

PAPER

Hybrid projection methods for large-scale inverse problems with mixed Gaussian priors

To cite this article: Taewon Cho *et al* 2021 *Inverse Problems* **37** 044002

View the [article online](#) for updates and enhancements.



IOP | ebooks™

Bringing together innovative digital publishing with leading authors from the global scientific community.

Start exploring the collection—download the first chapter of every title for free.

Hybrid projection methods for large-scale inverse problems with mixed Gaussian priors

Taewon Cho¹, Julianne Chung²  and Jiahua Jiang^{1,*} 

¹ Department of Mathematics, Virginia Tech, Blacksburg, VA, United States of America

² Department of Mathematics, Computational Modeling and Data Analytics Division, Academy of Integrated Science, Virginia Tech, Blacksburg, VA, United States of America

E-mail: jiahua@vt.edu

Received 30 March 2020, revised 2 December 2020

Accepted for publication 10 December 2020

Published 8 March 2021



Abstract

When solving ill-posed inverse problems, a good choice of the prior is critical for the computation of a reasonable solution. A common approach is to include a Gaussian prior, which is defined by a mean vector and a symmetric and positive definite covariance matrix, and to use iterative projection methods to solve the corresponding regularized problem. However, a main challenge for many of these iterative methods is that the prior covariance matrix must be known and fixed (up to a constant) before starting the solution process. In this paper, we develop hybrid projection methods for inverse problems with mixed Gaussian priors where the prior covariance matrix is a convex combination of matrices and the mixing parameter and the regularization parameter do not need to be known in advance. Such scenarios may arise when data is used to generate a sample prior covariance matrix (e.g., in data assimilation) or when different priors are needed to capture different qualities of the solution. The proposed hybrid methods are based on a mixed Golub–Kahan process, which is an extension of the generalized Golub–Kahan bidiagonalization, and a distinctive feature of the proposed approach is that *both* the regularization parameter and the weighting parameter for the covariance matrix can be estimated automatically during the iterative process. Furthermore, for problems where training data are available, various data-driven covariance matrices (including those based on learned covariance kernels) can be easily incorporated. Numerical examples from tomographic reconstruction demonstrate the potential for these methods.

*Author to whom any correspondence should be addressed.

Keywords: generalized Golub–Kahan, hybrid projection methods, Tikhonov regularization, Bayesian inverse problems, sample covariance matrix, tomography

(Some figures may appear in colour only in the online journal)

1. Introduction

For many imaging systems, the ability to obtain good image reconstructions from observed data requires the inclusion of a suitable prior. Priors provide a systematic and efficient means to describe in probabilistic terms any prior knowledge about the unknowns [5, 11]. Often-times prior knowledge will come from a *combination* of sources, and striking a good balance of information is critical. For example, priors may be learned from available training data, but bias in the reconstructions can be a big concern (e.g., when the training set is small or the desired image is very different from the training set) [2, 14, 28]. Thus, a safer approach is to include a prior that combines learned information with conventional smoothness properties. In other scenarios (e.g. in seismic tomography [1]), the desired solution may consist of components with different smoothness properties, and the correct mixture of smoothness priors can be difficult to know *a priori*. Using mixed Gaussian priors, where the prior covariance matrix can be represented as a convex combination of matrices, is a common approach to incorporate different prior covariance matrices [3]. However, various computational challenges arise for problems where the number of unknowns is very large and the regularization and mixing parameter are not known in advance. We address these challenges by developing hybrid iterative projection methods for the efficient computation of solutions to inverse problems with mixed Gaussian priors. By exploiting a project-then-regularize framework, we enable statistical optimization tools for selecting the regularization parameter and the mixing parameter automatically, which would be very costly for the original problem.

We are interested in linear inverse problems of the form,

$$\mathbf{d} = \mathbf{As} + \boldsymbol{\epsilon}, \quad (1)$$

where $\mathbf{d} \in \mathbb{R}^m$ contains the observed data, $\mathbf{A} \in \mathbb{R}^{m \times n}$ models the forward process, $\mathbf{s} \in \mathbb{R}^n$ represents the desired parameters, and $\boldsymbol{\epsilon} \in \mathbb{R}^m$ represents noise in the data. We assume that $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$, where \mathbf{R} is a symmetric positive definite matrix whose inverse and square root are inexpensive (e.g., a diagonal matrix). The goal of the inverse problem is to compute an approximation of \mathbf{s} , given \mathbf{d} and \mathbf{A} .

Due to ill-posedness, small errors in the data may lead to large errors in the computed approximation of \mathbf{s} , and regularization is required to stabilize the inversion process. We follow a Bayesian framework, where we assume a prior for \mathbf{s} . That is, we treat \mathbf{s} as a Gaussian random variable with mean vector $\boldsymbol{\mu} \in \mathbb{R}^n$ and covariance matrix $\mathbf{Q} \in \mathbb{R}^{n \times n}$. That is, $\mathbf{s} \sim \mathcal{N}(\boldsymbol{\mu}, \lambda^{-2}\mathbf{Q})$, where λ is a scaling parameter (yet to be determined) for the inverse of the covariance matrix.

In many applications, the choice of \mathbf{Q} is pre-determined (e.g., using expert knowledge) and is chosen to enforce smoothness or regularity conditions on the solution [9, 17, 31]. However, in some cases, there is not enough information to determine \mathbf{Q} completely or expensive procedures are needed to determine an informative subset of covariates from a set of candidates (e.g., in geophysical imaging [43, 45, 46]). These scenarios motivate us to consider mixed Gaussian priors, where the covariance matrix can be represented as a convex combination of matrices.

Without loss of generality we consider prior covariance matrices of the form,

$$\mathbf{Q} = \gamma \mathbf{Q}_1 + (1 - \gamma) \mathbf{Q}_2, \quad (2)$$

where \mathbf{Q}_1 is a symmetric positive definite matrix, \mathbf{Q}_2 is a symmetric positive semi-definite matrix, and mixing parameter $0 < \gamma \leq 1$. We consider the case where computing matrix-vector products with \mathbf{Q}_1 is easy, but accessing \mathbf{Q}_1^{-1} or its symmetric factorization (e.g., Cholesky or eigenvalue factorization) is not feasible. Such scenarios arise, for example, when the prior covariance matrix is modeled entry-wise using covariance kernels. In such cases, the main challenge is that the resulting covariance matrices are large and dense, and factorizing or inverting them can be computationally prohibitive. However, matrix-vector multiplications can be done efficiently (e.g., via FFT embedding). A wide range of kernels, including nonseparable spatio-temporal kernels [17], can be included. We assume that matrix-vector products with \mathbf{Q}_2 can be done efficiently.

Covariance matrices of the form (2) are becoming more common, especially in modern imaging applications where data (e.g., in the form of training images) are playing a larger role in the development of reconstruction algorithms [2]. Suppose we are given a dataset consisting of N samples, $\mathbf{s}^{(i)} \in \mathbb{R}^n, i = 1, 2, \dots, N$. Then the training data can be used to obtain an unbiased estimator of an $n \times n$ sample covariance matrix,

$$\hat{\mathbf{Q}} = \frac{1}{N} \sum_{i=1}^N (\mathbf{s}^{(i)} - \bar{\mathbf{s}})(\mathbf{s}^{(i)} - \bar{\mathbf{s}})^\top, \quad (3)$$

where $\bar{\mathbf{s}} = \frac{1}{N} \sum_{i=1}^N \mathbf{s}^{(i)}$ is the sample mean. Notice that $\hat{\mathbf{Q}} = \mathbf{S} \mathbf{S}^\top$, where the symmetric factor is defined as $\mathbf{S} = \frac{1}{\sqrt{N}} ([\mathbf{s}^{(1)} \ \dots \ \mathbf{s}^{(N)}] - \bar{\mathbf{s}} \otimes \mathbf{1}^\top)$ with $\mathbf{1} \in \mathbb{R}^N$ denoting the vector whose elements are all 1. For any vector $\mathbf{x} \in \mathbb{R}^n$, multiplication with $\hat{\mathbf{Q}}$ can be done efficiently if $N \ll n$, e.g., using the following order of operations $\mathbf{S}(\mathbf{S}^\top \mathbf{x})$. However, notice that $\hat{\mathbf{Q}}$ is likely positive semi-definite rather than positive definite, so it is common to use $\hat{\mathbf{Q}} + \gamma \mathbf{I}$ where γ is a nudging term. Such approaches are known as sample based priors [10]. Another common approach is to use a convex combination, i.e., the prior covariance matrix is given as

$$\mathbf{Q} = \gamma \mathbf{D} + (1 - \gamma) \hat{\mathbf{Q}}, \quad (4)$$

where \mathbf{D} is chosen to be the identity matrix or a suitably chosen diagonal or correlation matrix, which ensures that \mathbf{Q} is positive definite, and $\gamma \in \mathbb{R}$ is called the mixing parameter. The matrix in (4) is called a shrinkage estimator of the covariance matrix [39]. It is worth noting that covariance matrices of the form (4) are also used in hybrid methods for data assimilation that combine an ensemble Kalman filter system with a variational (e.g., 3D-Var) system [3]. These methods require careful tuning of the so-called blending parameter γ , and many of the existing approaches require γ to be fixed in advance. We do not assume this.

Previous works on combining training data with regularization techniques typically follow an optimal experimental design or empirical Bayes risk minimization framework [14, 28]. More recently, there has been significant work on using training data in the context of machine learning to learn regularization functionals (e.g., [33, 40]) or to learn the ‘invisible’ regions (e.g., [8]). The area of data-driven machine learning is currently a hot topic [2, 36], where the main goal is to determine new ways to combine physical models with deep learning techniques. In this work, we incorporate training data in a Bayesian framework and exploit tools from numerical linear algebra not only to compute solutions efficiently but also to determine the appropriate weighting of the training data.

In this paper we develop a hybrid iterative projection method that is based on a mixed, generalized Golub–Kahan process to approximate the MAP estimate,

$$\mathbf{s}_{\text{MAP}} = \arg \min_{\mathbf{s}} \frac{1}{2} \|\mathbf{A}\mathbf{s} - \mathbf{d}\|_{\mathbf{R}^{-1}}^2 + \frac{\lambda^2}{2} \|\mathbf{s} - \boldsymbol{\mu}\|_{\mathbf{Q}^{-1}}^2, \quad (5)$$

where \mathbf{Q} is of the form (2). Our approach can handle a wide range of scenarios, including data-informed regularization terms that use training or test images to define the prior. We assume that γ is not known in advance and neither the inverse nor a factorization of \mathbf{Q} is available. The proposed method has two distinctive features. First, we assume that *both* γ and λ are unknown *a priori* and we estimate them during the solution process. For large-scale problems where γ and hence \mathbf{Q} is fixed in advance, generalized hybrid methods [17] have been developed where λ can be estimated during the reconstruction process. On the other hand, previous works have used linear combinations of ‘simple’ covariance matrices and statistical methods to estimate mixing parameter γ , but the choice of regularization parameter λ is fixed [12, 32]. To the best of our knowledge, both problems have not been addressed simultaneously, and developing a hybrid method where both λ and γ can be selected adaptively is not an obvious extension of existing methods. We develop an iterative hybrid approach where the problem is projected onto generalized Krylov subspaces of small but increasing dimension and the regularization parameter and mixing parameter can be simultaneously and automatically selected. Second, we describe and investigate various scenarios where training data can be used to define \mathbf{Q}_1 and \mathbf{Q}_2 , so our approach can be considered a learning approach for the regularization term. There has been a large emphasis on using training data in the development of reconstruction techniques for modern imaging problems, but the ability to balance training data with physical priors remains an open problem. A key feature of the proposed hybrid methods is the ability to incorporate data-driven covariance matrices while simultaneously balancing existing priors. In our numerical experiments, we have verified that such methods can perform better than classical shrinkage algorithms, especially when the data contain artifacts.

An outline for the paper is as follows. In section 2 we provide some background on Gaussian priors and focus on various data-driven prior covariance matrices. Then in section 3, we describe mixed, generalized hybrid projection methods for approximating the MAP estimate (5), where \mathbf{Q} is of the form (2). The approach consists of two-steps: (1) project the problem onto a subspace of small but increasing dimension using an extension of the generalized Golub–Kahan bidiagonalization approach. (2) Solve the projected problem where the regularization parameter λ and mixing parameter γ can be selected automatically. Various regularization parameter selection techniques will be investigated, and some theoretical results will be provided. In section 4 numerical results on various image processing applications show the potential benefits and flexibility of these methods. Conclusions are provided in section 5.

2. Mixed Gaussian priors

In this section, we motivate the need for mixed Gaussian priors and draw some connections to existing works on multi-parameter Tikhonov regularization and shrinkage estimation.

To begin, we focus on using Gaussian random fields to represent prior information and summarize some common choices for the (unscaled) prior covariance matrix \mathbf{Q} . Oftentimes, the covariance matrix is generated using a covariance function (also called a kernel function). Covariance functions are crucial in many fields and encode assumptions about the form of

Table 1. Summary of commonly-used covariance functions. The covariance functions are written either as functions of \mathbf{z}_i and \mathbf{z}_j , or as a function of $r = |\mathbf{z}_i - \mathbf{z}_j|$ and depend on ℓ or ℓ and ν . Γ is the gamma function and $K_\nu(\cdot)$ is the modified Bessel function of the second kind of order ν .

Covariance kernel function	
Squared exponential	$\exp\left(-\frac{r^2}{2\ell^2}\right)$
Matérn	$\frac{1}{2^{\nu-1}\Gamma(\nu)}\left(\frac{\sqrt{2\nu}r}{\ell}\right)^\nu K_\nu\left(\frac{\sqrt{2\nu}r}{\ell}\right)$
γ -exponential	$\exp\left(-\left(\frac{r}{\ell}\right)^\gamma\right)$
Rational quadratic	$\left(1 + \frac{r^2}{2\nu\ell^2}\right)^{-\nu}$
Sinc	$\frac{\sin(\nu r)}{\nu r}$

the function that we are modeling. In most cases, the prior covariance matrix \mathbf{Q} is large and dense with entries directly computed as $\mathbf{Q}_{ij} = \kappa(\mathbf{z}_i, \mathbf{z}_j)$, where $\{\mathbf{z}_i\}_{i=1}^n$ are the spatial points in the domain and $\kappa(\cdot, \cdot)$ is a covariance kernel function. Some commonly used parametric covariance functions [37] are provided in table 1.

For some kernel choices, the precision matrix (i.e., the inverse of the covariance matrix) is sparse or structured, so working with \mathbf{Q}^{-1} or its symmetric factorization has obvious computational advantages. However, in many applications, the precision matrix is not readily available, and the aim is to develop computational methods that can work with \mathbf{Q} directly and avoid the need for the inverse or symmetric factorization. Such covariance kernels may arise in dynamic scenarios with nonseparable, spatio-temporal priors [18, 24, 34] or from spatially-variant priors [22, 44]. It is worth mentioning that in a truly Bayesian framework, the regularization parameter and the covariance kernel parameters could be included as hyperparameters and explored using MCMC methods [5], but the computational costs of this approach would be very high.

One reason to use Gaussian mixtures as prior distributions is that it allows greater flexibility in the definition of the prior. In this paper, we consider a mixture of two Gaussians, but one could consider more general mixtures. From a statistical viewpoint, a general formulation with N Gaussian random vectors would correspond to a sum of covariance matrices. That is, let $\mathbf{x}_1, \dots, \mathbf{x}_N$ be N mutually independent $n \times 1$ normal random vectors having means $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_N$ and covariance matrices $\mathbf{V}_1, \dots, \mathbf{V}_N$. Let $\mathbf{B}_1, \dots, \mathbf{B}_N$ be real $L \times n$ full rank matrices. Then the $L \times 1$ random vector

$$\mathbf{y} = \sum_{i=1}^N \mathbf{B}_i \mathbf{x}_i \quad (6)$$

has a normal distribution with mean $\mathbb{E}\mathbf{y} = \sum_{i=1}^N \mathbf{B}_i \boldsymbol{\mu}_i$ and covariance matrix of the form $\text{Cov}(\mathbf{y}) = \sum_{i=1}^N \mathbf{B}_i \mathbf{V}_i \mathbf{B}_i^\top$. Thus, a Gaussian mixture prior corresponds to an assumption that the desired solution can be represented as a linear combination of Gaussian realizations (e.g., with different smoothness properties).

In the context of inverse problems, we point out a connection between mixed Gaussian priors and multi-parameter Tikhonov regularization. The basic idea of multi-parameter Tikhonov regularization, see e.g. [6, 26, 35, 42], is to solve a problem of the form,

$$\min_{\mathbf{s}} \|\mathbf{A}\mathbf{s} - \mathbf{d}\|_{\mathbf{R}^{-1}}^2 + \sum_{i=1}^N \lambda_i^2 \|\mathbf{L}_i \mathbf{s}\|_2^2, \quad (7)$$

where $\lambda_i \in \mathbb{R}$ is the regularization parameter corresponding to regularization matrix \mathbf{L}_i for $i = 1, \dots, N$. By including multiple penalty terms, this approach can enforce different smoothness properties (e.g., at different frequency bands) and avoid difficulties in having to select just one regularization matrix. In a Bayesian framework, the multi-parameter Tikhonov solution can be interpreted as an MAP estimate, under the assumption of a Gaussian prior with mean $\mathbf{0}$ and covariance matrix $\left(\sum_{i=1}^N \lambda_i^2 \mathbf{L}_i^\top \mathbf{L}_i\right)^{-1}$. Notice that except for in very limited scenarios, this is not the same as using mixed Gaussian priors, since here the precision matrix (not the covariance matrix) is represented as a sum of matrices.

2.1. Data-driven prior covariance matrices

With the increasing amount of, and access to, data in many applications, an important and challenging task is to determine how to efficiently and effectively incorporate prior knowledge in the form of training data both in the solution computation process and the subsequent data analyses. In this section, we describe various examples where training data can be used to define the prior covariance matrix. For all cases, we assume that training data is provided and the sample covariance matrix (3) has the form $\hat{\mathbf{Q}} = \mathbf{S}\mathbf{S}^\top$.

As described in the introduction, the most common approach is to take $\mathbf{Q}_2 = \hat{\mathbf{Q}}$ and $\mathbf{Q}_1 = \mathbf{D}$ where \mathbf{D} is easy to invert (e.g., diagonal or identity matrix). In this case, a very popular approach called shrinkage estimation of covariance matrices, or more general biased estimation, can be used to reduce the variance of the estimator. Typical shrinkage targets are diagonal matrices (e.g., including the identity matrix), and approaches to estimate the optimal shrinkage intensity γ have been proposed by Ledoit and Wolf, Rao and Blackwell, and others [3, 13, 32, 39].

Another approach to incorporate training data is to force some structure or functional form on the prior covariance kernel function. For kernel functions that depend on a few parameters, the training data can be used to estimate these parameters. A similar idea was considered in [28] where training data was used to learn parameters defining the regularization functional. However, that approach requires solving an expensive constrained optimization problem, and the learned regularization functional is tailored to the forward operator and the noise level. We consider the case where the training data come from a prior defined by a covariance kernel function (e.g., for simplicity, we consider Matérn kernels). We use the training data to learn the parameters defining the prior. This reduces to an optimization problem where the goal is to learn two parameters ν and ℓ from the training data by solving the optimization problem,

$$(\hat{\nu}, \hat{\ell}) = \arg \min_{\nu > 0, \ell > 0} \|\mathbf{Q}(\nu, \ell) - \hat{\mathbf{Q}}\|_F^2. \quad (8)$$

Once the parameters are computed, they can be used to define $\mathbf{Q}_1 = \mathbf{Q}(\hat{\nu}, \hat{\ell})$, which can be used directly in generalized hybrid methods, or they can be combined with the sample covariance matrix, i.e., \mathbf{Q} as in (2) with $\mathbf{Q}_1 = \mathbf{Q}(\hat{\nu}, \hat{\ell})$ and $\mathbf{Q}_2 = \hat{\mathbf{Q}}$, and solvers described in section 3 can be used.

Next, we describe some computationally efficient methods to estimate $\hat{\nu}$ and $\hat{\ell}$. Notice that

$$\|\mathbf{Q}(\nu, \ell) - \hat{\mathbf{Q}}\|_F^2 = \text{tr}(\mathbf{Q}(\nu, \ell) - \hat{\mathbf{Q}})^\top (\mathbf{Q}(\nu, \ell) - \hat{\mathbf{Q}}) \quad (9)$$

$$= \mathbb{E}(\|(\mathbf{Q}(\nu, \ell) - \hat{\mathbf{Q}})\xi\|_2^2), \quad (10)$$

where ξ is a random variable such that $\mathbb{E}\xi = \mathbf{0}$ and $\mathbb{E}(\xi\xi^\top) = \mathbf{I}$. Although stochastic optimization methods [41] could be used here, we follow an approximation approach where we use a Hutchinson trace estimator. That is, we let $\xi^{(i)} \in \mathbb{R}^n$ for $i = 1, 2, \dots, M$ be realizations of a Rademacher distribution (i.e., ξ consists of ± 1 with equal probability), and we consider the approximate optimization problem,

$$(\check{\nu}, \check{\ell}) = \arg \min_{\nu > 0, \ell > 0} \frac{1}{M} \sum_{i=1}^M \|(\mathbf{Q}(\nu, \ell) - \hat{\mathbf{Q}})\xi^{(i)}\|_2^2. \quad (11)$$

We used an interior-point method (`fmincon.m` in MATLAB) to minimize (11) with $M = 100$. We observed no significant improvement for larger values of M . We mention that for problems without training data, semivariogram hyperparameters were investigated in [7] to estimate Matérn parameters from the data.

3. Hybrid projection methods for mixed Gaussian priors

In this section, we describe a hybrid projection method to approximate the MAP estimate (5). The distinguishing factor of this approach compared to generalized Golub–Kahan (genGK) hybrid methods [17] is that we address problems where the prior covariance matrix is of the form (2). That is, we consider priors of the form $\mathbf{s} \sim \mathcal{N}(\mu, \lambda^{-2}(\gamma\mathbf{Q}_1 + (1 - \gamma)\mathbf{Q}_2))$, and exploit a hybrid projection framework to enable tools for selecting both the regularization parameter λ and the mixing parameter γ simultaneously. Using the following change of variables,

$$\mathbf{x} = \mathbf{Q}^{-1}(\mathbf{s} - \mu), \quad \mathbf{b} = \mathbf{d} - \mathbf{A}\mu,$$

we see that solving (5) is equivalent to solving

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{A}\mathbf{Q}\mathbf{x} - \mathbf{b}\|_{\mathbf{R}^{-1}}^2 + \frac{\lambda^2}{2} \|\mathbf{x}\|_{\mathbf{Q}}^2. \quad (12)$$

If γ is known in advance, we can directly apply the genGK hybrid method and estimate λ automatically [17]. However, if γ is not known in advance, significant computations would be required for a different choice of γ . For this, we develop a variant of the genGK bidiagonalization which we call a *mixed* Golub–Kahan (mixGK) process, where both γ and λ can be estimated during the iterative process. Each iteration of the mixGK process requires two steps. The first step is to run one iteration of the genGK bidiagonalization process with \mathbf{Q}_1 . The second step incorporates \mathbf{Q}_2 so that the regularized problem can be iteratively projected onto a smaller subspace, and γ and λ can *both* be selected automatically. Next we describe the mixGK process in detail.

Given matrices \mathbf{A} , \mathbf{R} , \mathbf{Q}_1 , and vector \mathbf{b} , with initializations $\beta_1 = \|\mathbf{b}\|_{\mathbf{R}^{-1}}$, $\mathbf{u}_1 = \mathbf{b}/\beta_1$ and $\alpha_1 \mathbf{v}_1 = \mathbf{A}^\top \mathbf{R}^{-1} \mathbf{u}_1$, the k th iteration of the genGK bidiagonalization procedure with \mathbf{Q}_1 generates vectors \mathbf{u}_{k+1} and \mathbf{v}_{k+1} such that

$$\begin{aligned} \beta_{k+1} \mathbf{u}_{k+1} &= \mathbf{A}\mathbf{Q}_1 \mathbf{v}_k - \alpha_k \mathbf{u}_k \\ \alpha_{k+1} \mathbf{v}_{k+1} &= \mathbf{A}^\top \mathbf{R}^{-1} \mathbf{u}_{k+1} - \beta_{k+1} \mathbf{v}_k, \end{aligned}$$

Algorithm 1. Mixed Golub–Kahan (mixGK) process.

Require: matrices \mathbf{A} , \mathbf{R} , \mathbf{Q}_1 and \mathbf{Q}_2 , and vector \mathbf{b} .

- 1: $\beta_1 \mathbf{u}_1 = \mathbf{b}$, where $\beta_1 = \|\mathbf{b}\|_{\mathbf{R}^{-1}}$
- 2: $\alpha_1 \mathbf{v}_1 = \mathbf{A}^\top \mathbf{R}^{-1} \mathbf{u}_1$
- 3: **for** $k = 1, 2, \dots$ **do**
- 4: $\beta_{k+1} \mathbf{u}_{k+1} = \mathbf{A} \mathbf{Q}_1 \mathbf{v}_k - \alpha_k \mathbf{u}_k$, where $\beta_{k+1} = \|\mathbf{A} \mathbf{Q}_1 \mathbf{v}_k - \alpha_k \mathbf{u}_k\|_{\mathbf{R}^{-1}}$
- 5: $\alpha_{k+1} \mathbf{v}_{k+1} = \mathbf{A}^\top \mathbf{R}^{-1} \mathbf{u}_{k+1} - \beta_{k+1} \mathbf{v}_k$, where $\alpha_{k+1} = \|\mathbf{A}^\top \mathbf{R}^{-1} \mathbf{u}_{k+1} - \beta_{k+1} \mathbf{v}_k\|_{\mathbf{Q}_1}$
- 6: $[\mathbf{Y}_k, \mathbf{R}_k] = qr(\mathbf{I} - \tilde{\mathbf{U}}_{k+1} \tilde{\mathbf{U}}_{k+1}^\top) \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k, 0)$;
- 7: **end for**

where scalars $\alpha_i, \beta_i \geq 0$ are chosen such that $\|\mathbf{u}_i\|_{\mathbf{R}^{-1}} = \|\mathbf{v}_i\|_{\mathbf{Q}_1} = 1$. At the end of k steps, we have

$$\mathbf{B}_k \equiv \begin{bmatrix} \alpha_1 & & & & \\ \beta_2 & \alpha_2 & & & \\ & \ddots & \ddots & & \\ & & \beta_3 & \ddots & \\ & & & \ddots & \alpha_k \\ & & & & \beta_{k+1} \end{bmatrix}, \quad \mathbf{U}_{k+1} \equiv [\mathbf{u}_1, \dots, \mathbf{u}_{k+1}], \text{ and}$$

$$\mathbf{V}_k \equiv [\mathbf{v}_1, \dots, \mathbf{v}_k],$$

where the following relations hold up to machine precision,

$$\mathbf{U}_{k+1} \beta_1 \mathbf{e}_1 = \mathbf{b} \quad (13)$$

$$\mathbf{A} \mathbf{Q}_1 \mathbf{V}_k = \mathbf{U}_{k+1} \mathbf{B}_k \quad (14)$$

$$\mathbf{A}^\top \mathbf{R}^{-1} \mathbf{U}_{k+1} = \mathbf{V}_k \mathbf{B}_k^\top + \alpha_{k+1} \mathbf{v}_{k+1} \mathbf{e}_{k+1}^\top. \quad (15)$$

Furthermore, in exact arithmetic, matrices \mathbf{U}_{k+1} and \mathbf{V}_k satisfy the following orthogonality conditions

$$\mathbf{U}_{k+1}^\top \mathbf{R}^{-1} \mathbf{U}_{k+1} = \mathbf{I}_{k+1} \quad \text{and} \quad \mathbf{V}_k^\top \mathbf{Q}_1 \mathbf{V}_k = \mathbf{I}_k. \quad (16)$$

If we let $\tilde{\mathbf{U}}_{k+1} = \mathbf{L}_R \mathbf{U}_{k+1}$ where $\mathbf{R}^{-1} = \mathbf{L}_R^\top \mathbf{L}_R$, then $\tilde{\mathbf{U}}_{k+1}^\top \tilde{\mathbf{U}}_{k+1} = \mathbf{I}_{k+1}$.

Next, in order to incorporate \mathbf{Q}_2 , we additionally compute $m \times k$ matrix $\mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k$. Assuming that the columns of $\tilde{\mathbf{U}}_{k+1}$ and $\mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k$ are linearly independent, we can compute the skinny QR factorization, $(\mathbf{I} - \tilde{\mathbf{U}}_{k+1} \tilde{\mathbf{U}}_{k+1}^\top) \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k = \mathbf{Y}_k \mathbf{R}_k$ where $\mathbf{Y}_k \in \mathbb{R}^{m \times k}$ contains orthonormal columns and $\mathbf{R}_k \in \mathbb{R}^{k \times k}$ is upper triangular. Notice that since column vectors in \mathbf{Y}_k and $\tilde{\mathbf{U}}_{k+1}$ are orthogonal, we get the skinny QR factorization,

$$[\tilde{\mathbf{U}}_{k+1} \quad \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k] = [\tilde{\mathbf{U}}_{k+1} \quad \mathbf{Y}_k] \begin{bmatrix} \mathbf{I}_{k+1} & \tilde{\mathbf{U}}_{k+1}^\top \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k \\ \mathbf{0} & \mathbf{R}_k \end{bmatrix}. \quad (17)$$

The mixGK process is summarized in algorithm 1.

Notice that in addition to the computational cost of the genGK bidiagonalization, which includes one matrix-vector product with \mathbf{A} , one with \mathbf{A}^\top , two with \mathbf{Q}_1 , and two solves with \mathbf{R} , each iteration of the mixGK process requires one matrix-vector product with \mathbf{Q}_2 and a QR

factorization in step 6. Instead of performing a standard QR factorization on an m -by- k matrix, an efficient rank-one update strategy can be used to alleviate the computational cost. More specifically, we will describe it using mathematical induction. Let

$$(\mathbf{I} - \tilde{\mathbf{U}}_k \tilde{\mathbf{U}}_k^\top) \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_{k-1} = \mathbf{Y}_{k-1} \mathbf{R}_{k-1} \quad (18)$$

be the skinny QR factorization, where $\mathbf{Y}_{k-1}^\top \mathbf{Y}_{k-1} = \mathbf{I}_{k-1}$ and \mathbf{R}_{k-1} is an upper triangular matrix. Define $\tilde{\mathbf{U}}_{k+1} = [\tilde{\mathbf{U}}_k \ \tilde{\mathbf{u}}_{k+1}]$ and $\mathbf{V}_k = [\mathbf{V}_{k-1} \ \mathbf{v}_k]$. Then by (18), we have

$$\begin{aligned} (\mathbf{I} - \tilde{\mathbf{U}}_{k+1} \tilde{\mathbf{U}}_{k+1}^\top) \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k &= [(\mathbf{I} - \tilde{\mathbf{U}}_{k+1} \tilde{\mathbf{U}}_{k+1}^\top) \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_{k-1} \ (\mathbf{I} - \tilde{\mathbf{U}}_{k+1} \tilde{\mathbf{U}}_{k+1}^\top) \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{v}_k] \\ &= [(\mathbf{I} - \tilde{\mathbf{U}}_k \tilde{\mathbf{U}}_k^\top - \tilde{\mathbf{u}}_{k+1} \tilde{\mathbf{u}}_{k+1}^\top) \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_{k-1} \ (\mathbf{I} - \tilde{\mathbf{U}}_{k+1} \tilde{\mathbf{U}}_{k+1}^\top) \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{v}_k] \\ &= [\mathbf{Y}_{k-1} \mathbf{R}_{k-1} - \tilde{\mathbf{u}}_{k+1} \tilde{\mathbf{u}}_{k+1}^\top \mathbf{Y}_{k-1} \mathbf{R}_{k-1} \ (\mathbf{I} - \tilde{\mathbf{U}}_{k+1} \tilde{\mathbf{U}}_{k+1}^\top) \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{v}_k]. \end{aligned}$$

Since the first matrix is a rank-one update of a QR factorization, its QR factorization can be obtained in $\mathcal{O}(mk)$ operations [21]. That is, we have

$$\mathbf{Y}_{k-1} \mathbf{R}_{k-1} - \tilde{\mathbf{u}}_{k+1} (\mathbf{R}_{k-1}^\top \mathbf{Y}_{k-1}^\top \tilde{\mathbf{u}}_{k+1})^\top = \hat{\mathbf{Y}}_{k-1} \hat{\mathbf{R}}_{k-1},$$

where $\hat{\mathbf{Y}}_{k-1}^\top \hat{\mathbf{Y}}_{k-1} = \mathbf{I}_{k-1}$ and $\hat{\mathbf{R}}_{k-1}$ is an upper triangular matrix. Finally, let $\hat{\mathbf{v}}_k = (\mathbf{I} - \tilde{\mathbf{U}}_{k+1} \tilde{\mathbf{U}}_{k+1}^\top) \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{v}_k$, then one step of the Gram–Schmidt process gives the desired QR factorization,

$$[\hat{\mathbf{Y}}_{k-1} \hat{\mathbf{R}}_{k-1} \hat{\mathbf{v}}_k] = \mathbf{Y}_k \mathbf{R}_k.$$

3.1. Solving the projected problem

Using the mixGK process described above, we now describe a hybrid iterative projection method to solve (12). In particular, we consider the projected problem,

$$\min_{\mathbf{x} \in \mathcal{R}(\mathbf{V}_k)} \frac{1}{2} \|\mathbf{A} \mathbf{Q} \mathbf{x} - \mathbf{b}\|_{\mathbf{R}^{-1}}^2 + \frac{\lambda^2}{2} \|\mathbf{x}\|_{\mathbf{Q}}^2, \quad (19)$$

where $\mathcal{R}(\cdot)$ denotes the column space. Let $\mathbf{x} = \mathbf{V}_k \mathbf{y}$ where $\mathbf{y} \in \mathbb{R}^k$. Then using the relationships from the mixGK process, we obtain the equivalent problems,

$$\begin{aligned} &\min_{\mathbf{y}} \frac{1}{2} \|\gamma \mathbf{A} \mathbf{Q}_1 \mathbf{V}_k \mathbf{y} + (1 - \gamma) \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k \mathbf{y} - \mathbf{b}\|_{\mathbf{R}^{-1}}^2 \\ &\quad + \frac{\lambda^2}{2} \mathbf{y}^\top \mathbf{V}_k^\top (\gamma \mathbf{Q}_1 + (1 - \gamma) \mathbf{Q}_2) \mathbf{V}_k \mathbf{y} \end{aligned} \quad (20)$$

$$\begin{aligned} &\min_{\mathbf{y}} \frac{1}{2} \left\| \gamma \tilde{\mathbf{U}}_{k+1} \mathbf{B}_k \mathbf{y} + (1 - \gamma) \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k \mathbf{y} - \mathbf{L}_R \mathbf{b} \right\|_2^2 \\ &\quad + \frac{\lambda^2 \gamma}{2} \mathbf{y}^\top \mathbf{y} + \frac{\lambda^2 (1 - \gamma)}{2} \mathbf{y}^\top \mathbf{V}_k^\top \mathbf{Q}_2 \mathbf{V}_k \mathbf{y} \end{aligned} \quad (21)$$

$$\begin{aligned} &\min_{\mathbf{y}} \frac{1}{2} \left\| \begin{bmatrix} \tilde{\mathbf{U}}_{k+1} & \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k \end{bmatrix} \begin{bmatrix} \gamma \mathbf{B}_k \\ (1 - \gamma) \mathbf{I}_k \end{bmatrix} \mathbf{y} - \mathbf{L}_R \mathbf{b} \right\|_2^2 \\ &\quad + \frac{\lambda^2 \gamma}{2} \|\mathbf{y}\|_2^2 + \frac{\lambda^2 (1 - \gamma)}{2} \mathbf{y}^\top \mathbf{V}_k^\top \mathbf{Q}_2 \mathbf{V}_k \mathbf{y}. \end{aligned} \quad (22)$$

Using equation (17) and the fact that

$$\begin{bmatrix} \tilde{\mathbf{U}}_{k+1} & \mathbf{Y}_k \end{bmatrix} \begin{bmatrix} \beta_1 \mathbf{e}_1 \\ \mathbf{0} \end{bmatrix} = \tilde{\mathbf{U}}_{k+1}(\beta_1 \mathbf{e}_1) = \mathbf{L}_R \mathbf{b} \quad (23)$$

where $[\tilde{\mathbf{U}}_{k+1} \quad \mathbf{Y}_k]$ contains orthonormal columns (so it can be taken out of the norm), the projected, regularized problem becomes

$$\begin{aligned} \min_{\mathbf{y}} \frac{1}{2} & \left\| \begin{bmatrix} \mathbf{I}_{k+1} & \tilde{\mathbf{U}}_{k+1}^\top \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k \\ \mathbf{0} & \mathbf{R}_k \end{bmatrix} \begin{bmatrix} \gamma \mathbf{B}_k \\ (1-\gamma) \mathbf{I}_k \end{bmatrix} \mathbf{y} - \begin{bmatrix} \beta_1 \mathbf{e}_1 \\ \mathbf{0} \end{bmatrix} \right\|_2^2 \\ & + \frac{\lambda^2 \gamma}{2} \|\mathbf{y}\|_2^2 + \frac{\lambda^2 (1-\gamma)}{2} \mathbf{y}^\top \mathbf{V}_k^\top \mathbf{Q}_2 \mathbf{V}_k \mathbf{y}. \end{aligned} \quad (24)$$

Note that the solution subspace for \mathbf{x} does not depend on γ and λ , but the solution of the projection problem depends on both γ and λ . Let $\mathbf{y}_k(\lambda, \gamma)$ denote the solution to (24), then the k iterate of the mixGK method is given as

$$\mathbf{s}_k(\lambda, \gamma) = \boldsymbol{\mu} + (\gamma \mathbf{Q}_1 + (1-\gamma) \mathbf{Q}_2) \mathbf{V}_k \mathbf{y}_k(\lambda, \gamma). \quad (25)$$

In section 3.2 we describe some techniques for selecting λ and γ at each iteration, but first we provide a theoretical result. We show that for fixed regularization parameter λ and fixed mixing parameter γ , the proposed mixGK method converges in exact arithmetic to the desired regularized solution.

Theorem 3.1. *Assume $\lambda > 0$ and $0 < \gamma \leq 1$. Let $\mathbf{y}_k(\lambda, \gamma)$ be the exact solution to projected problem (24). Then the k th iterate of the mixGK approach, written as*

$$\mathbf{s}_k = \boldsymbol{\mu} + \mathbf{Q} \mathbf{V}_k \mathbf{y}_k(\lambda, \gamma) \quad (26)$$

converges to the MAP estimate given by

$$\mathbf{s}_{\text{MAP}} = \boldsymbol{\mu} + \mathbf{Q} (\mathbf{A}^\top \mathbf{R}^{-1} \mathbf{A} \mathbf{Q} + \lambda^2 \mathbf{I}_n)^{-1} \mathbf{A}^\top \mathbf{R}^{-1} \mathbf{b}. \quad (27)$$

Proof. The proof is provided in appendix A. \square

3.2. Regularization parameter selection methods

In this section, we describe two extensions of existing regularization parameter selection methods that can be used for selecting γ and λ at each iteration of the mixGK hybrid method. For all theoretical results, we assume no breakdown of the algorithms. Notice that the solution at the k th iteration can be written as

$$\mathbf{s}_k(\lambda, \gamma) = \boldsymbol{\mu} + (\gamma \mathbf{Q}_1 + (1-\gamma) \mathbf{Q}_2) \mathbf{V}_k \mathbf{y}_k(\lambda, \gamma), \quad (28)$$

where

$$\begin{aligned} \mathbf{y}_k(\lambda, \gamma) &= (\mathbf{D}_k(\gamma)^\top \mathbf{D}_k(\gamma) + \lambda^2 \gamma \mathbf{I}_k + \lambda^2 (1-\gamma) \mathbf{V}_k^\top \mathbf{Q}_2 \mathbf{V}_k)^{-1} \mathbf{D}_k(\gamma)^\top \begin{bmatrix} \beta_1 \mathbf{e}_1 \\ \mathbf{0} \end{bmatrix} \\ &= \mathbf{C}_k(\gamma, \lambda) \begin{bmatrix} \beta_1 \mathbf{e}_1 \\ \mathbf{0} \end{bmatrix} \end{aligned} \quad (29)$$

with

$$\begin{aligned}\mathbf{D}_k(\gamma) &= \begin{bmatrix} \mathbf{I}_{k+1} & \tilde{\mathbf{U}}_{k+1}^\top \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k \\ \mathbf{0} & \mathbf{R}_k \end{bmatrix} \begin{bmatrix} \gamma \mathbf{B}_k \\ (1-\gamma) \mathbf{I}_k \end{bmatrix} \\ &= \begin{bmatrix} \gamma \mathbf{B}_k + (1-\gamma) \tilde{\mathbf{U}}_{k+1}^\top \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k \\ (1-\gamma) \mathbf{R}_k \end{bmatrix}\end{aligned}\quad (30)$$

$$\mathbf{C}_k(\gamma, \lambda) = (\mathbf{D}_k(\gamma)^\top \mathbf{D}_k(\gamma) + \lambda^2 \gamma \mathbf{I}_k + \lambda^2 (1-\gamma) \mathbf{V}_k^\top \mathbf{Q}_2 \mathbf{V}_k)^{-1} \mathbf{D}_k(\gamma)^\top. \quad (31)$$

As with regularization parameter selection methods for standard hybrid methods, there is not one method that will work for all problems, so it is advised to try various approaches in practice.

In order to provide a comparison, we provide ‘optimal’ parameters which are computed as

$$(\gamma_{\text{opt}}, \lambda_{\text{opt}}) = \arg \min_{0 < \gamma \leq 1, \lambda} \|\mathbf{s}_k(\gamma, \lambda) - \mathbf{s}_{\text{true}}\|_2^2, \quad (32)$$

where \mathbf{s}_{true} is the true solution (that is not available in practice).

Unbiased predictive risk estimation (UPRE). We can select parameters γ, λ such that

$$\begin{aligned}(\gamma_{\text{u}}^{\text{proj}}, \lambda_{\text{u}}^{\text{proj}}) &= \arg \min_{0 < \gamma \leq 1, \lambda} \mathcal{U}_{\text{proj}}(\gamma, \lambda) \\ &= \frac{1}{2k+1} \|\mathbf{r}_k^{\text{proj}}(\gamma, \lambda)\|_2^2 + \frac{2\sigma^2}{2k+1} \text{tr}(\mathbf{D}_k(\gamma) \mathbf{C}_k(\gamma, \lambda)) - \sigma^2,\end{aligned}\quad (33)$$

where σ^2 is noise level, and

$$\mathbf{r}_k^{\text{proj}}(\gamma, \lambda) = \mathbf{D}_k(\gamma) \mathbf{y}_k(\gamma, \lambda) - \begin{bmatrix} \beta_1 \mathbf{e}_1 \\ \mathbf{0} \end{bmatrix} \quad (34)$$

and

$$\begin{aligned}\text{tr}(\mathbf{D}_k(\gamma) \mathbf{C}_k(\gamma, \lambda)) &= \text{tr}(\mathbf{C}_k(\gamma, \lambda) \mathbf{D}_k(\gamma)) \\ &= \text{tr} \left(((\mathbf{D}_k(\gamma))^\top \mathbf{D}_k(\gamma) + \lambda^2 \gamma \mathbf{I}_k \right. \\ &\quad \left. + \lambda^2 (1-\gamma) \mathbf{V}_k^\top \mathbf{Q}_2 \mathbf{V}_k)^{-1} (\mathbf{D}_k(\gamma))^\top \mathbf{D}_k(\gamma) \right).\end{aligned}\quad (35)$$

When the noise level σ^2 is not provided, a noise level estimation algorithm (e.g., based on a wavelet decomposition of the observation) can be utilized [23].

Generalized cross validation (GCV). Without *a priori* knowledge of the noise level, another option is to use an extension of the GCV method [27, 29]. The basic idea is to select parameters,

$$(\gamma_{\text{g}}^{\text{proj}}, \lambda_{\text{g}}^{\text{proj}}) = \arg \min_{0 < \gamma \leq 1, \lambda} \mathcal{G}_{\text{proj}}(\gamma, \lambda) = \frac{\|\mathbf{r}_k^{\text{proj}}(\gamma, \lambda)\|_2^2}{(\text{tr}(\mathbf{I}_{2k+1} - \mathbf{D}_k(\gamma) \mathbf{C}_k(\gamma, \lambda)))^2}, \quad (36)$$

where $\mathbf{r}_k^{\text{proj}}(\gamma, \lambda)$, $\mathbf{D}_k(\gamma)$, and $\mathbf{C}_k(\gamma, \lambda)$ are same as (33).

Notice that $\mathbf{r}_k^{\text{proj}}$ and $\text{tr}(\mathbf{D}_k(\gamma) \mathbf{C}_k(\gamma, \lambda))$ are functions of k in both the GCV and UPRE functions. In order to prove convergence of the parameters chosen by UPRE and GCV, we begin with a lemma that shows convergence of the projected residual $\mathbf{r}_k^{\text{proj}}$ and trace term

$\text{tr}(\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda))$ to their full counterparts. The derivations of (33) and (36) are provided in appendix C.

Lemma 3.2. *With (34) and (35), if $k = n$, then*

$$\begin{aligned} \mathbf{r}_k^{\text{proj}} &= \mathbf{r}^{\text{full}}(\gamma, \lambda) \\ \text{tr}(\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda)) &= \text{tr}(A(\gamma, \lambda)), \end{aligned} \quad (37)$$

where

$$\begin{aligned} \mathbf{r}^{\text{full}}(\gamma, \lambda) &= \mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{x}(\gamma, \lambda) - \mathbf{L}_R \mathbf{b} \\ A(\gamma, \lambda) &= \mathbf{L}_R \mathbf{A} \mathbf{Q} (\mathbf{Q}^\top \mathbf{A}^\top \mathbf{R}^{-1} \mathbf{A} \mathbf{Q} + \lambda^2 \mathbf{Q})^{-1} \mathbf{Q}^\top \mathbf{A}^\top \mathbf{L}_R^\top. \end{aligned} \quad (38)$$

and $\mathbf{r}^{\text{full}}(\gamma, \lambda) = \mathbf{r}_n^{\text{proj}}(\gamma, \lambda)$.

Proof. The proof is provided in appendix B. \square

Next we provide convergence results for the UPRE and GCV selected parameters that are similar to results provided in [38] but are extended to the mixed hybrid methods. In particular, we show in theorem 3.3 that the UPRE parameters for the projected problem converge to the UPRE parameters for the full problem. Then, we show that with an additional weighting parameter, the same result holds for GCV parameters.

Theorem 3.3. *The UPRE parameters for the full problem (12) are given by*

$$(\gamma_u^{\text{full}}, \lambda_u^{\text{full}}) = \arg \min_{0 < \gamma \leq 1, \lambda} \mathcal{U}_{\text{full}}(\gamma, \lambda) = \frac{1}{m} \|\mathbf{r}^{\text{full}}(\gamma, \lambda)\|_2^2 + \frac{2\sigma^2}{m} \text{tr}(A(\gamma, \lambda)) - \sigma^2. \quad (39)$$

For $k = n$,

$$(\gamma_u^{\text{proj}}, \lambda_u^{\text{proj}}) = (\gamma_u^{\text{full}}, \lambda_u^{\text{full}}). \quad (40)$$

Proof. Since when $k = n$, $\|\mathbf{r}_k^{\text{proj}}\|_2^2 = \|\mathbf{r}^{\text{full}}\|_2^2$ and $\text{tr}(\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda)) = \text{tr}(A(\gamma, \lambda))$ as shown in lemma 3.2,

$$\arg \min_{0 < \gamma \leq 1, \lambda} \mathcal{U}_{\text{proj}}(\gamma, \lambda) = \arg \min_{0 < \gamma \leq 1, \lambda} \mathcal{U}_{\text{full}}(\gamma, \lambda)$$

when $k = n$ for the same noise level σ^2 . \square

For the full problem, the GCV parameters are given by

$$(\gamma_g^{\text{full}}, \lambda_g^{\text{full}}) = \arg \min_{0 < \gamma \leq 1, \lambda} \mathcal{G}_{\text{full}}(\gamma, \lambda) = \frac{\|\mathbf{r}^{\text{full}}(\gamma, \lambda)\|_2^2}{(\text{tr}(\mathbf{I}_m - A(\gamma, \lambda)))^2}. \quad (41)$$

In contrast with UPRE, $(\gamma_g^{\text{proj}}, \lambda_g^{\text{proj}})$ does not minimize (41) when $k = n$ because the trace of $\mathbf{I}_{2k+1} - \mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda)$ is not equal to the trace of $\mathbf{I}_m - A(\gamma, \lambda)$. To compensate for this, we define a weighted GCV (WGCV) method by including an additional parameter ω in (36), and computing the WGCV parameters for the projected problem as,

$$(\lambda_\omega^{\text{proj}}, \gamma_\omega^{\text{proj}}) = \arg \min_{0 < \gamma \leq 1, \lambda} \mathcal{W}_{\text{proj}}(\gamma, \lambda) = \frac{\|\mathbf{r}_k^{\text{proj}}(\gamma, \lambda)\|_2^2}{(\text{tr}(\mathbf{I}_{2k+1} - \omega \mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda)))^2}, \quad (42)$$

where $\omega = \frac{2k+1}{m}$. Since

$$(\text{tr}(\mathbf{I}_{2k+1} - \omega \mathbf{D}_k(\gamma) \mathbf{C}_k(\gamma, \lambda)))^2 = \frac{2k+1}{m} (\text{tr}(\mathbf{I}_m - \mathbf{D}_k(\gamma) \mathbf{C}_k(\gamma, \lambda)))^2, \quad (43)$$

$\mathcal{G}_{\text{full}}(\gamma, \lambda)$ is minimized by $(\lambda_{\omega}^{\text{proj}}, \gamma_{\omega}^{\text{proj}})$ when $k = n$. Similar modified GCV functions were considered in [15, 38].

Remark. It is worth mentioning that although optimization problem (12) resembles the widely-studied multi-parameter Tikhonov problem, there are some important distinctions. For our proposed mixHyBR method, we have $\mathbf{Q} = \gamma \mathbf{Q}_1 + (1 - \gamma) \mathbf{Q}_2$. Under the additional assumption that \mathbf{Q}_2 is positive definite, (12) can be written as

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{AQx} - \mathbf{b}\|_{\mathbf{R}^{-1}}^2 + \frac{\lambda^2 \gamma}{2} \|\mathbf{x}\|_{\mathbf{Q}_1}^2 + \frac{\lambda^2 (1 - \gamma)}{2} \|\mathbf{x}\|_{\mathbf{Q}_2}^2 \quad (44)$$

$$= \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{AQx} - \mathbf{b}\|_{\mathbf{R}^{-1}}^2 + \frac{\tilde{\lambda}_1^2}{2} \|\mathbf{L}_1 \mathbf{x}\|^2 + \frac{\tilde{\lambda}_2^2}{2} \|\mathbf{L}_2 \mathbf{x}\|^2, \quad (45)$$

where $\mathbf{Q}_1 = \mathbf{L}_1^\top \mathbf{L}_1$, $\mathbf{Q}_2 = \mathbf{L}_2^\top \mathbf{L}_2$, $\tilde{\lambda}_1 = \lambda \sqrt{\gamma}$ and $\tilde{\lambda}_2 = \lambda \sqrt{1 - \gamma}$. We can see that although the regularization terms are similar in the multi-parameter Tikhonov problem (7) and in problem (45), the problems are very different. In multi-parameter Tikhonov, regularization parameters λ_1 and λ_2 only control the corresponding regularization terms. For mixHyBR, from (44), we can see that the weighting parameter γ affects both the regularization term and the data fit term, which is also evident in the projected problem (24). Furthermore, since mixHyBR does not require \mathbf{Q}_2 to be positive definite, more general covariance matrices such as data-driven matrices can be included.

4. Numerical results

In this section, we provide various numerical results from tomography to investigate our proposed hybrid method based on the mixGK process, which we denote as ‘mixHyBR’. First, in section 4.1 we investigate data-driven mixed Gaussian priors where we assume that training data are available, and we compare various hybrid methods to existing shrinkage algorithms. Then, we consider a seismic crosswell tomography reconstruction problem in section 4.2, where we show that using a combination of covariance kernels can result in improved reconstructions.

For selecting the UPRE, GCV, and WGCV regularization parameters for the mixGK approach, we solve nonlinear constrained optimization problems (33), (36) and (42) respectively. We use an interior-point method as implemented in MATLAB’s `fmincon.m` function with an initial guess of $\gamma_0 = 0.5$ and $\lambda_0 = 0$ at the first mixGK iteration. Then, for subsequent iterations, we used the computed values of γ and λ from the previous iteration as an initial guess. In general, the regularization parameter selection functions may have many minimizers, but we observed that by using an adaptive strategy, we are able to guide the optimization methods toward appropriate parameters. For the stopping criteria for mixHyBR, we use a combination of approaches described in [15–17], where the iterative process is terminated if either of the following three criteria is satisfied: (i) a maximum number of iterations is reached, (ii) depending on the chosen regularization parameter selection method, the function (33) for

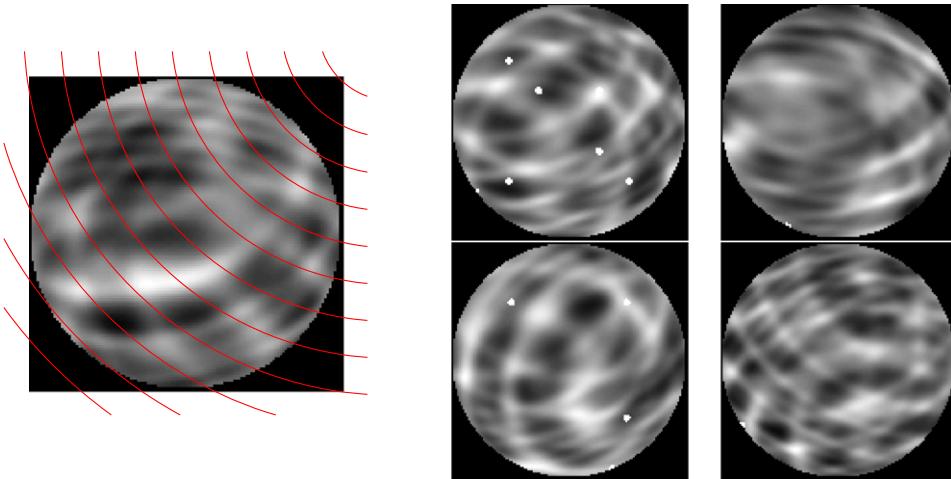


Figure 1. Spherical tomography example. On the left, the true image is provided, along with a few of the integration curves whose centers are located at 45° . Four sample images from the training dataset are provided on the right.

UPRE, (36) for GCV, or (42) for WGCV attains a minimum or flattens out, and (iii) tolerances on residuals are achieved.

4.1. Spherical tomography example

For our first example, we use a spherical means tomography reconstruction problem. Such models are often used in imaging problems from photoacoustic or optoacoustic imaging. The true image \mathbf{s}_{true} consists of 128×128 pixels and is provided in the left panel of figure 1. The forward model matrix \mathbf{A} was constructed using the IRTools toolbox [25, 30] and represents a ray-tracing operation along semi-circle curves where the angle of centers range from 0° to 90° at steps of $(90/64)^\circ$. The number of circles at each angle is 90. Thus the dimension of \mathbf{A} is $5,760 \times 16,384$ and the sinogram is 90×64 . The simulated observed sinogram was obtained as in (1), where we have included 3% additive Gaussian white noise, i.e., $\frac{\|\mathbf{e}\|}{\|\mathbf{A}\mathbf{s}_{\text{true}}\|} = 0.03$. Other conditions are chosen as the default settings provided by the toolbox; see [25] for details.

Next, we assume that we have a dataset of training images for this problem consisting of 49 images; four of the training images are provided in the right panel of figure 1. All of the images contain a circular mask to denote the region of interest or region of visibility. The inner regions of the images are generated using a linear combination of sine-squared functions, where the coefficients are random numbers uniformly distributed between 0.5 and 1, and the random numbers in sine-squared functions are uniformly distributed between 0 and 128. Furthermore, each image is contaminated by at most 8 ‘freckles’ generated as white disks, where 5 of them have radius 3 and the rest have radius 4. The freckles are randomly placed, where the origins of the freckles are uniformly distributed. The true image was generated in a similar manner, but notice that there are no freckles in the true image.

Given the training dataset $\{\mathbf{s}^{(1)}, \dots, \mathbf{s}^{(49)}\}$, we compute the (vectorized) mean image $\bar{\mathbf{s}}$ and the sample covariance matrix $\hat{\mathbf{Q}}$ is defined as in (3). Next, assuming that the prior covariance matrix represents a Matérn kernel, we solve optimization problem (11) to obtain ‘learned’ Matérn parameters $\check{\nu}$ and $\check{\ell}$ and consider the covariance matrix $\mathbf{Q}_{\text{learn}} = \mathbf{Q}(\check{\nu}, \check{\ell})$. To obtain $(\check{\nu}, \check{\ell})$, we used $\nu_0 = 1$ and $\ell_0 = 1$ as initial guess.

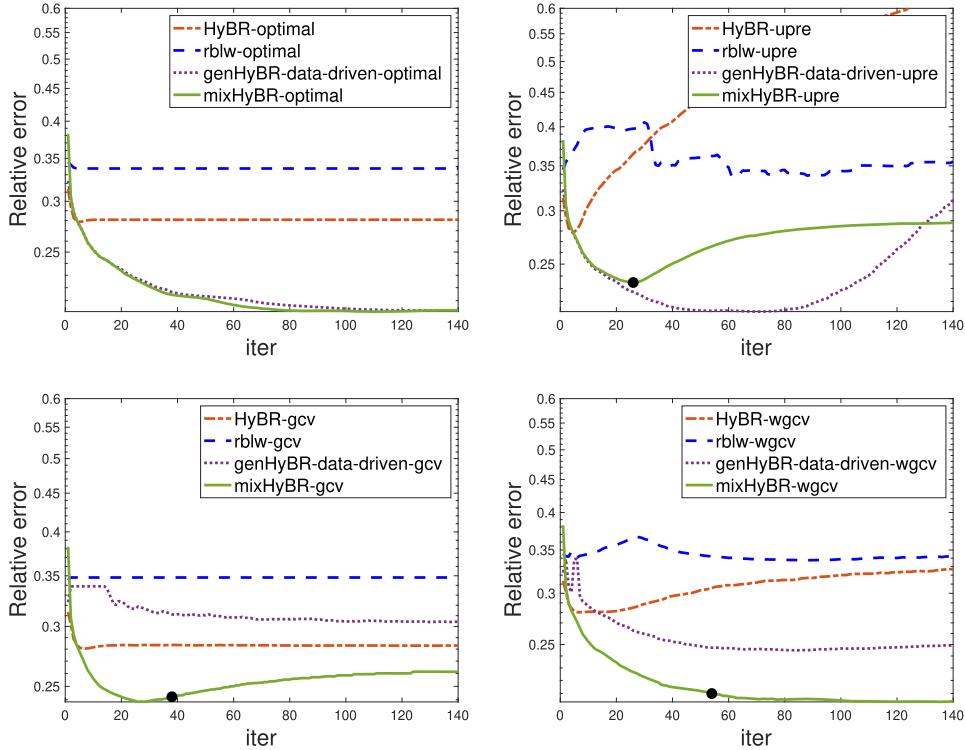


Figure 2. Comparison of relative reconstruction error norms for various iterative hybrid approaches for spherical tomography reconstruction. The top left plot corresponds to using the optimal regularization parameters. Other plots correspond to different methods to choose the regularization parameters, including UPRE, GCV, and WGCV.

We consider four hybrid iterative reconstruction methods, all with initial vector \bar{s} . Given the training data, we run the genHyBR algorithm with $\mathbf{Q} = \mathbf{Q}_{\text{learn}}$ which we denote as ‘genHyBR-data-driven’. We also provide results for ‘mixHyBR’ where $\mathbf{Q} = \gamma \mathbf{Q}_{\text{learn}} + (1 - \gamma) \hat{\mathbf{Q}}$ where γ and λ are selected during the iterative process. For comparison, we provide results for genHyBR with $\mathbf{Q} = \gamma \mathbf{I} + (1 - \gamma) \hat{\mathbf{Q}}$ where γ was pre-selected using the Rao–Blackwell Ledoit and Wolf estimator (rblw) [12, 13, 32]. We also provide results for HyBR where $\mathbf{Q} = \mathbf{I}$, but remark that this approach only uses the training data for the initial (sample mean) vector. Note that for all considered methods, the regularization parameter λ must be selected, and we investigate various approaches to do this.

In figure 2, we provide relative reconstruction error norms computed as $\|\mathbf{s}_k - \mathbf{s}_{\text{true}}\|_2 / \|\mathbf{s}_{\text{true}}\|_2$, where \mathbf{s}_k is the reconstruction at the k th iteration. Each plot corresponds to a different method for selecting the regularization parameters. For comparison, we provide in the top left plot results corresponding to the optimal regularization parameter, although these parameters cannot be computed in practice. We observe that both genHyBR-data-driven and mixHyBR result in small error norms and that even with the optimal regularization parameter λ , the rblw approach performs poorly because of the poorly-estimated mixing parameter γ . We remark that we also compared these results to a shrinkage algorithm based on the oracle approximating shrinkage (OAS) estimator [12, 13]

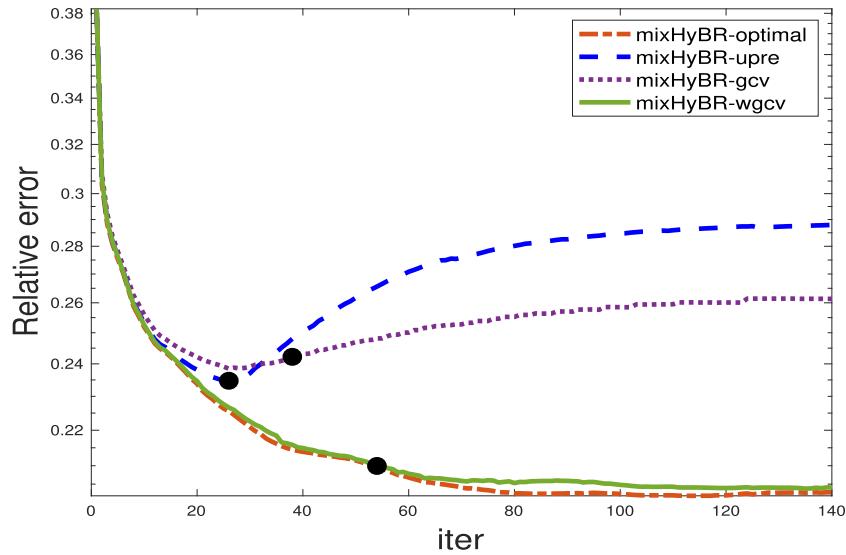


Figure 3. Relative reconstruction error norms per iteration of mixHyBR, for various regularization parameter choice methods. Black dots denote the automatically computed stopping iteration.

for obtaining γ . However, we observed very similar results as rblw, so we do not include them here.

For the automatic parameter selection methods, we observe that mixHyBR reconstructions with GCV and WGCV and genHyBR-data-driven reconstructions with UPRE have the smallest relative reconstruction error norms per iteration, compared to the other methods. Thus, we observe that including a data-driven covariance matrix, if done properly, can be beneficial. The black dots denote the (automatically-selected) stopping iteration for mixHyBR. Although one may wish to tweak the stopping criteria, all of the examples with mixHyBR resulted in a good reconstruction with the described stopping criteria. For a better comparison of the different parameter selection methods, we provide all relative reconstruction errors for mixHyBR in figure 3, where it is evident that relative errors for WGCV are very close to those for the optimal regularization parameter for this example.

Absolute error images, computed as $|\mathbf{s}_k - \mathbf{s}_{\text{true}}|$, reshaped as an image, and displayed in inverted colormap, are provided in figure 4. For better comparison, all error images have been put on the same scale, and dark regions corresponds to larger absolute errors. Relative reconstruction error norms are provided in the titles. In the top row, we compare reconstructions at iteration 140 using the optimal regularization parameter. Absolute error images in the bottom row correspond to mixHyBR reconstructions with automatic regularization parameter selection and correspond to the iteration determined by the stopping criteria. We notice that even with the optimal regularization parameter, the HyBR-optimal reconstruction suffers from the lack of sufficient prior information, and the rblw-optimal reconstruction contains large errors due to the poor choice of γ and disruptions due to freckles in the training data. Notice that the rblw-optimal reconstruction has a slight streak along the diagonal of the image, which is a result of the covariance matrix depending heavily on the training data. The mixHyBR and genHyBR-data-driven reconstructions have overall smaller absolute errors in the image. Moreover, we see that by including a combination of covariance matrices and allowing the observed

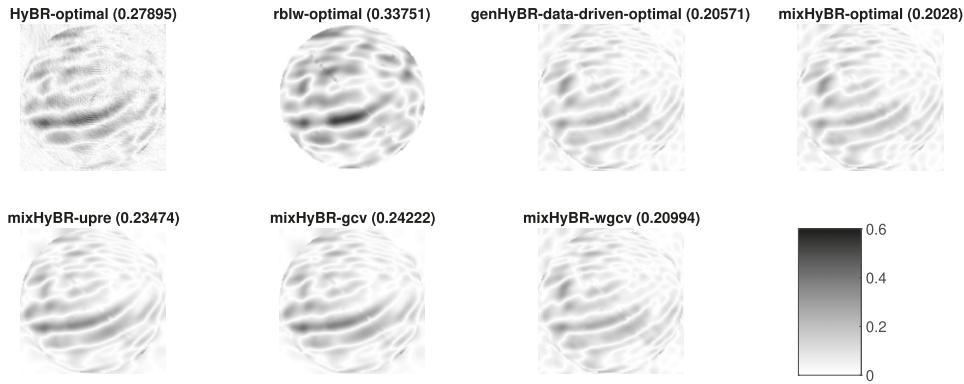


Figure 4. Absolute error images (in inverted colormap), with relative reconstruction error norms provided in the titles. The top row compares reconstructions using optimal regularization parameters, and the bottom row compares mixHyBR reconstructions with different parameter choice methods.

data to inform the choice of γ , mixHyBR methods can reduce the deleterious effects from corruptions in the training data. For this example, all parameter selection methods combined with the stopping iteration performed reasonably well.

4.2. Seismic tomography example

In this experiment, we consider a linear inversion problem from crosswell tomography [1]. Crosswell tomography is used to image the seismic wave speed in some region of interest, given data collected from multiple source–receiver pairs. The sources send out a seismic wave, and the receivers measure the travel time taken by the seismic wave to hit the receiver. The goal of the inverse problem is to image the slowness (reciprocal wave velocity) of the medium in the domain. We consider an example from continuous active source seismic monitoring (CASSM) [20], where the goal is to monitor the spatial development of a small scale injection of CO_2 into a high quality reservoir. We consider reconstruction at a single time point and investigate the impact of including mixed Gaussian priors on the reconstruction.

The inverse problem can be represented as (1) where the goal is to reconstruct the slowness $\mathbf{s} \in \mathbb{R}^{n \times 1}$ of the medium from the measured travel times $\mathbf{d} \in \mathbb{R}^{m \times 1}$ which are assumed to be corrupted by Gaussian white noise $\epsilon \in \mathbb{R}^{m \times 1}$. In our problem setup, the true slowness field was discretized into $n = 188,356$ cells, where the slowness within each cell is assumed to be constant. The true image (normalized between 0 and 1) is of size 434×434 and was obtained from [19]. For the observations, there were $m_s = 20$ sources and $m_r = 50$ receivers, so a total of $m = m_r m_s$ measurements. Each row of the forward model matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ corresponds to a source–receiver pair. Since the wave travels along a straight line from source to receiver, only the cells lying on the straight line contribute to the non-zero entries. Hence, \mathbf{A} is very sparse with $\mathcal{O}(\sqrt{mn})$ non-zero entries. The true image along with a schematic of the source–detector pairs are given in the left panel of figure 5. The observations, which contain 1% noise, are provided in the right panel of figure 5.

Next we investigate the impact of different choices of \mathbf{Q} on the reconstruction. First, we consider the genHyBR method with three different prior covariance matrices $\mathbf{Q}_1, \mathbf{Q}_2$, and \mathbf{Q}_3 defined by a Matérn kernel with $\nu = 1$ and $\ell = 0.2$, a rational quadratic with $\nu = 2$ and

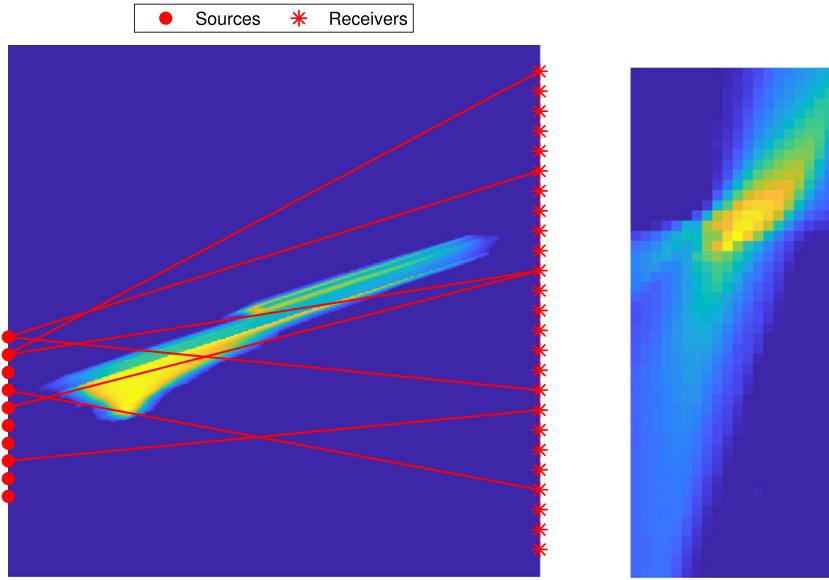


Figure 5. CASSM example. In the left panel, we provide the true slowness field image, along with some of the locations of the sources and the detectors. Seven of the source–receiver pairs are highlighted in the figure. In the right panel, we provide the observations corresponding to 20 sources and 50 receivers.

$\ell = 0.1$, and the identity matrix \mathbf{I} , respectively. These approaches are denoted by ‘genHyBR1’, ‘genHyBR2’, and ‘HyBR’ respectively. Then we consider two mixHyBR approaches that include mixed Gaussian priors, where $\text{mixHyBR}(\mathbf{Q}_1, \mathbf{Q}_2)$ uses covariance matrix $\mathbf{Q} = \gamma\mathbf{Q}_1 + (1 - \gamma)\mathbf{Q}_2$ and $\text{mixHyBR}(\mathbf{Q}_1, \mathbf{I})$ uses covariance matrix $\mathbf{Q} = \gamma\mathbf{Q}_1 + (1 - \gamma)\mathbf{I}$, where the mixing parameter γ is selected during the reconstruction process. For the optimally selected regularization parameters, we provide in figure 6 the relative reconstruction error norms per iteration.

We observe that if a good covariance matrix (in this case, \mathbf{Q}_1) is known in advance, stand-alone genHyBR can perform well and result in small relative reconstruction errors. Otherwise, the relative reconstruction errors may remain large, and multiple solves with different covariance matrices would be needed to determine a good prior. In this case, the mixHyBR approach can prove beneficial. The mixHyBR approaches produce reconstructions with overall smaller relative reconstruction errors than genHyBR with each covariance matrix alone. Image reconstructions, including a zoomed subregion, are provided in figure 7. Notice that the mixed Gaussian priors are better able to resolve some details of the true image. While the reconstructions of genHyBR1 and genHyBR2 are too smooth, $\text{mixHyBR}(\mathbf{Q}_1, \mathbf{Q}_2)$ and $\text{mixHyBR}(\mathbf{Q}_1, \mathbf{I})$ reconstructions reveal multiple layered regions that are present in the true field. Thus, incorporating mixed Gaussian priors can lead to improved reconstructions.

Next we investigate the performance of different regularization parameter selection methods within the mixHyBR methods. Reconstructions corresponding to GCV, WGV, and UPRE parameter choices are provided in figure 8, with stopping iterations and relative reconstruction errors provided in the titles. For the stopping criteria, we used a tolerance of 10^{-6} for the residual norm. Although the relative reconstruction error norms are larger than

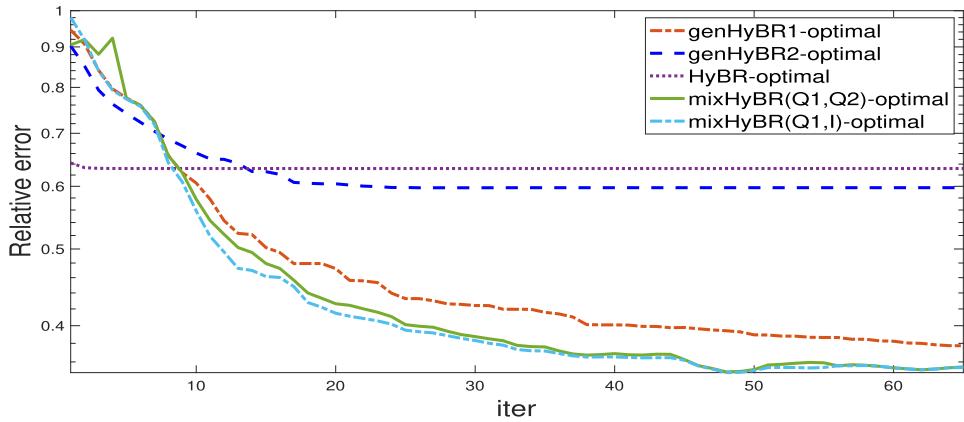


Figure 6. Comparison of relative reconstruction error norms for genHyBR and mixHyBR with optimal parameters γ and ℓ .

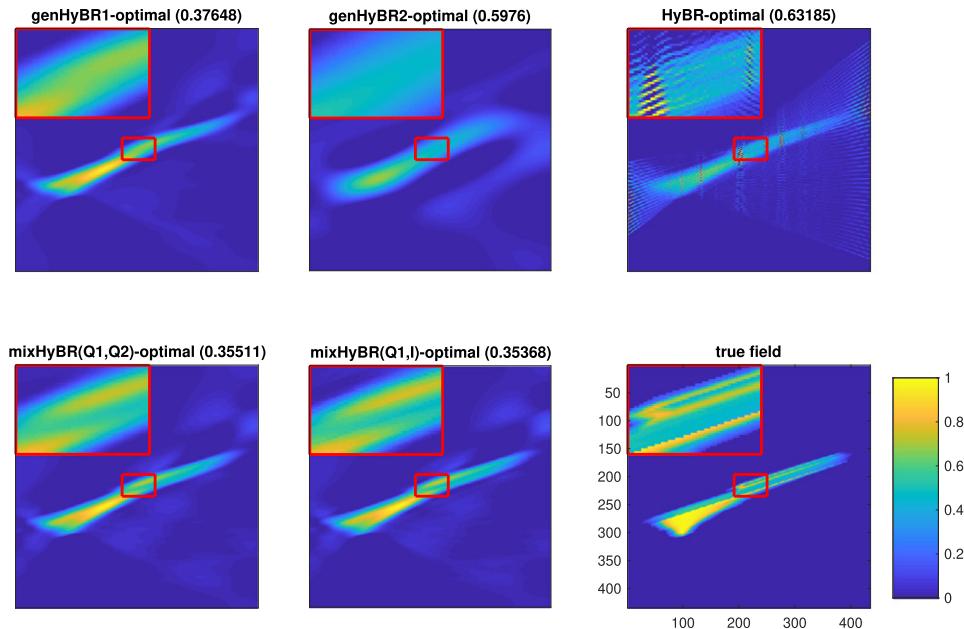


Figure 7. Reconstructions with zoomed subimages for CASSM example. All of the reconstructions use the optimal regularization parameter and relative reconstruction errors are provided in the titles.

those for the optimal regularization parameter, which is likely due to the increased errors in the surrounding flat regions, the reconstructed images can all distinguish the two yellow layers.

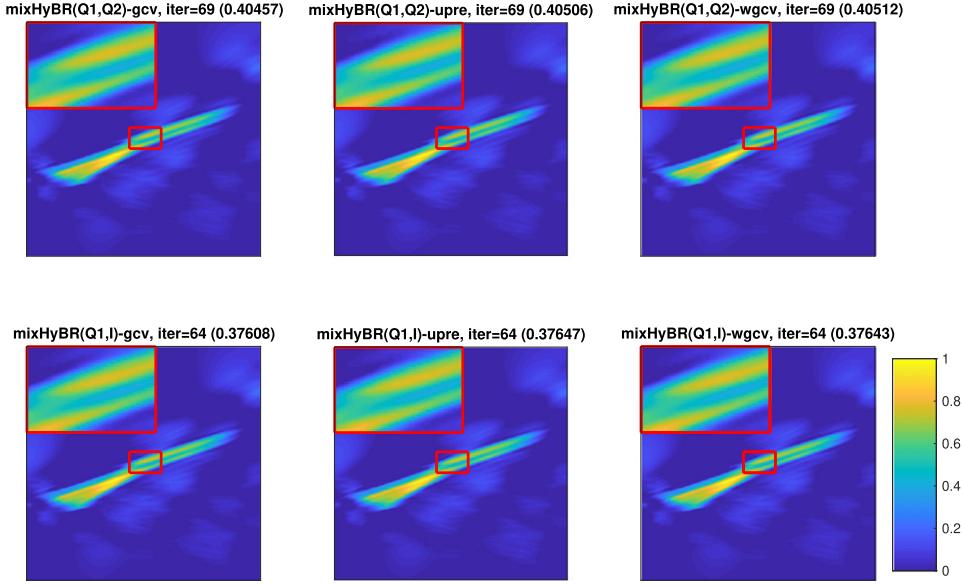


Figure 8. Reconstructions of mixHyBR($\mathbf{Q}_1, \mathbf{Q}_2$) (top row) and mixHyBR(\mathbf{Q}_1, \mathbf{I}) (bottom row) for different parameter choice methods. The automatically detected stopping iteration (iter) and corresponding relative reconstruction error norm are provided in the titles.

5. Conclusions

This paper describes a hybrid iterative projection method, dubbed mixHyBR, that is based on an extension of the generalized Golub–Kahan bidiagonalization and that can be used for solving inverse problems (i.e., computing MAP estimates) with mixed Gaussian priors. The main advantage of this approach is that the mixing or blending parameter does not need to be known *a priori*, but rather can be estimated during the iterative process along with the regularization parameter. Various methods for selecting these parameters were considered and evaluated. Furthermore, mixHyBR methods can easily incorporate data-driven priors where training data are used to define the prior covariance matrix itself (e.g., sample based priors) or to learn parameters for the covariance kernel function. Comparisons to widely-used shrinkage algorithms reveal that the mixed hybrid approaches are more robust under the presence of noise or artifacts in the data and enable greater flexibility when selecting suitable priors. Numerical results from both spherical and seismic tomography show the potential of these methods.

Acknowledgments

This work was partially supported by NSF DMS 1723005 and NSF DMS 1654175. J Chung would also like to acknowledge support from the Alexander von Humboldt Foundation.

Appendix A. Proof of theorem 3.1

Proof. Based on (29) and (30),

$$\mathbf{y}_k(\lambda, \gamma) = \mathbf{C}_k(\gamma, \lambda) \begin{bmatrix} \beta_1 \mathbf{e}_1 \\ \mathbf{0} \end{bmatrix} \quad (46)$$

With $k = n$, by (13)–(15), and (47),

$$\begin{aligned}
\mathbf{s}_n &= \boldsymbol{\mu} + \mathbf{Q}\mathbf{V}_n\mathbf{y}_n \\
&= \boldsymbol{\mu} + \mathbf{Q}\mathbf{V}_n\mathbf{C}_n(\gamma, \lambda) \begin{bmatrix} \beta_1 \mathbf{e}_1 \\ \mathbf{0} \end{bmatrix} \\
&= \boldsymbol{\mu} + \mathbf{Q}\mathbf{V}_n(\mathbf{V}_n^\top \mathbf{Q}^\top \mathbf{A}^\top \mathbf{L}_R^\top \mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{V}_n + \lambda^2 \mathbf{V}_n^\top \mathbf{Q} \mathbf{V}_n)^{-1} \mathbf{V}_n^\top \mathbf{Q} \mathbf{A}^\top \mathbf{R}^{-1} \mathbf{b} \\
&= \boldsymbol{\mu} + \mathbf{Q}(\mathbf{Q}^\top \mathbf{A}^\top \mathbf{L}_R^\top \mathbf{L}_R \mathbf{A} \mathbf{Q} + \lambda^2 \mathbf{Q})^{-1} \mathbf{Q} \mathbf{A}^\top \mathbf{R}^{-1} \mathbf{b} \\
&= \boldsymbol{\mu} + \mathbf{Q}(\mathbf{A}^\top \mathbf{R}^{-1} \mathbf{A} \mathbf{Q} + \lambda^2 \mathbf{I}_n)^{-1} \mathbf{A}^\top \mathbf{R}^{-1} \mathbf{b} \\
&= \mathbf{s}_{\text{MAP}}.
\end{aligned}$$

Therefore, the solution for (33) converges to the solution for (39) and the solution for (36) converges to the solution for (41) as k increases. \square

Appendix B. Proof of lemma 3.2

Proof. For the projected residual for \mathbf{x}_k ,

$$\begin{aligned}
\|\mathbf{r}_k^{\text{proj}}(\gamma, \lambda)\|_2^2 &= \left\| \mathbf{D}(\gamma) \mathbf{y}_k(\gamma, \lambda) - \begin{bmatrix} \beta_1 \mathbf{e}_1 \\ \mathbf{0} \end{bmatrix} \right\|_2^2 \\
&= \|(\gamma \mathbf{L}_R \mathbf{A} \mathbf{Q}_1 \mathbf{V}_k + (1 - \gamma) \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k) \mathbf{y}_k(\gamma, \lambda) - \mathbf{L}_R \mathbf{b}\|_2^2 \\
&= \left\| \begin{bmatrix} \mathbf{I}_{k+1} & \widetilde{\mathbf{U}}_{k+1} \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k \\ \mathbf{0} & \mathbf{R}_k \end{bmatrix} \begin{bmatrix} \gamma \mathbf{B}_k \\ (1 - \gamma) \mathbf{I}_k \end{bmatrix} \mathbf{y}_k(\gamma, \lambda) - \begin{bmatrix} \beta_1 \mathbf{e}_1 \\ \mathbf{0} \end{bmatrix} \right\|_2^2 \\
&= \left\| [\widetilde{\mathbf{U}}_{k+1} \quad \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k] \begin{bmatrix} \gamma \mathbf{B}_k \\ (1 - \gamma) \mathbf{I}_k \end{bmatrix} \mathbf{y}_k(\gamma, \lambda) - \mathbf{L}_R \mathbf{b} \right\|_2^2 \\
&= \|(\gamma \widetilde{\mathbf{U}}_{k+1} \mathbf{B}_k + (1 - \gamma) \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k) \mathbf{y}_k(\gamma, \lambda) - \mathbf{L}_R \mathbf{b}\|_2^2 \\
&= \|(\gamma \mathbf{L}_R \mathbf{A} \mathbf{Q}_1 \mathbf{V}_k + (1 - \gamma) \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k) \mathbf{y}_k(\gamma, \lambda) - \mathbf{L}_R \mathbf{b}\|_2^2 \\
&= \|\mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{x}_k(\gamma, \lambda) - \mathbf{L}_R \mathbf{b}\|_2^2.
\end{aligned}$$

When $k = n$, we have

$$\|\mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{x}_k(\gamma, \lambda) - \mathbf{L}_R \mathbf{b}\|_2^2 \rightarrow \|\mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{x}_n(\gamma, \lambda) - \mathbf{L}_R \mathbf{b}\|_2^2.$$

Since $\|\mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{x}_n(\gamma, \lambda) - \mathbf{L}_R \mathbf{b}\|_2^2 = \|\mathbf{r}^{\text{full}}(\gamma, \lambda)\|_2^2$, when $k = n$ we have

$$\|\mathbf{r}_k^{\text{proj}}(\gamma, \lambda)\|_2^2 = \|\mathbf{r}^{\text{full}}(\gamma, \lambda)\|_2^2.$$

For k th iteration in the projected problem (29),

$$\begin{aligned}
\mathbf{D}_k(\gamma)^\top \mathbf{D}_k(\gamma) &= \begin{bmatrix} \gamma \mathbf{B}_k + (1 - \gamma) \tilde{\mathbf{U}}_{k+1}^\top \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k \\ (1 - \gamma) \mathbf{R}_k \end{bmatrix}^\top \begin{bmatrix} \gamma \mathbf{B}_k + (1 - \gamma) \tilde{\mathbf{U}}_{k+1}^\top \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k \\ (1 - \gamma) \mathbf{R}_k \end{bmatrix} \\
&= \gamma^2 \mathbf{B}_k^\top \mathbf{B}_k + 2\gamma(1 - \gamma) \mathbf{B}_k^\top \mathbf{U}_{k+1}^\top \mathbf{L}_R^\top \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k \\
&\quad + (1 - \gamma)^2 \mathbf{V}_k^\top \mathbf{Q}_2^\top \mathbf{A}^\top \mathbf{L}_R^\top \mathbf{L}_R \mathbf{U}_{k+1} \mathbf{U}_{k+1}^\top \mathbf{L}_R^\top \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k + (1 - \gamma)^2 \mathbf{R}_k^\top \mathbf{R}_k \\
&= \gamma^2 \mathbf{V}_k^\top \mathbf{Q}_1^\top \mathbf{A}^\top \mathbf{L}_R^\top \mathbf{L}_R \mathbf{A} \mathbf{Q}_1 \mathbf{V}_k + 2\gamma(1 - \gamma) \mathbf{V}_k^\top \mathbf{Q}_1^\top \mathbf{A}^\top \mathbf{L}_R^\top \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k \\
&\quad + (1 - \gamma)^2 \mathbf{V}_k^\top \mathbf{Q}_2^\top \mathbf{A}^\top \mathbf{L}_R^\top \mathbf{L}_R \mathbf{U}_{k+1} \mathbf{U}_{k+1}^\top \mathbf{L}_R^\top \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k \\
&\quad + (1 - \gamma)^2 \mathbf{V}_k^\top \mathbf{Q}_2^\top \mathbf{A}^\top \mathbf{L}_R^\top (\mathbf{I}_{k+1} - \mathbf{L}_R \mathbf{U}_{k+1} \mathbf{U}_{k+1}^\top \mathbf{L}_R^\top) \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k \\
&= \gamma^2 \mathbf{V}_k^\top \mathbf{Q}_1^\top \mathbf{A}^\top \mathbf{L}_R^\top \mathbf{L}_R \mathbf{A} \mathbf{Q}_1 \mathbf{V}_k + 2\gamma(1 - \gamma) \mathbf{V}_k^\top \mathbf{Q}_1^\top \mathbf{A}^\top \mathbf{L}_R^\top \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k \\
&\quad + (1 - \gamma)^2 \mathbf{V}_k^\top \mathbf{Q}_2^\top \mathbf{A}^\top \mathbf{L}_R^\top \mathbf{L}_R \mathbf{A} \mathbf{Q}_2 \mathbf{V}_k \\
&= (\mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{V}_k)^\top \mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{V}_k
\end{aligned}$$

Therefore,

$$\begin{aligned}
\text{tr}(\mathbf{D}_k(\gamma) \mathbf{C}_k(\gamma, \lambda)) &= \text{tr}(\mathbf{D}_k(\gamma) (\mathbf{D}_k(\gamma)^\top \mathbf{D}_k(\gamma) + \lambda^2 \gamma \mathbf{I}_k + \lambda^2 (1 - \gamma) \mathbf{V}_k^\top \mathbf{Q}_2 \mathbf{V}_k)^{-1} \mathbf{D}_k(\gamma)^\top) \\
&= \text{tr}((\mathbf{D}_k(\gamma)^\top \mathbf{D}_k(\gamma) + \lambda^2 \gamma \mathbf{I}_k + \lambda^2 (1 - \gamma) \mathbf{V}_k^\top \mathbf{Q}_2 \mathbf{V}_k)^{-1} \mathbf{D}_k(\gamma)^\top \mathbf{D}_k(\gamma)) \\
&= \text{tr}((\mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{V}_k)^\top \mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{V}_k + \lambda^2 \mathbf{V}_k^\top \mathbf{Q} \mathbf{V}_k)^{-1} (\mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{V}_k)^\top (\mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{V}_k)) \\
&= \text{tr}((\mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{V}_k) ((\mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{V}_k)^\top \mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{V}_k + \lambda^2 \mathbf{V}_k^\top \mathbf{Q} \mathbf{V}_k)^{-1} (\mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{V}_k)^\top) \\
&\rightarrow \text{tr}((\mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{V}_n) ((\mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{V}_n)^\top \mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{V}_n + \lambda^2 \mathbf{V}_n^\top \mathbf{Q} \mathbf{V}_n)^{-1} (\mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{V}_n)^\top) \\
&= \text{tr}((\mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{V}_n) \mathbf{V}_n^{-1} ((\mathbf{L}_R \mathbf{A} \mathbf{Q})^\top \mathbf{L}_R \mathbf{A} \mathbf{Q} + \lambda^2 \mathbf{Q})^{-1} \mathbf{V}_n^{-\top} (\mathbf{L}_R \mathbf{A} \mathbf{Q} \mathbf{V}_n)^\top) \\
&= \text{tr}((\mathbf{L}_R \mathbf{A} \mathbf{Q}) ((\mathbf{L}_R \mathbf{A} \mathbf{Q})^\top \mathbf{L}_R \mathbf{A} \mathbf{Q} + \lambda^2 \mathbf{Q})^{-1} (\mathbf{L}_R \mathbf{A} \mathbf{Q})^\top) \\
&= \text{tr}(A(\gamma, \lambda))
\end{aligned} \tag{47}$$

with the invertible \mathbf{V}_n since $\mathbf{V}_n^\top \mathbf{Q}_1 \mathbf{V}_n = \mathbf{I}_n$ and $\mathbf{V}_n \in \mathbb{R}^{n \times n}$ is square matrix. \square

Appendix C. Derivation of regularization parameter selection methods for mixHyBR

The derivations for both UPRE and GCV follow derivations for the Tikhonov case, see e.g. [5]. For UPRE for mixHyBR, the goal is to select parameters λ and γ that minimize the predictive risk,

$$\mathbb{E} \|\mathbf{D}_k(\gamma) \mathbf{y}_k^{(\gamma, \lambda)} - \mathbf{D}_k(\gamma) \mathbf{y}_k(\gamma, \lambda)\|_2^2, \tag{48}$$

where $\mathbf{y}_k^{(\gamma, \lambda)}$ is the solution to (24) with fixed γ and λ , $\delta = \begin{bmatrix} \beta \mathbf{e}_1 \\ \mathbf{0} \end{bmatrix}$, $\mathbf{y}_k^{(\gamma, \lambda)} = \mathbf{C}_k(\gamma, \lambda) \delta$, and $\mathbf{D}_k(\gamma)$ and $\mathbf{C}_k(\gamma, \lambda)$ are defined in (30) and (31) respectively. For the projected problem (24), the vector

δ consists of a deterministic and a stochastic part as described in [38], so $\delta = \mathbf{D}_k(\gamma)\mathbf{y}_k(\gamma, \lambda) + \eta$ where $\eta \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ and $\eta \in \mathbb{R}^{2k+1}$. Then, (48) can be written as

$$\begin{aligned}
 & \mathbb{E}\|\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda)\delta - \mathbf{D}_k(\gamma)\mathbf{y}_k(\gamma, \lambda)\|_2^2 \\
 &= \mathbb{E}\|(\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda) - \mathbf{I}_{2k+1})\mathbf{D}_k(\gamma)\mathbf{y}_k(\gamma, \lambda) + \mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda)\eta\|_2^2 \\
 &= \|(\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda) - \mathbf{I}_{2k+1})\mathbf{D}_k(\gamma)\mathbf{y}_k(\gamma, \lambda)\|_2^2 + \mathbb{E}\eta^\top(\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda))^2\eta \\
 &= \|(\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda) - \mathbf{I}_{2k+1})\mathbf{D}_k(\gamma)\mathbf{y}_k(\gamma, \lambda)\|_2^2 + \sigma^2 \text{tr}((\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda))^2).
 \end{aligned} \tag{49}$$

Next the expectation of the projected residual can be written as

$$\begin{aligned}
 & \mathbb{E}\|\mathbf{D}_k(\gamma)\mathbf{y}_k^{(\gamma, \lambda)} - \delta\|_2^2 \\
 &= \mathbb{E}\|\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda)\delta - \delta\|_2^2 \\
 &= \mathbb{E}\|(\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda) - \mathbf{I}_{2k+1})\delta\|_2^2 \\
 &= \mathbb{E}\|(\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda) - \mathbf{I}_{2k+1})\mathbf{D}_k(\gamma)\mathbf{y}_k(\gamma, \lambda) + (\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda) - \mathbf{I}_{2k+1})\eta\|_2^2 \\
 &= \|(\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda) - \mathbf{I}_{2k+1})\mathbf{D}_k(\gamma)\mathbf{y}_k(\gamma, \lambda)\|_2^2 + \mathbb{E}(\eta^\top(\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda) - \mathbf{I}_{2k+1})^2\eta) \\
 &= \|(\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda) - \mathbf{I}_{2k+1})\mathbf{D}_k(\gamma)\mathbf{y}_k(\gamma, \lambda)\|_2^2 + \sigma^2 \text{tr}((\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda))^2) \\
 &\quad - 2\sigma^2 \text{tr}(\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda)) + \sigma^2(2k+1).
 \end{aligned} \tag{50}$$

By combining (49) and (50), we get

$$\begin{aligned}
 & \mathbb{E}\|\mathbf{D}_k(\gamma)\mathbf{y}_k^{(\gamma, \lambda)} - \mathbf{D}_k(\gamma)\mathbf{y}_k(\gamma, \lambda)\|_2^2 \\
 &= \mathbb{E}\|\mathbf{D}_k(\gamma)\mathbf{y}_k^{(\gamma, \lambda)} - \delta\|_2^2 + 2\sigma^2 \text{tr}(\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda)) - \sigma^2(2k+1) \\
 &\approx \|\mathbf{D}_k(\gamma)\mathbf{y}_k^{(\gamma, \lambda)} - \delta\|_2^2 + 2\sigma^2 \text{tr}(\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda)) - \sigma^2(2k+1).
 \end{aligned} \tag{51}$$

By dividing the function in (51) by $2k+1$, we get the following UPRE function at the k th iteration of the mixHyBR method,

$$\mathcal{U}_{\text{proj}}(\gamma, \lambda) = \frac{1}{2k+1} \|\mathbf{r}_k^{\text{proj}}(\gamma, \lambda)\|_2^2 + \frac{2\sigma^2}{2k+1} \text{tr}(\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda)) - \sigma^2, \tag{52}$$

where $\mathbf{r}_k^{\text{proj}}(\gamma, \lambda) = \mathbf{D}_k(\gamma)\mathbf{y}_k^{(\gamma, \lambda)} - \delta$.

The GCV method for selecting a regularization parameter is an approximation of a leave-one-out cross validation (LOOCV) method. However, for large-scale problems, the direct application of LOOCV and GCV for parameter selection is not feasible. Following the derivation from [4], we get

$$V(\gamma, \lambda) = \frac{1}{2k+1} \sum_{i=1}^{2k+1} \left(\frac{[\mathbf{D}_k(\gamma)\mathbf{y}_k^{(\gamma, \lambda)}]_i - [\delta]_i}{1 - [\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda)]_{ii}} \right)^2, \tag{53}$$

where $[\cdot]_i$ represents the i th component of a vector and $[\cdot]_{ij}$ represents the i th row and j th column component of a matrix. Next, since computing diagonal entries of $\mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda)$ can be expensive, the entries are replaced by the average of diagonal entries such that we get the GCV function for the mixHyBR method,

$$G(\gamma, \lambda) = \frac{(2k+1)\|\mathbf{D}_k(\gamma)\mathbf{y}_k^{(\gamma, \lambda)} - \boldsymbol{\delta}\|_2^2}{(\text{tr}(\mathbf{I}_{2k+1} - \mathbf{D}_k(\gamma)\mathbf{C}_k(\gamma, \lambda)))^2}. \quad (54)$$

Notice that this is a scalar multiple of the GCV function (with constant $2k+1$) as given in (36).

ORCID iDs

Julianne Chung  <https://orcid.org/0000-0002-6760-4736>
 Jiahua Jiang  <https://orcid.org/0000-0002-4000-8846>

References

- [1] Ambikasaran S, Li J Y, Kitanidis P K and Darve E 2013 Large-scale stochastic linear inversion using hierarchical matrices *Comput. Geosci.* **17** 913–27
- [2] Arridge S, Maass P, Öktem O and Schönlieb C-B 2019 Solving inverse problems using data-driven models *Acta Numer.* **28** 1–174
- [3] Asch M, Bocquet M and Nodet M 2016 *Data Assimilation: Methods, Algorithms, and Applications* vol 11 (Philadelphia, PA: SIAM)
- [4] Aster R C, Borchers B and Thurber C H 2019 Chapter four - Tikhonov regularization *Parameter Estimation and Inverse Problems* 3rd edn ed R C Aster, B Borchers and C H Thurber (Amsterdam: Elsevier) pp 93–134 <http://sciencedirect.com/science/article/pii/B9780128046517000092>
- [5] Bardsley J M 2018 *Computational Uncertainty Quantification for Inverse Problems* vol 19 (Philadelphia, PA: SIAM)
- [6] Bazán F S V, Borges L S and Francisco J B 2012 On a generalization of Regińska's parameter choice rule and its numerical realization in large-scale multi-parameter Tikhonov regularization *Appl. Math. Comput.* **219** 2100–13
- [7] Brown R D, Bardsley J M and Cui T 2020 Semivariogram methods for modeling Whittle–Matérn priors in Bayesian inverse problems *Inverse Problems* **36** 055006
- [8] Bubba T A, Kutyniok G, Lassas M, Maerz M, Samek W, Siltanen S and Srinivasan V 2019 Learning the invisible: a hybrid deep learning-shearlet framework for limited angle computed tomography *Inverse Problems* **35** 064002
- [9] Buccini A, Donatelli M and Reichel L 2017 Iterated Tikhonov regularization with a general penalty term *Numer. Linear Algebra Appl.* **24** e2089
- [10] Calvetti D and Somersalo E 2005 Priorconditioners for linear systems *Inverse Problems* **21** 1397
- [11] Calvetti D and Somersalo E 2007 *An Introduction to Bayesian Scientific Computing: Ten Lectures on Subjective Computing* vol 2 (Berlin: Springer)
- [12] Chen Y, Wiesel A, Eldar Y C and Hero A O 2010 Shrinkage algorithms for MMSE covariance estimation *IEEE Trans. Signal Process.* **58** 5016–29
- [13] Chen Y, Wiesel A and Hero A O 2009 Shrinkage estimation of high dimensional covariance matrices 2009 *IEEE Int. Conf. on Acoustics, Speech and Signal Processing* (IEEE) pp 2937–40
- [14] Chung J, Chung M and O'Leary D P 2011 Designing optimal spectral filters for inverse problems *SIAM J. Sci. Comput.* **33** 3132–52
- [15] Chung J, Nagy J G and O'Leary D P 2008 A weighted GCV method for Lanczos hybrid regularization *Electron. Trans. Numer. Anal.* **28** 149–67
- [16] Chung J and Palmer K 2015 A hybrid LSMR algorithm for large-scale Tikhonov regularization *SIAM J. Sci. Comput.* **37** S562–80

[17] Chung J and Saibaba A K 2017 Generalized hybrid iterative methods for large-scale Bayesian inverse problems *SIAM J. Sci. Comput.* **39** S24–46

[18] Chung J, Saibaba A K, Brown M and Westman E 2018 Efficient generalized Golub–Kahan based methods for dynamic inverse problems *Inverse Problems* **34** 024005

[19] Daley T and Ajo-Franklin J 2019 Frio 2: raw CASSM datasets <https://osti.gov/biblio/1490697-frio-raw-cassm-datasets> (accessed)

[20] Daley T M, Ajo-Franklin J B and Doughty C 2011 Constraining the reservoir model of an injected CO₂ plume with crosswell CASSM at the Frio-II brine pilot *Int. J. Greenh. Gas Control* **5** 1022–30

[21] Daniel J W, Gragg W B, Kaufman L and Stewart G W 1976 Reorthogonalization and stable algorithms for updating the Gram–Schmidt QR factorization *Math. Comput.* **30** 772–95

[22] Dong Y and Schönlieb C-B 2020 Tomographic reconstruction with spatially varying parameter selection *Inverse Problems* **36** 054002

[23] Donoho D L 1995 De-noising by soft-thresholding *IEEE Trans. Inform. Theory* **41** 613–27

[24] Galka A, Yamashita O and Ozaki T 2004 Spatiotemporal Kalman filtering: a new approach to solving dynamical inverse problems *Int. Symp. on Nonlinear Theory and its Applications*

[25] Gazzola S, Hansen P C and Nagy J G 2018 IR tools: a MATLAB package of iterative regularization methods and large-scale test problems *Numer. Algorithms* **81** 773–811

[26] Gazzola S and Novati P 2013 Multi-parameter Arnoldi–Tikhonov methods *Electronic Transactions on Numerical Analysis* **40** 452–75

[27] Golub G H, Heath M and Wahba G 1979 Generalized cross-validation as a method for choosing a good ridge parameter *Technometrics* **21** 215–23

[28] Haber E and Tenorio L 2003 Learning regularization functionals a supervised training approach *Inverse Problems* **19** 611

[29] Hansen P C 2010 *Discrete Inverse Problems: Insight and Algorithms* (Philadelphia, PA: SIAM)

[30] Hansen P C and Jorgensen J S 2018 AIR tools II: algebraic iterative reconstruction methods, improved implementation *Numer. Algorithms* **79** 107–37

[31] Hochstenbach M E and Reichel L 2010 An iterative method for Tikhonov regularization with a general linear regularization operator *J. Integr. Equ. Appl.* **22** 465–82

[32] Ledoit O and Wolf M 2004 A well-conditioned estimator for large-dimensional covariance matrices *J. Multivariate Anal.* **88** 365–411

[33] Li H, Schwab J, Antholzer S and Haltmeier M 2020 NETT: solving inverse problems with deep neural networks *Inverse Problems* **36** 065005

[34] Long C J, Purdon P L, Temereanca S, Desai N U, Hämäläinen M S and Brown E N 2011 State-space solutions to the dynamic magnetoencephalography inverse problem using high performance computing *Ann. Appl. Stat.* **5** 1207

[35] Lu S and Pereverzev S V 2011 Multi-parameter regularization and its numerical realization *Numer. Math.* **118** 1–31

[36] Lucas A, Iliadis M, Molina R and Katsaggelos A K 2018 Using deep neural networks for inverse problems in imaging: beyond analytical methods *IEEE Signal Process. Mag.* **35** 20–36

[37] Rasmussen C E 2003 *Gaussian Processes in Machine Learning (Summer School on Machine Learning)* (Berlin: Springer) pp 63–71

[38] Renaut R A, Vatankhah S and Ardestani V E 2017 Hybrid and iteratively reweighted regularization by unbiased predictive risk and weighted GCV for projected systems *SIAM J. Sci. Comput.* **39** B221–43

[39] Schäfer J and Strimmer K 2005 A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics *Stat. Appl. Genet. Mol. Biol.* **4** 32

[40] Schwab J, Antholzer S and Haltmeier M 2018 Deep null space learning for inverse problems: convergence analysis and rates *Inverse Problems* **35** 025008

[41] Shapiro A, Dentcheva D and Ruszczyński A 2009 *Lectures on Stochastic Programming: Modeling and Theory* (Philadelphia, PA: SIAM)

[42] Wang Z 2012 Multi-parameter Tikhonov regularization and model function approach to the damped Morozov principle for choosing regularization parameters *J. Comput. Appl. Math.* **236** 1815–32

[43] Yadav V, Michalak A M, Ray J and Shiga Y P 2016 A statistical approach for isolating fossil fuel emissions in atmospheric inverse problems *J. Geophys. Res. Atmos.* **121** 12–490

[44] Yang S, Chen H, Li J and Wang W 2016 A spatially-variant morphological regularization method for inverse problems in image processing *2016 Chinese Control and Decision Conf. (CCDC)* (IEEE) pp 3789–94

- [45] Yao Z S, Roberts R G and Tryggvason A 1999 Calculating resolution and covariance matrices for seismic tomography with the LSQR method *Geophys. J. Int.* **138** 886–94
- [46] Zhang J and McMechan G A 1995 Estimation of resolution and covariance for large matrix inversions *Geophys. J. Int.* **121** 409–26