical goal of accelerating convergence (provided that it does not exacerbate the semi-convergence phenomenon). Sometimes additional vectors (that are hopefully meaningful to recover known features of the solution) can be added to the approximation subspace, where the goal is not necessarily to accelerate convergence, but rather to improve the solution subspace. This is the idea behind the so-called enriched or recycling methods (see, for instance, [69, 128]); these are described in more detail in subsection 4.2.

In Figure 3.2, we show a few of the basis vectors for the solution generated by the Arnoldi and the GKB algorithms applied to the image deblurring and the tomography reconstruction problems described in subsection 1.1. The displayed images are reshaped columns of $\mathbf{V}_k$. For the deblurring example, we observe that, as expected, the first basis vector for the GKB process (i.e., $\mathbf{A}^\top\mathbf{b}/\|\mathbf{A}^\top\mathbf{b}\|$) is smoother or more blurred than that from the Arnoldi process (i.e., $\mathbf{b}/\|\mathbf{b}\|$). However, we notice that the 4th basis vector computed by Arnoldi may have more details but is more noisy than that for GKB. It is interesting to note that, although any vector $\mathbf{v}_k$ spans a Krylov subspace that has been obtained from the previous one by adding an increasingly blurred vector of the form $\mathbf{A}\mathbf{v}_{k-1}$ (for Arnoldi, see Algorithm 3.2) or $\mathbf{A}^\top\mathbf{A}\mathbf{v}_{k-1}$ (for GKB, see Algorithm 3.3), a linear combination of the columns of $\mathbf{V}_k$ results in a restored image where the blur has been removed: this is because Krylov methods compute polynomial (and regularized) approximations to a solution of (2.4). We refer to [125, Chapter 6], [146], and the references therein for more details and insight. For the tomography example, we provide some of the GKB solution basis vectors, for

746 which we note that the basis vector for the first iteration is the scaled vector $\mathbf{A}^\top \mathbf{b}$, which can be interpreted as an unfiltered backprojection image; see [174].

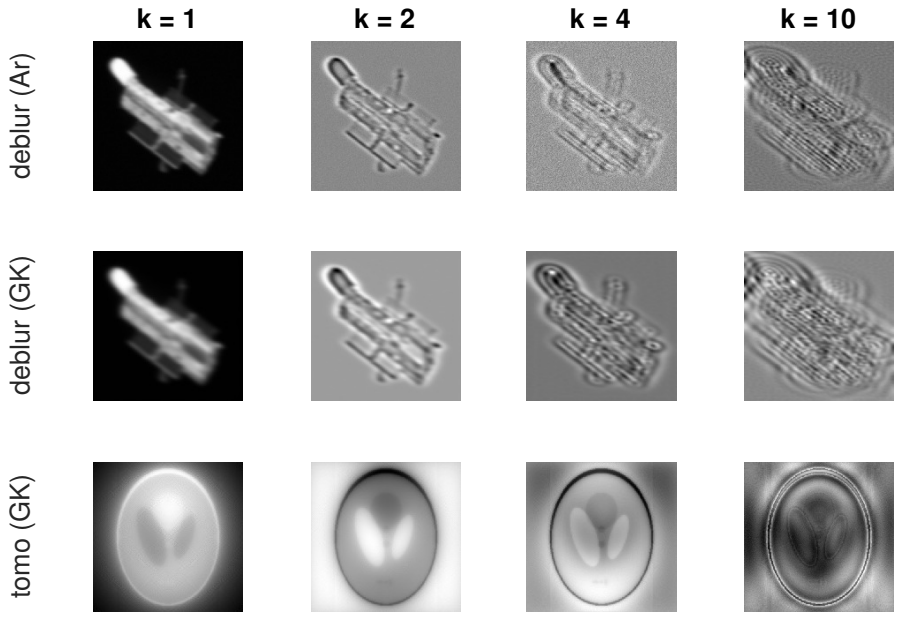


FIG. 3.2. *For the deblurring example and the tomography example, we provide basis images (i.e., the basis vectors $\mathbf{v}_k$ for the solution that have been reshaped into images) for iterations $k = 1, 2, 4, 10$. Notice that, since the coefficient matrix in the deblurring example is square, we can compare the Arnoldi and GKB basis vectors, while for the tomography example, we consider only GKB approaches.*

747

748 **3.2.2. Solving the regularized, projected problem.** In subsection 3.2.1, we
749 described two iterative projection methods that can be used to generate and expand
750 the projection subspace $\text{ran}(\mathbf{V}_k)$ (step 2 in Algorithm 3.1). Although projection
751 onto subspaces of increasing dimension (e.g., Krylov subspaces) can have an inherent
752 regularizing effect (recall the discussion in subsection 2.2), a key component of hybrid
753 projection methods is the combination of iterative and variational regularization, i.e.,
754 the inclusion of a regularization term within the projected problem. We remark that
755 Steps 3 and 4 in Algorithm 3.1 are closely intertwined and could easily be addressed
756 together. However, for clarity of presentation, methods to select the regularization
757 parameter λ_k and the stopping iteration k will be addressed in detail in subsection 3.3,
758 and this section will focus on solving the projected regularized problem (3.1). Notice
759 that at step 4 of Algorithm 3.1, we have computed $\mathbf{V}_k$ and λ_k, and the stopping
760 criterion is not yet satisfied.
761 Computing a solution to (3.1) can be done efficiently by exploiting components
762 and relationships from the projection process ((3.2) for Arnoldi and (3.3) for GKB).
763 In this section, we denote the solution subspaces for Arnoldi and Golub-Kahan as
764 $\mathbf{V}_k^{\text{Ar}}$ and $\mathbf{V}_k^{\text{GK}}$ respectively. Recall that for a matrix $\mathbf{V}_k$ that contains orthonormal
765 columns, $\mathbf{V}_k^\top \mathbf{V}_k = \mathbf{I}_k$ and $\|\mathbf{V}_k \mathbf{y}\| = \|\mathbf{y}\|$ for any $\mathbf{y} \in \mathbb{R}^k$.

The *hybrid GMRES iterate* at the kth iteration is given by

$$\begin{aligned}\mathbf{x}_k(\lambda_k) = \mathbf{V}_k^{\mathrm{Ar}}\mathbf{y}_k(\lambda_k), \quad \text{where} \quad \mathbf{y}_k(\lambda_k) &= \arg\min_{\mathbf{y}\in\mathbb{R}^k} \|\mathbf{b} - \mathbf{A}\mathbf{V}_k^{\mathrm{Ar}}\mathbf{y}\|^2 + \lambda_k \|\mathbf{y}\|^2 \\ &= \arg\min_{\mathbf{y}\in\mathbb{R}^k} \left\|\mathbf{V}_{k+1}^{\mathrm{Ar}} \|\mathbf{b}\| \mathbf{e}_1 - \mathbf{V}_{k+1}^{\mathrm{Ar}}\mathbf{H}_k\mathbf{y}\right\|^2 + \lambda_k \|\mathbf{y}\|^2 \\ &= \arg\min_{\mathbf{y}\in\mathbb{R}^k} \|\|\mathbf{b}\| \mathbf{e}_1 - \mathbf{H}_k\mathbf{y}\|^2 + \lambda_k \|\mathbf{y}\|^2 . \end{aligned} \tag{3.6}$$

Since $\mathbf{V}_k^{\mathrm{Ar}}$ contains orthonormal columns, we can equivalently write

$$\mathbf{x}_k(\lambda_k) = \arg\min_{\mathbf{x}\in\mathrm{ran}(\mathbf{V}_k^{\mathrm{Ar}})} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2 + \lambda_k \|\mathbf{x}\|^2 = \mathbf{V}_k^{\mathrm{Ar}}\mathbf{y}_k(\lambda_k) .$$

Oftentimes, the hybrid GMRES approach is referred to as an Arnoldi-Tikhonov approach [38, 94, 166].

Similar derivations can be done for the Golub-Kahan projection method. In particular, the *hybrid LSQR iterate* is computed as

$$\begin{aligned}\mathbf{x}_k(\lambda_k) = \mathbf{V}_k^{\mathrm{GK}}\mathbf{y}_k(\lambda_k), \quad \text{where} \quad \mathbf{y}_k(\lambda_k) &= \arg\min_{\mathbf{y}\in\mathbb{R}^k} \|\mathbf{b} - \mathbf{A}\mathbf{V}_k^{\mathrm{GK}}\mathbf{y}\|^2 + \lambda_k \|\mathbf{y}\|^2 \\ &= \arg\min_{\mathbf{y}\in\mathbb{R}^k} \|\beta_1\mathbf{e}_1 - \mathbf{B}_k\mathbf{y}\|^2 + \lambda_k \|\mathbf{y}\|^2 \end{aligned} \tag{3.7}$$

or, equivalently,

$$\mathbf{x}_k(\lambda_k) = \arg\min_{\mathbf{x}\in\mathrm{ran}(\mathbf{V}_k^{\mathrm{GK}})} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2 + \lambda_k \|\mathbf{x}\|^2 = \mathbf{V}_k^{\mathrm{GK}}\mathbf{y}_k(\lambda_k) .$$

The *hybrid LSMR iterate* [60] is computed as

$$\begin{aligned}\mathbf{x}_k(\lambda_k) = \mathbf{V}_k^{\mathrm{GK}}\mathbf{y}_k(\lambda_k), \quad \text{where} \quad \mathbf{y}_k(\lambda_k) &= \arg\min_{\mathbf{y}\in\mathbb{R}^k} \left\|\mathbf{A}^\top(\mathbf{b} - \mathbf{A}\mathbf{V}_k^{\mathrm{GK}}\mathbf{y})\right\|^2 + \lambda_k \|\mathbf{y}\|^2 \\ &= \arg\min_{\mathbf{y}\in\mathbb{R}^k} \left\|\bar{\beta}_1\mathbf{e}_1 - \begin{bmatrix}\mathbf{B}_k^\top\mathbf{B}_k \\ \bar{\beta}_{k+1}\mathbf{e}_k^\top\end{bmatrix}\mathbf{y}\right\|^2 + \lambda_k \|\mathbf{y}\|^2 , \end{aligned} \tag{3.8}$$

where $\bar{\beta}_k = \alpha_k\beta_k$.

Table 3.1 provides a summary of common methods with their defining subspace and corresponding subproblem.

TABLE 3.1
Solution subspaces $\mathbf{V}_k$ and corresponding subproblems defining $\mathbf{y}_k(\lambda_k)$ for different hybrid projection methods.

method	**subspace**	**subproblem**
hybrid GMRES	Arnoldi, $\mathrm{ran}(\mathbf{V}_k^{\mathrm{Ar}})$	$\min \|\beta_1\mathbf{e}_1 - \mathbf{H}_k\mathbf{y}\|^2 + \lambda_k \|\mathbf{y}\|^2$
hybrid LSQR	Golub-Kahan, $\mathrm{ran}(\mathbf{V}_k^{\mathrm{GK}})$	$\min \|\beta_1\mathbf{e}_1 - \mathbf{B}_k\mathbf{y}\|^2 + \lambda_k \|\mathbf{y}\|^2$
hybrid LSMR	Golub-Kahan, $\mathrm{ran}(\mathbf{V}_k^{\mathrm{GK}})$	$\min \left\|\bar{\beta}_1\mathbf{e}_1 - \begin{bmatrix}\mathbf{B}_k^\top\mathbf{B}_k \\ \bar{\beta}_{k+1}\mathbf{e}_k^\top\end{bmatrix}\mathbf{y}\right\|^2 + \lambda_k \|\mathbf{y}\|^2$

One important property to highlight is that, for these methods, the generated subspace is independent of the choice of the regularization parameter λ_k. This is

not always true (e.g., in the flexible methods presented in subsection 4.3). A very desirable consequence of this is that one can avoid computing the regularized solution at each iteration. For many of the regularization parameter selection methods, the choice of the parameter λ_k at each iteration does not depend on previous or later iterates. Thus, it is possible to delay regularization parameter selection and solution computation until a solution is needed or some stopping criterion is satisfied.

Next, we draw some connections and distinctions to standard iterative methods for least-squares problems (i.e., for the case where $\lambda_k = 0$ in subsection 2.2 or $\lambda_k = \lambda$, fixed along the iterations, in subsection 2.3). For problems where no regularization is imposed on the projected problem (i.e., $\lambda_k = 0$), we recover standard iterative methods GMRES for Arnoldi (solving (3.6)) and LSQR for GKB (solving (3.7)).

For hybrid projection methods, a potential concern is the need to store all of the basis vectors $\mathbf{V}_k$. For Arnoldi based methods for solving Tikhonov regularized problems, this requirement is the same as for standard iterative methods. However, for Golub-Kahan based methods, this potential caveat warrants a discussion. Indeed, it is well-known that the LSQR iterates can be computed efficiently using a three-term-recurrence property by exploiting an efficient QR factorization of $\mathbf{B}_k$. As shown in [185], such computational efficiencies can also be exploited for standard Tikhonov regularization if λ is fixed a priori. This can be done by exploiting the fact that the Krylov subspace is shift invariant with respect to λ. However, semi-convergence will be an issue if $\lambda = 0$ and the entire process must be restarted from scratch if a different λ is desired. This computational flaw is even more severe when using iterative methods such as LSQR for general-form Tikhonov regularization, since the solution subspace typically depends on the regularization parameter and the regularization matrix (although some strategies to ease this dependence are described in subsection 4.1). For hybrid projection methods based on GKB, storage of the basis vectors is the main additional cost associated with the ability to select λ adaptively during the hybrid projection procedure. For problems that require many iterations, potential remedies include developing a good preconditioner, compression and/or augmentation techniques [152].

3.2.3. A unifying framework. All the methods described so far can be expressed by this partial factorization,

$$\mathbf{A}\mathbf{V}_k = \mathbf{U}_{k+1}\mathbf{G}_k\,, \tag{3.9}$$

where the columns of $\mathbf{V}_k$ are orthonormal and span the k-dimensional approximation subspace for the solution, the columns of $\mathbf{U}_{k+1}$ are orthonormal with $\mathbf{u}_1 = \mathbf{b}/\left\|\mathbf{b}\right\|$ and span a $(k+1)$-dimensional subspace (e.g., associated to $\mathbf{A}^\top$), and $\mathbf{G}_k$ is a $(k+1)\times k$ matrix that has some structure and represents the projected problem. For hybrid GMRES, $\mathbf{V}_k = \mathbf{V}_k^{\text{Ar}}$, $\mathbf{U}_{k+1} = \mathbf{V}_{k+1} = \mathbf{V}_{k+1}^{\text{Ar}}$ and $\mathbf{G}_k = \mathbf{H}_k$ are given in (3.2). For hybrid LSQR and hybrid LSMR, $\mathbf{V}_k = \mathbf{V}_k^{\text{GK}}$, $\mathbf{G}_k = \mathbf{B}_k$, and $\mathbf{U}_{k+1}$ are given in (3.3). Let the SVD of the matrix $\mathbf{G}_k$ be given by

$$\mathbf{G}_k = \mathbf{U}^{\text{G}}\mathbf{\Sigma}^{\text{G}}(\mathbf{V}^{\text{G}})^\top, \tag{3.10}$$

where $\mathbf{U}^{\text{G}} \in \mathbb{R}^{k+1\times k+1}$ and $\mathbf{V}^{\text{G}} \in \mathbb{R}^{k\times k}$ are orthogonal, and $\mathbf{\Sigma}^{\text{G}} = \text{diag}(\sigma_1^{\text{G}}, \ldots, \sigma_k^{\text{G}}) \in \mathbb{R}^{(k+1)\times k}$ contains the singular values of $\mathbf{G}_k$. Note that, for notational convenience, we have dropped the subscript k.

For hybrid GMRES and hybrid LSQR with standard Tikhonov regularization,

the solution to the regularized projected problems (3.6) and (3.7) have the form,

$$\begin{aligned} \mathbf{y}_k(\lambda_k) &= \arg\min_{\mathbf{y}} \|\mathbf{b} - \mathbf{A}\mathbf{V}_k\mathbf{y}\|^2 + \lambda_k \|\mathbf{y}\|^2 \\ &= \arg\min_{\mathbf{y}} \left\|\mathbf{U}_{k+1}^\top \mathbf{b} - \mathbf{G}_k\mathbf{y}\right\|^2 + \lambda_k \|\mathbf{y}\|^2 , \end{aligned} \tag{3.11}$$

and thus one may express the kth iterate of the hybrid projection method as

$$\mathbf{x}_k(\lambda_k) = \underbrace{\mathbf{V}_k(\mathbf{G}_k^\top\mathbf{G}_k + \lambda_k\mathbf{I}_k)^{-1}\mathbf{G}_k^\top\mathbf{U}_{k+1}^\top}_{=:\mathbf{A}_{\text{reg}}^\dagger(\lambda_k,k)}\mathbf{b}. \tag{3.12}$$

Note that $\mathbf{U}_{k+1}^\top\mathbf{b} = \|\mathbf{b}\|\mathbf{e}_1$ by construction.

For many of the regularization parameter selection methods described in subsection 3.3 and the theoretical results in subsection 3.4, it will be helpful to define the so-called 'influence matrix',

$$\mathbf{A}\mathbf{A}_{\text{reg}}^\dagger(\lambda_k, k) = \mathbf{U}_{k+1}\mathbf{G}_k(\mathbf{G}_k^\top\mathbf{G}_k + \lambda_k\mathbf{I}_k)^{-1}\mathbf{G}_k^\top\mathbf{U}_{k+1}^\top . \tag{3.13}$$

For purely iterative methods (where $\lambda_k = 0$), the influence matrix is given by

$$\mathbf{A}\mathbf{A}_{\text{reg}}^\dagger(k) = \mathbf{U}_{k+1}\mathbf{G}_k\mathbf{G}_k^\dagger\mathbf{U}_{k+1}^\top , \quad \text{where} \quad \mathbf{A}_{\text{reg}}^\dagger(k) = \mathbf{V}_k\mathbf{G}_k^\dagger\mathbf{U}_{k+1}^\top.$$

Also, it will be helpful to define the so-called 'discrepancy' (or 'regularized residual') at the kth iteration as

$$\mathbf{r}(\mathbf{x}_k(\lambda_k)) = \mathbf{b} - \mathbf{A}\mathbf{x}_k(\lambda_k) = (\mathbf{I}_m - \mathbf{A}\mathbf{A}_{\text{reg}}^\dagger(\lambda_k, k))\mathbf{b} , \tag{3.14}$$

for hybrid methods; the same definition applies to iterative methods, with $\lambda_k = 0$. In the following, we will adopt the notation $\mathbf{A}_{\text{reg}}^\dagger(\lambda)$ and $\mathbf{r}(\mathbf{x}(\lambda))$ to denote the regularized inverse and the discrepancy associated to (direct) Tikhonov regularization, coherently to (3.12) and (3.14), respectively. When no regularization method is specified, $\mathbf{x}_{\text{reg}}$, $\mathbf{A}_{\text{reg}}^\dagger$ and $\mathbf{r}_{\text{reg}}$ denote a generic regularized solution, inverse, and residual, respectively.

3.3. Regularization parameter selection methods. The success of any regularization method depends on the choice of one (or more) regularization parameter(s). This was illustrated in section 2 for Tikhonov regularization and for iterative regularization methods; also, when the Tikhonov-regularized problem is solved using an iterative method, both the Tikhonov regularization parameter and the number of iterations should be accurately tuned (see subsection 2.3). Similarly, for hybrid projection methods, there are inherently two regularization parameters to tune: (1) the number of iterations k (i.e., the dimension of the projection subspace), and (2) the regularization parameter for the projected problem (e.g., λ_k for Tikhonov). It is important to note that, when carelessly applied to hybrid projection methods where *both* regularization parameters must be determined, standard regularization parameter choice strategies may not produce good results. Instead, a two-pronged approach is typically adopted, where a well-established parameter choice strategy is used to determine λ_k and one or more stopping rules are used to terminate the iteration, typically by monitoring the stabilization of some relevant quantities; see, for instance, [34, 59, 91, 160, 198].

In subsection 3.3.1 we comment on a few factors that are unique to selecting regularization parameters in a hybrid projection method, and that should be considered

TABLE 3.2
Summary of regularization parameter choice rules described in subsection 3.3.

assumption	**method**
requires noise magnitude (subsection 3.3.2)	discrepancy principle (DP)
	unbiased predictive risk estimation (UPRE)
does not require noise magnitude (subsection 3.3.3)	L-curve
	generalized cross-validation (GCV)
	normalized cumulative periodogram (NCP)

when determining which parameter selection methods to employ. Parameter choice rules represent a large and still growing body of literature in the field of inverse problems, with papers ranging from theoretical developments of regularization methods to papers focused on methods specific to applications. Although there are extensive survey papers describing parameter choice methods in the continuous setting (see, for instance, [12, 13, 191]), we focus on parameter selection strategies for discrete inverse problems that have proven successful in conjunction with hybrid projection methods. We describe two main classes of parameter choice methods: (1) those that require knowledge of the noise magnitude in subsection 3.3.2 and (2) those that do not require knowledge of the noise magnitude in subsection 3.3.3. These are also listed in Table 3.2. In almost all the considered strategies for hybrid methods, there are common quantities that must be monitored. These include:

1. the norm of the approximate solution, $\|\mathbf{x}_k(\lambda_k)\|$,
2. the norm of the residual, $\|\mathbf{r}(\mathbf{x}_k(\lambda_k))\|$, and
3. the trace of the influence matrix, $\operatorname{trace}\big(\mathbf{A}\mathbf{A}^{\dagger}_{\text{reg}}(\lambda_k, k)\big)$.

Computing these quantities can be done very efficiently by monitoring the corresponding projected quantities and exploiting the orthogonal invariance of the 2-norm. This fact along with the relatively cheap computation of (3.10) make the adaptation of standard parameter choice strategies particularly appealing in the setting of hybrid methods. Simplified formulations for these quantities using the SVD of the projected matrix can be found in Appendix A.

Appendix B unfolds some connections between specific parameter choice rules for (unprojected) Tikhonov regularization (2.1) and corresponding strategies for hybrid methods. These leverage the fact that many parameter choice rules for setting λ in (2.1) involve at least one evaluation of a quadratic form of the matrix $\mathbf{A}^\top\mathbf{A}$ or $\mathbf{A}\mathbf{A}^\top$, which can be expressed as particular Riemann-Stieltjes integrals (written in terms of the SVD of $\mathbf{A}$). When dealing with large matrices $\mathbf{A}$ whose SVD is not available, the latter can be approximated by Gauss or Gauss-Radau quadrature rules, which are related to the symmetric Lanczos and the GKB algorithms; see [101] for full explanations of these relations. Exploiting these links is crucial to provide computationally affordable approximations of the original quadratic forms in terms of quantities (such as the upper bidiagonal matrix $\mathbf{B}_k$) generated at the kth iteration of GKB-based hybrid methods that, under some assumptions, can be proven to provide upper or lower bounds for λ that get increasingly accurate as k increases.

3.3.1. General considerations on selecting λ_k and k.

Selecting λ_k. Assume that k is fixed and consider the case of standard form Tikhonov regularization. First, for standard Krylov subspace projection methods including those described in subsection 3.2.1, the projection subspace is independent of the current value of the Tikhonov regularization parameter. This is due to the

so-called 'shift-invariance' property of the computed Krylov subspaces, and this has the effect that the approximate solution at the kth iteration only depends on the current regularization parameter λ_k. Second, notice that, for a fixed k, the projected problem (3.11) is a standard Tikhonov-regularized problem. A natural idea would be to directly apply well-established (e.g, SVD-based) parameter choice strategies that were developed for Tikhonov regularization; however, care must be taken to ensure accurate results in the hybrid setting. Third, since the number of iterations k is significantly smaller than the size of the original problem (i.e., $k \ll \min\{m, n\}$) and the projected problem has order k (i.e., the coefficient matrix $\mathbf{G}_k$ has size $(k+1) \times k$), computations with $\mathbf{G}_k$ can be performed easily. For many methods, there are computational advantages to using the SVD of $\mathbf{G}_k$ (3.10). Indeed, for most problems, the computational cost of obtaining (3.10) is negligible compared to the computational cost of performing matrix-vector products with $\mathbf{A}$ (and possibly $\mathbf{A}^\top$) to expand the approximation subspace. We will see how some common parameter choice methods can be efficiently formulated using the SVD.

Selecting k. Various rules have been developed for selecting the stopping iteration k, and these rules can be employed quite generally (and independently of the parameter choice rule for λ_k). The main idea is to terminate iterations when a maximum number of iterations is achieved or when one or more of the following conditions are satisfied:

$$\begin{array}{lll} |\lambda_k - \lambda_{k-1}| & < & \tau_\lambda \lambda_{k-1}\,, \\ \|\mathbf{r}(\mathbf{x}_k(\lambda_k)) - \mathbf{r}(\mathbf{x}_{k-1}(\lambda_{k-1}))\| & < & \tau_r \|\mathbf{r}(\mathbf{x}_{k-1}(\lambda_{k-1}))\|\,, \\ \|\mathbf{x}_k(\lambda_k) - \mathbf{x}_{k-1}(\lambda_{k-1})\| & < & \tau_x \|\mathbf{x}_{k-1}(\lambda_{k-1})\|\,, \end{array} \tag{3.15}$$

where τ_λ, τ_r, τ_x are positive user-specified thresholds. The rationale behind these approaches is that, often and broadly speaking, when stabilization happens, the selected value for the regularization parameter is suitable for the full-dimensional Tikhonov problem and the approximated solution cannot significantly improve with more iterations. Although this argument is mainly empirical, in some cases it is supported by theoretical results; see Appendix B. This property is also heavily exploited in [56, 106, 197], where the regularization parameter (and other relevant quantities) for the original Tikhonov problem are estimated by projecting the original problem onto subspaces of smaller dimension. Once the regularization parameter is estimated, it is fixed and any iterative method can be used to solve the resulting Tikhonov problem; therefore, these methods are not considered hybrid projection methods as defined in this manuscript; see section 1 and subsection 2.3. The left frame of Figure 3.1 displays the typical behavior of adaptively chosen regularization parameters where, relevant to the first criterion in (3.15), some stabilization is visible as the iterations proceed. Returning to the issue of selecting a stopping iteration for hybrid projection methods, we remark that, with a suitable choice of λ_k, hybrid methods can overcome semi-convergent behavior, as illustrated in Figure 2.2. Thus, an imprecise (over-)estimate of the stopping iteration does not significantly degrade the reconstruction quality. In fact, one can typically afford a few more iterations without experiencing deterioration of the solution (on the contrary, the solution may improve because it is computed by solving a well-posed problem in a larger approximation subspace). This insight is also linked to the ability of the considered Krylov projection methods to 'capture' the dominant (i.e., relevant) truncated right singular vector subspace information to reconstruct the solution; we present more details about this in subsection 3.4.

3.3.2. Methods that require knowledge of the noise magnitude. The *discrepancy principle* selects a regularized solution $\mathbf{x}_{\text{reg}}$ satisfying

$$\|\mathbf{b} - \mathbf{A}\mathbf{x}_{\text{reg}}\| = \eta\varepsilon\,, \tag{3.16}$$

where ε is an estimate of the norm of the noise $\|\mathbf{e}\|$ and $\eta > 1$ is a safety factor (the larger η, the more uncertainty on ε). Note that if the noise $\mathbf{e}$ is assumed Gaussian with covariance matrix $\sigma^2\mathbf{I}_m$, i.e., when

$$\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_m)\,, \tag{3.17}$$

then the expected value of $\|\mathbf{e}\|^2$ is $\mathbb{E}(\|\mathbf{e}\|^2) = m\sigma^2$. The discrepancy principle typically works well if a good estimate of ε is available. For Tikhonov regularization (subsection 2.1), $\mathbf{x}_{\text{reg}} = \mathbf{x}(\lambda)$ and (3.16) is a nonlinear equation in $\lambda > 0$ (that should be solved by employing a zero-finder, typically of the Newton family [193]); for iterative regularization methods (subsection 2.2), $\mathbf{x}_{\text{reg}} = \mathbf{x}_k$ and the iterative method should stop as soon as the kth iterate satisfies

$$\|\mathbf{b} - \mathbf{A}\mathbf{x}_k\| \leq \eta\varepsilon\,. \tag{3.18}$$

Using the discrepancy principle, one can typically prove regularization properties such as $\mathbf{x}_{\text{reg}} \to \mathbf{x}_{\text{true}}$ as $\|\mathbf{e}\| \to 0$; see [40, 73, 119].

When considering hybrid solvers, the most common approach is to solve the nonlinear equation,

$$\|\mathbf{b} - \mathbf{A}\mathbf{x}_k(\lambda_k)\| = \|\mathbf{U}_{k+1}^\top\mathbf{b} - \mathbf{G}_k\mathbf{y}_k(\lambda_k)\| = \eta\varepsilon \tag{3.19}$$

to determine the Tikhonov regularization parameter λ_k to be employed at the kth iteration. Notice that once the discrepancy principle is satisfied (typically after a few iterations are performed), one may stop the iterative process. However, it has been observed that the quality of the solutions improves if more iterations are performed, as the regularized solutions belong to richer approximation subspaces. This phenomenon is described at length in [166, 192]. In this case, one may resort to one (or more) stopping criterion of the form (3.15). Another approach is to update the regularization parameter for the projected problem in such a way that stopping by the discrepancy principle is ensured [86, 91]. That is, the discrepancy principle can be used for setting both k and λ_k, but only one iteration of a root-finder algorithm for (3.19) is applied at each iteration of a hybrid method. The approach is derived in [91] and its underlying theory is analyzed in [86], by leveraging the tools described in more details in Appendix B. Note that, thanks to the first equality in (3.19), when applying the discrepancy principle within hybrid methods, one can exclusively perform computations with projected quantities, which are cheap when $k \ll \min\{m, n\}$ (see Appendix A for more details).

UPRE (*unbiased predictive risk estimation*) selects the regularization parameter that minimizes the expectation of the mean squared norm of the predictive error, $\mathbb{E}(\|\mathbf{p}(\mathbf{x}_{\text{reg}})\|^2)$, associated to the regularized solution $\mathbf{x}_{\text{reg}} = \mathbf{A}_{\text{reg}}^\dagger\mathbf{b}$, where the predictive error is defined as

$$\mathbf{p}(\mathbf{x}_{\text{reg}}) = \mathbf{A}\mathbf{x}_{\text{reg}} - \mathbf{b}_{\text{true}} = \mathbf{A}\mathbf{A}_{\text{reg}}^\dagger\mathbf{b} - \mathbf{b}_{\text{true}} = (\mathbf{A}\mathbf{A}_{\text{reg}}^\dagger - \mathbf{I}_n)\mathbf{b}_{\text{true}} + \mathbf{A}\mathbf{A}_{\text{reg}}^\dagger\mathbf{e}\,,$$

where $\mathbf{A}\mathbf{A}_{\text{reg}}^\dagger$ is assumed to be symmetric (this is the case for all the regularization methods considered so far). When considering (direct) Tikhonov regularization and

when $\mathbf{e}$ represents Gaussian white noise with standard deviation σ (3.17), one should compute the regularization parameter to be the minimizer of

$$\mathbb{E}\|\mathbf{p}(\mathbf{x}(\lambda))\|^2 = \|(\mathbf{A}\mathbf{A}^\dagger_{\text{reg}}(\lambda) - \mathbf{I}_m)\mathbf{b}_{\text{true}}\|^2 + \sigma^2 \text{trace}\big((\mathbf{A}\mathbf{A}^\dagger_{\text{reg}}(\lambda))^\top(\mathbf{A}\mathbf{A}^\dagger_{\text{reg}}(\lambda))\big) \ .$$

To circumvent the fact that the first term in the above expression of $\mathbf{p}(\mathbf{x}(\lambda))$ is in practice unavailable, one can perform some algebraic manipulations and approximations. The UPRE parameter is then given by

$$\lambda^* = \arg\min_{\lambda\in\mathbb{R}_+} \underbrace{\|\mathbf{r}(\mathbf{x}(\lambda))\|^2 + 2\sigma^2 \text{trace}\big(\mathbf{A}\mathbf{A}^\dagger_{\text{reg}}(\lambda)\big) - m\sigma^2}_{=:\,U(\lambda)} \ . \tag{3.20}$$

Note that the expected value of $U(\lambda)$ is the expected value of the predictive risk, i.e., $\mathbb{E}(U(\lambda)) = \mathbb{E}(\|\mathbf{p}(\mathbf{x}(\lambda))\|^2)$. We refer to [234] for more details on the derivation of the UPRE method. UPRE can also be used as a stopping rule for iterative methods, e.g., for nonnegatively constrained Poisson inverse problems [9]. When considering hybrid methods, applying UPRE to the projected problem is quite straightforward, and was first considered in [198]: it is essentially a matter of replacing $\mathbf{r}(\mathbf{x}(\lambda))$ and $\mathbf{A}^\dagger_{\text{reg}}(\lambda)$ in (3.20) by $\mathbf{r}(\mathbf{x}_k(\lambda_k))$ and $\mathbf{A}^\dagger_{\text{reg}}(\lambda_k, k)$, respectively. Since the influence matrix (3.13) is still symmetric, performing some algebraic manipulations leads to the following projected UPRE functional

$$\begin{aligned} U_k(\lambda) = \quad & \|(\mathbf{G}_k(\mathbf{G}_k^\top\mathbf{G}_k + \lambda\mathbf{I}_k)^{-1}\mathbf{G}_k^\top - \mathbf{I}_{k+1})\mathbf{U}_{k+1}^\top\mathbf{b}\|^2 \\ & + 2\sigma^2 \text{trace}\big(\mathbf{G}_k(\mathbf{G}_k^\top\mathbf{G}_k + \lambda\mathbf{I}_k)^{-1}\mathbf{G}_k^\top\big) - (k+1)\sigma^2 \, , \end{aligned}$$

which is minimized at each iteration of a hybrid method to find λ_k. Note that the functional $U_k(\lambda)$ is expressed with respect to projected quantities only, which are not computationally expensive to evaluate as far as $k \ll \min\{m, n\}$ (see Appendix A for more details). An interesting (and still partially open) question (common to other parameter choice strategies) is determining wether the regularization parameter λ_k so obtained is a good approximation of λ. To answer this question, the authors of [198] first consider the direct regularization method to be obtained by combining TSVD and Tikhonov regularization methods (sometimes referred to as 'FTSVD', i.e., filtered TSVD), so that the variational regularized solution belongs to the subspace spanned by the dominant right singular vectors; they deduce that, if the k-dimensional projection subspace generated by the hybrid method captures the relevant (i.e., dominant) spectral information about the original problem, then $\text{trace}\big(\mathbf{G}_k(\mathbf{G}_k^\top\mathbf{G}_k + \lambda\mathbf{I}_k)^{-1}\mathbf{G}_k^\top\big) \approx \text{trace}\big(\mathbf{A}\mathbf{A}^\dagger_{\text{reg}}\big)$ (the latter being specified for the FTSVD), and $\lambda_k \approx \lambda^*$.

We conclude this subsection by mentioning that, although it may seem to be a disadvantage that these methods require knowledge of the noise magnitude, there are actually various approaches for estimating the noise level from the data: here we describe a couple of them. A first approach uses statistical tools and performs quite well at estimating the variance [70]. Assuming Gaussian white noise (3.17), an estimate $\hat{\sigma}$ of the standard deviation σ can be obtained from the highest coefficients of the noisy data under some transformation (e.g., wavelet). For instance, given an observation vector $\mathbf{b}$, the following MATLAB code can be used to estimate the noise variance

```
[~, cD]   =  dwt(b,'db1');
sigmahat  =  median(abs(cD(:)))/.67; '
```

(3.21)

where the first line returns the detail coefficients of a single-level 1D discrete wavelet transform of $\mathbf{b}$ using the Haar wavelet and the second line computes an estimate of the variance. In many inverse problems, it is common to work with a relative noise level rather than the noise variance directly. For a simulated example, a noisy measurement can be generated as

$$\mathbf{b} = \mathbf{b}_{\text{true}} + \texttt{noiseLevel} * \frac{\|\mathbf{b}_{\text{true}}\|}{\|\mathbf{e}\|}\mathbf{e}\,,$$

where $\mathbf{e}$ is a realization from a standard Gaussian and the relative noise level is given in `noiseLevel`. Thus, an estimate of the relative `noiseLevel` obtained running (3.21) is $\hat{\sigma}\sqrt{m}/\,\|\mathbf{b}\|$.

Still assuming Gaussian white noise, a second approach to estimate relevant noise information (including $\|\mathbf{e}\|$) is to leverage some theoretical properties of the Krylov subspaces generated as projection subspaces for the solution: this is especially relevant in the context of hybrid regularization. To the best of our knowledge, the first detailed analysis of how the noise affects the approximation subspace generated by the GKB algorithm can be found in [139], where the authors show how to estimate at a negligible cost the (assumed) unknown amount of noise in the original data. By exploiting the connections between GKB and Gaussian quadrature rules (recalled in Appendix B), theoretical estimates prove that the norm of the residual computed by LSQR stabilizes around the noise magnitude. This can lead to the construction of stopping criteria for the bidiagonalization process as well as to the application of any of the parameter choice rules described in this section, using the information gathered about the noise. Extensions of the analysis in [139] to projection methods based on the Arnoldi algorithm can be found in [93].

3.3.3. Methods that do not require knowledge of the noise magnitude. In many situations, assuming that an accurate estimate of $\varepsilon = \|\mathbf{e}\|$ is available is unrealistic, and one cannot confidently apply the methods described in subsection 3.3.2. However, a number of strategies can be adopted when dealing with direct, iterative and hybrid methods: for the latter, most of these parameter choice rules can be regarded as the projected variants of their full-dimensional counterparts.

The *L-curve* criterion was popularized by [137]: it relies on the intuition that a good choice of the regularization parameter should balance the contribution of the so-called perturbation error (i.e., the error due to overfitting the noisy data, which would lead to an under-regularized solution) and regularization error (i.e., the error due to replacing the original problem with a related one, which would lead to an over-regularized solution). The L-curve is a plot of the norm of the regularized solution $\|\mathbf{x}_{\text{reg}}\|$ versus the norm of the regularized residual $\|\mathbf{r}_{\text{reg}}\|$ for varying values of the regularization parameter, and it is named after the desirable shape of its graph. When Tikhonov regularization is considered (i.e., when the L-curve is a parametric curve with respect to λ), one can prove that the L-curve is convex (see, for instance, [126, Chapter 4]). The ideally steep or vertical part of the curve corresponds to small amounts of regularization, so that such solutions are dominated by perturbation error. The ideally flat or horizontal part corresponds to too much regularization, so that such solutions are dominated by regularization error. Therefore the corner represents the point on the L-curve where both errors are balanced. The L-curve is commonly plotted in logarithmic scale, e.g., $(\log_{10}(\|\mathbf{r}_{\text{reg}}\|), \log_{10}(\|\mathbf{x}_{\text{reg}}\|))$, to better highlight the corner (and also because of the large range of values of the plotted quantities). For the L-curve to be effective it is necessary to have monotonicity in

$\|\mathbf{r}_{\text{reg}}\|$ and $\|\mathbf{x}_{\text{reg}}\|$ (this is not always the case, e.g., alternative L-curves have been devised for GMRES [38]). Moreover, it can be experimentally noted that, when the decay of the singular values, as well as the impact of noise on the computed solution happen slowly, the L-curve does not exhibit a distinct corner, making the L-curve criterion less successful; this is often the case for large-scale inverse problems. For hybrid projection methods, since we need to select two parameters (namely, λ_k and k), the L-curve can be interpreted as a surface, which can be challenging to analyze. However, by exploiting connections to Gaussian quadrature rules (see Appendix B), a variant of the L-curve called the L-ribbon has been considered that inexpensively constructs a ribbon-like region that contains the L-curve of the (direct) Tikhonov regularization applied to the full-dimensional problem [35, 41, 45]. Note that the points on the L-curve can be computed using the projected quantities only, which are not computationally expensive to evaluate as far as $k \ll \min\{m, n\}$ (see Appendix A for more details).

An approach related to the L-curve, which still aims at finding the right balance between the regularization error and the perturbation error, was described in [14, 233], where the estimated regularization parameter λ for (direct) Tikhonov regularization is obtained as the fixed point of the ratio between the residual norm $\|\mathbf{r}(\mathbf{x}(\lambda))\|$ and the solution norm $\|\mathbf{x}(\lambda)\|$. This same strategy can be used to select a regularization parameter at each iteration of hybrid projection methods, namely, by computing the fixed point λ_k of the function $\frac{\|\mathbf{r}(\mathbf{x}_k(\lambda))\|}{\|\mathbf{x}_k(\lambda)\|}$: this is done in [231] for a hybrid method based on GKB and Tikhonov regularization. The fixed point approach can be regarded as a realization of a parameter choice rule due to Regińska [190], and its performance strongly depends on a good initial guess of λ.

The generalized cross-validation (GCV) method is another popular approach for selecting regularization parameters when the noise level is unknown. The GCV method is a 'leave-one-out' prediction method. That is, the basic idea behind GCV is that, if an arbitrary element of the observed data is left out, a good choice of the regularization parameters should be able to predict the missing observation [98]. For (direct) Tikhonov regularization applied to the full-dimensional problem (2.1), the parameter computed by GCV is the one that minimizes the GCV function

$$G(\lambda) = \frac{n \left\|\mathbf{r}(\mathbf{x}(\lambda))\right\|^2}{\left(\text{trace}\Big(\mathbf{I}_m - \mathbf{A}\mathbf{A}^{\dagger}_{\text{reg}}(\lambda)\Big)\right)^2}. \tag{3.22}$$

In [20] Björck suggested using GCV in conjunction with TSVD for hybrid projection methods and using GCV to determine an appropriate stopping iteration as well. However, it was observed in [59] that the GCV method tended to perform poorly when used within hybrid methods based on GKB and Tikhonov regularization, due to over-smoothing. To remedy this, a weighted GCV (wGCV) method was introduced, which can be interpreted as a weighted 'leave-one-out' approach; an adaptive approach to estimate the new weight parameter was also described. At each iteration k, the weighted GCV functional for estimating λ is given by

$$G_w(\lambda, k) = \frac{n \left\|\mathbf{r}(\mathbf{x}_k(\lambda))\right\|^2}{\left(\text{trace}\Big(\mathbf{I}_m - w\mathbf{A}\mathbf{A}^{\dagger}_{\text{reg}}(\lambda, k)\Big)\right)^2}, \tag{3.23}$$

where we get the standard GCV function for $w = 1$. By minimizing the above functional with respect to both λ and k, it is possible to set both regularization parameters

involved in a hybrid method. One can use an alternating approach to sample the 2D GCV surface (3.23) as follows: first, a projected version of the GCV functional (3.23) is minimized for fixed k to get λ_k (this is sampling the GCV functional along lines); second, the GCV functional (3.23) is minimized for fixed λ_k. Note that both the numerator and the denominator of the functional $G_w(\lambda, k)$ can be expressed with respect to projected quantities only, which are not computationally expensive to evaluate as far as $k \ll \min\{m, n\}$ (see Appendix A for more details).

The *NCP (normalized cumulative periodogram)* approach for selecting regularization parameters does not require an estimate of the noise level, and it uses the residual components rather than the residual norm to estimate the regularization parameter [201]. This approach was used within a projection framework in [135], although this was not a hybrid projection method according to the criteria given in the present paper. It is used a stopping criterion for purely iterative methods (namely, the so-called algebraic iterative reconstruction techniques for computed tomography) in [134].

3.3.4. Illustrations on the behavior of parameter choice in hybrid methods. We use the image deblurring and tomography test problems from subsection 1.1 to show how different regularization parameter selection methods can perform within GKB-based hybrid projection methods.

In Figure 3.3 we display the so-called 'RRE surfaces', i.e., the values of the relative reconstructions error (RRE) norms $\|\mathbf{x}_k(\lambda_k) - \mathbf{x}_{\text{true}}\|/\|\mathbf{x}_{\text{true}}\|$ versus the pair (k, λ_k), at $k = 1, \ldots, 100$ and at logarithmically equispaced values of λ_k between 10^{-6} and 10^2. We can clearly see that, for both the image deblurring and the computed tomography problems, there are combinations of values of k and λ_k that deliver minimal relative reconstructions error norms, and that both the discrepancy principle (DP) and the weighted GCV (wGCV) method are able to compute couples (k, λ_k) that eventually (i.e., for k big enough) lay in such regions.

In Figure 3.4 we only consider the computed tomography problem and we illustrate the behavior (in terms of relative reconstruction errors and selected values of λ) of different regularization parameter choice strategies (DP, wGCV, optimal parameter minimizing the relative error at each iteration) as the number of iterations progresses. We can clearly see that the behavior of the GKB-based hybrid method is quite desirable for all the regularization parameter rules: namely, all approaches are successful in avoiding semiconvergence and, after a very rapid decrease in the relative error norms, at about the 20th iteration both the quality of the solution and the value of the regularization parameter stabilize (only the parameter set by wGCV displays slight variations). We observe that the first stopping criterion in (3.15) (stabilization of λ_k at consecutive iterations, denoted by an asterisk), applied with $\tau_\lambda = 10^{-4}$, is very effective when the DP is used (i.e., it prescribes to stop very close to the iteration where the relative error is minimal (denoted by a circle). This is not the case for wGCV, although there is no adverse impact on the quality of the solution computed when the solver stops, thanks to the long-term stable behavior of relative errors.

Finally, for both the image deblurring and computed tomography test problems, we generated observations with 100 noise realizations at noise level 10^{-2} and used GKB-based hybrid methods where the regularization parameter is chosen using the DP with the true noise level and wGCV with automatically selected weighting parameter. The computed values are displayed as histogram plots in Figure 3.5: looking at them we can infer that such strategies exhibit consistent behavior within different

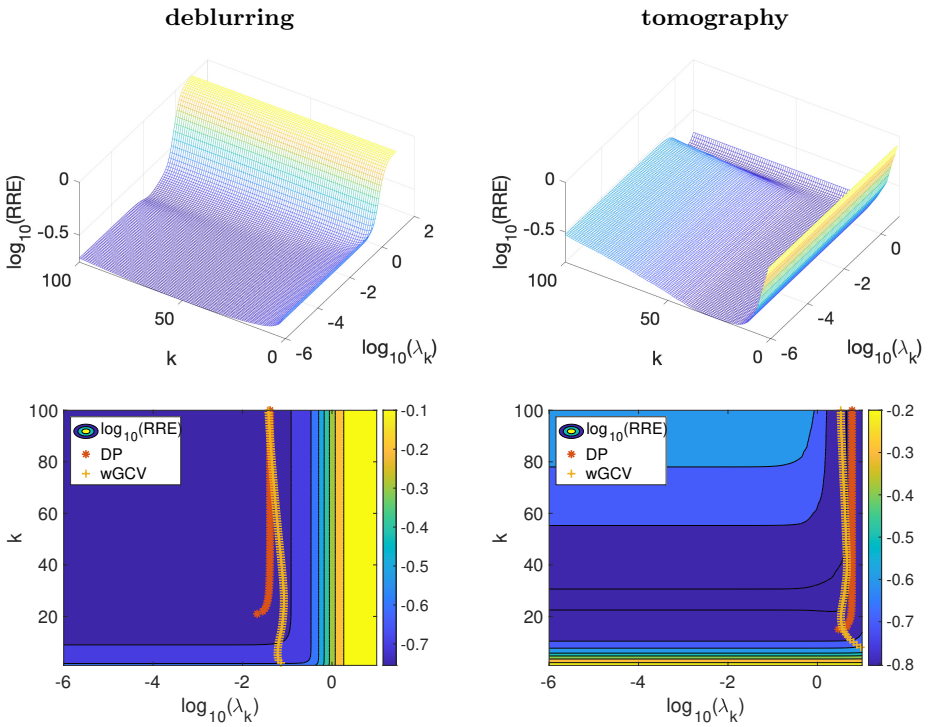


FIG. 3.3. *Illustration of the behavior of the GKB-based hybrid methods when the regularization parameter is chosen according to the discrepancy principle and the wGCV criterion. First and second row: the relative error norm for each sampled* (k, λ_k) *is recorded, to form the so-called* 'RRE surfaces'*, which are displayed as 3D surface plots (first row) and as 2D contour plots (second row), where the computed couples* (k, λ_k) *obtained by applying the discrepancy principle and wGCV are highlighted by special markers.*

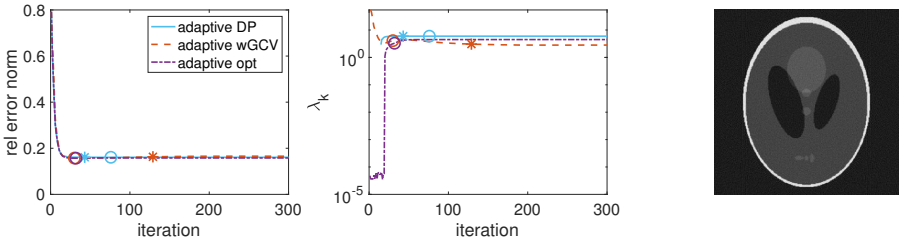


FIG. 3.4. *Illustration of the GKB-based hybrid method with different adaptive parameter choice strategies, applied to the computed tomography test problem. In the left panel, we provide the history of the relative reconstruction error norms per iteration* k *obtained when using the discrepancy principle (DP) (3.19) and the weighted GCV (wGCV) (3.23). The optimal (opt) parameter is the* λ_k *that minimizes the relative error at each iteration* k*. In the middle panel we provide the computed regularization parameters at each iteration. Special markers highlight the iteration satisfying the first stopping criterion in (3.15) with* $\tau_\lambda = 10^{-4}$ *(asterisks), and the iteration delivering the best relative error (circle), for each approach (DP, wGCV, opt). In the right panel we display the best solution computed when applying wGCV, having relative reconstruction error* $\|\mathbf{x}_{30} - \mathbf{x}_{\text{true}}\|_2 / \|\mathbf{x}_{\text{true}}\|_2 = 0.1574$*. We refer to Figure 3.1 for a similar illustration for the image deblurring problem.*

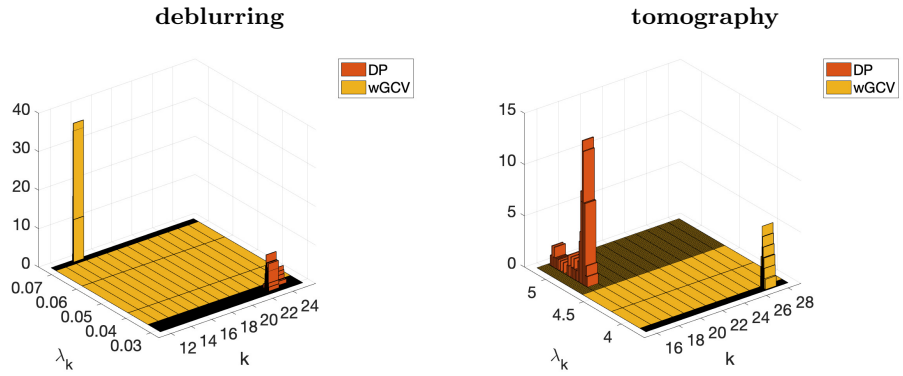


FIG. 3.5. *An illustration of the behavior of the GKB-based hybrid methods when the regularization parameter is chosen according to the discrepancy principle and the wGCV criterion. The test problems are run 100 times, each with a different noise realization (noise level is always* 10^{-2}*). Histogram plots of the stopping iteration* k *and the corresponding chosen regularization parameter* λ_k *are displayed.*

noise realizations.

To conclude, we emphasize that every problem is different, and there is not one approach that will work for all problems. Thus, it is good to have a variety of methods that can guide one in selecting a suitable set of parameters.

3.4. Theoretical insights. The goal of this section is to give the reader a flavor of the kinds of theoretical results that have been investigated for hybrid projection methods. This section is organised in subsections, which present specific classes of results and elaborate on their significance. Specifically:

- We can characterize the kth iterate of a hybrid projection method for a fixed regularization parameter; see subsection 3.4.1
- We can study how noise in the data propagates to the projected problem using subspace approximation properties, and we can use this information for parameter selection and analysis; see subsection 3.4.2
- We can study the behavior of the projected problem and draw connections to TSVD; see subsection 3.4.3
- We can investigate Krylov subspace methods for continuous formulations to get insight into the adopted SVD approximations and the convergence of hybrid methods; see subsection 3.4.4

3.4.1. Project-then-regularize versus regularize-then-project. For standard Tikhonov regularization, an insightful result deals with characterizing the iterates from a hybrid projection method. For a fixed regularization parameter, there is a result regarding the equivalence of iterates from two approaches: 'first-regularize-then-project' and 'first-project-then-regularize' [121, 126]. See Figure 3.6 and the following theorem.

THEOREM 3.1. *Fix* $\lambda > 0$ *and define* $\mathbf{x}_k(\lambda) \in \mathbb{R}^n$ *to be the* k*th iterate of conjugate gradient applied to the Tikhonov problem,*

$$\min_{\mathbf{x}} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2 + \lambda \|\mathbf{x}\|^2 .$$

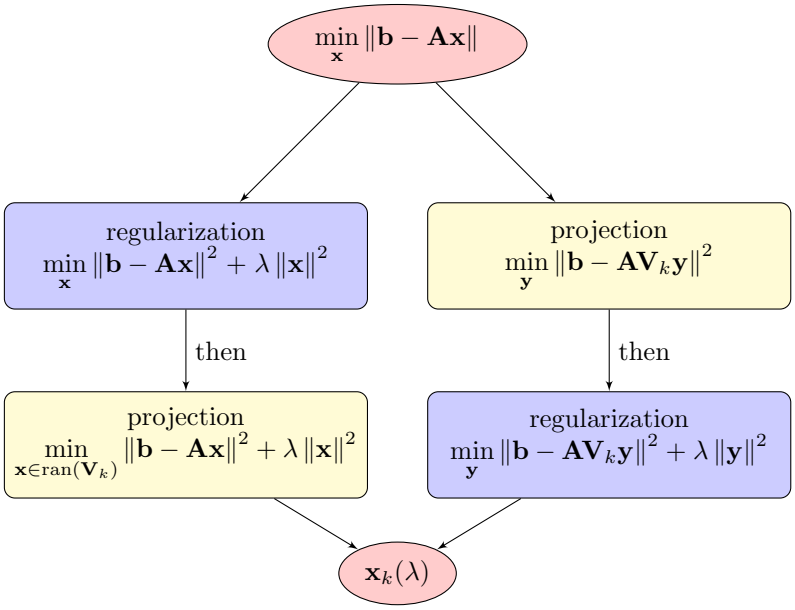


FIG. 3.6. *For fixed regularization parameter λ and in exact arithmetic, both approaches 'first-regularize-then-project' and 'first-project-then-regularize' result in the same solution approximation. A similar figure can be found in [126].*

Let $\mathbf{y}_k(\lambda) \in \mathbb{R}^k$ be the exact solution to the regularized, projected problem

$$\min_{\mathbf{y}} \|\beta_1 \mathbf{e}_1 - \mathbf{B}_k \mathbf{y}\|^2 + \lambda \|\mathbf{y}\|^2,$$

where $\mathbf{B}_k$ and $\mathbf{V}_k$ are derived from GKB applied to the original problem. Then

$$\mathbf{z}_k(\lambda) = \mathbf{V}_k \mathbf{y}_k(\lambda) = \mathbf{x}_k(\lambda).$$

The significance of this result is that, for large-scale problems where the 'first-regularize-then-project' approach is not feasible (e.g., because obtaining a good regularization parameter is too expensive a priori, see Figure 2.3), the hybrid projection methods that follow the 'first-project-then-regularize' approach produce, at the kth iteration, the same regularized solution (in exact arithmetic and with the same regularization parameter). Extensions of this result to the case where TSVD regularization is used instead of Tikhonov can be found in [160].

3.4.2. Subspace approximation and noise propagation. Since the approximation subspace for the solution plays a pivotal role in the success of both purely iterative and hybrid projection methods, there have been recent investigations on subspace approximation properties, including results quantifying how much the approximation subspace can be expanded and how the noise in the data affects the approximation subspace.

The authors of [94] prove that, when GKB is applied to a severely ill-posed problem with noiseless data, the product of the entries of the last column of the bidiagonal matrix $\mathbf{B}_k$ decays as $(k\sigma_k^{\mathrm{A}})^2$; note that these entries are somewhat related to how much the Krylov subspace $\mathcal{K}_k(\mathbf{A}^\top\mathbf{A}, \mathbf{A}^\top\mathbf{b})$ is expanded at the kth iteration. Under similar assumptions, when employing the Arnoldi algorithm, the rate of decay of the last entry of the Hessenberg matrix $\mathbf{H}_k$ (i.e., $[\mathbf{H}_k]_{k+1,k}$) is comparable to $k\sigma_k^{\mathrm{A}}$. The authors

of [139] study how noise in the measurements (right-hand side of (1.1)) propagates to the projected problem and, in particular, how it enters the basis vectors spanning the approximation subspace. This leads also to efficient estimates of the noise level that can be used to set the regularization parameter λ_k for hybrid methods, or a stopping criterion (see subsection 3.3). Specifically, assuming that the Discrete Picard Condition holds, the noise amplification (i.e., the appearance of high frequency noise) is described as an effect of damping the smooth components due to convergence of Ritz values to large eigenvalues of $\mathbf{A}^\top\mathbf{A}$ and the orthogonalization in the GKB algorithm. After the noise has 'revealed' itself in the so called 'noise-revealing' iteration, the LSQR relative residual can be used as a noise level estimator. This investigation is extended in [138], to consider the most popular iterative solvers based on GKB and to establish explicit relations between the noise-contaminated bidiagonalization vectors and the residuals of the considered regularization methods. In particular, it is shown that the coefficients of the linear combination of the computed bidiagonalization vectors reflect the amount of propagated noise in each of these vectors; influence of the loss of orthogonality is also discussed. Some of these investigations (most relevantly, the analysis of the residual vector) have been also performed in [92, 93] for methods based on the Arnoldi algorithm, mainly using tools from approximation theory.

Finally, the propagation of noise in the approximation subspace and the residuals associated to purely iterative methods based on the GKB algorithm (such as LSQR) and the Arnoldi algorithm (such as GMRES and RRGMRES) is studied in [148]. The latter may not be successful in filtering the noise out, because of the so-called 'mixing' of the SVD components in the approximation subspace; this issue may actually be irrelevant, if features of the solution can be reconstructed anyway thanks to the presence of favourable vectors in the Krylov approximation subspace and it can be mitigated using some 'preconditioning' [130]; see also subsection 4.1. Some 'preconditioners' that can be used together with (RR)GMRES to correct for severe asymmetries in the discretized forward operator are proposed and analyzed in [90]. Focusing on image deblurring problems, and exploiting the two-dimensional discrete cosine transform, [131] also analyzes how noise from the data enters the solution computed by LSQR, GMRES, RRGMRES, MINRES, and MR-II, concluding that the noise mainly affects the reconstructions in the form of low-pass filtered white noise, and discouraging the use of GMRES and MINRES for image deblurring.

3.4.3. The projected problem and connections to TSVD. Beyond investigations into the approximation subspace, another class of results pair the analysis of the approximation subspace for the solution with the analysis of the behavior of the projected problem, or focus solely on the latter, including results on the approximation of the singular value decomposition. These results shed light on how the ill-conditioning of the original problem transfers to the projected problem, on the semi-convergence phenomenon, and on how properties such as the so-called discrete Picard condition are 'inherited' by the projected problem.

Bounds on the 'residuals' associated to the SVD approximations obtained by the GKB and Arnoldi decompositions (i.e., the matrix 2-norm of the difference between the original coefficient matrix (2.2) and the approximated SVD obtained at each step of a Krylov subspace method) are presented in [94], where it is concluded that, depending on the method and on the residuals, these could either be 0, or their norm may decay as the singular values of the original coefficient matrix. The authors of [92] prove that, starting from a full-dimensional problem satisfying the discrete Picard condition (i.e., typically assuming noise-free data), the discrete Picard condition

holds when the problem is projected using the Arnoldi or the GKB algorithms: this is achieved by combining some SVD update rules and a backward induction argument (i.e., going from the full-dimensional problem to projected problems of decreasing dimension). Naturally, much analysis focuses on establishing theoretical and heuristic links between the performance of TSVD (which can be regarded as a least-squares solver projecting the full-dimensional problem (2.4) onto the subspace spanned by the dominant right singular vectors) and Krylov projection methods. Some early investigations, such as [120], find that, since the projection attained with the LSQR method is tailored to the specific right-hand side $\mathbf{b}$, this results in a more rapid convergence of the solver. Also, when it comes to computing TSVD-like approximations, the authors of [16] exploit a classical property of GKB (namely, the approximation quality depends on the relative distance between the singular values of $\mathbf{A}$ [203]) to conclude that, since for discrete inverse problems the relative gap between large singular values of $\mathbf{A}$ is generally much larger than the relative gap between its small singular values, the singular values of $\mathbf{B}_k$ converge very quickly to the largest singular values of $\mathbf{A}$. Also, links between the TSVD and Lanczos algorithms (for symmetric problems) are investigated in [95] where, thanks to the speed at which the nonnegative subdiagonal entries of $\mathbf{T}_k$ decay to zero, the solution of the discrete ill-posed problems can be efficiently and effectively (i.e., without significant, if any, reduction of the quality of the computed solution) expressed in terms of the Lanczos basis vectors rather than the eigenvectors of $\mathbf{A}$; this extends to the solution subspace determined by LSQR and the right singular vector subspace.

More recently, a body of papers by Jia and collaborators puts some renewed emphasis on the investigation of the relations between the TSVD solutions and the solutions computed by some popular purely iterative regularization methods (such as LSQR, LSMR, and MINRES). In particular, assuming distinct singular values, [144] establishes bounds for the distance between the k-dimensional Krylov subspace associated to LSQR and the k-dimensional dominant right singular space associated to TSVD, and concludes that the former better 'captures' the latter for severely and moderately ill-posed problems than for mildly ill-posed problems. This implies that LSQR performs better as a stand-alone regularization method for severely and moderately ill-posed problems, but should be paired with additional regularization (i.e., in a hybrid framework) for successfully handling mildly ill-posed problems. Estimates for the accuracy of the rank-k approximation generated by GKB are also provided. This topic is further investigated in [149, 150], where an analysis of the approximation of the large singular values of $\mathbf{A}$ by the Ritz values is also performed to assess if this happens in natural order (i.e., largest singular values first). The paper [151] extends the same investigations to the case of multiple singular values, with similar findings.

3.4.4. Continuous formulations. Although the present survey paper considers discrete inverse problems, in many situations (especially when analyzing some convergence properties of solvers or inferring the behavior of the projected problems) looking at the continuous formulation (i.e., within a Hibert space $\mathcal{H}$) can be beneficial. Indeed, it is well known that some discretization schemes (e.g., those based on boundary element methods) actually project the continuous problem onto finite dimensional vector spaces, and that such a projection has a regularizing effect; see [121, 160]. Indeed, with 'proper conversion', many of the presented derivations for Krylov methods in finite dimension can be translated to the continuous setting; see [73]. CG was historically among the first Krylov methods to be analyzed in a continuous setting: [102] reports relevant results and references. More recently, the authors

of [40] show that, when the error-free data vector $\mathbf{b}_{\text{true}}$ lies in a finite-dimensional Krylov subspace, then the (continuous) GMRES method is a regularization method if the iterations are terminated by a stopping rule based on the discrepancy principle. Methods based on the Arnoldi algorithm (either FOM or GMRES) have been analyzed in a continuous setting [178]: here, some of the properties described (and somewhat heuristically justified) in a discrete setting in [94] are fully established (and sometimes strengthened) in a continuous setting. More specifically, assuming that we are dealing with Hilbert-Schmidt operators (like Fredholm integral equations of the first kind with square-integrable kernels), it can be shown that the rate of convergence of these solvers is related to the extendibility of the Krylov subspace $\mathcal{K}_k(\mathbf{A}, \mathbf{b})$, which is also comparable with the rate of decay of the singular values of $\mathbf{A}$. Moreover, it is proven that the dominant singular values can be approximated with improved accuracy as the iterations proceed, further supporting the statement that methods based on the Arnoldi algorithm are regularization methods. This investigation has an impact on the finite-dimensional setting, too: for instance, having proven convergence of the SVD approximation obtained by the Arnoldi algorithm, implies that many parameter choice rules that are intrinsically based on the success of this approximation (see subsection 3.3) are meaningful when used within hybrid methods. An analogous investigation appeared in [51], where the infinite-dimensional GKB algorithm and LSQR are considered. In [179] the case of Krylov solvers applied to general-form Tikhonov regularization (still in a continuous setting), with a fixed regularization parameter; both solvers based on the Arnoldi algorithm and on the GKB algorithm are considered and the resulting scheme is dubbed ' Krylov-Tikhonov'. Denoting by $u^\dagger$ and u_k the solution to the continuous problem and the Krylov solution, respectively, the main investigation in [179] is concerned with proving that such Krylov methods are orthogonal projection methods for the linear operator equation associated to the normal equations for the continuous Tikhonov formulation: because of this, there exists a (semi)norm $E(\cdot)$ in $\mathcal{H}$ such that $E(u_k - u^\dagger)$ converges to zero as k goes to infinity.

4. Extensions. The goal of this section is to describe some extensions and recent advancements in hybrid projection methods. In particular, in subsection 4.1 we describe various hybrid projection methods that have been developed for general-form Tikhonov regularization. In subsection 4.2 we describe hybrid projection approaches that go beyond the standard projection subspaces by enrichment, augmentation and recycling of approximation subspaces for the solution. Then in subsection 4.3 we go beyond 2-norm regularization and consider the ℓ_p regularized problem. In particular, we focus on recent advancements in flexible iterative methods (which are related to flexible preconditioning where the preconditioner changes during the iterative process) and extensions to hybrid frameworks. In subsection 4.4 we describe the increasingly important role that hybrid iterative methods play in the field of large-scale computational uncertainty quantification. We first give a brief introduction to draw connections between traditional variational regularization and statistical (Bayesian) inverse problems, and we describe various scenarios where hybrid projection methods have enabled researchers to go beyond point estimates and perform efficient UQ. In subsection 4.5, we describe the role that hybrid projection methods have played in solving nonlinear inverse problems, where the forward model is nonlinear. Table 4.1 presents a compact summary of the problem formulations considered in the following subsections, highlighting similarities and differences in the choice of the fit-to-data and regularization terms.

TABLE 4.1
Variational regularization methods of the form $\min_{\mathbf{x}\in\mathbb{R}^n} \mathcal{J}(\mathbf{x},\mathbf{b}) + \lambda\mathcal{R}(\mathbf{x})$*, where* $\mathcal{J}$ *and* $\mathcal{R}$ *are a fit-to-data and regularization term, respectively, considered in each subsections.*

subsection	$\mathcal{J}(\mathbf{b},\mathbf{x})$	$\mathcal{R}(\mathbf{x})$	**notes**
4.1	$\|\mathbf{b}-\mathbf{A}\mathbf{x}\|_2^2$	$\|\mathbf{L}\mathbf{x}\|_2^2$	$\mathbf{L}\in\mathbb{R}^{r\times n}$
4.2	$\|\mathbf{b}-\mathbf{A}\mathbf{x}\|_2^2$	$\|\mathbf{x}\|_2^2$	projection on sums of subspaces
4.3	$\|\mathbf{b}-\mathbf{A}\mathbf{x}\|_2^2$	$\|\mathbf{x}\|_p^p$	extends to $\mathcal{J}(\mathbf{b},\mathbf{x}) = \|\mathbf{b}-\mathbf{A}\mathbf{x}\|_q^q$
4.4	$\|\mathbf{b}-\mathbf{A}\mathbf{x}\|^2_{\mathbf{R}^{-1}}$	$\|\mathbf{x}\|^2_{\mathbf{Q}^{-1}}$	
4.5	$\|\mathbf{b}-F(\mathbf{x})\|_2^2$	$\|\mathbf{x}\|_2^2$	$F:\mathbb{R}^n\to\mathbb{R}^m$ nonlinear operator

A common theme in all the following subsections is that extensions of hybrid projection methods beyond the standard approaches described in section 3, in order to be effective in a large-scale setting, require the efficient computation of an appropriate solution subspace and adaptive regularization parameter choice for the projected problem (i.e., within the solution subspace), giving rise to hybrid formulations. Although the computed solution subspaces may not be as straightforward as standard methods, they can incorporate some prior knowledge or meaningful information about the solution or the kind regularization functional to be considered, eventually leading to superior reconstructions with respect to the ones delivered by standard methods. Most often this means that the standard GKB or Arnoldi decompositions have to be modified, for example, to formally work with variable preconditioning, to enforce orthogonality or optimality properties in a different norm, or to incorporate a low-dimensional subspace representing prior information. Also, most often, such tools have already been developed for use in other settings, e.g., for the solution of well-posed problems (e.g., PDEs). Therefore, a considerable part of this section is devoted to reviewing such tools, tailoring them to the problems at hand, and drawing possible analogies between them.

4.1. Beyond standard-form Tikhonov: Hybrid projection methods for general-form Tikhonov. Thus far we have focused mainly on the standard-form Tikhonov problem (2.1), where the regularizer $\|\mathbf{x}\|^2$ promotes smoothness by penalizing solutions with large norm; however, other smoothness properties may be desired. Many researchers have considered the *general-form Tikhonov problem*,

$$\min_{\mathbf{x}} \|\mathbf{b}-\mathbf{A}\mathbf{x}\|^2 + \lambda\,\|\mathbf{L}\mathbf{x}\|^2\,, \tag{4.1}$$

where $\mathbf{L}\in\mathbb{R}^{p\times n}$ is called the regularization matrix. We assume that the null spaces of $\mathbf{A}$ and $\mathbf{L}$ intersect trivially, so that $[\mathbf{A}^\top, \mathbf{L}^\top]^\top$ has full column rank and the solution to (4.1) is unique. Typical choices of $\mathbf{L}$ include discretizations of a differential operator (e.g., the discrete first or second derivative, which promote further smoothness by penalizing derivatives of solutions with large norms). An important observation is that the general-form Tikhonov solution can be written as a spectral filtered solution, where both the filter factors and the basis for the solution are determined by the generalized singular value decomposition (GSVD) of $\{\mathbf{A},\mathbf{L}\}$; see, e.g., [126, Chapter 8]. However computing the GSVD is not always feasible, e.g., for large-scale unstructured problems. In these settings, the most immediate approach is to apply an iterative solver (e.g., any of the solvers listed in subsection 2.2) to the equivalent least-squares formulation of (4.1): however, a suitable value of λ must be fixed ahead of the iterative method. Since such λ is typically not available, one should apply an iterative solver to (4.1) repeatedly (once for every tested value of λ),

as dictated by some well-known parameter choice strategies (such as the ones listed in subsection 3.3): this eventually results in a costly strategy, and alternatives have been sought in the literature.

A common approach to handle the general-form Tikhonov problem is to transform it to standard form [71]. That is, one computes

$$\bar{\mathbf{y}}_{\mathbf{L}}(\lambda) = \arg\min_{\bar{\mathbf{y}}\in\mathbb{R}^p} \|\mathbf{A}\mathbf{L}_{\mathbf{A}}^{\dagger}\bar{\mathbf{y}} - \bar{\mathbf{b}}\|^2 + \lambda\|\bar{\mathbf{y}}\|^2 \,, \text{ where } \begin{array}{rcl} \bar{\mathbf{b}} & = & \mathbf{b} - \mathbf{A}\mathbf{x}_0^{\mathbf{L}} \\ \mathbf{x}_{\mathbf{L}}(\lambda) & = & \mathbf{L}_{\mathbf{A}}^{\dagger}\bar{\mathbf{y}}_{\mathbf{L}}(\lambda) + \mathbf{x}_0^{\mathbf{L}} \end{array} , \tag{4.2}$$

and where $\mathbf{x}_0^{\mathbf{L}}$ is the component of the regularized solution $\mathbf{x}_{\mathbf{L}}(\lambda)$ in the null space of $\mathbf{L}$. Here $\mathbf{L}_{\mathbf{A}}^{\dagger}$ is the $\mathbf{A}$-weighted generalized inverse of $\mathbf{L}$, defined by $\mathbf{L}_{\mathbf{A}}^{\dagger} = (\mathbf{I}_n - (\mathbf{A}(\mathbf{I}_n - \mathbf{L}^{\dagger}\mathbf{L}))^{\dagger}\mathbf{A})\mathbf{L}^{\dagger}$; see [125, Section 2.3] and [127] for a geometrical interpretation (in terms of projections). Note that, if $\mathbf{L}$ is invertible, then $\mathbf{L}_{\mathbf{A}}^{\dagger} = \mathbf{L}^{-1}$, and, if $\mathbf{L}$ has full column rank, then $\mathbf{L}_{\mathbf{A}}^{\dagger} = \mathbf{L}^{\dagger}$. If $\mathbf{L}$ is an underdetermined matrix, then $\mathbf{L}_{\mathbf{A}}^{\dagger}$ may not be the same as $\mathbf{L}^{\dagger}$. In the Bayesian inverse problems literature, this particular change of variables is referred to as *priorconditioning* [33, 36, 47, 48], because the matrix $\mathbf{L}$ is constructed from the covariance matrix of the solution modeled as a random variable (see also subsection 4.4.1). The main idea is to interpret $\mathbf{L}$ as a preconditioner; however, while preconditioning for iterative methods is often discussed in the context of accelerating iterative methods (typically via clustering of the eigenvalues), priorconditioners in (4.2) change the subspace for the solution by including information from the prior (recall also the discussion in subsection 3.2.1). Indeed, in earlier work that leveraged this idea without collocating it within the Bayesian framework, this approach was dubbed 'smoothing norm preconditioning' and was first adopted in connection with CGLS [117, 121], and then GMRES [130]. Once formulation (4.2) is established, the hybrid projection methods of section 3 can be applied, which define the approximation subspace for the solution with respect to the matrix $\mathbf{A}\mathbf{L}_{\mathbf{A}}^{\dagger}$. For instance, in the case of GKB, the kth approximation subspace for the solution is defined as follows:

$$\mathcal{K}_k((\mathbf{A}\mathbf{L}_{\mathbf{A}}^{\dagger})^{\top}\mathbf{A}\mathbf{L}_{\mathbf{A}}^{\dagger}, (\mathbf{A}\mathbf{L}_{\mathbf{A}}^{\dagger})^{\top}\bar{\mathbf{b}}) = \text{span}\{((\mathbf{A}\mathbf{L}_{\mathbf{A}}^{\dagger})^{\top}\mathbf{A}\mathbf{L}_{\mathbf{A}}^{\dagger})^{i-1}(\mathbf{A}\mathbf{L}_{\mathbf{A}}^{\dagger})^{\top}\bar{\mathbf{b}}\}_{i=0,\dots,k-1} \,.$$

For some problems, multiplication with $\mathbf{A}\mathbf{L}_{\mathbf{A}}^{\dagger}$ can be done by solving a least-squares problem followed by an oblique projection; see [124, 130] for details. Furthermore, there are some regularization operators $\mathbf{L}$ for which sparsity or structure can be exploited so that multiplications with $\mathbf{L}_{\mathbf{A}}^{\dagger}$ can be done efficiently (e.g., banded matrices, circulant matrices, and orthogonal projections [46, 72, 169, 196]). However, in general, performing multiplications with $\mathbf{L}_{\mathbf{A}}^{\dagger}$ may be computationally difficult especially for large-scale problems (e.g., when $\mathbf{L}$ is defined as a product of matrices or as a sum of Kronecker products).

In the framework of hybrid projection methods, alternative approaches to approximate the solution of (4.1) have been devised without resorting to a standard-form transformation. For instance, Kilmer, Hansen and Español [159] developed a hybrid projection method that simultaneously bidiagonalizes both $\mathbf{A}$ and $\mathbf{L}$ (here $\mathbf{L}$ can be rectangular and does not need to be full rank), using a joint bidiagonalization algorithm inspired by Zha [240]. More precisely, a partial joint bidiagonalization for $\mathbf{A}$ and $\mathbf{L}$ is updated at each iteration: at the kth iteration, it reads

$$\mathbf{A}\mathbf{Z}_k = \mathbf{U}_{k+1}^{\mathbf{A}}\mathbf{B}_k^{\mathbf{A}}, \qquad \mathbf{L}\mathbf{Z}_k = \mathbf{U}_k^{\mathbf{L}}\mathbf{B}_k^{\mathbf{L}}, \tag{4.3}$$

where $\mathbf{Z}_k = [\mathbf{z}_1, \dots, \mathbf{z}_k]$, $\mathbf{U}_{k+1}(\beta\mathbf{e}_1) = \mathbf{b}$, and $\beta = \|\mathbf{b}\|$. Although this algorithm only requires that products with $\mathbf{A}, \mathbf{L}$ and their transposes can be computed, the cost of

generating the partial factorization (4.3) is somewhat difficult to quantify: indeed, the kth step requires one call to LSQR to produce orthogonal projections that are needed to update the partial joint bidiagonalization (4.3), and each LSQR iteration requires four matrix-vector products (with each of $\mathbf{A}$, $\mathbf{A}^\top$, $\mathbf{L}$, and $\mathbf{L}^\top$). Taking $\mathbf{x} \in \mathrm{ran}(\mathbf{Z}_k)$, i.e., $\mathbf{x} = \mathbf{Z}_k\mathbf{w}$, in (4.1) results in the following equivalent problems

$$\min_{\mathbf{w}\in\mathbb{R}^k} \left\| \begin{bmatrix} \mathbf{A}\mathbf{Z}_k \\ \sqrt{\lambda}\mathbf{L}\mathbf{Z}_k \end{bmatrix} \mathbf{w} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\| = \min_{\mathbf{w}\in\mathbb{R}^k} \left\| \begin{bmatrix} \mathbf{B}_k^{\mathbf{A}} \\ \sqrt{\lambda}\mathbf{B}_k^{\mathbf{L}} \end{bmatrix} \mathbf{w} - \begin{bmatrix} \beta\mathbf{e}_1 \\ \mathbf{0} \end{bmatrix} \right\|. \tag{4.4}$$

Since the last problem in the above equations is a Tikhonov problem of dimension $O(k)$ involving only bidiagonal matrices, it is significantly cheaper to solve than (4.1), especially for k small. Let us denote by $\mathbf{w}_k(\lambda)$ the minimizer of (4.4), then $\mathbf{x}_k(\lambda) = \mathbf{Z}_k\mathbf{w}_k(\lambda)$ is the minimizer of (4.1), constrained to $\mathrm{ran}(\mathbf{Z}_k)$. For fixed λ, the joint bidiagonal structure leads to short recurrence updates (in k) for $\mathbf{w}_k(\lambda)$, for the residual norm $\|\mathbf{A}\mathbf{x}_k(\lambda) - \mathbf{b}\|$, and for the regularization norm term $\|\mathbf{L}\mathbf{x}_k(\lambda)\|$: this is relevant if the solution $\mathbf{x}_k(\lambda) = \mathbf{Z}_k\mathbf{w}_k(\lambda)$ has to be computed at fixed pre-sampled values of λ, e.g., when applying the L-curve criterion. The discrepancy principle to set $\lambda = \lambda_k$ adaptively at each iteration can also be efficiently employed. It can be shown that the approximation subspace $\mathrm{ran}(\mathbf{Z}_k)$ is meaningful, as $\mathbf{x}_k(\lambda)$ 'resembles' a truncated GSVD regularized solution to (1.1).

Reichel, Sgallari, and Ye [192] describe an iterative approach that simultaneously reduces square matrices $\mathbf{A}$ and $\mathbf{L}$ to generalized Hessenberg matrices using a generalized Arnoldi process, originally introduced in [167]. The formal full factorizations associated to this method consist of

$$\begin{array}{rcll} \mathbf{A}\mathbf{Q} &=& \mathbf{Q}\mathbf{H}^{\mathbf{A}}, & \text{where } h_{i,j}^{\mathbf{A}} = 0 \quad \text{if } i \geq 2j+1\,, \\ \mathbf{L}\mathbf{Q} &=& \mathbf{Q}\mathbf{H}^{\mathbf{L}}, & \text{where } h_{i,j}^{\mathbf{L}} = 0 \quad \text{if } i \geq 2j+2\,, \end{array} \tag{4.5}$$

and where $\mathbf{Q}$ is orthogonal. In practice, k iterations of the generalized Arnoldi algorithm can be computed with initial vector $\mathbf{q}_1 = \mathbf{b}/\|\mathbf{b}\| = \mathbf{b}/\beta$ to generate the first k columns of the generalized Hessenberg matrices $\mathbf{H}^{\mathbf{A}}$ and $\mathbf{H}^{\mathbf{L}}$ appearing above, and the corresponding columns of $\mathbf{Q}$. The columns of $\mathbf{Q}$ span a so-called generalized Krylov subspace obtained by multiplying $\mathbf{q}_1$ by $\mathbf{A}$ and $\mathbf{L}$ in a periodic fashion. Let $\mathbf{H}_k^{\mathbf{A}} \in \mathbb{R}^{(2k+1)\times k}$ and $\mathbf{H}_k^{\mathbf{L}} \in \mathbb{R}^{(2k+2)\times k}$ be the principal submatrices of the matrices $\mathbf{H}^{\mathbf{A}}$ and $\mathbf{H}^{\mathbf{L}}$, and let $\mathbf{Q}_k = [\mathbf{q}_1, \ldots, \mathbf{q}_k] \in \mathbb{R}^{n\times k}$ be obtained by taking the first k columns of $\mathbf{Q}$. Taking $\mathbf{x} \in \mathrm{ran}(\mathbf{Q}_k)$, i.e., $\mathbf{x} = \mathbf{Q}_k\mathbf{w}$, in (4.1) results in the following equivalent problems

$$\min_{\mathbf{w}\in\mathbb{R}^k} \left\| \begin{bmatrix} \mathbf{A}\mathbf{Q}_k \\ \sqrt{\lambda}\mathbf{L}\mathbf{Q}_k \end{bmatrix} \mathbf{w} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\| = \min_{\mathbf{w}\in\mathbb{R}^k} \left\| \begin{bmatrix} \mathbf{H}_k^{\mathbf{A}} \\ \sqrt{\lambda}\mathbf{H}_k^{\mathbf{L}} \end{bmatrix} \mathbf{w} - \begin{bmatrix} \beta\mathbf{e}_1 \\ \mathbf{0} \end{bmatrix} \right\|.$$

Note that the approximation subspace for the solution of (4.1) can alternatively be generated starting from $\mathbf{q}_1 = \mathbf{Q}\mathbf{e}_1 = \mathbf{A}\mathbf{b}/\|\mathbf{A}\mathbf{b}\|$, leading to minor adjustments in the above projected problem (analogous to the range-restricted approach described in subsection 3.2.1). Similar to the method in [159], the present approach allows adaptive selection of the regularization parameter $\lambda = \lambda_k$ at the kth iteration, with a negligible computational overhead. Different from [159], the present method can work only if both $\mathbf{A}$ and $\mathbf{L}$ are square (but this restriction can be overcome, e.g., by zero-padding). Furthermore, the present method does not require the action of $\mathbf{A}^\top$ and $\mathbf{L}^\top$, and no (inner) calls to LSQR are needed to generate the factorization (4.5). While these are clear advantages of the solver presented in [192] compared to that presented

in [159], the downsides are that the structure of the generalized Hessenberg matrices $\mathbf{H}_k^{\mathbf{A}}$ and $\mathbf{H}_k^{\mathbf{L}}$ cannot be immediately exploited (as was possible for bidiagonal matrices $\mathbf{B}_k^{\mathbf{A}}$ and $\mathbf{B}_k^{\mathbf{L}}$ in (4.3)); moreover, to the best of our knowledge, no heuristic insight is available to justify the choice of the approximation subspace $\mathrm{ran}(\mathbf{Q}_k)$ for the solution (in particular, no links with truncated GSVD have been established). Despite this, some specific features of the solution of (4.1) can be enhanced by pairing the present method with so-called 'selective regularization', i.e., by augmenting the approximation subspace for the solution associated to this specific method; generic (i.e., not specific to the generalized Arnoldi algorithm) schemes for achieving augmentation are described in subsection 4.2.

Lampe, Reichel, and Voss [162] propose an alternative projection method for approximating the solution (4.1), which is initiated by picking an approximation subspace $\mathcal{V} \subset \mathbb{R}^n$ of small dimension $k \ll \min\{m, n\}$ spanned by the orthonormal columns of a matrix $\mathbf{V}_k \in \mathbb{R}^{n\times k}$. This can be done, for instance, by running k GKB iterations, so that $\mathrm{ran}(\mathbf{V}_k) = \mathcal{K}_k(\mathbf{A}^\top\mathbf{A}, \mathbf{A}^\top\mathbf{b})$. Denote by $\mathbf{x}_k(\lambda)$ the solution of the constrained Tikhonov problem obtained by imposing $\mathbf{x} \in \mathcal{V}$ in (4.1), i.e., take $\mathbf{x}_k(\lambda) = \mathbf{V}_k\mathbf{y}_k(\lambda)$, where

$$\mathbf{y}_k(\lambda) = \underset{\mathbf{y}\in\mathbb{R}^k}{\arg\min} \left\| \begin{bmatrix} \mathbf{A}\mathbf{V}_k \\ \sqrt{\lambda}\mathbf{L}\mathbf{V}_k \end{bmatrix} \mathbf{w} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\| = (\mathbf{V}_k^\top \underbrace{(\mathbf{A}^\top\mathbf{A} + \lambda\mathbf{L}^\top\mathbf{L})}_{=:\mathbf{T}(\lambda)} \mathbf{V}_k)^{-1}\mathbf{V}_k^\top\mathbf{A}^\top\mathbf{b}$$

Note that $\mathbf{y}_k(\lambda)$ can be computed efficiently by using a 'skinny' QR factorization of $[(\mathbf{A}\mathbf{V}_k)^\top, (\mathbf{L}\mathbf{V}_k)^\top]^\top$, which can be updated as k increases. A suitable value of the regularization parameter $\lambda = \lambda_k$ can be determined using any of the strategies described in subsection 3.3. Once $\mathbf{x}_k(\lambda)$ is computed, the original search space $\mathcal{V} \in \mathbb{R}^{n\times k}$ is expanded by the normalized gradient of the functional in (4.1) evaluated at $\mathbf{x}_k(\lambda)$. Namely, one considers

$$\mathbf{V}_{k+1} = [\mathbf{V}_k, \mathbf{v}_{new}], \quad \text{where} \quad \mathbf{v}_{new} = \left(\mathbf{T}(\lambda_k)\mathbf{x}_k(\lambda_k) - \mathbf{A}^\top\mathbf{b}\right) / \|\mathbf{T}(\lambda_k)\mathbf{x}_k(\lambda_k) - \mathbf{A}^\top\mathbf{b}\|.$$

Although $\mathbf{v}_{new}$ so defined is orthogonal to $\mathrm{ran}(\mathbf{V}_k)$, this can be enforced numerically using reorthogonalization techniques, if loss of orthogonality is a concern. In general, the approximation subspace $\mathrm{ran}(\mathbf{V}_k)$ for the solution of (4.1) is not a Krylov subspace and it is therefore called a 'generalized Krylov subspace': note however that it is yet a different generalized Krylov subspace than the one used in [192]. The process of solving problem (4.1) constrained to the updated approximation subspace $\mathcal{V}$ is repeated, and expansions of $\mathcal{V}$ are generated until a stopping criterion is satisfied.

Hochstenbach and Reichel [140] use a partial GKB built with respect to $\mathbf{A}$ and $\mathbf{b}$ to generate, at the kth iteration, the solution subspace $\mathcal{K}_k(\mathbf{A}^\top\mathbf{A}, \mathbf{A}^\top\mathbf{b})$, onto which they project the regularization matrix $\mathbf{L}$ in order to compute an approximate solution. A similar idea was used in [94, 232]. Note that, contrary to all the strategies summarized so far in the present section, when using these approaches, the matrix $\mathbf{L}$ does not enter the approximation subspace for the solution.

4.2. Beyond standard projection subspaces: Enrichment and recycling. The success of Krylov projection methods, and in particular the regularizing properties of Krylov methods, depends on the ability of the associated Krylov subspace to capture the most salient features of the solution. In the previous section, we surveyed approaches to solve the general-form Tikhonov problem (4.1), where the choice of the matrix $\mathbf{L} \neq \mathbf{I}_n$ may affect the solution subspace for the iterative method (e.g., via subspace preconditioning or priorconditioning with a change of variables). Such

transformations result in Krylov subspaces that are 'better suited' for the problem [47, 130]. This means that the basis vectors can capture the essence or important features of the solution, which can lead to faster convergence properties since fewer iterations are needed to obtain an accurate solution. However, in some applications, the regularized solution can be further improved by extending or enhancing the approximation subspace.

In this section, we assume that we are given a low-dimensional subspace that represents prior information or expert knowledge. More specifically, we assume that we are given a suitable set of basis vectors, denoted $\mathbf{W}_p \in \mathbb{R}^{n\times p}$, for which $\mathcal{W}_p = \text{ran}(\mathbf{W}_p)$ should be used in addition to a Krylov subspace in order to improve the solution. We call $\mathcal{W}_p$ the 'prior solution subspace' (not to be confused with priors or priorconditioning methods). The main assumption is that the solution has a significant component in the given subspace $\mathcal{W}_p$, and the goal is to incorporate the provided subspace judiciously in order to obtain improved regularized solutions. These methods are often referred to as enrichment, augmentation, and recycling techniques.

We begin with some motivating scenarios where such techniques may be desired. First, experienced practitioners may have solution basis vectors containing important information about the desired solution (e.g., a low-dimensional subspace) from previous experiments or theoretical analyses. By incorporating these solution basis vectors, the solution accuracy can be *significantly* improved, depending on the quality of the provided vectors. Second, for very large inverse problems where many iterations are needed or where there are many unknowns, one of the main computational disadvantages of most hybrid methods compared to standard iterative methods is the need to store the basis vectors for solution computation, namely $\mathbf{V}_k$ (using the same notations as in (3.9)). Therefore, for problems with slow convergence, extracting important information from existing Krylov subspaces and restarting the iterative method can speed up convergence while keeping memory requirements low. Third, and this is related to subsection 4.5, hybrid methods may be embedded within nonlinear optimization frameworks, e.g., optimal experimental design [113, 115] or nonlinear inverse problems [112]. In these cases, a sequence of inverse problems must be solved, e.g., where the forward model may be parameterized such that the change in the model from one problem to the next is relatively small, or the goal may be to compute and update solutions from streaming data. Rather than start each solution computation from scratch, techniques described in this section can be used to improve the given subspace and to compute a regularized solution efficiently in the improved subspace. In the context of solving inverse problems, we describe extensions that go beyond the standard projection subspaces.

Enrichment methods. We begin with enrichment methods, which we define as methods that, at the kth iteration, compute regularized solutions in a solution subspace that is a direct sum of the prior solution subspace and a k-dimensional Krylov subspace $\mathcal{K}_k$ corresponding to a standard iterative approach, see e.g. subsection 2.2. That is, hybrid projection methods with enrichment construct iterates defined as

$$\mathbf{x}_k(\lambda_k) = \underset{\mathbf{x}\in\mathbf{S}_{p,k}}{\arg\min} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2 + \lambda_k \|\mathbf{x}\|^2, \tag{4.6}$$

where

$$\mathbf{S}_{p,k} = \mathcal{W}_p \oplus \mathcal{K}_k = \{\mathbf{x} + \mathbf{y} \mid \mathbf{x} \in \mathcal{W}_p \text{ and } \mathbf{y} \in \mathcal{K}_k\}. \tag{4.7}$$

Enrichment methods can improve the solution accuracy by incorporating information about the desired solution into the solution process, i.e., by enriching the solution

subspace. Methods for enriching Krylov subspaces have been described in the context of both Arnoldi and Lanczos projection methods.

For Arnoldi based methods, the Regularized Range-Restricted GMRES method (R^{3}GMRES) [69] extends the RRGMRES approach (see subsection 3.2.1) to include a subspace that represents prior information about the solution where the approach is to compute $\mathbf{x}_k(\lambda_k)$ as in (4.6) with $\mathbf{S}_{p,k} = \mathcal{W}_p \oplus \mathcal{K}_k(\mathbf{A}, \mathbf{Ab})$ and $\lambda_k = 0$. Efficient implementations are described in [69, 218], but these are not considered hybrid methods. If additional regularization is desired, hybrid versions can be developed.

For Lanczos-based projection methods, enrichment methods seek solutions in the solution space (4.7) with $\mathcal{K}_k(\mathbf{A}^\top\mathbf{A}, \mathbf{A}^\top\mathbf{b})$. A modification of CGLS method (sometimes also denoted in the literature with the acronym CGNR) was described in [44], where Krylov subspaces are combined with vectors containing important information about the desired solution (e.g., a low-dimensional subspace). Given an enrichment subspace $\mathcal{W}_p$, the enriched CGLS method determines the kth iterate in the subspace $\mathcal{W}_p \oplus \mathcal{K}_k(\mathbf{A}^\top\mathbf{A}, \mathbf{A}^\top\mathbf{b})$. The enriched CGLS method was used for Tikhonov problems (e.g., penalized least-squares problems), where several linear systems were solved for different choices of the regularization parameter as described in Frommer and Maass [81]. More specifically, a sequence of Tikhonov problems for regularization parameters $\lambda = \lambda_j = 2^{-j}$ for $j = 0, 1, \ldots$ are solved, until for some λ_j, the corresponding solution satisfies the discrepancy principle. In [128], a hybrid enriched bidiagonalization method was developed. For a fixed regularization parameter, the iterates of the enriched bidiagonalization method take the form (4.6) with $\mathbf{S}_{p,k} = \mathcal{W}_p \oplus \mathcal{K}_k(\mathbf{A}^\top\mathbf{A}, \mathbf{A}^\top\mathbf{b})$ and are mathematically equivalent to those of the enriched CGLS method [44]. The main contribution of [128] was to turn the basic algorithm into a hybrid method with automatic selection of λ_k, where the Lanczos bidiagonalization algorithm was used to compute an orthonormal basis for $\mathcal{K}_k(\mathbf{A}^\top\mathbf{A}, \mathbf{A}^\top\mathbf{b})$ that is augmented by a user-defined low-dimensional subspace $\mathcal{W}_p$ in each step of the algorithm.

Finally, we remark that a subspace-restricted SVD was described in [141], where a modification of the singular value decomposition permits a specified linear subspace to be contained in the singular vector subspaces. That is, the user can prescribe some of the columns of the $\mathbf{U}$ and $\mathbf{V}$ matrices for all truncations, and truncated versions as well as Tikhonov problems can be solved. For inverse problems, such approaches can give more accurate approximations of the solution when compared to standard TSVD and Tikhonov.

Recycling techniques. Hybrid methods have been considered in the context of subspace recycling methods, which were developed for solving sequences of linear systems with slowly changing coefficient matrices, multiple right hand sides, or both. Krylov subspace recycling methods can exhibit accelerated convergence by reusing subspace information that is typically generated during a Krylov method on one or more of the systems. We point the interested reader to recent surveys [219, 220] and references therein.

A general framework to combine recycling techniques with tools from compression in a hybrid context is described in [152]. More specifically, the approach consists of three steps. First, a suitable set of orthonormal basis vectors $\mathbf{W}_p$ is provided (e.g., from a related problem or from expert knowledge) or may need to be determined (e.g., via compression of previous solutions). This prior solution subspace may even include an initial guess $\mathbf{x}^*$ for the solution. Let

$$\mathbf{AW}_p = \mathbf{V}_p\mathbf{R} \tag{4.8}$$

be the 'skinny' QR factorization, where $\mathbf{V}_p \in \mathbb{R}^{m\times p}$ contains orthonormal columns

and $\mathbf{R} \in \mathbb{R}^{p\times p}$ is upper triangular. 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 the Krylov subspace

$$\mathcal{K}_k(\mathbf{A}^\top \mathbf{P}^\perp_{\mathbf{V}_p}\mathbf{A}, \mathbf{A}^\top \mathbf{P}^\perp_{\mathbf{V}_p}\mathbf{r}),$$

where $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}^*$ and $\mathbf{P}^\perp_{\mathbf{V}_p} = \mathbf{I} - \mathbf{V}_p\mathbf{V}_p^\top$ is the orthonormal projector onto $\mathcal{V}_p^\perp$ where $\mathcal{V}_p = \text{ran}(\mathbf{V}_p)$. The third step is to find a suitable regularization parameter λ_k and compute a solution to the regularized projected problem in the extended solution space. The recycled, hybrid iterate is given by

$$\mathbf{x}_k(\lambda_k) = \arg\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 + \lambda_k \|\mathbf{x}\|^2, \quad \mathbf{x} \in \mathcal{W}_p \oplus \mathcal{K}_k(\mathbf{A}^\top \mathbf{P}^\perp_{\mathbf{V}_p}\mathbf{A}, \mathbf{A}^\top \mathbf{P}^\perp_{\mathbf{V}_p}\mathbf{r}).$$

This process can be repeated in a cyclic fashion, and it is successful in solving a broad class of inverse problems (e.g., solving very large problems where the number of basis vectors becomes too large for memory storage, solving a sequence of regularized problems (e.g., changing regularization terms or nonlinear solvers), and solving problems with streaming data). We also mention that there have been various works on developing hybrid methods for solving systems with multiple right hand sides. In [182], the authors suggest using the block Lanczos algorithm [100]. Recent works extend hybrid methods for multiple right hand sides via global iterative methods [226], block reduction [3], and tensor-tensor products [194, 195], with applications to color image and video restoration.

The development of recycling techniques is deeply intertwined with the development of augmented iterative methods. For example, the recycling GKB process is similar to that used in the augmented LSQR method [7, 8]. For well-posed least-squares problems that require many LSQR iterations, augmented LSQR methods use harmonic Ritz vectors that approximate singular vectors associated with the small singular values, thereby reducing computational cost and improving convergence. However, when applied to ill-posed inverse problems, the augmented LSQR method without an explicit regularization term exhibits semi-convergent behavior.

For all of the methods described in this section, the improvement in solution accuracy *significantly* depends on the quality of the vectors in $\mathbf{W}_p$. Some suggestions for subspaces are provided in [141, 152]. For problems with unsuitable augmentation spaces, an adaptive augmented GMRES was considered in [161], which automatically selects a suitable subspace from a set of user-specified candidates.

4.3. Beyond the 2-norm: Sparsity-enforcing hybrid projection methods for ℓ_p regularization. Although widely used, it is well known that standard-form Tikhonov regularization (2.1) and even its general-form counterpart (4.1) can be rather restrictive and that other regularization terms can yield better approximations of $\mathbf{x}_{\text{true}}$ in (1.1). For example, if $\mathbf{x}_{\text{true}}$ is known to be sparse (i.e., when most of its entries $[\mathbf{x}_{\text{true}}]_i$, $i = 1, \ldots, n$, are expected to be zero), the variational formulation

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_p^p, \quad \text{with } 0 < p \leq 1, \tag{4.9}$$

is commonly considered in the literature, especially since the advent of the compressed sensing theory; see, e.g., [50]. Indeed, while sparse vectors have a small ℓ_0-'norm' (denoting the number of nonzero elements), replacing the ℓ_p-norm in (4.9) by an ℓ_0-'norm' would yield an NP-hard optimization problem (see, for instance, [80]). A well-established practice to remedy this drawback is to consider ℓ_p regularization terms with $0 < p \leq 1$ to promote sparsity, noting that the objective function in (4.9)

is non-differentiable at the origin and it is nonconvex when $0 < p < 1$ (see, e.g., [143, 163, 164]). Note that, if sparsity of the solution in a different domain (e.g., wavelets, discrete cosine transform, gradient) is desired, a regularization term of the form $\|L(\mathbf{x})\|_p^p$ should be considered, where L is a linear operator that, in this setting, is typically called 'sparsity' transformation (i.e., it maps $\mathbf{x}$ from $\mathbb{R}^n$ to a domain where $\mathbf{x}$ is sparse); this situation will be considered in subsection 4.3.4. Also, sometimes, an ℓ_q-norm with $q \neq 2$ has to be used to evaluate the fit-to-data term in (4.9): this is appropriate, for instance, when the noise term in (1.1) is modeled as impulse noise; see, e.g., [28].

Optimization problems involving the ℓ_p norm, such as the ℓ_2-ℓ_p regularized problem (4.9), can be solved using a variety of general nonlinear solvers and optimization methods; see, for instance, [15, 96, 183, 211, 238] and the references therein. A common drawback of these approaches is that a suitable value of the regularization parameter λ should be set a priori implying that, if such value is not available, several instances of problem (4.9) with different values of λ should typically be solved to determine an appropriately regularized solution. Here we focus on solvers that approximate the regularization term in (4.9) by a sequence of weighted ℓ_2 terms (i.e., iteratively reweighted schemes); as we will highlight in the next section, these methods all support an adaptive choice of the regularization parameter. Such iteratively reweighted schemes intrinsically rely on the interpretation of problem (4.9) as a nonlinear weighted least-squares problem of the form

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_p^p = \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{W}^{(p)}(\mathbf{x})\mathbf{x}\|_2^2 \,, \tag{4.10}$$

where the diagonal weights in $\mathbf{W}^{(p)}(\mathbf{x})$ are defined as

$$\mathbf{W}^{(p)}(\mathbf{x}) = \operatorname{diag}\left((|x_i|^{\frac{p-2}{2}})_{i=1,\dots,n} \right) . \tag{4.11}$$

Since we are considering $0 < p \leq 1$, division by zero might occur if $x_i = 0$ for any $i \in \{1, ..., n\}$ and, in fact, this is common situation in case of sparse solutions. For this reason, instead of (4.11), we can consider the following closely related weights

$$\widetilde{\mathbf{W}}^{(p,\tau)}(\mathbf{x}) = \operatorname{diag}\left(((x_i{}^2 + \tau^2)^{\frac{p-2}{4}})_{i=1,\dots,n} \right) , \tag{4.12}$$

where τ is a parameter that, to keep the derivations simple, we will consider fixed and chosen ahead of the iterations (although some authors adaptively choose it). Problem (4.10) is then replaced by

$$\min_{\mathbf{x}} \underbrace{\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\widetilde{\mathbf{W}}^{(p,\tau)}(\mathbf{x})\mathbf{x}\|_2^2}_{T^{(p,\tau)}(\mathbf{x})}, \tag{4.13}$$

where $\tau \neq 0$ ensures that $T^{(p,\tau)}(\mathbf{x})$ is differentiable at the origin for $p > 0$. Note that problem (4.10) can be recovered from problem (4.13) by setting $\tau = 0$.

A well-established framework to solve problem (4.13) is the local approximation of $T^{(p,\tau)}$ by a sequence of quadratic functionals $T_k(\mathbf{x})$ that, for $k = 1, 2, \dots$, gives rise to a sequence of quadratic problems of the form

$$\mathbf{x}_{k,*} = \arg\min_{\mathbf{x}} \underbrace{\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{W}_k\mathbf{x}\|_2^2 + c_k}_{=:T_k(\mathbf{x})}, \tag{4.14}$$

where $\mathbf{W}_k = \widetilde{\mathbf{W}}^{(p,\tau)}(\mathbf{x}_{k-1,*})$, c_k is a constant (with respect to $\mathbf{x}$) term for the kth problem in the sequence, and λ has absorbed other possible multiplicative constants so that $T_k(\mathbf{x})$ in (4.14) corresponds to a quadratic tangent majorant of $T^{(p,\tau)}(\mathbf{x})$ in (4.13) at $\mathbf{x} = \mathbf{x}_{k-1,*}$, i.e., $T_k(\mathbf{x}) \geq T^{(p,\tau)}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^n$, $T_k(\mathbf{x}_{k-1,*}) = T^{(p,\tau)}(\mathbf{x}_{k-1,*})$, and $\nabla T_k(\mathbf{x}_{k-1,*}) = \nabla T^{(p,\tau)}(\mathbf{x}_{k-1,*})$; see also [143, 199]. Because of this, iteratively reweighted schemes (4.14) can be regarded as particular instances of majorization-minimization (MM) schemes, and convergence to a stationary point of the objective function in problem (4.13) can be guaranteed; see [143, 145, 163].

The vector $\mathbf{x}_{k,*}$ denotes the solution of (4.14). For moderate-sized problems, or for large-scale problems where $\mathbf{A}$ has some exploitable structure, $\mathbf{x}_{k,*}$ may be obtained by applying a direct solver to (4.14). However, for large unstructured problems, iterative solvers can be used in different fashions to approximate the solution of (4.14): this leads to either schemes based on nested (inner-outer) iteration cycles for the sequence of problems (4.13) or schemes that adaptively incorporate updated weights at each iteration of a single iteration cycle. The former are detailed in subsection 4.3.1, while the latter are detailed in subsections 4.3.2 and 4.3.3.

4.3.1. Strategies based on inner-outer iterations. Iteratively Reweighted Least Squares (IRLS, [21, Chapter 4], [67]) or Iteratively Reweighted Norm (IRN, [109, 199]) methods are very popular schemes that can handle the smoothed reformulation (4.13) of problem (4.9). Let $\mathbf{x}_{k,l}$ denote the approximate solution at the lth iteration (of an inner cycle of iterations) for the kth problem of the form (4.14) (i.e., at the kth iteration of an outer cycle of iterations); one typically takes $\mathbf{x}_{k,*} = \mathbf{x}_{k,l}$ when a stopping criterion is satisfied for the inner iteration cycle. IRLS or IRN methods based on an inner-outer iteration scheme are very popular and have been used in combination with different inner solvers, such as steepest descent and CGLS [79, 199].

If $\mathbf{W}_k$ is square and invertible (note that this can be assumed when the weights are defined as in (4.12) with $\tau > 0$ for any fixed $p > 0$), problem (4.14) can be easily and conveniently transformed into standard form as follows

$$\bar{\mathbf{x}}_{k,*} = \arg\min_{\bar{\mathbf{x}}} \|\mathbf{A}\mathbf{W}_k^{-1}\bar{\mathbf{x}} - \mathbf{b}\|_2^2 + \lambda\|\bar{\mathbf{x}}\|_2^2, \quad \text{so that} \quad \mathbf{x}_{k,*} = \mathbf{W}_k^{-1}\bar{\mathbf{x}}_{k,*}. \tag{4.15}$$

The interpretation of the matrix $\mathbf{W}_k^{-1}$ as a right preconditioner for problem (4.14) can be exploited under the framework of priorconditioning; see subsection 4.1 and [33]. Similar to the approaches mentioned for (4.14), Krylov methods can be straightforwardly adopted to compute an approximate solution of the equivalent kth problem of the form (4.15) (i.e., at the kth iteration of an outer cycle of iterations) through an inner cycle of iterations. In particular, variants based on hybrid projection methods (such as hybrid GMRES or hybrid LSQR) have been successfully considered, with the added benefit that adaptive regularization parameter choice strategies (based, e.g., on the discrepancy principle, UPRE, and GCV) can be considered, effectively overcoming a common drawback of other popular optimization methods for ℓ_p regularization; see [198]. Specifically, at the lth iteration of a Krylov method for (4.15), let $\mathbf{V}_{k,l} \in \mathbb{R}^{n\times l}$ be the matrix whose columns span a (preconditioned) Krylov Subspace $\mathcal{K}_{k,l}$ of dimension l; the precise definition of $\mathcal{K}_{k,l}$ depends on the selected projection method. Problem (4.15) can be projected and solved in $\mathcal{K}_{k,l}$ by computing

$$\bar{\mathbf{y}}_{k,l} = \arg\min_{\bar{\mathbf{y}}} \|\mathbf{A}\underbrace{\mathbf{W}_k^{-1}\mathbf{V}_{k,l}}_{=:\mathbf{Z}_{k,l}}\bar{\mathbf{y}} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{V}_{k,l}\bar{\mathbf{y}}\|_2^2, \tag{4.16}$$

so that $\bar{\mathbf{x}}_{k,l} = \mathbf{V}_{k,l}\,\bar{\mathbf{y}}_{k,l}$, and

$$\mathbf{x}_{k,l} = \mathbf{W}_k^{-1}\,\bar{\mathbf{x}}_{k,l} = \mathbf{W}_k^{-1}\,\mathbf{V}_{k,l}\,\bar{\mathbf{y}}_{k,l} = \mathbf{Z}_{k,l}\,\bar{\mathbf{y}}_{k,l}\,.$$

Once $\mathbf{x}_{k,l}$ has been computed and once the weights $\mathbf{W}_{k+1} = \widetilde{\mathbf{W}}^{(p,\tau)}(\mathbf{x}_{k,l})$ have been updated according to (4.12), the $(k+1)$st problem of the form (4.15), with k replaced by $k+1$, needs to be solved: to achieve this, one needs to compute a new Krylov subspace from scratch, defined with respect to the new coefficient matrix $\mathbf{A}\mathbf{W}_{k+1}^{-1}$. For this reason, this approach may become computationally expensive.

4.3.2. Strategies based on generalized Krylov methods. One way to avoid inner-outer iterations when applying the IRN method described in subsection 4.3.1 is to exploit generalized Krylov subspaces (GKS), like the ones described in subsection 4.1: this approach underlies the so-called GKSpq [163] and MM-GKS [143] algorithms, which we summarize below.

Given the initial subspace $\mathcal{K}_h(\mathbf{A}^\top\mathbf{A}, \mathbf{A}^\top\mathbf{b}) = \mathrm{ran}(\mathbf{V}_0)$ generated by the GKB algorithm, where h is small (typically $h \leq 5$) and $\mathbf{V}_0 \in \mathbb{R}^{n\times h}$ has orthonormal columns, at the kth iteration of the MM-GKS method, a GKS $\mathcal{V}_k$ spanned by the orthonormal columns of $\mathbf{V}_k \in \mathbb{R}^{n\times(h+k)}$ is computed, and it is used to project problem (4.14) as follows

$$\mathbf{y}_k = \operatorname*{arg\,min}_{\mathbf{y}\in\mathbb{R}^{h+k}} \|\mathbf{A}\mathbf{V}_k\mathbf{y} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{W}_k\mathbf{V}_k\mathbf{y}\|_2^2.$$

Equivalently, introducing the 'skinny' QR factorizations

$$\mathbf{A}\mathbf{V}_k = \mathbf{Q}^{\mathbf{A}}\mathbf{R}^{\mathbf{A}}\,, \quad \mathbf{W}_k\mathbf{V}_k = \mathbf{Q}^{\mathbf{W}}\mathbf{R}^{\mathbf{W}}\,, \tag{4.17}$$

the above problem can be reformulated as the low-dimensional (if $h+k \ll \min\{m,n\}$) problem

$$\mathbf{y}_k = \operatorname*{arg\,min}_{\mathbf{y}\in\mathbb{R}^{h+k}} \|\mathbf{R}^{\mathbf{A}}\mathbf{y} - (\mathbf{Q}^{\mathbf{A}})^\top\mathbf{b}\|_2^2 + \lambda\|\mathbf{R}^{\mathbf{W}}\mathbf{y}\|_2^2\,.$$

After the kth approximation to problem (4.14) is formed by taking $\mathbf{x}_k = \mathbf{V}_k\mathbf{y}_k$, one computes the residual of the normal equations associated to (4.14)

$$\mathbf{r}_k = (\mathbf{A}^\top\mathbf{A} + \lambda\mathbf{W}_k^2)\mathbf{x}_k - \mathbf{A}^\top\mathbf{b}\,.$$

The subspace $\mathcal{V}_{k+1} = \mathrm{ran}(\mathbf{V}_{k+1})$ is then expanded by adding the vector $\mathbf{r}_k/\|\mathbf{r}_k\|_2$; reorthogonalization of the columns of $\mathbf{V}_{k+1}$ may be advisable to enforce accuracy. An initial guess $\mathbf{x}_0$ for $\mathbf{x}_{\mathrm{true}}$ is needed to define the first weights $\mathbf{W}_1$, and the authors of [143] suggest to take $\mathbf{x}_0 = \mathbf{b}$, although this only works when $\mathbf{A}$ is square. We emphasize that the regularization parameter λ can be adaptively set during the GKS iterations, effectively overcoming a common drawback of other popular optimization methods for ℓ_p regularization; see [27, 28]. Also, an expression for the quadratic tangent majorant (4.14) different from $T_k(\mathbf{x})$ can be devised, which allows for a cheaper GKS computation, in that smart updates of the QR factorization (4.17) are performed: we refer to [143] for the details. Finally, such strategies based on GKS can also handle generalizations of problem (4.9) where weights are also included in the fit-to-data term.

4.3.3. Strategies based on flexible Krylov methods. Another approach that avoids inner-outer iterations when applying the IRN method described in subsection 4.3.1 is to exploit flexible Krylov subspaces (FKS): this approach underlies

hybrid methods based on the flexible Arnoldi algorithm [88] and the flexible Golub-Kahan algorithm [54], as well as the more recent IRN-FKS methods [89]. The starting point of this class of methods is the formulation (4.15) of the kth reweighted problem. Flexible Krylov methods, which are classically employed to handle iteration-dependent preconditioning [177, 204, 216], provide a natural framework to update the inverted weights $\mathbf{W}_k^{-1}$ as soon as a new approximation of $\mathbf{x}_{\text{true}}$ is computed. That is, at each iteration, the updated inverted weights are immediately incorporated within the approximation subspace for the solution to the next problem in the sequence stemming from (4.15), for $k = 1, 2, \dots$.

Formally, both the flexible Arnoldi and the flexible Golub-Kahan algorithms look very similar to their standard counterparts (3.2) and (3.3), respectively. If $\mathbf{A}$ is square, taking $\mathbf{v}_1 = \mathbf{b}/\left\|\mathbf{b}\right\|_2$ and defining $\mathbf{W}_1$ using an initial guess $\mathbf{x}_0$ ($\mathbf{W}_1 = \mathbf{I}_n$ if $\mathbf{x}_0 = \mathbf{0}$), the ith iteration of the flexible Arnoldi algorithm [204] applied to problem (4.15) (with $k = i$) computes

$$\mathbf{z}_i = \mathbf{W}_i^{-1}\mathbf{v}_i\,, \quad \mathbf{v} = \mathbf{A}\mathbf{z}_i\,,\ \mathbf{v} = (\mathbf{I}_n - \mathbf{V}_i\mathbf{V}_i^\top)\mathbf{v}\,, \quad \mathbf{v}_{i+1} = \mathbf{v}/\left\|\mathbf{v}\right\|_2\,, \tag{4.18}$$

where $\mathbf{W}_i = \widetilde{\mathbf{W}}^{(p,\tau)}(\mathbf{x}_{i-1})$ is defined as in (4.12), $\mathbf{x}_{i-1}$ being the solution obtained at the previous iteration (approximately solving problem (4.15) with $k = i-1$), and where $\mathbf{V}_i = [\mathbf{v}_1, ..., \mathbf{v}_i] \in \mathbb{R}^{n\times i}$ has orthonormal columns. k iterations of the flexible Arnoldi algorithm can be equivalently expressed as the partial matrix factorization

$$\mathbf{A}\mathbf{Z}_k = \mathbf{V}_{k+1}\mathbf{H}_k\,, \quad \text{where} \quad \begin{array}{rcll} \mathbf{Z}_k &=& [\mathbf{W}_1^{-1}\mathbf{v}_1, ..., \mathbf{W}_k^{-1}\mathbf{v}_k] & \in \mathbb{R}^{n\times k} \\ \mathbf{H}_k &=& \mathbf{V}_{k+1}^\top\mathbf{A}\mathbf{Z}_k & \in \mathbb{R}^{(k+1)\times k} \end{array}\,. \tag{4.19}$$

Here the columns of $\mathbf{Z}_k$ span the approximation subspace for the solution, $\mathbf{V}_{k+1}$ is defined as in (4.18) and, by construction, $\mathbf{H}_k$ is upper Hessenberg. If $\mathbf{A}$ is either square or rectangular, taking $\mathbf{u}_1 = \mathbf{b}/\left\|\mathbf{b}\right\|_2$, the ith iteration of the flexible Golub-Kahan algorithm [54, 89] applied to problem (4.15) computes

$$\begin{array}{lll} \mathbf{v} = \mathbf{A}^\top\mathbf{u}_i\,, & \mathbf{v} = (\mathbf{I}_n - \mathbf{V}_{i-1}\mathbf{V}_{i-1}^\top)\mathbf{v}\,, & \mathbf{v}_i = \mathbf{v}/\left\|\mathbf{v}\right\|_2\,, \\ \mathbf{z}_i = (\mathbf{W}_i^{-1})^2\mathbf{v}_i\,, & \mathbf{u} = \mathbf{A}\mathbf{z}_i\,, & \mathbf{u} = (\mathbf{I}_m - \mathbf{U}_i\mathbf{U}_i^\top)\mathbf{u}\,, \qquad \mathbf{u}_{i+1} = \mathbf{u}/\left\|\mathbf{u}\right\|_2\,, \end{array} \tag{4.20}$$

where the inverted weights $\mathbf{W}_i^{-1}$ are updated as in the flexible Arnoldi case, and both $\mathbf{V}_i = [\mathbf{v}_1, ..., \mathbf{v}_i] \in \mathbb{R}^{n\times i}$ and $\mathbf{U}_i = [\mathbf{u}_1, ..., \mathbf{u}_i] \in \mathbb{R}^{m\times i}$ have orthonormal columns. k iterations of the flexible Golub-Kahan algorithm can be equivalently expressed as the partial matrix factorizations

$$\begin{array}{l} \mathbf{A}\mathbf{Z}_k = \mathbf{U}_{k+1}\mathbf{M}_k \\ \mathbf{A}^\top\mathbf{U}_{k+1} = \mathbf{V}_{k+1}\mathbf{S}_{k+1} \end{array}\,, \text{ where } \begin{array}{rcll} \mathbf{Z}_k &=& [(\mathbf{W}_1^{-1})^2\mathbf{v}_1, ..., (\mathbf{W}_k^{-1})^2\mathbf{v}_k] & \in \mathbb{R}^{n\times k} \\ \mathbf{M}_k &=& \mathbf{U}_{k+1}^\top\mathbf{A}\mathbf{Z}_k & \in \mathbb{R}^{(k+1)\times k} \\ \mathbf{S}_{k+1} &=& \mathbf{V}_{k+1}^\top\mathbf{A}\mathbf{U}_{k+1} & \in \mathbb{R}^{(k+1)\times k} \end{array}\,. \tag{4.21}$$

Similar to (4.19), the columns of $\mathbf{Z}_k$ span the approximation subspace for the solution $\mathbf{x}_k$, $\mathbf{V}_{k+1}$ and $\mathbf{U}_{k+1}$ are defined as in (4.20) and, by construction, $\mathbf{M}_k$ and $\mathbf{S}_{k+1}$ are upper Hessenberg and upper triangular, respectively. We emphasize that, in both (4.19) and (4.21), $\text{ran}(\mathbf{Z}_k)$ is not necessarily a Krylov subspace (see [216]), and interpreting it as standard preconditioned Krylov subspaces is not straightforward (see [177]). As in section 3, we assume that both (4.19) and (4.21) are breakdown-free, i.e., at iteration $k \leq \min\{m, n\}$, $\text{ran}(\mathbf{Z}_k)$ has dimension k. The approximate solution $\mathbf{x}_k$ computed at the kth iteration is used to update the matrix $\mathbf{W}_{k+1}$ to be employed to expand the solution subspace at the $(k+1)$st iteration, as in (4.18) and (4.20).

In summary, all the available strategies [54, 88, 89] that approximate the solution of problem (4.15) through flexible Krylov methods compute, at the kth iteration,

$$\mathbf{y}_k = \arg\min_{\mathbf{y}} \|\mathbf{A}\mathbf{Z}_k\mathbf{y} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{P}_k\mathbf{y}\|_2^2, \quad \mathbf{x}_k = \mathbf{Z}_k\mathbf{y}_k \in \mathrm{ran}(\mathbf{Z}_k)\,, \tag{4.22}$$

and essentially differ in the choice of the regularization matrix $\mathbf{P}_k \in \mathbb{R}^{p\times k}$. Specifically,

(i) The method in [88] and one of the methods in [54] (dubbed 'hybrid-I') take $\mathbf{P}_k = \mathbf{I}_k$.

(ii) One of the methods in [54] (dubbed 'hybrid-R') takes $\mathbf{P}_k = \mathbf{R}_k \in \mathbb{R}^{k\times k}$, where $\mathbf{R}_k$ is the upper triangular factor in the 'skinny' QR factorization of the basis vectors, i.e., $\mathbf{Z}_k = \mathbf{Q}_k\mathbf{R}_k$, which can be efficiently updated for consecutive k.

(iii) The methods in [89] take $\mathbf{P}_k = \mathbf{R}_k^{\mathbf{W}} \in \mathbb{R}^{k\times k}$, where $\mathbf{R}_k^{\mathbf{W}}$ is the upper triangular factor in the 'skinny' QR factorization of the transformed basis vectors, i.e., $\mathbf{W}_k\mathbf{Z}_k = \mathbf{Q}_k^{\mathbf{W}}\mathbf{R}_k^{\mathbf{W}}$. Note that, unless $\mathbf{W}_k$ is a multiple of the identity matrix, the QR factorization cannot be efficiently updated for consecutive values of k, but it is still feasible to compute if $k \ll \min\{m, n\}$.

The three options above reveal that, when using hybrid projection methods based on flexible Krylov subspaces, even if problem (4.14) is transformed into standard form (4.15), projecting and regularizing are not interchangeable anymore. Indeed, option (i) corresponds to a 'first-project-then-regularize' scheme (i.e., it penalizes the 2-norm of the projected solution $\mathbf{y}_k$) while option (iii) corresponds to a 'first-regularize-then-project' scheme; option (ii) penalizes the 2-norm of the full-dimensional solution $\mathbf{x}_k$. When option (iii) is used, exploiting the majorization-minimization framework guarantees that the sequence of approximate solutions $\{\mathbf{x}_k\}_{k\geq 1}$ to (4.22) converges to a stationary point of problem (4.13). We refer to [89] for more details. Note that, by exploiting the properties of the matrices appearing in (4.19) and (4.21), the fit-to-data term is equivalently expressed as $\|\mathbf{H}_k\mathbf{y} - \|\mathbf{b}\|_2\mathbf{e}_1\|_2^2$ in the flexible Arnoldi case, or $\|\mathbf{M}_k\mathbf{y} - \|\mathbf{b}\|_2\mathbf{e}_1\|_2^2$ in the flexible Golub-Kahan case. Although the regularization parameter λ in (4.22) is displayed as independent of k, a heuristical adaptive choice is possible employing any of the strategies described in subsection 3.3, effectively overcoming a common drawback of other popular optimization methods for ℓ_p regularization. As illustrated below, a suitable approximation to a solution of (4.9) is efficiently computed if, for small k, the columns of $\mathbf{Z}_k$ can be used to capture the main features thereof.

An illustration. We use the tomography test problem from section 1 to show the behavior of a hybrid method based on the flexible Golub-Kahan algorithm (4.20), for the solution of problem (4.9) with $p = 1$; specifically, the regularization term in the projected problem (4.22) is chosen as in (i) and the regularization parameter is adaptively set at each iteration using wGCV (3.23). We run the method for 100 iterations. In the top row of Figure 4.1 we display the weights $\mathbf{W}_k$ at iterations $k = 4, 10, 12, 20$, reshaped as images. We can clearly see that, as the iterations proceed, the weights become increasingly accurate and structured. Namely, for $k = 4$, the weights almost uniformly penalize each entry of the solution $\mathbf{x}_4$ and therefore are not so effective in enforcing sparsity in $\mathbf{x}_4$. Starting at $k = 10$, larger weights are located in spatial positions corresponding to the zero background or the zero-valued pixels at the center of the phantom to be reconstructed (see also frame **(a)** in Figure 1.2), and therefore further penalise the corresponding entries of $\mathbf{x}_{10}$. This forces them towards zero. This trend continues and improves in future iterations, e.g., at $k = 20$, the zero-valued regions and their spatial boundaries are heavily penalised

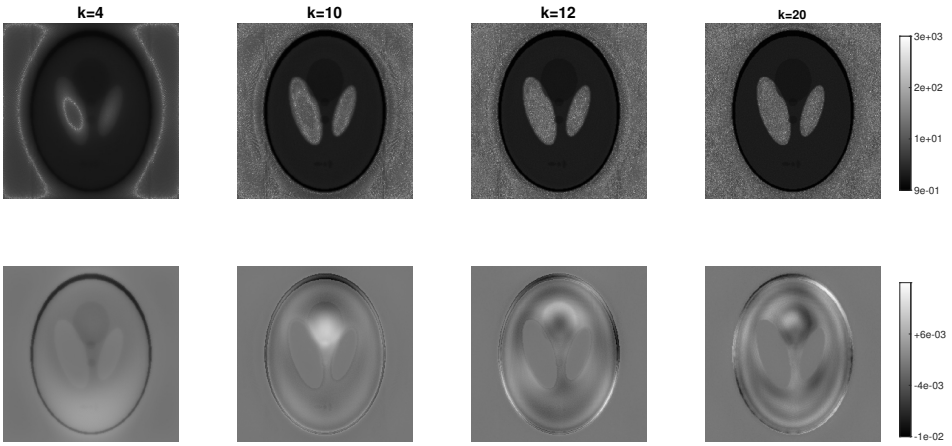


FIG. 4.1. *For the tomography example, and for a hybrid method based on the flexible Golub-Kahan algorithm (4.20) for ℓ_1-norm regularization, we display the weights $\mathbf{W}_k$ (upper rows) and the basis vectors $\mathbf{z}_k$ (lower rows), all reshaped as images, generated at iteration $k = 4, 10, 12, 20$.*

by the weighting matrix $\mathbf{W}_{20}$. In the bottom row of Figure 4.1 we display the flexible Golub-Kahan basis vectors $\mathbf{z}_k$ generated at iterations $k = 4, 10, 12, 20$, reshaped as images. All of these vectors capture the zero spatial locations of the phantom to be reconstructed, and hence a linear combination of these vectors can produce an accurate solution, where information about the phantom sparsity is encoded. In contrast, the basis vectors computed by the standard GKB-based hybrid method applied to the ℓ_2-norm Tikhonov problem (2.1) are not able to capture the sparsity in the phantom: we refer to Figure 3.2, where the GKB basis vectors generated at iterations $k = 1, 2, 4, 10$ are displayed, and Figure 3.4, where a GKB-based hybrid reconstruction is displayed. The reconstruction computed at the 91st iteration of the hybrid flexible Golub-Kahan method is displayed in Figure 4.2; looking at the upper-right corner of the phantom displayed as a surface, it is evident that zero pixels can be accurately reconstructed with a flexible method.

4.3.4. Sparsity under transform. Many of the methods described above for problem (4.9) can be generalized to approximate a solution to the more general variational regularization problem,

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \mathcal{R}(\mathbf{x}), \quad \text{for many choices of the functional } \mathcal{R}(\cdot). \tag{4.23}$$

All the methods based on inner-outer iterations described in subsection 4.3.1 and the methods based on GKS described in subsection 4.3.2 can handle regularization terms of the form $\mathcal{R}(\mathbf{x}) = \|L(\mathbf{x})\|_p^p$, where L is a linear operator: this can be done efficiently if, e.g., $L(\mathbf{x}) = \mathbf{L}\mathbf{x}$, and matrix-vector products with $\mathbf{L}$ are cheap to compute. Note that the discrete total variation (TV) regularization functional can be expressed in this framework. Specifically, discrete anisotropic TV can be expressed as

$$\mathrm{TV}(\mathbf{x}) = \|L(\mathbf{x})\|_1, \quad \text{where} \quad L(\mathbf{x}) = \begin{bmatrix} \mathbf{I}_{\hat{n}} \otimes \mathbf{D}_{1\mathrm{d}} \\ \mathbf{D}_{1\mathrm{d}} \otimes \mathbf{I}_{\hat{n}} \end{bmatrix} \mathbf{x}$$

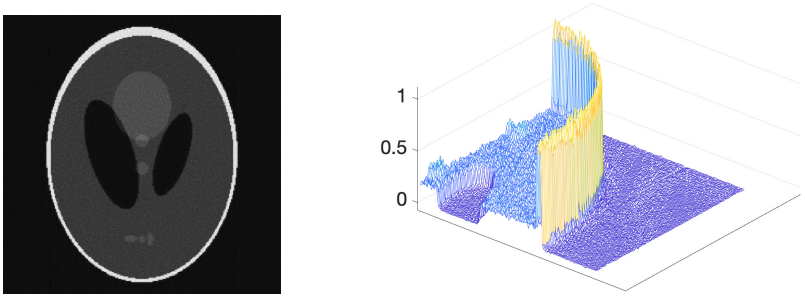


FIG. 4.2. *For the tomography example, we display the reconstruction obtained at the 91th iteration of a hybrid method based on the flexible Golub-Kahan algorithm (4.20) for ℓ_1-norm regularization. The relative reconstruction error is $\|\mathbf{x}_{\text{true}} - \mathbf{x}_{91}\|_2/\|\mathbf{x}_{\text{true}}\|_2 = 0.1046$. On the left the reconstructed phantom is displayed as a grayscale image; one the right, the the top right corner of the reconstructed phantom is displayed as a surface.*

and where $\hat{n}^2 = n$, $\otimes$ is the Kronecker product, and $\mathbf{D}_{1\text{d}}$ is a one-dimensional finite difference discretization of the derivative operator (so that the upper and lower block in $\mathbf{D}$ represent discretizations of the derivatives in the vertical and the horizontal directions of the vectorized image $\mathbf{x}$, respectively. The discrete isotropic TV can be expressed as

$$\text{TV}(\mathbf{x}) = \|L(\mathbf{x})\|_{2,1} = \sum_i \|\mathbf{e}_i^\top L(\mathbf{x})\|_2, \; L(\mathbf{x}) = \left[(\mathbf{I}_{\hat{n}} \otimes \mathbf{D}_{1\text{d}})\mathbf{x}, (\mathbf{D}_{1\text{d}} \otimes \mathbf{I}_{\hat{n}})\mathbf{x}\right]. \tag{4.24}$$

Both the discrete TV functionals have the effect of enforcing sparsity of the unknown $L(\mathbf{x})$ (i.e., the (magnitude of the) discrete gradient of $\mathbf{x}$).

An IRN method for handling TV-like regularizers was proposed in [237] that employs CGLS during the inner iterations, while [6] uses preconditioned LSQR applied to a problem that is effectively transformed into standard form. The transformed problem amplifies directions spanning the columns of the prior covariance matrix, thereby improving convergence. The authors of [83] propose an inner-outer iterative method that enhances edges through multiplicative updates of the weights, and which exploits the hybrid projection method based on joint bidiagonalization method (4.3). The methods based on flexible Krylov subspaces described in subsection 4.3.3 can be as well extended to handle problem (4.23), but the strategies adopted to achieve this are regularizer-dependent: in other words, no universal approach is possible, even for the special case $\mathcal{R}(\mathbf{x}) = \|\mathbf{L}\mathbf{x}\|_p^p$. For instance, [54] describes an approach that works when $\mathbf{L}$ is an orthogonal wavelet transform for methods based on both the flexible Arnoldi and the flexible Golub-Kahan algorithms. Also, [84] describes an extension of FGMRES that handles total variation (4.24) by first transforming the original problem into standard form. Finally, hybrid Krylov projection methods based on either an inner-outer iterative scheme or a flexible scheme, have been used for nuclear norm regularization $\mathcal{R}(\mathbf{x}) = \|\mathbf{x}\|_*$, which is meaningful when $\mathbf{x}$ is a vectorization of a low-rank 2D quantity (such as a low-rank 2D image); see [87].

4.4. Beyond deterministic inversion: Hybrid projection methods in a Bayesian setting. Recently, hybrid projection methods have found utility in the framework of statistical inverse problems, within a Bayesian approach. Our goal in

this section is to draw important connections between deterministic and statistical inverse problems, so that the reader can understand the role that hybrid projection methods can play in the Bayesian setting. For more detailed descriptions, we refer the reader to excellent books and reviews on Bayesian inverse problems [42, 48, 49, 155] and computational uncertainty quantification [10], and references therein.

Traditionally, regularization can be interpreted as a *practical* approach to replace an ill-posed problem by a nearby well-posed one. For example, Tikhonov regularization was originally motivated as a tool to stabilize the solution process by computing one single deterministic solution of a modified problem [224, 225]; see also section 2. However, the Bayesian framework provides a systematic way for solving and analyzing inverse problems by modelling the unknown as a random variable. Although Bayesian inverse problems can be analyzed in a continuous setting [222], here we focus on discrete inverse problems, where the underlying mathematical models have been discretized before applying Bayesian analysis.

In subsection 4.4.1 we will draw connections between regularization and Bayesian inversion. Then, in subsections 4.4.2 and 4.4.3, we describe some computational tools for UQ that can exploit hybrid projection methods. These tools bridge developments in the numerical linear algebra community with those in the UQ community for more efficient and practical inversion that can benefit a wide range of applications.

4.4.1. Connections between regularization and Bayesian inversion. Consider the stochastic extension of (1.1) where $\mathbf{b}, \mathbf{x}$, and $\mathbf{e}$ are random variables. For notational clarity, note that we have dropped the subscript on $\mathbf{x}$. Assume that $\mathbf{x}$ and $\mathbf{e}$ are independent and normally distributed,

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \alpha^2 \mathbf{Q}) \quad \text{and} \quad \mathbf{e} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{R}) \tag{4.25}$$

where $\boldsymbol{\mu} \in \mathbb{R}^n$ is the prior mean and $\mathbf{Q} \in \mathbb{R}^{n \times n}$ and $\mathbf{R} \in \mathbb{R}^{m \times m}$ are symmetric positive definite (SPD) covariance matrices for the prior and noise respectively. Here we assume that α and σ are known. Although it is typically not necessary to include these parameters (i.e., they can be absorbed in the definition of $\mathbf{Q}$ and $\mathbf{R}$), we include them for clarity and to draw connections to hybrid projection methods in subsection 4.4.3.

The probability density function for $\mathbf{b}$ given $\mathbf{x}$ is given by the *likelihood function*,

$$\pi_{\text{like}}(\mathbf{b} \mid \mathbf{x}) = \left(\frac{1}{2\pi\sigma^2 |\mathbf{R}|} \right)^{m/2} \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2_{\mathbf{R}^{-1}} \right),$$

where $|\cdot|$ denotes the determinant of a matrix and $\|\mathbf{z}\|^2_{\mathbf{M}} = \mathbf{z}^\top \mathbf{M} \mathbf{z}$ for any $\mathbf{z} \in \mathbb{R}^n$ and $\mathbf{M} \in \mathbb{R}^{n \times n}$ SPD. The estimate that maximizes the likelihood function (dubbed the maximum likelihood estimator or MLE) is the solution to an unregularized *weighted* least-squares problem, i.e.,

$$\begin{aligned} \mathbf{x}_{\text{MLE}} &= \arg\max_{\mathbf{x}} \pi_{\text{like}}(\mathbf{b} \mid \mathbf{x}) \\ &= \arg\min_{\mathbf{x}} \frac{1}{2\sigma^2} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2_{\mathbf{R}^{-1}} = \arg\min_{\mathbf{x}} \frac{1}{2\sigma^2} \|\mathbf{L}_{\mathbf{R}}(\mathbf{b} - \mathbf{A}\mathbf{x})\|_2^2 \end{aligned}$$

where maximizing the likelihood function is equivalent to minimizing the negative log-likelihood function and $\mathbf{R}^{-1} = \mathbf{L}_{\mathbf{R}}^\top \mathbf{L}_{\mathbf{R}}$ is a symmetric (e.g., Cholesky) factorization. Note that if $\mathbf{R} = \mathbf{I}_m$, then we get a standard least-squares problem. The likelihood function is defined solely based on the assumptions for the noise model and extends naturally for nonlinear problems, e.g., for nonlinear forward model, $\mathbf{b} = F(\mathbf{x}) + \mathbf{e}$

where $F(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$ and $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$, the MLE estimate is the solution to the nonlinear least-squares problem, $\min_{\mathbf{x}} \frac{1}{2\sigma^2}\|\mathbf{b} - F(\mathbf{x})\|^2_{\mathbf{R}^{-1}}$. Additive Gaussian noise is the most common noise model used in the literature, but it is worth mentioning that in many medical and atmospheric imaging applications, a Poisson noise model is appropriate. Specifically, in the linear case and assuming independence, $[\mathbf{b}]_i \sim Poisson([\mathbf{Ax}]_i)$; the likelihood model takes the form

$$\pi_{\text{like}}(\mathbf{b} \mid \mathbf{x}) = \prod_{i=1}^{m} \frac{[\mathbf{Ax}]_i^{[\mathbf{b}]_i}}{[\mathbf{b}]_i!} \exp(-[\mathbf{Ax}]_i),$$

where the notation $[\cdot]_i$ stands for the ith entry of a vector. For general data-fit terms, hybrid projection methods described in subsection 4.3 can be used to approximate the MLE. For the remainder of this section, we consider additive Gaussian noise.

Next, we consider assumptions on prior information about $\mathbf{x}$, as described in (4.25), where the prior density function is given by

$$\pi_{\text{prior}}(\mathbf{x}) = \left(\frac{1}{2\pi\alpha^2|\mathbf{Q}|}\right)^{n/2} \exp\left(-\frac{1}{2\alpha^2}\|\mathbf{x} - \boldsymbol{\mu}\|^2_{\mathbf{Q}^{-1}}\right).$$

Using Bayes' Theorem, we can derive the posterior density function

$$\begin{aligned} \pi_{\text{post}}(\mathbf{x} \mid \mathbf{b}) &= \frac{\pi_{\text{like}}(\mathbf{b} \mid \mathbf{x})\pi_{\text{prior}}(\mathbf{x})}{\pi(\mathbf{b})} \\ &\propto \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{b} - \mathbf{Ax}\|^2_{\mathbf{R}^{-1}} - \frac{1}{2\alpha^2}\|\mathbf{x} - \boldsymbol{\mu}\|^2_{\mathbf{Q}^{-1}}\right) \end{aligned} \tag{4.26}$$

where $\propto$ stands for 'proportional to'. *A crucial point to note is that for Bayesian inverse problems, the posterior density function is the solution to the inverse problem.* The posterior distribution provides the full information about the distribution of parameters in $\mathbf{x}$, given the data $\mathbf{b}$. However, for practical interpretation and data analysis, it is necessary to describe various characteristics of the posterior distribution [223]; thus leading to the field of *uncertainty quantification*.

For example, a common goal is to compute the realization of largest a posteriori probability corresponding to the maximum of (4.26), i.e., the so-called maximum a posteriori (MAP) estimate. We denote this as

$$\begin{aligned} \mathbf{x}_{\text{MAP}} &= \arg\max_{\mathbf{x}} \pi_{\text{post}}(\mathbf{x} \mid \mathbf{b}) \\ &= \arg\min_{\mathbf{x}} \frac{1}{2\sigma^2}\|\mathbf{b} - \mathbf{Ax}\|^2_{\mathbf{R}^{-1}} + \frac{1}{2\alpha^2}\|\mathbf{x} - \boldsymbol{\mu}\|^2_{\mathbf{Q}^{-1}} \end{aligned} \tag{4.27}$$

$$= \left(\mathbf{A}^\top\mathbf{R}^{-1}\mathbf{A} + \lambda\mathbf{Q}^{-1}\right)^{-1}\left(\mathbf{A}^\top\mathbf{R}^{-1}\mathbf{b} + \lambda\mathbf{Q}^{-1}\boldsymbol{\mu}\right), \tag{4.28}$$

where $\lambda = \frac{\sigma^2}{\alpha^2}$ and the closed form solution in (4.28) can be obtained by setting the derivative to zero.

Furthermore, it can be shown that, under the above assumptions (linear problem with Gaussian noise and prior), the posterior is Gaussian, where the posterior covariance matrix and the posterior mean (i.e., the MAP estimate) are given as

$$\boldsymbol{\Gamma} \equiv \left(\tfrac{1}{\alpha^2}\mathbf{Q}^{-1} + \tfrac{1}{\sigma^2}\mathbf{A}^\top\mathbf{R}^{-1}\mathbf{A}\right)^{-1} \quad \text{and} \quad \mathbf{x}_{\text{MAP}} = \boldsymbol{\Gamma}\left(\tfrac{1}{\sigma^2}\mathbf{A}^\top\mathbf{R}^{-1}\mathbf{b} + \tfrac{1}{\alpha^2}\mathbf{Q}^{-1}\boldsymbol{\mu}\right), \tag{4.29}$$

respectively [155]. In the following sections, we describe approaches that can exploit hybrid projection methods for MAP estimation and subsequent UQ.

4.4.2. Hybrid projection methods for MAP estimation. Recall that the solution to a Bayesian inverse problem is the posterior distribution, and the goal is to describe and explore it. For linear inverse problems with Gaussian priors, if the regularization parameters σ and α are set in advance, the posterior is Gaussian with covariance matrix (4.29), and various tools can be used for UQ. However, if the regularization parameters are *not* known in advance, one may consider a fully Bayesian framework, where all unknown parameters (including the regularization parameters) are treated as random variables, and use hierarchical models [10], also defining hyperpriors for these hyperparameters. For example, for Gaussian priors, since σ and α are unknown, we can assume that they are random variables with hyperpriors $\pi(\sigma)$ and $\pi(\alpha)$, respectively. Then, using again Bayes' law, we have the full posterior density,

$$\pi_{\text{post}}(\mathbf{x}, \sigma, \alpha|\mathbf{b}) \propto \pi_{\text{like}}(\mathbf{b}|\mathbf{x})\pi_{\text{prior}}(\mathbf{x})\pi(\sigma)\pi(\alpha). \tag{4.30}$$

For linear inverse problems with Gaussian priors, common choices are Gamma hyperpriors, i.e.,

$$\pi(\sigma) \propto \sigma^{\beta_\sigma - 1} \exp(-\gamma_\sigma \sigma),$$
$$\pi(\alpha) \propto \alpha^{\beta_\alpha - 1} \exp(-\gamma_\alpha \alpha),$$

where $\beta_\sigma, \gamma_\sigma, \beta_\alpha$, and γ_α are parameters. In addition to requiring the user to predefine these additional parameters for the hyperpriors, a potential computational disadvantage of this approach is that the posterior (4.30) is no longer Gaussian, so more sophisticated sampling techniques (e.g., hierarchical Gibbs sampling) should be used.

Instead, we can consider an alternative approach by observing that (4.27) is a general-form Tikhonov problem, thus establishing a main connection between hybrid projection methods and Bayesian inverse problems. Therefore, a simple *practical* alternative to adopting a full Bayesian paradigm is to first use a hybrid projection method to compute a MAP estimate efficiently while selecting the regularization parameter automatically (often based on some statistical tools, see subsection 3.3), and, once this is done, perform standard UQ with fixed hyperparameters [209].

Next we discuss some examples of priors and hybrid methods that can be used to approximate the corresponding MAP estimates. *Gaussian priors* are the most common priors used in the literature. We split the discussion into three scenarios. First, for cases where the symmetric factorization of the precision matrix (i.e., the inverse of the covariance matrix) is computationally feasible, i.e., $\mathbf{Q}^{-1} = \mathbf{L}_\mathbf{Q}^\top \mathbf{L}_\mathbf{Q}$, the MAP estimate is the solution to the optimization problem,

$$\min_{\mathbf{x}} \frac{1}{2\sigma^2}\|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2_{\mathbf{R}^{-1}} + \frac{1}{2\alpha^2}\|\mathbf{L}_\mathbf{Q}(\mathbf{x} - \boldsymbol{\mu})\|_2^2.$$

With a simple change of variables, $\mathbf{y} = \mathbf{x} - \boldsymbol{\mu}$ and $\mathbf{d} = \mathbf{b} - \mathbf{A}\boldsymbol{\mu}$, the problem reduces to solving

$$\min_{\mathbf{y}} \frac{1}{2\sigma^2}\|\mathbf{d} - \mathbf{A}\mathbf{y}\|^2_{\mathbf{R}^{-1}} + \frac{1}{2\alpha^2}\|\mathbf{L}_\mathbf{Q}\mathbf{y}\|_2^2.$$

Let $\mathbf{R} = \mathbf{I}_m$: then, this is general-form Tikhonov problem (4.1), and hybrid projection methods to solve this problem were discussed in subsection 4.1. For this scenario, a prevalent approach in the literature models $\mathbf{L}_\mathbf{Q}$ as a sparse discretization of a differential operator (for example, the Laplacian or the biharmonic operator). This choice corresponds to a covariance matrix that represents a Gauss-Markov random field.

Since for these choices, the precision matrix is sparse, working with $\mathbf{L_Q}$ directly has obvious computational advantages.

Second, for the case where application of $\mathbf{Q}^{-1}$ is feasible (e.g, it is a sparse matrix) but computing the symmetric factorization is too expensive, a factorization-free preconditioned LSQR approach called MLSQR was proposed in [6]. The approach is analytically equivalent to LSQR applied to a preconditioned least-squares problem with the preconditioner $\mathbf{L}_{\mathbf{Q}}^{-1}$, but avoids the factorization by using weighted inner products to implicitly apply the preconditioner. The authors considered applications in the context of nonlinear regularizers.

Third, there are many prior models where the precision matrix $\mathbf{Q}^{-1}$ and its factorization are not available, but matrix-vector multiplications with $\mathbf{Q}$ can be done efficiently. For example, for Gaussian random fields, entries of the covariance matrix are computed directly as $q_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$, where $\{\mathbf{x}_i\}_{i=1}^n$ are the spatial points in the domain and κ is a covariance kernel function (e.g., γ-exponential, or Matérn class). Although there are many modeling advantages, the main challenge is that the resulting prior covariance matrices are often very large and dense; explicitly forming and factorizing these matrices is prohibitively expensive. For such prior models, efficient matrix-free techniques (e.g., FFT embedding and $\mathcal{H}$-matrix approaches) can be used, for example, to compute matrix-vector products with the prior covariance matrix $\mathbf{Q}$. Generalized hybrid projection methods were proposed in [61] to compute Tikhonov regularized solutions (i.e., MAP estimate (4.27)) effectively. The approach relies on a change of variables and uses a generalized GKB algorithm for projection [5]; note that this is a fundamentally different algorithm than the one presented in subsection 4.3.2 for $\ell_2 - \ell_p$ regularization that is based on GKB (for initialization) and some generalization of Krylov projection methods.

Non-Gaussian priors are also common in the literature, but are more challenging to handle. One class of priors contains the so-called *sparsity-promoting priors* or ℓ_p-priors, which have the density function,

$$\pi_{\text{prior}}(\mathbf{x}) = \exp(-\alpha\|\mathbf{x}\|_p^p), \quad \text{for} \quad 0 < p \leq 1. \tag{4.31}$$

One of the main challenges in considering (4.31) is that the posterior distribution is no longer Gaussian. However, one could still compute the MAP estimate, which is the solution to optimization problem,

$$\min_{\mathbf{x}} \frac{1}{2\sigma^2}\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_{\mathbf{R}^{-1}}^2 + \alpha\|\mathbf{x}\|_p^p.$$

Note that $p = 1$ corresponds to a prior where the components of $\mathbf{x}$ are independent and follow the univariate Laplace$(0, \alpha^{-1})$ distribution [10, Chapter 4.3]; the Laplacian prior promotes sparsity in $\mathbf{x}$. On a related note, *Besov priors* can preserve sharp discontinuous interfaces in Bayesian inversion, and when combined with a wavelet-based approach, they can promote sparsity in the MAP estimates [30, 66, 229] but, to the best of our knowledge, hybrid projection methods have not been used in this setting. For many non-Gaussian priors, the methods described in subsection 4.3 may be used to approximate the MAP estimate.

Similar approaches have been considered for hierarchical Bayesian models, where hyperparameters (e.g., from hyperpriors from generalized gamma distributions) are estimated during MAP estimation [43]. For example, priorconditioned Krylov subspace solvers with early stopping (e.g., via the Morozov discrpancy principle) have been used within alternating optimization schemes for reconstructing sparse solutions

[42, 47]. Such approaches can be used to estimate a very large number of hyperparameters (e.g, prior variances for individual components of the solution), but require solving a sequence of optimization problems similar to reweighted inner-outer iterations, c.f. subsection 4.3.

4.4.3. Exploiting hybrid projection methods for variance-covariance estimation and sampling. Handling unknown hyperparameters is not the only task where hybrid projection methods can be beneficial for efficient UQ. In the remaining part of this section, we consider other *practical* scenarios that combine hybrid projection methods (in particular, low-rank approximations) with UQ. We still focus on the Gaussian, linear case where the posterior is given in (4.26) and consider the simple case where $\mathbf{R} = \mathbf{I}_m$ and $\mathbf{Q} = \mathbf{I}_n$.

Assume that a hybrid projection method was used to compute an estimate $\mathbf{x}_k(\lambda_k)$ of $\mathbf{x}_{\text{MAP}}$ where λ_k was determined using techniques in subsection 3.3. The posterior variances corresponding to the diagonals of the posterior covariance matrix $\mathbf{\Gamma}$ in (4.29) provide a measure of the spread of the posterior distribution around the posterior mean. For many problems, $\mathbf{\Gamma}$ is large and dense, so forming it explicitly to obtain the diagonal entries may be infeasible. Instead, we can exploit available information from the subspace projection process (c.f., subsection 3.2.1) to estimate the posterior variances.

Using notation from subsection 3.2.3, after k iterations of the projection process, we have matrices $\mathbf{G}_k, \mathbf{U}_{k+1}$ and $\mathbf{V}_k$ satisfying (3.9). Let $\mathbf{G}_k^\top \mathbf{G}_k = \mathbf{V}^{\text{G}}(\mathbf{\Sigma}^{\text{G}})^\top \mathbf{\Sigma}^{\text{G}}(\mathbf{V}^{\text{G}})^\top$ be the eigenvalue decomposition with eigenvalues $(\sigma_1^{\text{G}})^2, \dots, (\sigma_k^{\text{G}})^2$ and let $\mathbf{Z}_k = \mathbf{V}_k \mathbf{V}^{\text{G}}$, then we get the following low-rank approximation,

$$\mathbf{A}^\top \mathbf{A} \approx \mathbf{V}_k \mathbf{G}_k^\top \mathbf{G}_k \mathbf{V}_k^\top = \mathbf{Z}_k (\mathbf{\Sigma}^{\text{G}})^\top \mathbf{\Sigma}^{\text{G}} \mathbf{Z}_k^\top . \tag{4.32}$$

Assume we have an estimate of the noise variance σ^2 and fix $\lambda = \lambda_k$, then using (4.32) and the Woodbury formula, we obtain the approximation

$$\begin{aligned} \mathbf{\Gamma} = \sigma^2(\lambda \mathbf{I}_n + \mathbf{A}^\top \mathbf{A})^{-1} &\approx \sigma^2(\lambda \mathbf{I}_n + \mathbf{Z}_k (\mathbf{\Sigma}^{\text{G}})^\top \mathbf{\Sigma}^{\text{G}} \mathbf{Z}_k^\top)^{-1} \\ &= \sigma^2(\lambda^{-1}\mathbf{I}_n - \lambda^{-1}\mathbf{Z}_k(\mathbf{I}_k + \lambda((\mathbf{\Sigma}^{\text{G}})^\top \mathbf{\Sigma}^{\text{G}})^{-1})^{-1}\mathbf{Z}_k^\top) \\ &= \sigma^2(\lambda^{-1}\mathbf{I}_n - \mathbf{Z}_k \mathbf{\Delta}_k \mathbf{Z}_k^\top) \equiv \mathbf{\Gamma}_k \end{aligned}$$

where

$$\mathbf{\Delta}_k \equiv \lambda^{-1} \begin{bmatrix} \frac{(\sigma_1^{\text{G}})^2}{(\sigma_1^{\text{G}})^2+\lambda} & & \\ & \ddots & \\ & & \frac{(\sigma_k^{\text{G}})^2}{(\sigma_k^{\text{G}})^2+\lambda} \end{bmatrix} \in \mathbb{R}^{k \times k}. \tag{4.33}$$

Notice that we have an efficient representation of $\mathbf{\Gamma}_k$ as a low-rank perturbation of the prior covariance matrix, $\sigma^2\lambda^{-1}\mathbf{I}_n$. Thus, diagonal entries of $\mathbf{\Gamma}_k$ can provide estimates of diagonal entries of $\mathbf{\Gamma}$, where the main computational requirement is to obtain the diagonals of the rank-k perturbation. In addition, one can approximate the sum of all values in the posterior covariance matrix $\mathbf{1}^\top \mathbf{\Gamma}\, \mathbf{1}$ as

$$\mathbf{1}^\top \mathbf{\Gamma}_k\, \mathbf{1} = \sigma^2(\lambda^{-1} n - \mathbf{1}^\top \mathbf{Z}_k \mathbf{\Delta}_k \mathbf{Z}_k^\top \mathbf{1}) \tag{4.34}$$

where $\mathbf{1}$ is an $n \times 1$ vector of ones. These approximations were considered for dynamic inverse problems using generalized Golub-Kahan approaches in [52, 62].

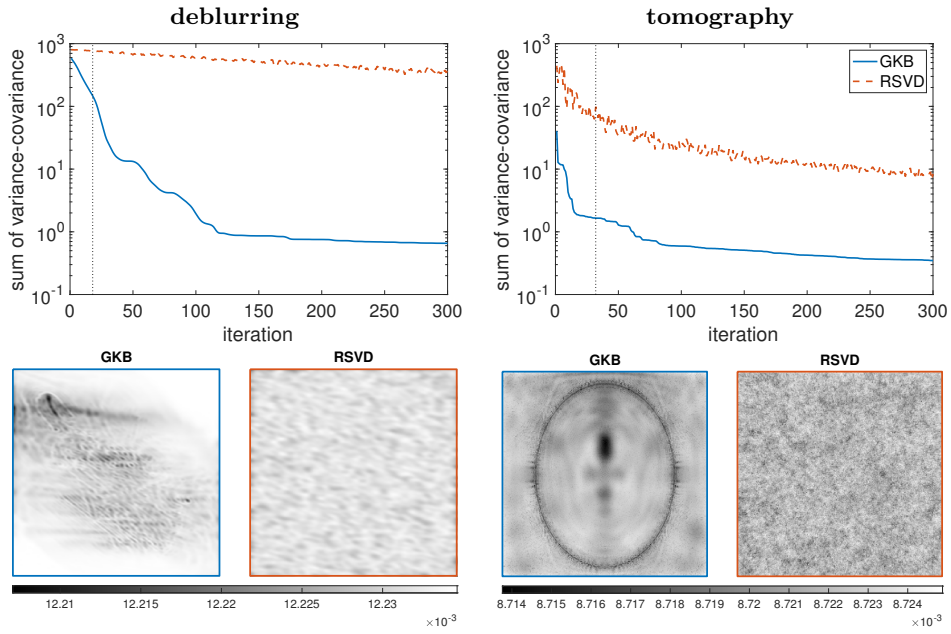


FIG. 4.3. *For the deblurring and tomography examples, we provide estimates of* $\mathbf{1}^\top \mathbf{\Gamma}\, \mathbf{1}$ *using both the GKB approximation and the RSVD approximation. Solution variances computed at the stopping iterate (corresponding to the vertical line) are provided for GKB. Solution variances that can be obtained using an RSVD approximation with comparable computational cost are provided as well.*

In fact, there has been plenty of work at the intersection of Krylov methods and UQ. For example, the idea of using low-rank perturbative approximations for the posterior covariance matrix previously appeared in [29, 31, 77, 221]. Along these lines, there has also been some work on using randomized approaches to efficiently compute a low-rank approximation for use in UQ settings [210]; however, such methods are not ideal for inverse problems where the decay of the singular values is not sufficiently rapid (e.g., in tomography applications). In Figure 4.3, we provide estimates of (4.34) at various iterations k of the GKB process. In exact arithmetic, the values converge to $\mathbf{1}^\top \mathbf{\Gamma}\, \mathbf{1}$. We also provide estimates that use a randomized SVD approximation of the original matrix $\mathbf{A}$, where to ensure that the main computational cost is the same, the iteration corresponds to the number of matrix-vector multiplications with $\mathbf{A}$ and $\mathbf{A}^\top$. See Appendix C for details on the RSVD approximation. We observe that for both the deblurring and tomography examples, the GKB approximations of the sum of elements in the posterior covariance matrix exhibit faster decay (and hence better approximations) than the RSVD approximations. For the automatically selected stopping iteration, denoted with a vertical line, we provide an image of the solution variances (corresponding to the diagonal entries of $\mathbf{\Gamma}_k$) for both GKB and RSVD.

Theoretical results on approximating the posterior distribution and posterior covariance matrix can be found in [209, 215]. Beyond variance-covariance estimation, previous work on Lanczos methods for sampling from Gaussian distributions can be found in, e.g., [53, 188, 212, 217], but these algorithms are meant for sampling from

generic Gaussian distributions and do not exploit the structure of the posterior covariance matrix (4.26). Low-rank approximations for proposal sampling are commonly used for efficient UQ. In [26], the authors consider a fully Bayesian approach with assigned hyperpriors and use MCMC methods with Metropolis-Hastings independence sampling with a proposal distribution based on a low-rank approximation of the prior-preconditioned Hessian. In [208], these ideas are combined with marginalization. We point the interested reader to [10, 49] for more examples where Krylov methods are used for UQ.

4.5. Beyond linear forward models: Hybrid projection methods for nonlinear inverse problems. In this section, we consider the role that hybrid projection methods have played in the context of solving nonlinear inverse problems. We remark that the literature on nonlinear inverse problems is vast, and it is not our intention to survey this topic. We point the interested reader to books on this topic, see e.g., [74, 171, 213], and we focus on frameworks and methodologies for solving nonlinear inverse problems that have successfully utilized or incorporated hybrid projection methods.

We are concerned with nonlinear inverse problems, where the forward model depends nonlinearly on the desired parameters. Consider the *discrete* nonlinear inverse problem,

$$\mathbf{b} = F(\mathbf{x}_{\text{true}}) + \mathbf{e} \tag{4.35}$$

where $F : \mathbb{R}^n \to \mathbf{R}^m$ is a transformation representing the forward model. Notice that the linear model (1.1) is a special case of (4.35) with $F(\mathbf{x}_{\text{true}}) = \mathbf{A}\mathbf{x}_{\text{true}}$.

Regularization methods for nonlinear inverse problems typically follow a variational approach, where the goal is to solve an optimization problem of the form,

$$\min_{\mathbf{x}} \mathcal{J}(\mathbf{b}, F(\mathbf{x})) + \lambda \mathcal{R}(\mathbf{x}) \tag{4.36}$$

where similar to (1.5), $\mathcal{J}$ is some loss function, $\mathcal{R}$ is a regularization operator, and $\lambda \geq 0$ is a regularization parameter.

There are two main computational difficulties in solving problems like (4.36), especially for large-scale problems. First, the regularization parameter is unknown, and estimating it may require a significant computational effort to solve the same nonlinear optimization problem multiple times for various parameter choices, which can be a very expensive and time-consuming task; see [73]. Second, since the optimization functional to be minimized is nonlinear, gradient-based iterative techniques (even Newton-type approaches) are needed, but computing derivatives can be costly [112]. We remark that similar challenges arise for linear inverse problems with nonlinear regularization terms (see, e.g., (4.10)), where sophisticated nonlinear optimization schemes are required; we point the reader to the discussion in subsection 4.3 for some efficient approaches to deal with these kinds of nonlinearities. Alternatively, a two-stage method could also be used that splits the inversion process. First the misfit $\mathcal{J}$ is reduced to some target misfit value. In the second stage, the target misfit is kept constant and the regularization term $\mathcal{R}$ is reduced. Although very popular in practice, this approach is not guaranteed to converge (in fact it diverges in some cases) and appropriate safety steps and ad hoc parameters must be used; see [65, 189].

Specifically for nonlinear least-squares problems, other solvers can be used, such as the one described in Haber and Oldenburg [114], which combines a damped Gauss-Newton method for local regularization with a GCV method for adaptively selecting

the global regularization parameter. To the best of our knowledge, this is the first paper that exploits hybrid projection methods for selecting the regularization parameter for solving large-scale nonlinear inverse problems. Thus, we further describe the approach. Consider problem (4.36) where $\mathcal{J}(\mathbf{b}, F(\mathbf{x})) = \|F(\mathbf{x}) - \mathbf{b}\|_2^2$ and $\mathcal{R}(\mathbf{x}) = \|\mathbf{x}\|_2^2$, such that we get the Tikhonov-regularized nonlinear problem,

$$\min_{\mathbf{x}} \|F(\mathbf{x}) - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_2^2 . \tag{4.37}$$

Let's begin by assuming that λ is fixed. If we differentiate the objective function in (4.37) with respect to $\mathbf{x}$ and set the result to zero, we get

$$\mathbf{g}(\mathbf{x}) = 2(\mathbf{J}(\mathbf{x})^\top (F(\mathbf{x}) - \mathbf{b}) + \lambda \mathbf{x}) = \mathbf{0}, \tag{4.38}$$

where $\mathbf{g}(\mathbf{x})$ is the gradient and $\mathbf{J}(\mathbf{x}) = \frac{\partial F}{\partial \mathbf{x}} \in \mathbb{R}^{m \times n}$ is the Jacobian matrix that contains the partial derivatives or sensitivities of the forward model. If we find a solution to (4.38), we have the desired solution to (4.37) for fixed λ. Rather than using Newton's method, which requires calculating the second derivative of F with respect to $\mathbf{x}$ (i.e., differentiation of $\mathbf{g}(\mathbf{x})$), a damped Gauss-Newton approach can be used. That is, let $F(\mathbf{x} + \delta\mathbf{x}) = F(\mathbf{x}) + \mathbf{J}(\mathbf{x})\delta\mathbf{x} + R(\mathbf{x}, \delta\mathbf{x})$, be a linearization of F. Since $R(\mathbf{x}, \delta\mathbf{x})$ is assumed small, we aim to solve the Gauss-Newton equations,

$$(\mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x}) + \lambda \mathbf{I}_n)\delta\mathbf{x} = -\mathbf{J}(\mathbf{x})^\top (F(\mathbf{x}) - \mathbf{b}) - \lambda \mathbf{x}. \tag{4.39}$$

or equivalently the least-squares problem,

$$\min_{\delta\mathbf{x}} \left\| \begin{bmatrix} \mathbf{J}(\mathbf{x}) \\ \sqrt{\lambda}\mathbf{I}_n \end{bmatrix} \delta\mathbf{x} - \begin{bmatrix} \mathbf{b} - F(\mathbf{x}) \\ \sqrt{\lambda}\mathbf{x} \end{bmatrix} \right\|. \tag{4.40}$$

The Gauss-Newton method is an iterative approach where given an initial guess $\mathbf{x}_0$, the update at the kth iteration is computed by solving (4.40) for the perturbation $\delta\mathbf{x}$ and updating the solution estimate as $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \delta\mathbf{x}$, where α_k is a line-search parameter. Notice that, for a given λ, Krylov iterative methods could be used to efficiently estimate the solution to (4.40). However, the authors of [114] propose a reformulation that enables the direct use of hybrid projection methods described in Section 3. The astute observation is that at each iteration of the Gauss-Newton method, we have the option to solve directly for the step $\delta\mathbf{x}$ or to solve for the updated solution $\mathbf{x}_{k+1}$. To see the latter option, substitute $\mathbf{x}_{k+1} = \mathbf{x}_k + \delta\mathbf{x}$ into (4.39) to get least-squares problem,

$$\min_{\mathbf{x}} \left\| \begin{bmatrix} \mathbf{J}(\mathbf{x}_k) \\ \sqrt{\lambda}\mathbf{I}_n \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b} - F(\mathbf{x}_k) - \mathbf{J}(\mathbf{x}_k)\mathbf{x}_k \\ \mathbf{0} \end{bmatrix} \right\|_2^2 . \tag{4.41}$$

At this point, (4.41) is just a standard-form, linear Tikhonov problem, and hybrid projection methods can be used in each nonlinear step to simultaneously estimate the regularization parameter λ and compute an approximate solution. Although the hybrid projection method solves the linearized problem with automatic regularization parameter selection, a major concern is that the objective function changes from iteration to iteration because the regularization parameter is changing. Furthermore, we are not guaranteed to reduce the nonlinear function that we seek to minimize. A remedy is to consider the solution to (4.41) as a proposal $\mathbf{x}_{k+1}^p$ and to take the step direction to be a weighted version of the perturbation $\delta\mathbf{x} = \mathbf{x}_{k+1}^p - \mathbf{x}_k$. The

Algorithm 4.1 Gauss-Newton with hybrid projection for nonlinear problem (4.37)

Input: F, $\mathbf{b}$, $\mathbf{x}_0$, $k = 0$

1: **while** stopping criterion not satisfied **do**
2: Calculate $\mathbf{J}(\mathbf{x}_k)$ and $\mathbf{r}(\mathbf{x}_k)$
3: Compute $\mathbf{x}_{k+1}^p$ by solving (4.41) using a hybrid projection method
4: Calculate the perturbation $\delta\mathbf{x} = \mathbf{x}_{k+1}^p - \mathbf{x}_k$
5: Update $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha\delta\mathbf{x}$ where α ensures reduction of the nonlinear function
6: $k = k + 1$
7: **end while**

Output: $\mathbf{x}_k$

general damped Gauss-Newton approach with a hybrid Krylov solver is summarized in Algorithm 4.1.

Another adaptive approach to select the regularization parameter when solving nonlinear inverse problems is based on continuation or cooling approaches [112]. By reformulating the Tikhonov problem as a constrained optimization problem where the objective function is given in terms of the regularizer, and the constraints are given in terms of the data misfit, and exploiting the connection between the regularization parameter and the inverse of the Lagrange multiplier for the constraint, the regularization parameter can be selected to be a large value initially, and then gradually reduced to an appropriate value.

Finally, in the context of solving large-scale nonlinear inverse problems, there is another class of nonlinear inverse problems that have greatly benefited from recent developments in hybrid projection methods. These are *separable nonlinear inverse problems*, where the unknown parameters can be separated into two distinct components such that the forward model is linear in one set of parameters and nonlinear in another set of parameters. This situation corresponds to problem (4.35) with

$$F(\mathbf{x}_{\text{true}}) = \mathbf{A}\left(\mathbf{x}_{\text{true}}^{(nl)}\right)\ \mathbf{x}_{\text{true}}^{(l)},$$

where $\mathbf{x}_{\text{true}} = \begin{bmatrix}\mathbf{x}_{\text{true}}^{(nl)} & \mathbf{x}_{\text{true}}^{(l)}\end{bmatrix}^\top$, with $\mathbf{x}_{\text{true}}^{(nl)} \in \mathbb{R}^\ell$ and $\mathbf{x}_{\text{true}}^{(l)} \in \mathbb{R}^{n-\ell}$ and $\mathbf{A}$ is a linear operator defined by a set of nonlinear parameters $\mathbf{x}_{\text{true}}^{(nl)}$. Therefore F is linear in $\mathbf{x}_{\text{true}}^{(l)}$ and nonlinear in $\mathbf{x}_{\text{true}}^{(nl)}$. Furthermore, it is often the case that $\ell \ll n - \ell$. Such problems arise in imaging applications such as blind deconvolution, super-resolution imaging, and motion correction [55, 58], but also in other inverse problems; see [97]. Although one could treat separable nonlinear inverse problems using standard optimization techniques, approaches that exploit their separable structure have been successfully used to improve convergence. For example, a decoupled or alternating optimization approach is a simple approach where, for fixed $\mathbf{x}^{(nl)}$, a hybrid projection method could be used to solve the resulting linear inverse problem, and, for fixed $\mathbf{x}^{(l)}$, a nonlinear optimization scheme can be used to solve the smaller dimensional optimization problem. However, a potential disadvantage of these alternating approaches is slow convergence and sensitivity to initializations. An alternative approach is to use the variable projection method [97, 103, 104, 157, 184], where the linear variables are implicitly eliminated and a reduced optimization problem is solved. Consider the

full Tikhonov problem,

$$\min_{\mathbf{x}^{(nl)},\mathbf{x}^{(l)}} \|\mathbf{A}\left(\mathbf{x}^{(nl)}\right)\mathbf{x}^{(l)} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}^{(l)}\|_2^2 = \left\| \begin{bmatrix} \mathbf{A}\left(\mathbf{x}^{(nl)}\right) \\ \sqrt{\lambda}\mathbf{I}_n \end{bmatrix} \mathbf{x}^{(l)} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_2^2. \tag{4.42}$$

A variable projection approach applied to (4.42) would seek the solution to the reduced optimization problem,

$$\min_{\mathbf{x}^{(nl)}} \|\mathbf{A}(\mathbf{x}^{(nl)})\,\mathbf{x}^{(l)}(\mathbf{x}^{(nl)}) - \mathbf{b}\|_2^2, \tag{4.43}$$

where

$$\mathbf{x}^{(l)}(\mathbf{x}^{(nl)}) = \arg\min_{\mathbf{x}^{(l)}} \left\| \begin{bmatrix} \mathbf{A}(\mathbf{x}^{(nl)}) \\ \sqrt{\lambda}\mathbf{I}_n \end{bmatrix} \mathbf{x}^{(l)} - \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \right\|_2^2. \tag{4.44}$$

Each iteration of a nonlinear optimization method to solve (4.43) would require solving linear subproblems (4.44). Since this is a linear inverse problem, hybrid projection methods can be used (especially in the large-scale setting), and the regularization parameter can be estimated automatically, giving rise to an inner-outer iterative scheme [55, 58]. A more recent hybrid regularization method, which avoids inner-outer iterations by exploiting inexact Krylov solvers, is presented in [86]. For solving inverse problems with coupled variables, an approach called Linearize and Project (LAP) was described in [173] as an alternative to variable projection methods. LAP can support different regularization strategies as well as equality and inequality constraints, and thus is more broadly applicable than variable projection. Hybrid regularization methods are used in LAP to simultaneously compute the search direction and automatically select an appropriate regularization parameter at each iteration.

Although hybrid projection methods have found practical uses in various nonlinear scenarios, there are still open questions and yet-to-be-explored uses of hybrid projection methods for solving nonlinear inverse problems. Recent advancements in flexible hybrid methods and iteratively reweighted approaches for solving linear inverse problems with nonlinear regularization terms have opened the door to new approaches for handling nonlinear optimization problems. Furthermore, the training of many state-of-the-art deep neural networks can be interpreted as a separable nonlinear inverse problems [176], and there is great potential for the use of hybrid projection methods for training deep neural networks [156].

5. Software. We start this section by listing some software packages for inverse problems that contain implementations of some hybrid projection methods. To the best of our knowledge, one of the most recent ones is called IR Tools [82], and it runs in MATLAB. This toolbox provides implementations of a range of iterative solvers, including many of the hybrid projection methods discussed in this paper and extending the GKB-based hybrid solver originally implemented in [172]. Various examples (including image deblurring and tomography reconstruction) are also provided as test problems. For specific details, we point the reader to the IR Tools software package and documentation [82]. Here we only highlight a few important design objectives and considerations for the implementation of hybrid projection methods for inverse problems, which are also adopted within IR Tools:

- The implementation should work for the common scenarios where the coefficient matrix is only available as a (sparse) matrix, a function handle, or an object. Matrix-vector multiplications with the coefficient matrix represent a

core component (and oftentimes the computational bottleneck) of iterative projection methods. High performance or distributed computing tools can be used to accelerate the task of matrix-vector multiplication. For example, hybrid projection methods were used with message passing tools for cryo-EM reconstruction [63] and were included in a high-performance java library in [236].

- The software should enable automatic regularization parameter selection and automatic stopping criteria, but at the same time provide easy-to-use functionalities for users to manually change any algorithm parameter.
- Compared to some standard iterative methods, one of the main disadvantages of hybrid methods is the need to store the basis vectors for solution computation. Thus, it is often assumed that solutions can be captured in relatively few iterations or that appropriate preconditioning can be used. Thus, it is desirable that implementations of hybrid projection methods can incorporate preconditioning techniques or exploit recycling techniques. Another potential issue is the computational costs associated with reorthogonalization, especially with GKB-based hybrid methods [11, 22].
- The software should provide basic outputs (e.g., the solution automatically computed with default solvers' settings) as well as a range of outputs that provide relevant information (e.g., for subsequent UQ or prediction).

The illustrations for this paper were made using the IR Tools software, and MATLAB scripts for reproducing the experiments reported herein are available at https://github.com/juliannechung/surveyhybridprojection. Although IR Tools contains implementations of many state-of-the-art hybrid projection methods, it is by no means exhaustive and the authors encourage further extensions. For some recent work on generalized hybrid projection methods and subsequent UQ (including those described in subsection 4.4), and hybrid methods with recycling (including those described in subsection 4.2), see https://github.com/juliannechung.

Leaving aside hybrid methods' implementations, but still considering software packages written in MATLAB, we point the reader to Regularization Tools [124] for getting familiar with some small-scale test problems and basic direct and iterative regularization methods, Restore Tools [172] for image deblurring problems, and AIR Tools II [132] for tomographic reconstruction problems. For computational UQ, we refer to the accompanying MATLAB codes for the book [10].

Leaving aside MATLAB, a number of general purpose or application-specific libraries for the solution of inverse problems exists, which are mainly written in Python or C++. To the best of our knowledge, at present, such software collections do not contain implementations of hybrid projection methods, and therefore providing an exhaustive account of such packages is outside the scope of this paper. We mention anyway a few of them, which currently appear to be among the most popular ones within communities of researchers in inverse problems. Concerning Python-based general-purpose ones, we list:

- The Operator Discretization Library (ODL) [1], which mainly provides a unified framework for the discretization of a variety of forward operators and optimization algorithms for variational regularization.
- The Inverse Problem PYthon library (hIPPYlib) [230], which implements state-of-the-art algorithms for PDE-based deterministic and Bayesian inverse problems.

Concerning application-specific software, many recent libraries target tomography reconstruction problems, to match the latest technological advances in the field that

enable enhanced and innovative imaging modalities: these packages typically provide extensive modular optimization toolboxes to be used in connection with variational regularization methods, powerful data processing and visualization tools, as well as new challenging test cases and datasets. We mention the Python-based Core Imaging Library (CIL) [154, 187], the Python-based platform tomoPy [110] for the analysis of tomographic datasets acquired with synchroton light sources. Aside from Python, the toolboxes TIGRE [18] and ASTRA [228], and the CCPi-Regularisation toolkit [158], all feature building blocks in C or C++ and user interfaces in MATLAB (TIGRE, ASTRA, CCPi) and Python (CCPi and ASTRA only).

6. Concluding remarks and future outlook. This survey provides an overview of hybrid projection methods for solving large-scale inverse problems. Our aim is to provide a gentle introduction to the area for budding scientists and interested researchers working in related fields. We explain the general principles underlying hybrid projection methods for standard Tikhonov regularization (Algorithm 3.1), and carefully describe the most common algorithmic choices (including subspace projection approaches and parameter choice techniques), and finish with a summary of extensions that go *beyond* standard approaches and uses.

The field of hybrid projection methods has gained significant momentum in the past few years, and there are many new areas of exploration. Large-scale inverse problems continue to motivate the development of more efficient and accurate solvers, which often leverage tools from different areas of numerical linear algebra, optimization, and statistics. We envisage that a lot of scientific research will be devoted to using hybrid projection methods to incorporate more sophisticated priors and for performing subsequent UQ, e.g., for further prediction and forecasting, in forthcoming years. There is still need for the development and analysis of new parameter selection rules, and strategies that can incorporate supervised learning techniques in optimal experimental design and deep learning frameworks seem promising [2]. Efficient implementations of hybrid methods will be needed for advanced distributed computing architectures and sampling techniques for massive or streaming data problems. With the plethora of examples of inverse problems in areas ranging from biomedical imaging to atmospheric monitoring for threat detection, it is inevitable that many scientific and engineering applications will benefit from recent and forthcoming advancements in hybrid projection methods for inverse problems.

7. Acknowledgements. The authors would like to thank Carola-Bibiane Schönlieb for encouraging us to write this review article. This work was partially supported by the National Science Foundation (NSF) under grant DMS-1654175 and 1723005 (J. Chung) and by the Engineering and Physical Sciences Research Council (EPSRC) under grant EP/T001593/1 (S. Gazzola). The authors are especially grateful to Per Christian Hansen, Jim Nagy, Dianne O'Leary, and the anonymous referees for helpful feedback.

Appendix A. SVD formulations of quantities used for regularization parameter selection. As emphasised in subsection 3.3, many of the regularization parameter selection strategies for hybrid projection methods require solving a nonlinear optimization or root finding problem to estimate λ_k. As far as $k \ll \min\{m, n\}$, one can exploit the SVD of the matrix $\mathbf{G}_k$ appearing in the projected problem (3.11) for efficient computation of the norm of the approximate solution, the norm of the residual, and the trace of the influence matrix, which all depend on λ_k. We remark that these efficiencies are especially impactful when these quantities need to be com-

puted for many regularization parameters (e.g., to obtain a representative L-curve, or when many iterations of the nonlinear optimization solver or zero finder for λ_k have to be performed). In this section, we provide SVD formulations for these quantities for Tikhonov regularization, but remark that similar derivations can be made for any spectral filtering method (e.g., TSVD) used to solve the projected problem (3.11) in a hybrid projection method.

Let the SVD of $\mathbf{G}_k \in \mathbb{R}^{(k+1)\times k}$ be given as in (3.10) and let

$$\widetilde{\mathbf{U}} = \mathbf{U}_{k+1}\mathbf{U}^{\mathrm{G}} = \begin{bmatrix} \widetilde{\mathbf{u}}_1 & \cdots & \widetilde{\mathbf{u}}_{k+1} \end{bmatrix} \quad \text{and} \quad \widetilde{\mathbf{V}} = \mathbf{V}_k\mathbf{V}^{\mathrm{G}} = \begin{bmatrix} \widetilde{\mathbf{v}}_1 & \cdots & \widetilde{\mathbf{v}}_k \end{bmatrix};$$

notice that these contain orthonormal columns. Then the solution of the regularized, projected problem (3.12) has the form

$$\begin{aligned} \mathbf{x}_k(\lambda_k) &= \widetilde{\mathbf{V}}((\mathbf{\Sigma}^{\mathrm{G}})^\top\mathbf{\Sigma}^{\mathrm{G}} + \lambda_k\mathbf{I}_k)^{-1}(\mathbf{\Sigma}^{\mathrm{G}})^\top\widetilde{\mathbf{U}}^\top\mathbf{b} \\ &= \sum_{i=1}^{k} \phi_i^{\mathrm{G}} \frac{\widetilde{\mathbf{u}}_i^\top\mathbf{b}}{\sigma_i^{\mathrm{G}}}\widetilde{\mathbf{v}}_i, \quad \text{where} \quad \phi_i^{\mathrm{G}} = \frac{(\sigma_i^{\mathrm{G}})^2}{(\sigma_i^{\mathrm{G}})^2 + \lambda_k}. \end{aligned}$$

The influence matrix (3.13) can then be represented as

$$\mathbf{A}\mathbf{A}^{\dagger}_{\mathrm{reg}}(\lambda_k, k) = \widetilde{\mathbf{U}}\mathbf{\Sigma}^{\mathrm{G}}((\mathbf{\Sigma}^{\mathrm{G}})^\top\mathbf{\Sigma}^{\mathrm{G}} + \lambda_k\mathbf{I}_k)^{-1}(\mathbf{\Sigma}^{\mathrm{G}})^\top\widetilde{\mathbf{U}}^\top.$$

Using the above formulas, the squared norm of the solution can be represented as

$$\|\mathbf{x}_k(\lambda_k)\|^2 = \sum_{i=1}^{k} \left(\phi_i^{\mathrm{G}} \frac{\widetilde{\mathbf{u}}_i^\top\mathbf{b}}{\sigma_i^{\mathrm{G}}}\right)^2$$

and the trace of the influence matrix takes the form

$$\mathrm{trace}\big(\mathbf{A}\mathbf{A}^{\dagger}_{\mathrm{reg}}(\lambda_k, k)\big) = \sum_{i=1}^{k} \phi_i^{\mathrm{G}}.$$

To obtain an expression for the norm of the residual, we exploit the fact that $\widetilde{\mathbf{U}}$ has orthonormal columns to get

$$\begin{aligned} \|\mathbf{r}(\mathbf{x}_k(\lambda_k))\|^2 &= \|\widetilde{\mathbf{U}}\mathbf{\Sigma}^{\mathrm{G}}((\mathbf{\Sigma}^{\mathrm{G}})^\top\mathbf{\Sigma}^{\mathrm{G}} + \lambda\mathbf{I}_k)^{-1}(\mathbf{\Sigma}^{\mathrm{G}})^\top\widetilde{\mathbf{U}}^\top\mathbf{b} - \widetilde{\mathbf{U}}\widetilde{\mathbf{U}}^\top\mathbf{b}\|^2 \\ &= \|(\mathbf{\Sigma}^{\mathrm{G}}((\mathbf{\Sigma}^{\mathrm{G}})^\top\mathbf{\Sigma}^{\mathrm{G}} + \lambda_k\mathbf{I}_k)^{-1}(\mathbf{\Sigma}^{\mathrm{G}})^\top - \mathbf{I}_k)\widetilde{\mathbf{U}}^\top\mathbf{b}\|^2. \end{aligned}$$

Let $\widetilde{\mathbf{b}} = \widetilde{\mathbf{U}}^\top\mathbf{b} \in \mathbb{R}^{k+1}$ with entries $\widetilde{b}_i$, then

$$\|\mathbf{r}(\mathbf{x}_k(\lambda_k))\|^2 = \widetilde{b}_{k+1}^2 + \sum_{i=1}^{k} (\phi_i^{\mathrm{G}} - 1)^2\widetilde{b}_i^2.$$

Thus, computing the norms of candidate solution and residual vectors, and the trace of the influence matrix can be done efficiently exploiting the SVD of $\mathbf{G}_k$, without forming the solution, the residual or the influence matrix.

Appendix B. Parameter choice rules that exploit the links between GKB and Gaussian quadrature rules.

As mentioned in subsection 3.3, connections between GKB and Gaussian quadrature rules can be naturally exploited in the framework of hybrid projection methods,

for both defining the kth projected problem (3.7) or (3.8) and computing an approximation λ_k of the regularization parameter λ for the Tikhonov problem (2.1). Such connections are underpinned by a solid and elegant theory: in this appendix we will mention without proof some basic facts, which are fully unfolded in, e.g., [101]. Since all the parameter choice rules described for Tikhonov regularization in subsection 3.3.2 and subsection 3.3.3 involve the discrepancy norm $\|\mathbf{r}(\mathbf{x}(\lambda))\|$, and since

$$\begin{aligned} \|\mathbf{r}(\mathbf{x}(\lambda))\|^2 &= \lambda^2 \mathbf{b}^\top (\mathbf{A}\mathbf{A}^\top + \lambda \mathbf{I}_m)^{-2} \mathbf{b} \\ &= \lambda^2 \mathbf{b}^\top \psi(\mathbf{A}\mathbf{A}^\top)\mathbf{b}\,, \text{ with } \psi(t) = (t+\lambda)^{-2}\,, \end{aligned} \tag{B.1}$$

in the following we provide the details of the approximation of such a quadratic form and for an overdetermined $\mathbf{A}$ (i.e., $m \geq n$) only. However, the derivations below can be extended to consider other quadratic forms commonly appearing in regularization parameter choice strategies, such as $\|\mathbf{x}(\lambda)\|$.

Considering the eigendecomposition $\mathbf{A}\mathbf{A}^\top = \mathbf{U}^{\text{A}}\mathbf{\Sigma}^{\text{A}}(\mathbf{\Sigma}^{\text{A}})^\top(\mathbf{U}^{\text{A}})^\top$ given in terms of the SVD of $\mathbf{A}$ (2.2), and setting

$$\mathbf{\Lambda}^{\text{A}} := \operatorname{diag}(\lambda_1^{\text{A}}, \dots, \lambda_n^{\text{A}}, 0, \dots, 0) = \mathbf{\Sigma}^{\text{A}}(\mathbf{\Sigma}^{\text{A}})^\top \in \mathbb{R}^{m\times m}, \quad \bar{\mathbf{b}} := \lambda(\mathbf{U}^{\text{A}})^\top \mathbf{b},$$

it can be shown that

$$\lambda^2 \mathbf{b}^\top \psi(\mathbf{A}\mathbf{A}^\top)\mathbf{b} = \bar{\mathbf{b}}^\top \psi(\mathbf{\Lambda}^{\text{A}})\bar{\mathbf{b}} = \int_0^{\lambda_1^{\text{A}}} \psi(t)\, d\omega(t) =: I(\psi)\,, \tag{B.2}$$

where the discrete measure $\omega(t)$ is a non-decreasing step function with jump discontinuities at the λ_i^{A}'s and at the origin. It is well-known that the above integral can be numerically approximated using a Gaussian quadrature rule, whose nodes are the zeros of a family of orthonormal polynomials with respect to the inner product induced by the measure $\omega(t)$. Such polynomials satisfy a three-term recurrence relation whose coefficients coincide with the entries of the tridiagonal matrix computed by the symmetric Lanczos algorithm applied to $\mathbf{A}\mathbf{A}^\top$ with initial vector $\mathbf{b}$ (specifically, the entries on the kth column of such tridiagonal matrix define the kth orthonormal polynomial). Note that such decomposition can be conveniently obtained from the partial GKB decomposition, as shown in (3.4).

Consider the eigendecomposition $\mathbf{T}_{k,k} = \mathbf{U}^{\text{B}}(\mathbf{\Sigma}^{\text{B}})^2(\mathbf{U}^{\text{B}})^\top$ given in terms of the SVD of the matrix $\mathbf{B}_{k,k}$ appearing in (3.3), i.e., $\mathbf{B}_{k,k} = \mathbf{U}^{\text{B}}\mathbf{\Sigma}^{\text{B}}(\mathbf{V}^{\text{B}})^\top$. Then it is well-known that the k-point Gauss quadrature rule for approximating (B.2) is given by

$$\begin{aligned} \mathcal{G}_k(\psi) &= \lambda^2\|\mathbf{b}\|^2 \sum_{j=1}^{k} \psi((\sigma_i^{\text{B}})^2)(\mathbf{e}_1^\top \mathbf{U}^{\text{B}}\mathbf{e}_k)^2 = \lambda^2\|\mathbf{b}\|^2 \mathbf{e}_1^\top \mathbf{U}^{\text{B}} \psi((\mathbf{\Sigma}^{\text{B}})^2)(\mathbf{U}^{\text{B}})^\top \mathbf{e}_1 \\ &= \lambda^2\|\mathbf{b}\|^2 \mathbf{e}_1^\top \psi(\mathbf{T}_k)\mathbf{e}_1 = (\lambda\|\mathbf{b}\|\mathbf{e}_1)^\top \psi(\mathbf{B}_{k,k}\mathbf{B}_{k,k}^\top)(\lambda\|\mathbf{b}\|\mathbf{e}_1)\,, \end{aligned} \tag{B.3}$$

i.e., the quadratic form ψ evaluated with respect to the projected matrices. The k-point Gauss-Radau quadrature rule for approximating (B.2), with one node at the origin, can be evaluated similarly, replacing the matrix $\mathbf{B}_{k,k}\mathbf{B}_{k,k}^\top$ by the matrix $\mathbf{B}_{k-1}\mathbf{B}_{k-1}^\top$, where $\mathbf{B}_{k-1} \in \mathbb{R}^{k\times(k-1)}$ is computed at the $(k-1)$st GKB iteration. That is,

$$\mathcal{R}_k(\psi) = (\lambda\|\mathbf{b}\|\mathbf{e}_1)^\top \psi(\mathbf{B}_{k-1}\mathbf{B}_{k-1}^\top)(\lambda\|\mathbf{b}\|\mathbf{e}_1)\,. \tag{B.4}$$

Since ψ defined in (B.1) is a $2k$-times differentiable function, the quadrature errors $\mathcal{E}_{\mathcal{Q}_k}(\psi) := I(\psi) - \mathcal{Q}_k(\psi)$ associated with the k-point Gauss and Gauss-Radau quadrature rules (i.e., with $\mathcal{Q}_k(\psi) = \mathcal{G}_k(\psi)$ and $\mathcal{Q}_k(\psi) = \mathcal{R}_k(\psi)$, respectively), are given by

$$\mathcal{E}_{\mathcal{G}_k}(\psi) = \frac{\psi^{(2k)}(\zeta_{\mathcal{G}_k})}{(2k)!} \int_0^{+\infty} \prod_{i=1}^{k} (t - \zeta_i)^2 d\omega(t)\,,$$

$$\mathcal{E}_{\mathcal{R}_k}(\psi) = \frac{\psi^{(2k-1)}(\bar{\zeta}_{\mathcal{R}_k})}{(2k-1)!} \int_0^{+\infty} t \prod_{i=2}^{k} (t - \bar{\zeta}_i)^2 d\omega(t)\,,$$

where $\zeta_{\mathcal{G}_k}$, $\bar{\zeta}_{\mathcal{R}_k} \in [0, \lambda_1^{\mathrm{A}}]$, and the ζ_i's and the $\bar{\zeta}_i$'s denote the nodes of the Gauss and the Gauss-Radau quadrature rules, respectively. Since $\psi^{(2k-1)}(t) < 0$ and $\psi^{(2k)}(t) > 0$ for $t \geq 0$, the Gauss-Radau quadrature rule (B.4) is an upper bound for $\|\mathbf{r}(\lambda)\|^2$, while the Gauss quadrature rule (B.3) is a lower bound for $\|\mathbf{r}(\lambda)\|^2$.

The idea of exploiting Gaussian quadrature to approximate functionals used to set the Tikhonov regularization parameter was first presented by Golub and Von Matt [107], with derivations specific for GCV. This investigation can be regarded as a particular case of the broader explorations of the links between some Krylov methods and Gaussian quadrature conducted by Golub and collaborators; see again [101] for a complete overview of the accomplishments in this area. An important remark concerning GCV is that a different treatment is needed to derive bounds for the numerator and denominator of the GCV function (3.22): since a trace term appears in the denominator, a randomized trace estimator can be used to compute its approximation (see, e.g., [107, 207]), while Gauss-quadrature bounds can be used to handle the numerator, which is essentially of the form (B.1). Note, however, that the approach in [107] cannot be regarded as a hybrid solver according to the framework adopted in this paper, since a Krylov projection method is adopted to set the regularization parameter λ for the full-dimensional Tikhonov problem (2.1), which is then solved using an iterative method. To the best of our knowledge, Gaussian-quadrature-based regularization parameter choice strategies were first adapted to work within hybrid solvers in [34]: here the algorithm involved in setting the regularization parameter according to GCV is identical to the one described in [107], and the computations performed to approximate the numerator of the GCV functional in (3.22) are then also employed to define approximation subspaces (of increasing dimensions) for the solution, effectively making this approach a hybrid solver. A similar strategy, still tailored to GCV, is presented in [76].

Regularization parameter choice rules for hybrid methods that exploit the links between GKB and Gaussian quadrature rules are especially successful and well-understood when the L-curve is employed. Gauss and Gauss-Radau quadrature rules provide computationally inexpensive upper and lower bounds (i.e., boxes) for each point on the L-curve associated to the full-dimensional Tikhonov problem, creating a so-called 'L-ribbon'; see, e.g., [35, 45]. Since the L-curve bounds are nested (i.e., they become tighter as the number of GKB iterations increases), when the 'L-ribbon' is narrow it is possible to infer the approximate location of the 'vertex' of the L-curve from its shape. For hybrid methods, the 'L-ribbon' may first conveniently determine narrow bounds for the (direct) Tikhonov L-curve (thereby performing a suitable number of GKB iterations), and then easily locate the approximate vertex of the L-curve. Most recently, the authors of [85] introduce a new principled and adaptive algorithmic approach for regularization, which provides reliable parameter choice rules by leverag-

ing the framework of bilevel optimization, GKB-based hybrid methods, and the links between Gauss quadrature and GKB.

To conclude, exploiting the connections between GKB and Gaussian quadrature rules can lead to the derivation of new regularization parameter choice rules that can naturally be employed within GKB-based hybrid methods; error estimates for Gaussian quadrature rules can allow further theoretical analysis of the approximation properties of GKB-based hybrid methods.

Appendix C. Low-rank Approximation of the Posterior Covariance using RSVD. In subsection 4.4.3, we described low-rank approximation techniques based on hybrid projection methods that can be used for efficient computation of desired variance-covariance estimates. In this section, we describe an approach based on a randomized version of the SVD (henceforth called RSVD) for computing a rank k approximation of an $m \times n$ matrix $\mathbf{A}$ [116]. Then, we use the RSVD approximation to approximate the posterior covariance matrix (see subsection 4.4.3).

The idea underlying the RSVD approach is to find a matrix $\mathbf{Q}$ whose range approximates that of $\mathbf{A}$. This is done by first drawing a standard Gaussian random matrix $\mathbf{G} \in \mathbb{R}^{n\times(k+p)}$, where k is the desired target rank and $p \geq 0$ is an oversampling parameter. Then, the matrix $\mathbf{Y} = \mathbf{AG}$, computed with $k+p$ matrix-vector multiplications with $\mathbf{A}$, contains random linear combinations of columns of $\mathbf{A}$. We can obtain a matrix $\mathbf{Q}$ such that $\text{ran}(\mathbf{Q}) \approx \text{ran}(\mathbf{A})$ using a thin QR factorization $\mathbf{Y} = \mathbf{QR}$. A low-rank approximation can be obtained as

$$\mathbf{A} \approx \mathbf{Q}\underbrace{\mathbf{Q}^\top\mathbf{A}}_{\mathbf{B}} = \mathbf{Q}\mathbf{U}^{\text{B}}\mathbf{\Sigma}^{\text{B}}(\mathbf{V}^{\text{B}})^\top.$$

The process is summarized in Algorithm C.1.

Algorithm C.1 RSVD

Input: matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ with target rank k, and oversampling parameter p such that $k+p \leq \min\{m,n\}$

Output: $\widehat{\mathbf{U}} \in \mathbb{R}^{m\times k}, \widehat{\mathbf{\Sigma}} \in \mathbb{R}^{k\times k}$, and $\widehat{\mathbf{V}} \in \mathbb{R}^{n\times k}$

1: Draw standard Gaussian random matrix $\mathbf{G} \in \mathbb{R}^{n\times(k+p)}$
2: Multiply $\mathbf{Y} = \mathbf{AG}$
3: Compute thin QR factorization $\mathbf{Y} = \mathbf{QR}$
4: Form $\mathbf{B} = \mathbf{Q}^\top\mathbf{A}$
5: Calculate thin SVD $\mathbf{B} = \mathbf{U}^{\text{B}}\mathbf{\Sigma}^{\text{B}}(\mathbf{V}^{\text{B}})^\top$
6: Set $\widehat{\mathbf{U}} = \mathbf{QU}^{\text{B}}(:,1:k), \widehat{\mathbf{\Sigma}} = \mathbf{\Sigma}^{\text{B}}(1:k,1:k)$, and $\widehat{\mathbf{V}} = \mathbf{V}^{\text{B}}(:,1:k)$

Notice that the total number of matrix-vector multiplications with $\mathbf{A}$ is $k+p$ and with $\mathbf{A}^\top$ is at most $k+p$. If the singular values of $\mathbf{A}$ decay rapidly or if the rank of $\mathbf{A}$ is exactly k, then $\text{ran}(\mathbf{Q})$ is a good approximation of $\text{ran}(\mathbf{A})$. Moreover, the above algorithm can be combined with a power iteration (replacing line 2 of Algorithm C.1 with $\mathbf{Y} = (\mathbf{AA}^\top)^q\mathbf{AG}$ for $q \geq 0$) for tighter approximation bounds, but this would require additional matrix-vector multiplications with $\mathbf{A}$ and $\mathbf{A}^\top$. Assuming that we a rank-k RSVD approximation $\mathbf{A} = \widehat{\mathbf{U}}\widehat{\mathbf{\Sigma}}\widehat{\mathbf{V}}^\top$, an approximation of the posterior covariance matrix can be obtained as

$$\mathbf{\Gamma} \approx \sigma^2(\lambda\mathbf{I} + \widehat{\mathbf{V}}\widehat{\mathbf{\Sigma}}^\top\widehat{\mathbf{\Sigma}}\widehat{\mathbf{V}}^\top)^{-1}. \tag{C.1}$$

Results comparing the GKB approximation and the RSVD approximation for variance-covariance estimation are provided in Figure 4.3 for the deblurring and tomography examples.

REFERENCES

[1] J. Adler, H. Kohr, and O. Öktem. Operator discretization library (ODL), 2017. See also: https://zenodo.org/record/556409#.YHw3rz_ONPY.

[2] B. M. Afkham, J. Chung, and M. Chung. Learning regularization parameters of inverse problems via deep neural networks. *Inverse Problems*, 37(10):105017, 2021.

[3] A. Alqahtani, S. Gazzola, L. Reichel, and G. Rodriguez. On the block Lanczos and block Golub–Kahan reduction methods applied to discrete ill-posed problems. *Numerical Linear Algebra with Applications*, page e2376, 2021.

[4] H. Andrews and B. Hunt. *Digital Image Restoration*. Prentice Hall, Englewood Cliffs, NJ, 1977.

[5] M. Arioli. Generalized Golub–Kahan bidiagonalization and stopping criteria. *SIAM J. Matrix Anal. Appl.*, 34(2):571–592, 2013.

[6] S. R. Arridge, M. M. Betcke, and L. Harhanen. Iterated preconditioned LSQR method for inverse problems on unstructured grids. *Inverse Problems*, 30(7):075009, 2014.

[7] J. Baglama and L. Reichel. Augmented implicitly restarted Lanczos bidiagonalization methods. *SIAM J. Sci. Comput.*, 27(1):19–42, 2005.

[8] J. Baglama, L. Reichel, and D. Richmond. An augmented LSQR method. *Numerical Algorithms*, 64(2):263–293, 2013.

[9] J. M. Bardsley. Stopping rules for a nonnegatively constrained iterative method for ill-posed Poisson imaging problems. *BIT*, 48(4):651–664, 2008.

[10] J. M. Bardsley. *Computational Uncertainty Quantification for Inverse Problems*, volume 19. SIAM, 2018.

[11] J. L. Barlow. Reorthogonalization for the Golub–Kahan–Lanczos bidiagonal reduction. *Numerische Mathematik*, 124(2):237–278, 2013.

[12] F. Bauer, M. Gutting, and M. A. Lukas. Evaluation of parameter choice methods for regularization of ill-posed problems in geomathematics. In W. Freeden, M. Z. Nashed, and T. Sonar, editors, *Handbook of Geomathematics*, pages 1713–1774. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[13] F. Bauer and M. A. Lukas. Comparing parameter choice methods for regularization of ill-posed problems. *Mathematics and Computers in Simulation*, 81(9):1795–1841, 2011.

[14] F. S. V. Bazán. Fixed-point iterations in determining the Tikhonov regularization parameter. *Inverse Problems*, 24(3):035001, 2008.

[15] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

[16] S. Berisha and J. G. Nagy. Iterative methods for image restoration. In *Academic Press Library in Signal Processing*, volume 4, pages 193–247. Elsevier, 2014.

[17] M. Bertero and P. Boccacci. *Introduction to Inverse Problems in Imaging*. IOP Pulishing Ltd., London, 1998.

[18] A. Biguri, M. Dosanjh, S. Hancock, and M. Soleimani. TIGRE: a MATLAB-GPU toolbox for CBCT image reconstructions. *Biomed. Phys. Eng. Express*, 2(5), 2016.

[19] Å. Björck. A bidiagonalization algorithm for solving ill-posed systems of equations. Dept. Mathematics, Linköping University, Sweden, Rep. LiTHMAT-80-33, 1980.

[20] Å. Björck. A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations. *BIT*, 28(3):659–670, 1988.

[21] Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, 1996.

[22] Å. Björck, E. Grimme, and P. van Dooren. An implicit shift bidiagonalization algorithm for ill-posed systems. *BIT*, 34(4):510–534, 1994.

[23] A. F. Boden, D. C. Redding, R. J. Hanisch, and J. Mo. Massively parallel spatially-variant maximum likelihood restoration of Hubble Space Telescope imagery. *J. Opt. Soc. Am. A*, 13:1537–1545, 1996.

[24] N. K. Bose and K. J. Boo. High-resolution image reconstruction with multisensors. *Int. J. Imaging Syst. Technol.*, 9:294–304, 1998.

[25] C. Brezinski, P. Novati, and M. Redivo-Zaglia. A rational Arnoldi approach for ill-conditioned linear systems. *J. Comput. Appl. Math.*, 236(8):2063–2077, 2012.

[26] D. A. Brown, A. Saibaba, and S. Vallélian. Low-rank independence samplers in hierarchical Bayesian inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*,

6(3):1076–1100, 2018.
[27] A. Buccini and L. Reichel. An $\ell_2 - \ell_q$ regularization method for large discrete ill-posed problems. *J. Sci. Comput.*, 78:1526–1549, 2019.
[28] A. Buccini and L. Reichel. An ℓ^p-ℓ^q minimization method with cross-validation for the restoration of impulse noise contaminated images. *J. Comput. Appl. Math.*, 375:112824, 2020.
[29] T. Bui-Thanh, C. Burstedde, O. Ghattas, J. Martin, G. Stadler, and L. C. Wilcox. Extreme-scale UQ for Bayesian inverse problems governed by PDEs. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, page 3. IEEE Computer Society Press, 2012.
[30] T. Bui-Thanh and O. Ghattas. A scalable algorithm for MAP estimators in Bayesian inverse problems with Besov priors. *Inverse Problems and Imaging*, 9(1):27–53, 2015.
[31] T. Bui-Thanh, O. Ghattas, J. Martin, and G. Stadler. A computational framework for infinite-dimensional Bayesian inverse problems Part I: The linearized case, with application to global seismic inversion. *SIAM J. Sci. Comput.*, 35(6):A2494–A2523, 2013.
[32] T. M. Buzug. *Computed Tomography*. Springer, Berlin, 2008.
[33] D. Calvetti. Preconditioned iterative methods for linear discrete ill-posed problems from a Bayesian inversion perspective. *J. Comput. Appl. Math.*, 198:378–395, 2007.
[34] D. Calvetti, G. H. Golub, and L. Reichel. Estimation of the L-curve via Lanczos bidiagonalization. *BIT*, 39:603–619, 1999.
[35] D. Calvetti, P. C. Hansen, and L. Reichel. L-curve curvature bounds via Lanczos bidiagonalization. *Electron. Trans. Numer. Anal*, 14:20–35, 2002.
[36] D. Calvetti, J. P. Kaipio, and E. Somersalo. Inverse problems in the Bayesian framework. *Inverse Problems*, 30(11):110301, 2014.
[37] D. Calvetti, B. Lewis, and L. Reichel. On the choice of subspace for iterative methods for linear discrete ill-posed problems. *Applied Mathematics and Computer Science*, 11(5):1069–1092, 2001.
[38] D. Calvetti, B. Lewis, and L. Reichel. GMRES, L-curves, and discrete ill-posed problems. *BIT Numerical Mathematics*, 42(1):44–65, 2002.
[39] D. Calvetti, B. Lewis, and L. Reichel. A hybrid GMRES and TV-norm-based method for image restoration. In *Advanced Signal Processing Algorithms, Architectures, and Implementations XII*, volume 4791, pages 192–200. International Society for Optics and Photonics, 2002.
[40] D. Calvetti, B. Lewis, and L. Reichel. On the regularizing properties of the GMRES method. *Numerische Mathematik*, 91(4):605–625, 2002.
[41] D. Calvetti, S. Morigi, L. Reichel, and F. Sgallari. Tikhonov regularization and the L-curve for large discrete ill-posed problems. *J. Comput. Appl. Math.*, 123:423–446, 2000.
[42] D. Calvetti, F. Pitolli, E. Somersalo, and B. Vantaggi. Bayes meets Krylov: Statistically inspired preconditioners for CGLS. *SIAM Review*, 60(2):429–461, 2018.
[43] D. Calvetti, M. Pragliola, E. Somersalo, and A. Strang. Sparse reconstructions from few noisy data: analysis of hierarchical Bayesian models with generalized gamma hyperpriors. *Inverse Problems*, 36(2):025010, 2020.
[44] D. Calvetti, L. Reichel, and A. Shuibi. Enriched Krylov subspace methods for ill-posed problems. *Linear Algebra Appl.*, 362:257–273, 2003.
[45] D. Calvetti, L. Reichel, and A. Shuibi. L-curve and curvature bounds for Tikhonov regularization. *Numer. Algorithms*, 35(2):301–314, Apr 2004.
[46] D. Calvetti, L. Reichel, and A. Shuibi. Invertible smoothing preconditioners for linear discrete ill-posed problems. *Applied Numerical Mathematics*, 54(2):135–149, 2005.
[47] D. Calvetti and E. Somersalo. Priorconditioners for linear systems. *Inverse Problems*, 21(4):1397, 2005.
[48] D. Calvetti and E. Somersalo. *An Introduction to Bayesian Scientific Computing: Ten Lectures on Subjective Computing*, volume 2. Springer Science & Business Media, 2007.
[49] D. Calvetti and E. Somersalo. Inverse problems: From regularization to Bayesian inference. *Wiley Interdisciplinary Reviews: Computational Statistics*, 10(3):e1427, 2018.
[50] E. J. Candés, J. K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measuerments. *Comm. Pure Appl. Math.*, LIX:1207–1223, 2006.
[51] N. A. Caruso and P. Novati. Convergence analysis of LSQR for compact operator equations. *Linear Algebra Appl.*, 583:146–164, 2019.
[52] T. Cho, J. Chung, S. M. Miller, and A. K. Saibaba. Computationally efficient methods for large-scale atmospheric inverse modeling. *Geoscientific Model Development Discussions*, pages 1–30, 2022.
[53] E. Chow and Y. Saad. Preconditioned Krylov subspace methods for sampling multivariate Gaussian distributions. *SIAM J. Sci. Comput.*, 36(2):A588–A608, 2014.

[54] J. Chung and S. Gazzola. Flexible Krylov methods for ℓ_p regularization. *SIAM J. Sci. Comput.*, 41(5):S149–S171, 2019.
[55] J. Chung, E. Haber, and J. Nagy. Numerical methods for coupled super-resolution. *Inverse Problems*, 22(4):1261–1272, 2006.
[56] J. Chung, M. E. Kilmer, and D. P. O'Leary. A framework for regularization via operator approximation. *SIAM J. Sci. Comput.*, 37:B332–B359, 2015.
[57] J. Chung, S. Knepper, and J. G. Nagy. *Large-Scale Inverse Problems in Imaging*, pages 47–90. Springer New York, NY, 2015.
[58] J. Chung and J. Nagy. An efficient iterative approach for large-scale separable nonlinear inverse problems. *SIAM J. Sci. Comput.*, 31(6):4654–4674, 2010.
[59] J. Chung, J. G. Nagy, and D. P. O'Leary. A weighted GCV method for Lanczos hybrid regularization. *Elec. Trans. Numer. Anal.*, 28:149–167, 2008.
[60] J. Chung and K. Palmer. A hybrid LSMR algorithm for large-scale Tikhonov regularization. *SIAM J. Sci. Comput.*, 37(5):S562–S580, 2015.
[61] J. Chung and A. Saibaba. Generalized hybrid iterative methods for large-scale Bayesian inverse problems. *SIAM J. Sci. Comput.*, 39(5):S24–S46, 2017.
[62] J. Chung, A. K. Saibaba, M. Brown, and E. Westman. Efficient generalized Golub–Kahan based methods for dynamic inverse problems. *Inverse Problems*, 34(2):024005, 2018.
[63] J. Chung, P. Sternberg, and C. Yang. High performance 3-D image reconstruction for molecular structure determination. *International Journal of High Performance Computing Applications*, 24(2):117–135, 2010.
[64] S. Cipolla, M. Donatelli, and F. Durastante. Regularization of inverse problems by an approximate matrix-function technique. *Numerical Algorithms*, pages 1–34, 2021.
[65] S. C. Constable, R. L. Parker, and C. G. Constable. Occam's inversion: A practical algorithm for generating smooth models from electromagnetic sounding data. *Geophysics*, 52(3):289–300, 1987.
[66] M. Dashti, S. Harris, and A. Stuart. Besov priors for Bayesian inverse problems. *Inverse Problems and Imaging*, 6(2):183–200, 2012.
[67] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Comm. Pure Appl. Math.*, 63(1):1–38, 2010.
[68] M. Donatelli, D. Martin, and L. Reichel. Arnoldi methods for image deblurring with anti-reflective boundary conditions. *Applied Mathematics and Computation*, 253:135–150, 2015.
[69] Y. Dong, H. Garde, and P. C. Hansen. R^{3}GMRES: including prior information in GMRES-type methods for discrete inverse problems. *Electronic Transactions on Numerical Analysis*, 42:136–146, 2014.
[70] D. L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.
[71] L. Eldén. Algorithms for the regularization of ill-conditioned least squares problems. *BIT Numerical Mathematics*, 17(2):134–145, 1977.
[72] L. Eldén. A weighted pseudoinverse, generalized singular values, and constrained least squares problems. *BIT Numerical Mathematics*, 22(4):487–502, 1982.
[73] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*, volume 375. Springer Science & Business Media, 1996.
[74] H. W. Engl and P. Kügler. Nonlinear inverse problems: Theoretical aspects and some industrial applications. In *Multidisciplinary methods for analysis optimization and control of complex systems*, pages 3–47. Springer, 2005.
[75] C. L. Epstein. *Introduction to the Mathematics of Medical Imaging, Second Edition.* SIAM, Philadelphia, PA, 2007.
[76] C. Fenu, L. Reichel, and G. Rodriguez. GCV for Tikhonov regularization via global Golub–Kahan decomposition. *Numer. Linear Algebra Appl.*, 23, 02 2016.
[77] H. Flath, L. Wilcox, V. Akçelik, J. Hill, B. van Bloemen Waanders, and O. Ghattas. Fast algorithms for Bayesian uncertainty quantification in large–scale linear inverse problems based on low-rank partial Hessian approximations. *SIAM J. Sci. Comput.*, 33(1):407–432, 2011.
[78] D. C.-L. Fong and M. Saunders. LSMR: An iterative algorithm for sparse least-squares problems. *SIAM J. Sci. Comput.*, 33(5):2950–2971, 2011.
[79] M. Fornasier, S. Peter, H. Rauhut, and S. Worm. Conjugate gradient acceleration of iteratively re-weighted least squares methods. *Computational optimization and applications*, 65(1):205–259, 2016.
[80] M. Fornasier and H. Rauhut. Compressive sensing. In O. Scherzer, editor, *Handbook of Mathematical Methods in Imaging*, pages 187–228. Springer New York, New York, NY,

2011.
[81] A. Frommer and P. Maass. Fast CG-based methods for Tikhonov-Phillips regularization. *SIAM J. Sci. Comput.*, 20:1831–1850, 1999.
[82] S. Gazzola, P. C. Hansen, and J. G. Nagy. IR Tools: a MATLAB package of iterative regularization methods and large-scale test problems. *Numerical Algorithms*, 81(3):773–811, 2019. Software available at https://github.com/jnagy1/IRtools.
[83] S. Gazzola, M. E. Kilmer, J. G. Nagy, O. Semerci, and E. L. Miller. An inner-outer iterative method for edge preservation in image restoration and reconstruction. *Inverse Problems*, 36:124004, 2020.
[84] S. Gazzola and M. S. Landman. Flexible GMRES for total variation regularization. *BIT Numerical Mathematics*, 59(3):721–746, 2019.
[85] S. Gazzola and M. S. Landman. Krylov methods for inverse problems: Surveying classical, and introducing new, algorithmic approaches. *Mitteilungen der Gesellschaft für Angewandte Mathematik und Mechanik*, 43(4), 2020.
[86] S. Gazzola and M. S. Landman. Adaptive regularization parameter choice rules for large-scale problems, 2021. To appear in SIAM J. Matrix Anal. Appl.
[87] S. Gazzola, C. Meng, and J. G. Nagy. Krylov methods for low-rank regularization. *SIAM Journal on Matrix Analysis and Applications*, 41(4):1477–1504, 2020.
[88] S. Gazzola and J. G. Nagy. Generalized Arnoldi–Tikhonov method for sparse reconstruction. *SIAM J. Sci. Comput.*, 36(2):B225–B247, 2014.
[89] S. Gazzola, J. G. Nagy, and M. Sabaté Landman. Iteratively reweighted FGMRES and FLSQR for sparse reconstruction. *SIAM J. Sci. Comput.*, pages S47–S69, 2021.
[90] S. Gazzola, S. Noschese, P. Novati, and L. Reichel. Arnoldi decomposition, GMRES, and preconditioning for linear discrete ill-posed problems. *Applied Numerical Mathematics*, 142:102–121, 2019.
[91] S. Gazzola and P. Novati. Automatic parameter setting for Arnoldi-Tikhonov methods. *J. Comput. Appl. Math.*, 256:180–195, 2014.
[92] S. Gazzola and P. Novati. Inheritance of the discrete Picard condition in Krylov subspace methods. *BIT Numerical Mathematics*, 56(3):893–918, 2016.
[93] S. Gazzola, P. Novati, and M. R. Russo. Embedded techniques for choosing the parameter in Tikhonov regularization. *Numer. Linear Algebra Appl.*, 21(6):796–812, 2014.
[94] S. Gazzola, P. Novati, and M. R. Russo. On Krylov projection methods and Tikhonov regularization. *Electron. Trans. Numer. Anal.*, 44:83–123, 2015.
[95] S. Gazzola, E. Onunwor, L. Reichel, and G. Rodriguez. On the Lanczos and Golub-Kahan reduction methods applied to discrete ill-posed problems. *Numer. Linear Algebra Appl.*, 23:187–204, 2016.
[96] T. Goldstein and S. Osher. The split Bregman method for L1-regularized problems. *SIAM J. Imaging Sci.*, 2(2):323–343, 2009.
[97] G. Golub and V. Pereyra. Separable nonlinear least squares: the variable projection method and its applications. *Inverse Problems*, 19(2):R1–R26, 2003.
[98] G. H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.
[99] G. H. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, 2(2):205–224, 1965.
[100] G. H. Golub, F. T. Luk, and M. L. Overton. A block Lanczos method for computing the singular values and corresponding singular vectors of a matrix. *ACM Trans. Math Software*, 7(2):149–169, 1981.
[101] G. H. Golub and G. Meurant. *Matrices, moments, and quadrature with applications*. Princeton University Press, Princeton, NJ, 2010.
[102] G. H. Golub and D. P. O'Leary. Some history of the conjugate gradient and Lanczos algorithms: 1948–1976. *SIAM review*, 31(1):50–102, 1989.
[103] G. H. Golub and V. Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on numerical analysis*, 10(2):413–432, 1973.
[104] G. H. Golub and V. Pereyra. Separable nonlinear least squares: the variable projection method and its applications. *Inverse Problems*, 19:R1–R26, 2003.
[105] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 2013.
[106] G. H. Golub and U. Von Matt. Quadratically constrained least squares and quadratic problems. *Numer. Math.*, 59:561–580, 1991.
[107] G. H. Golub and U. Von Matt. Generalized cross-validation for large-scale problems. *Journal*

of Computational and Graphical Statistics, 6(1):1–34, 1997.
[108] R. A. Gonsalves. Phase diversity in adaptive optics. *Opt. Eng.*, 21:829–832, 1982.
[109] I. F. Gorodnitsky and B. D. Rao. A new iterative weighted norm minimization algorithm and its applications. In *IEEE Sixth SP Workshop on Statistical Signal and Array Processing*, pages 412–415, 1992.
[110] D. Gürsoy, F. D. Carlo, X. Xiao, and C. Jacobsen. TomoPy: a framework for the analysis of synchrotron tomographic data. *J. Synchrotron Radiat.*, 21:1188–1193, 2014.
[111] E. Haber. *Computational Methods in Geophysical Electromagnetics*. SIAM, Philadelphia, PA, 2014.
[112] E. Haber, U. M. Ascher, and D. Oldenburg. On optimization techniques for solving nonlinear inverse problems. *Inverse Problems*, 16(5):1263, 2000.
[113] E. Haber, L. Horesh, and L. Tenorio. Numerical methods for experimental design of large-scale linear ill-posed inverse problems. *Inverse Problems*, 24(5):055012, 2008.
[114] E. Haber and D. Oldenburg. A GCV based method for nonlinear ill-posed problems. *Computational Geosciences*, 4(1):41–63, 2000.
[115] E. Haber and L. Tenorio. Learning regularization functionals—a supervised training approach. *Inverse Problems*, 19(3):611, 2003.
[116] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.
[117] M. Hanke. Regularization with differential operators: An iterative approach. *Numerical Functional Analysis and Optimization*, 13(5-6):523–540, 1992.
[118] M. Hanke. *Conjugate Gradient Type Methods for Ill-Posed Problems*. Pitman Research Notes in Mathematics, Longman Scientific & Technical, Harlow, Essex, 1995.
[119] M. Hanke. The minimal error conjugate gradient method is a regularization method. *Proceedings of the American Mathematical Society*, 123(11):3487–3497, 1995.
[120] M. Hanke. On Lanczos based methods for the regularization of discrete ill-posed problems. *BIT*, 41(5):1008–1018, 2001.
[121] M. Hanke and P. C. Hansen. Regularization methods for large-scale problems. *Surv. Math. Ind*, 3(4):253–315, 1993.
[122] M. Hanke, J. G. Nagy, and R. J. Plemmons. Preconditioned iterative regularization for ill-posed problems. In L. Reichel, A. Ruttan, and R. S. Varga, editors, *Numerical Linear Algebra*, pages 141–163. de Gruyter, Berlin, 1993.
[123] P. C. Hansen. The discrete picard condition for discrete ill-posed problems. *BIT*, 30:658—672, 1990.
[124] P. C. Hansen. Regularization Tools: A MATLAB package for analysis and solution of discrete ill-posed problems. *Numerical Algorithms*, 6(1):1–35, 1994.
[125] P. C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems*. SIAM, Philadelphia, PA, 1998.
[126] P. C. Hansen. *Discrete Inverse Problems: Insight and Algorithms*. SIAM, 2010.
[127] P. C. Hansen. Oblique projections and standard-form transformations for discrete inverse problems. *Numerical linear algebra with applications*, 20(2):250–258, 2013.
[128] P. C. Hansen, Y. Dong, and K. Abe. Hybrid enriched bidiagonalization for discrete ill-posed problems. *Numerical Linear Algebra with Applications*, 26(3):e2230, 2019.
[129] P. C. Hansen, K. Hayami, and K. Morikuni. GMRES methods for tomographic reconstruction with an unmatched back projector. *Journal of Computational and Applied Mathematics*, 413:114352, 2022.
[130] P. C. Hansen and T. K. Jensen. Smoothing-norm preconditioning for regularizing minimum-residual methods. *SIAM J. Matrix Anal. Appl.*, 29(1):1–14, 2007.
[131] P. C. Hansen and T. K. Jensen. Noise propagation in regularizing iterations for image deblurring. *Electron. Trans. Numer. Anal.*, 31:204–220, 2008.
[132] P. C. Hansen and J. S. Jørgensen. AIR Tools II: Algebraic Iterative Reconstruction Methods, Improved Implementation. *Numerical Algorithms*, 2018. Sotfware available at https://github.com/jakobsj/AIRToolsII/.
[133] P. C. Hansen, J. S. Jørgensen, and W. R. B. Lionheart. *Computed Tomography: Algorithms, Insight, and Just Enough Theory*. SIAM, Philadelphia, PA, 2021.
[134] P. C. Hansen, J. S. Jørgensen, and P. W. Rasmussen. Stopping rules for algebraic iterative reconstruction methods in computed tomography. In *2021 21st International Conference on Computational Science and Its Applications (ICCSA)*, pages 60–70. IEEE, 2021.
[135] P. C. Hansen, M. E. Kilmer, and R. Kjeldsen. Exploiting residual information in the parameter choice for discrete ill-posed problems. *BIT Numerical Mathematics*, 46(1):41–59, 2006.
[136] P. C. Hansen, J. G. Nagy, and D. P. O'Leary. *Deblurring Images: Matrices, Spectra and*

Filtering. SIAM, Philadelphia, PA, 2006.
[137] P. C. Hansen and D. P. O'Leary. The use of the L-curve in the regularization of discrete ill-posed problems. *SIAM J. Sci. Comput.*, 14:1487–1503, 1993.
[138] I. Hnětynková, M. Kub´inová, and M. Plešinger. Noise representation in residuals of LSQR, LSMR, and CRAIG regularization. *Linear Algebra Appl.*, 533:357–379, 2017.
[139] I. Hnětynková, M. Plešinger, and Z. Strakos. The regularizing effect of the Golub-Kahan iterative bidiagonalization and revealing the noise level in the data. *BIT*, 49:669–696, 2009.
[140] M. E. Hochstenbach and L. Reichel. An iterative method for Tikhonov regularization with a general linear regularization operator. *The Journal of Integral Equations and Applications*, pages 465–482, 2010.
[141] M. E. Hochstenbach and L. Reichel. Subspace-restricted singular value decompositions for linear discrete ill-posed problems. *J. Comput. Appl. Math.*, 235(4):1053–1064, 2010.
[142] B. Hofmann. Regularization of nonlinear problems and the degree of ill-posedness. In G. Anger, R. Gorenflo, H. Jochmann, H. Moritz, and W. Webers, editors, *Inverse Problems: Principles and Applications in Geophysics, Technology, and Medicine*. Akademie Verlag, Berlin, 1993.
[143] G. Huang, A. Lanza, S. Morigi, L. Reichel, and F. Sgallari. Majorization–minimization generalized Krylov subspace methods for $\ell_p - \ell_q$ optimization applied to image restoration. *BIT Numerical Mathematics*, 57(2):351–378, 2017.
[144] Y. Huang and Z. Jia. Some results on the regularization of LSQR for large-scale discrete ill-posed problems. *Science China Mathematics*, 60(4):701–718, 2017.
[145] D. Hunter and K. Lange. A tutorial on MM algorithms. *Am. Stat.*, 58:20–37, 2004.
[146] I. C. F. Ipsen and C. D. Meyer. The idea behind krylov methods. *The American mathematical monthly*, 105(10):889, 1998.
[147] S. M. Jefferies, K. J. Schulze, C. L. Matson, K. Stoltenberg, and E. K. Hege. Blind deconvolution in optical diffusion tomography. *Optics Express*, 10(1):46–53, 2002.
[148] T. K. Jensen and P. C. Hansen. Iterative regularization with minimum-residual methods. *BIT*, 47:103–120, 2007.
[149] Z. Jia. Approximation accuracy of the Krylov subspaces for linear discrete ill-posed problems. *J. Comput. Appl. Math.*, 374:112786, 2020.
[150] Z. Jia. The low rank approximations and Ritz values in LSQR for linear discrete ill-posed problem. *Inverse Problems*, 36(4):45013, 2020.
[151] Z. Jia. Regularization properties of LSQR for linear discrete ill-posed problems in the multiple singular value case and best, near best and general low rank approximations. *Inverse Problems*, 36(8):85009, 2020.
[152] J. Jiang, J. Chung, and E. de Sturler. Hybrid projection methods with recycling for inverse problems. *SIAM J. Sci. Comput.*, pages S146–S172, 2021.
[153] M. Jiang, L. Xia, G. Shou, F. Liu, and S. Crozier. Two hybrid regularization frameworks for solving the electrocardiography inverse problem. *Physics in Medicine & Biology*, 53(18):5151, 2008.
[154] J. S. Jørgensen, E. Ametova, G. Burca, G. Fardell, E. Papoutsellis, E. Pasca, K. Thielemans, M. Turner, R. Warr, W. R. B. Lionheart, and P. J. Withers. Core Imaging Library - Part I: a versatile Python framework for tomographic imaging. *Phil. Trans. R. Soc. A*, 379, 2021.
[155] J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*, volume 160. Springer Science & Business Media, 2006.
[156] K. Kan, J. G. Nagy, and L. Ruthotto. Avoiding the double descent phenomenon of random feature models using hybrid regularization. *arXiv preprint arXiv:2012.06667*, 2020.
[157] L. Kaufman. A variable projection method for solving separable nonlinear least squares problems. *BIT Numerical Mathematics*, 15(1):49–57, 1975.
[158] D. Kazantsev, E. Pasca, M. J. Turner, and P. J. Withers. CCPi-Regularisation toolkit for computed tomographic image reconstruction with proximal splitting algorithms. *SoftwareX*, 9:317–323, 2019.
[159] M. E. Kilmer, P. C. Hansen, and M. I. Español. A projection-based approach to general-form Tikhonov regularization. *SIAM J. Sci. Comput.*, 29(1):315–330, 2007.
[160] M. E. Kilmer and D. P. O'Leary. Choosing regularization parameters in iterative methods for ill-posed problems. *SIAM J. Matrix Anal. Appl.*, 22(4):1204–1221, 2001.
[161] N. Kuroiwa and T. Nodera. The adaptive augmented GMRES method for solving ill-posed problems. *ANZIAM Journal*, 50:654–667, 2008.
[162] J. Lampe, L. Reichel, and H. Voss. Large-scale Tikhonov regularization via reduction by orthogonal projection. *Linear Algebra Appl.*, 436(8):2845–2865, 2012.

[163] A. Lanza, S. Morigi, L. Reichel, and F. Sgallari. A generalized Krylov subspace method for $\ell_p - \ell_q$ minimization. *SIAM J. Sci. Comput.*, 37(5):S30–S50, 2015.
[164] A. Lanza, S. Morigi, I. Selesnick, and F. Sgallari. Nonconvex nonsmooth optimization via convex - nonconvex majorization - minimization. *Numerische Mathematik*, 136(2):343–381, 2017.
[165] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. SIAM, Philadelphia, PA, 1995.
[166] B. Lewis and L. Reichel. Arnoldi–Tikhonov regularization methods. *J. Comput. Appl. Math.*, 226(1):92–102, 2009.
[167] R. C. Li and Q. Ye. A Krylov subspace method for quadratic matrix polynomials with application to constrained least squares problems. *SIAM J. Matrix Anal. Appl.*, 25:405–428, 2003.
[168] J. L. Mead and R. A. Renaut. A Newton root-finding algorithm for estimating the regularization parameter for solving ill-conditioned least squares problems. *Inverse Problems*, 25(2):025002, 2008.
[169] S. Morigi, L. Reichel, and F. Sgallari. Orthogonal projection regularization operators. *Numerical Algorithms*, 44(2):99–114, 2007.
[170] V. A. Morozov. On the solution of functional equations by the method of regularization. *Soviet Math. Doklady*, 7:414–417, 1966.
[171] J. L. Mueller and S. Siltanen. *Linear and Nonlinear Inverse Problems with Practical Applications*. SIAM, Philadelphia, 2012.
[172] J. Nagy, K. Palmer, and L. Perrone. Iterative methods for image deblurring: A MATLAB object oriented approach. *Numerical Algorithms*, 36(1):73–93, 2004.
[173] J. G. Nagy and L. Ruthotto. LAP: a linearize and project method for solving inverse problems with coupled variables. *Imaging (MRI)*, 1(6), 2018.
[174] F. Natterer. *The Mathematics of Computerized Tomography*. SIAM, Philadelphia, PA, 2001.
[175] A. S. Nemirovskii. The regularizing properties of the adjoint gradient method in ill-posed problems. *USSR Computational Mathematics and Mathematical Physics*, 26(2):7–16, 1986.
[176] E. Newman, L. Ruthotto, J. Hart, and B. van Bloemen Waanders. Train like a (Var) Pro: Efficient training of neural networks with variable projection. *SIAM Journal on Mathematics of Data Science*, 3(4):1041–1066, 2021.
[177] Y. Notay. Flexible conjugate gradients. *SIAM J. Sci. Comput.*, 22(4):1444–1460, 2000.
[178] P. Novati. Some properties of the Arnoldi based methods for linear ill-posed problems. *SIAM J. Matrix Anal. Appl.*, 55:1437–1455, 2017.
[179] P. Novati. A convergence result for some Krylov-Tikhonov methods in Hilbert spaces. *Numerical functional analysis and optimization*, 39(6):655–666, 2018.
[180] P. Novati, M. Redivo-Zaglia, and M. R. Russo. Preconditioning linear systems via matrix function evaluation. *Appl. Numer. Math.*, 62(12):2063—-2077, 2012.
[181] D. P. O'Leary. *Scientific Computing with Case Studies*, volume 109. SIAM, 2009.
[182] D. P. O'Leary and J. A. Simmons. A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems. *SIAM J. Sci. Stat. Comp.*, 2(4):474–489, 1981.
[183] D. P. O'Leary. Robust regression computation using iteratively reweighted least squares. *SIAM Journal on Matrix Analysis and Applications*, 11(3):466–480, 1990.
[184] D. P. O'Leary and B. W. Rust. Variable projection for nonlinear least squares problems. *Computational Optimization and Applications*, 54(3):579–593, 2013.
[185] C. C. Paige and M. A. Saunders. Algorithm 583 LSQR: Sparse linear equations and least squares problems. *ACM Trans. Math. Soft.*, 8(2):195–209, 1982.
[186] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Soft.*, 8(1):43–71, 1982.
[187] E. Papoutsellis, E. Ametova, C. Delplancke, G. Fardell, J. S. Jørgensen, E. Pasca, M. Turner, R. Warr, W. R. B. L. William, and P. J. Withers. Core Imaging Library - Part II: multichannel reconstruction for dynamic and spectral tomography. *Phil. Trans. R. Soc. A*, 379, 2021.
[188] A. Parker and C. Fox. Sampling Gaussian distributions in Krylov spaces with conjugate gradients. *SIAM J. Sci. Comput.*, 34(3):B312–B334, 2012.
[189] R. L. Parker and R. L. Parker. *Geophysical inverse theory*, volume 1. Princeton university press, 1994.
[190] T. Regińska. A regularization parameter in discrete ill-posed problems. *SIAM J. Sci. Comput.*, 17(3):740–749, 1996.
[191] L. Reichel and G. Rodriguez. Old and new parameter choice rules for discrete ill-posed problems. *Numer. Algorithms*, 63:65–87, 2013.

[192] L. Reichel, F. Sgallari, and Q. Ye. Tikhonov regularization based on generalized Krylov subspace methods. *Applied Numerical Mathematics*, 62(9):1215–1228, 2012.
[193] L. Reichel and A. Shyshkov. A new zero-finder for tikhonov regularization. *BIT Numerical Mathematics*, 48(3):627–643, 2008.
[194] L. Reichel and U. O. Ugwu. Tensor Krylov subspace methods with an invertible linear transform product applied to image processing. *Applied Numerical Mathematics*, 2021.
[195] L. Reichel and U. O. Ugwu. The tensor Golub–Kahan–Tikhonov method applied to the solution of ill-posed problems with a t-product structure. *Numerical Linear Algebra with Applications*, 29(1):e2412, 2022.
[196] L. Reichel and Q. Ye. Simple square smoothing regularization operators. *Electronic Transactions on Numerical Analysis*, 33:63, 2009.
[197] R. A. Renaut, M. Horst, Y. Wang, D. Cochran, and J. Hansen. Efficient estimation of regularization parameters via downsampling and the singular value expansion. *BIT Numerical Mathematics*, 57(2):499–529, 2017.
[198] R. A. Renaut, S. Vatankhah, and V. E. Ardestani. Hybrid and iteratively reweighted regularization by unbiased predictive risk and weighted GCV for projected systems. *SIAM J. Sci. Comput.*, 39(2):B221–B243, 2017.
[199] P. Rodriguez and B. Wohlberg. An efficient algorithm for sparse representations with ℓ^p data fidelity term. In *Proceedings of 4th IEEE Andean Technical Conference (ANDESCON)*, 2008.
[200] M. C. Roggemann and B. Welsh. *Imaging Through Turbulence*. CRC Press, Boca Raton, FL, 1996.
[201] B. W. Rust and D. P. O'Leary. Residual periodograms for choosing regularization parameters for ill-posed problems. *Inverse Problems*, 24(3):034005, 2008.
[202] L. Ruthotto, J. Chung, and M. Chung. Optimal experimental design for inverse problems with state constraints. *SIAM J. Sci. Comput.*, 40(4):B1080–B1100, 2018.
[203] Y. Saad. On the rates of convergence of the Lanczos and the block-Lanczos methods. *SIAM J. Numer. Anal.*, 17(5):687–706, 1980.
[204] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Comput.*, 14(2):461–469, 1993.
[205] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2003.
[206] M. D. Sacchi. Inversion of large ill-posed seismic imaging problems. In *SEG Technical Program Expanded Abstracts 1998*, pages 1657–1660. Society of Exploration Geophysicists, 1998.
[207] A. K. Saibaba, A. Alexanderian, and I. C. F. Ipsen. Randomized matrix-free trace and log-determinant estimators. *Numer. Math.*, 137(5):353–395, 2017.
[208] A. K. Saibaba, J. Bardsley, D. A. Brown, and A. Alexanderian. Efficient marginalization-based MCMC methods for hierarchical Bayesian inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*, 7(3):1105–1131, 2019.
[209] A. K. Saibaba, J. Chung, and K. Petroske. Efficient Krylov subspace methods for uncertainty quantification in large Bayesian linear inverse problems. *Numerical Linear Algebra with Applications*, 27(5):e2325, 2020.
[210] A. K. Saibaba and P. K. Kitanidis. Fast computation of uncertainty quantification measures in the geostatistical approach to solve inverse problems. *Advances in Water Resources*, 82(0):124 – 138, 2015.
[211] M. Schmidt, G. Fung, and R. Rosales. Fast optimization methods for l1 regularization: A comparative study and two new approaches. In *European Conference on Machine Learning*, pages 286–297. Springer, 2007.
[212] M. K. Schneider and A. S. Willsky. A Krylov subspace method for covariance approximation and simulation of random processes and fields. *Multidimensional Systems and Signal Processing*, 14(4):295–318, 2003.
[213] J. K. Seo and E. J. Woo. *Nonlinear inverse problems in imaging*. John Wiley & Sons, 2012.
[214] J. A. Simmons. *Deconvolution for acoustic emission*, pages 727–735. Springer-Verlag US, 1986.
[215] H. D. Simon and H. Zha. Low-rank matrix approximation using the Lanczos bidiagonalization process with applications. *SIAM J. Sci. Comput.*, 21(6):2257–2274, 2000.
[216] V. Simoncini and D. B. Szyld. Recent computational developments in Krylov subspace methods for linear systems. *Numerical Linear Algebra with Applications*, 14(1):1–59, 2007.
[217] D. P. Simpson. *Krylov subspace methods for approximating functions of symmetric positive definite matrices with applications to applied statistics and anomalous diffusion*. PhD thesis, Queensland University of Technology, 2008.
[218] K. M. Soodhalter. A note on augmented unprojected Krylov subspace methods. *Electronic Transactions on Numerical Analysis*, 55:532–546, 2022.

[219] K. M. Soodhalter, E. de Sturler, and M. E. Kilmer. A survey of subspace recycling iterative methods. *GAMM-Mitteilungen*, 43(4):e202000016, 2020.
[220] K. M. Soodhalter, D. B. Szyld, and F. Xue. Krylov subspace recycling for sequences of shifted linear systems. *Applied Numerical Mathematics*, 81:105–118, 2014.
[221] A. Spantini, A. Solonen, T. Cui, J. Martin, L. Tenorio, and Y. Marzouk. Optimal low-rank approximations of Bayesian linear inverse problems. *SIAM J. Sci. Comput.*, 37(6):A2451–A2487, 2015.
[222] A. M. Stuart. Inverse problems: a Bayesian perspective. *Acta Numerica*, 19:451–559, 2010.
[223] L. Tenorio. *An Introduction to Data Analysis and Uncertainty Quantification for Inverse Problems*. SIAM, 2017.
[224] A. N. Tikhonov. Regularization of incorrectly posed problems. *Soviet Math. Doklady*, 4:1624–1627, 1963.
[225] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Doklady*, 4:1035–1038, 1963.
[226] F. Toutounian and S. Karimi. Global least squares method (Gl-LSQR) for solving general linear systems with several right-hand sides. *Applied Mathematics and Computation*, 178(2):452–460, 2006.
[227] V. F. Turchin. Solution of Fredholm equations of the first kind in a statistical ensemble of smooth functions. *USSR Comput. Math. Math. Phys.*, 7:79–101, 1967.
[228] W. van Aarle, W. J. Palenstijn, J. Cant, E. Janssens, F. Bleichrodt, A. Dabravolski, J. D. Beenhouwer, K. J. Batenburg, and J. Sijbers. Fast and flexible x-ray tomography using the ASTRA toolbox. *Opt. Express*, 24(22):25129–25147, 2016.
[229] S. Vänskä, M. Lassas, and S. Siltanen. Statistical x-ray tomography using empirical Besov priors. *International Journal of Tomography & Statistics*, 11(S09):3–32, 2009.
[230] U. Villa, N. Petra, and O. Ghattas. hIPPYlib: An extensible software framework for large-scale inverse problems governed by PDEs: Part I: Deterministic inversion and linearized Bayesian inference. *ACM Trans. Math. Softw.*, 47(2), 2021.
[231] B. F. Viloche and L. S. Borges. GKB-FP: an algorithm for large-scale discrete ill-posed problems. *BIT Numerical Mathematics*, 50(3):481–507, 2010.
[232] B. F. Viloche, M. Cunha, and L. S. Borges. Extension of GKB-FP algorithm to large-scale general-form Tikhonov regularization. *Numerical Linear Algebra with Applications*, 21(3):316–339, 2014.
[233] F. S. Viloche Bazan, L. S. Borges, J. B. Francisco, et al. On a generalization of Reginska's parameter choice rule and its numerical realization in large-scale multi-parameter Tikhonov regularization. *Applied Mathematics and Computation*, 219(4):2100, 2012.
[234] C. R. Vogel. *Computational Methods for Inverse Problems*. SIAM, Philadelphia, PA, 2002.
[235] Y. Wang. Regularization for inverse models in remote sensing. *Progress in Physical Geography*, 36:38–59, 2015.
[236] P. Wendykier and J. G. Nagy. Parallel colt: A high-performance java library for scientific computing and image processing. *ACM transactions on mathematical software (TOMS)*, 37(3):1–22, 2010.
[237] B. Wohlberg and P. Rodriguez. An iteratively reweighted norm algorithm for minimization of total variation functionals. *IEEE Signal Processing Letters*, 14(12):948–951, 2007.
[238] S. J. Wright, R. D. Nowak, and M. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Trans. Signal Process.*, 57(7):2479–2493, 2009.
[239] Z. S. Yao and R. G. Roberts. A practical regularization for seismic tomography. *Geophysical Journal International*, 138(2):293–299, 1999.
[240] H. Zha. Computing the generalized singular values/vectors of large sparse or structured matrix pairs. *Numerische Mathematik*, 72(3):391–417, 1996.