Hyper-differential sensitivity analysis with respect to model discrepancy: Calibration and optimal solution updating

Joseph Hart\textsuperscript{a,1}, Bart van Bloemen Waanders\textsuperscript{b,1,}\textsuperscript{*}

\textsuperscript{a}Sandia National Laboratories, P.O. Box 5800, Albuquerque, NM 87185, United States
\textsuperscript{b}Sandia National Laboratories, P.O. Box 5800, Albuquerque, NM 87185, United States

Abstract

Optimization constrained by computational models is common across science and engineering. However, in many cases a high-fidelity numerical emulation of systems cannot be optimized due to complexity and computational costs. Rather, low-fidelity models are constructed to enable intrusive algorithms for large-scale optimization. As a result of the discrepancy between high and low-fidelity models, optimal solutions determined using low-fidelity models are frequently far from true optimality. In this article we introduce a novel approach that uses post-optimality sensitivities with respect to model discrepancy to enable solutions representative of the true system. Limited high-fidelity data is used to calibrate the model discrepancy in a Bayesian framework which in turn is propagated through the optimization problem. The result provides significant improvement in optimal solutions with uncertainty characterizations. Our formulation exploits structure in the post-optimality sensitivity operator to achieve computational scalability. Numerical results demonstrate how an optimal solution computed using a low-fidelity model may be significantly improved with limited evaluations of a high-fidelity model.

Keywords: Hyper-differential sensitivity analysis, post-optimality sensitivity analysis, PDE-constrained optimization, model discrepancy

1. Introduction

Optimization problems constrained by computational models are ubiquitous in science and engineering and have been the topic of extensive research. Nonetheless, optimization problems are only as useful as the models that constrain them, and in the case of large-scale science and engineering applications, the models are frequently flawed due to the complexity of the system being modeled. Errors arise from many sources such as incomplete knowledge of the system physics or a lack of computational resources to accurately simulate the model. Hence the use of such models in optimization algorithms mandate rigorous analysis of how modeling errors influence the optimal solution.

We use the terminology model form error (or model error) to describe errors in the mathematical representation of the model, for instance, missing an operator in a differential equation or using a phenomenological model for a mathematically undefined component within a system. The presence of model form error causes the solution predicted by the model to differ from reality. The difference between a model’s prediction and reality is referred to as the model discrepancy. This article focuses on how model discrepancy influences optimization problems, assuming that a high-fidelity data source is available to calibrate the discrepancy. We focus on problems where high-fidelity data is generated from limited number of computational model evaluations. Models rarely predict reality perfectly, but in the scope of this article we consider a high-fidelity computational model as the “gold standard” for comparison.

In many applications, high-fidelity models exist but cannot be instrumented for optimization due to their software complexity and computational cost. This is common in production codes

\textsuperscript{*}Corresponding author

Email addresses: johart@sandia.gov (Joseph Hart), bartv@sandia.gov (Bart van Bloemen Waanders)
which are developed over many years by large teams with the goal of achieving model fidelity rather
than enabling optimization or other intrusive analysis approaches. In such cases, a logical option
is to construct low-fidelity models and thereby enable efficient optimization. Assuming that the
optimization problem can be solved using a low-fidelity model, we pose the question: “How can a
small number of evaluations from the high-fidelity model be used to improve the optimal solution?”.
Assuming an algorithmic approach can be created to address this challenge, this would benefit a
variety of complex optimization problems for applications with multi-scale, multi-physics, nonlinear,
and multi-component features (such as climate, additive manufacturing, and experimental fusion).
Another potential benefit is that low-fidelity models may arise from model order reduction methods
or other approximation approaches, all of which yield efficient models but suffer from model-form
error.

In this paper we present new capabilities to make use of high-fidelity data to improve the opti-
mal solution by incorporating hyper-differential sensitivity analysis (HDSA) with respect to model
discrepancy [22]. Given that the space of optimal solutions is large and the number of high-fidelity
solves is small, it is critical that we also characterize uncertainty due to the data sparsity. Accord-
ingly we formulate a Bayesian inverse problem whose solution is coupled with HDSA to update the
optimal solution with a statistical determination of the uncertainty. Our approach depends on PDE-
constrained optimization with efficient supporting components, including adjoint-based derivative
computation, Krylov and Newton iterative solves, and parallel numerical linear algebra. The reader
is referred to [43, 4, 20, 44, 30, 7, 31, 27, 25, 6, 9, 28, 5, 2] for additional details.

Post-optimality sensitivities determines the influence of parametric uncertainty with respect to
the solution of optimization problems [8, 15]. It was originally developed in the context of opera-
tions research and then extended to optimization constrained by partial differential equations (PDEs)
targeting stability analysis [19, 18, 10]. HDSA was specifically created to evaluate the influence of
uncertainties by scaling post-optimality sensitivities to high-dimensions through a coupling of tools
from PDE-constrained optimization and numerical linear algebra [23, 39, 42, 21, 41, 24]. Building on
these developments, [22] applies HDSA with respect to model discrepancy. A combination of special
Kronecker product representation and innovative numerical linear algebra, enabled computational
efficient methods which demonstrated new insight of model discrepancy effects on optimization so-
lutions. The subject of this article is to leverage these model discrepancy sensitivities and strive to
improve the solution of optimization problems constrained by low-fidelity models. We show that
optimization solutions can be achieved without instrumenting high-fidelity models with optimization
algorithms. This new capability is made possible by introducing a Bayesian formulation to

calibrate the model discrepancy and efficiently propagate its posterior through the post-optimality
sensitivity operator. A systematic integration of prior knowledge, high-fidelity data, and low-fidelity
optimization is enabled in a pragmatic way for large-scale applications.

Kennedy and O’Hagan [29] calibrated model discrepancy alongside parameters in a Bayesian
framework to account for uncertainty and correlations between the discrepancy and parameters. This
was followed by many works which analyzed and extended their approach, [36, 11, 26, 3, 33, 32, 16].
In the calibration context, the model discrepancy is accessed through observed data with an implicit
dependence on the calibration parameters which are not known a priori. Our work is inspired
by these concepts but is significantly different in our assumption that a high-fidelity model may be
queried for different values of the optimization variables. We consider general optimization problems
and assume an ability to control the parameter settings in the high-fidelity model evaluations. For
this reason, we focus on optimal control and design problems; however, model calibration may still
be considered from a different perspective than Kennedy and O’Hagan.

Multi-fidelity methods are motivated by a need to enable outer loop analysis such as optimization
and uncertainty quantification. They seek to develop algorithms that combine evaluations of high
and low-fidelity models to achieve accuracy commensurate to high-fidelity analysis using a smaller
number of high-fidelity model evaluations [37]. This has seen considerable interest in the context
of accelerating Monte Carlo type analysis [17] for general uncertainty quantification applications as
well as in the specific context of optimization under uncertainty [35, 34]. There has also been work to
combine high and low-fidelity model evaluations within each iterate of a deterministic optimization
algorithm [12, 1]. Analogously to these ideas, we assume access to both a high and low-fidelity
model. However, unlike the previous references we seek a framework where the optimization problem does not require access to the high-fidelity model. Furthermore, our focus is on cases where the number of high-fidelity solves is significantly constrained and must be preformed offline. In what follows, section 2 introduces the optimization problem under consideration and section 3 reviews the model discrepancy sensitivities introduced in [22]. Section 4 presents the article’s contributions to formulate and solve the discrepancy calibration problem and propagate its posterior through the post-optimality sensitivity operator. The proposed algorithm with computational cost requirements are analyzed in Section 5. We show numerical results in Section 6 for an illustrative thermal control example and a more complex fluid flow example where the Stokes equation is used as a low-fidelity approximation of the Navier-Stokes equation. Using only one forward Navier-Stokes solve, we are able to improve the quality of the Stokes optimal solution by two orders of magnitude in the objective function. We conclude in Section 7 with a forward looking perspective on how this research may impact a variety of science and engineering applications.

2. Optimization Formulation

Consider optimization problems of the form

$$\min_{z \in \mathcal{Z}} J(S(z), z)$$  \hspace{1cm} (1)

where $z$ denotes an optimization variable in the Hilbert space $\mathcal{Z}$, $S: \mathcal{Z} \to \mathcal{U}$ denotes the solution operator for a PDE (partial differential equation) $c(u, z) = 0$ with state variable $u$ in a Hilbert space $\mathcal{U}$, and $J: \mathcal{U} \times \mathcal{Z} \to \mathbb{R}$ is the objective function. We allow for the possibility that $\mathcal{Z}$ and $\mathcal{U}$ are infinite dimensional. This formulation is general and applicable to design, control, and inverse formulations. For simplicity we will call $z$ the “parameters” though its name varies depending upon the context.

The high-fidelity PDE governing the system may be too complex and/or computationally intensive to optimize, so in practice we solve

$$\min_{z \in \mathcal{Z}} J(\tilde{S}(z), z)$$  \hspace{1cm} (2)

where $\tilde{S}: \mathcal{Z} \to \mathcal{U}$ is the solution operator for a simpler model $\tilde{c}(u, z) = 0$. The solution of the low-fidelity optimization problem (2) is used to approximate the solution of the high-fidelity optimization problem (1). The goal of this article is to analyze how uncertainty in the discrepancy, $S(z) - \tilde{S}(z)$, may be modeled and propagated through (2) to improve the optimal solution with a characterization of uncertainty.

Focusing on the discretized problem, let $\{\phi_i\}_{i=1}^m$ and $\{\psi_j\}_{j=1}^n$ be bases for finite dimensional subspaces $\mathcal{U}_h \subset \mathcal{U}$ and $\mathcal{Z}_h \subset \mathcal{Z}$, respectively. As an example, these may be finite element basis functions. Let $u \in \mathbb{R}^m$ and $z \in \mathbb{R}^n$ denote coordinates in basis representations and let

$$(M_u)_{i,j} = (\phi_i, \phi_j)_{\mathcal{U}} \quad \text{and} \quad (M_z)_{i,j} = (\psi_i, \psi_j)_{\mathcal{Z}}$$

be mass matrices that define inner products in $\mathcal{U}_h$ and $\mathcal{Z}_h$ using the coordinate vectors. The objective $J$ and solution operator $\tilde{S}$ are discretized by $J: \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ and $\tilde{S}: \mathbb{R}^n \to \mathbb{R}^m$ to arrive at a finite dimensional optimization problem.

3. Model discrepancy and post-optimality sensitivities

To propagate model discrepancy through the optimization problem, we build on [22] defining a representation of the discrepancy as $\delta(z, \theta) \approx S(z) - \tilde{S}(z)$, a function of $z$ parameterized by $\theta$. We assume that $\delta(z, \theta_{\text{nom}}) = 0 \forall z$ when evaluated at $\theta_{\text{nom}} = 0$. Consider the parameterized optimization problem

$$\min_{z \in \mathbb{R}^n} \tilde{J}(z, \theta) := J(\tilde{S}(z) + \delta(z, \theta), z)$$  \hspace{1cm} (3)
which we assume has a local minimum \( \mathbf{z} \in \mathbb{R}^n \) that satisfies the first and second order optimality conditions when model discrepancy is zero, i.e. \( \theta = \theta_{\text{nom}} \).

Applying the Implicit Function Theorem to the equation \( \nabla_z \hat{J}(\mathbf{z}, \theta_{\text{nom}}) = 0 \), we consider an operator \( F : \mathcal{N}(\theta_{\text{nom}}) \to \mathcal{N}(\mathbf{z}) \), defined on neighborhoods of \( \theta_{\text{nom}} \) and \( \mathbf{z} \), such that \( F(\theta) \) is a stationary point of (3), i.e.

\[
\nabla_z \hat{J}(F(\theta), \theta) = 0 \quad \forall \theta \in \mathcal{N}(\theta_{\text{nom}}).
\]

Assuming positive definiteness of the Hessian for all \( (F(\theta), \theta) \), \( F \) maps the model discrepancy to the optimal solution. The Jacobian of \( F \) with respect to \( \theta \), i.e. the sensitivity of the optimal solution to the model discrepancy, is given by

\[
\nabla_\theta F(0) = -H^{-1}B \in \mathbb{R}^{n \times p}
\]

where \( H = \nabla_{z,z} \hat{J}(\mathbf{z}, 0) \in \mathbb{R}^{n \times n} \) is the Hessian of \( \hat{J} \) and \( B = \nabla_{z,\theta} \hat{J}(\mathbf{z}, 0) \in \mathbb{R}^{n \times p} \) is the Jacobian of \( \nabla_z \hat{J} \) with respect to \( \theta \). Both are evaluated at the nominal solution \((\mathbf{z}, 0)\). We interpret \( F_\theta'(0) \) as a Newton update of the optimal solution \( \mathbf{z} \) when the model discrepancy is perturbed.

To parameterize the discrepancy \( \delta \), observe that the difference in solution operators \( S(z) - \hat{S}(z) \) is, in general, a nonlinear function of \( z \). However, for post-optimality sensitivity analysis, it is sufficient to consider a discrepancy which is linear in \( z \). Following the derivation in [22], we consider a linear operator that inputs \( z \) and outputs the discrepancy (an element of the state space). Starting from a general linear operator on function spaces, parameterizing the operator with \( \theta \), and transforming to the coordinate space, we define the discrete model discrepancy \( \delta : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}^m \) as

\[
\delta(z, \theta) = \left( I_m \ I_m \otimes z^T M_z \right) \theta
\]

where \( \theta_0 = (\theta_{0,0}, \theta_{1,0}, \ldots, \theta_{m,0})^T \in \mathbb{R}^m \), \( \theta_i = (\theta_{i,1}, \theta_{i,2}, \ldots, \theta_{i,n})^T \in \mathbb{R}^n \), \( i = 1, 2, \ldots, m \), and \( \theta = (\theta_0^T, \theta_1^T, \ldots, \theta_m^T)^T \in \mathbb{R}^p \), \( p = m(n+1) \), and \( I_m \in \mathbb{R}^{m \times m} \) is the identity matrix. Our ordering of \( \theta \)'s components enables an efficient expression of \( \delta \) using the Kronecker product \( \otimes \). We emphasize that \( \theta \in \mathbb{R}^p \) may be extremely high dimensional and our subsequent analysis relies on representing all computation as Kronecker products to execute linear algebra in \( \mathbb{R}^m \) and \( \mathbb{R}^n \) rather than \( \mathbb{R}^p \).

To simplify the analysis which follows we assume that \( \nabla_{z,u} J = 0 \). This holds on a wide range of problems in practice where the objective is defined by a performance criteria (tracking type, data misfit, etc.) on the state plus a regularization on the parameters. Differentiating the objective function \( \hat{J}(z, \theta) \), the resulting mixed-second derivative can be written in a Kronecker form as

\[
B = \tilde{S}_z^T \nabla_{u,u} J \left( I_m \ I_m \otimes z^T M_z \right) + \left( 0 \ \nabla_u J \otimes M_z \right) \in \mathbb{R}^{n \times p}
\]

where \( \tilde{S}_z \), \( \nabla_{u,u} J \), and \( \nabla_{u,u} J \) are evaluated at \((\tilde{\mathbf{u}}, \mathbf{z})\), \( \tilde{\mathbf{u}} = \hat{S}(\mathbf{z}) \). By expressing \( B \) with Kronecker products, which arise from the Jacobians of \( \delta(z, \theta) \), the sensitivity computation is compressed into \( \mathbb{R}^m \) and \( \mathbb{R}^n \), thus avoiding computation in \( \mathbb{R}^p \).

4. Updating and quantifying uncertainty in the optimal solution

Assume that the low-fidelity optimization problem (2) has been solved to determine \( \mathbf{z} \), and that we have access to \( N \) forward solves of the high-fidelity model \( \mathbf{S} \) for different inputs \( \{z_\ell\}_{\ell=1}^N \). Let \( y_\ell = \mathbf{S}(z_\ell) - \hat{S}(z_\ell), \ell = 1, 2, \ldots, N \), denote the corresponding evaluations of the model discrepancy at these inputs. In practice, the number of evaluations \( N \) will be small since evaluating \( \mathbf{S}(z_\ell) \) is computationally costly. Our goal is to use these limited high-fidelity evaluations to: (i) improve the optimal solution \( \mathbf{z} \) without requiring model modifications or additional nonlinear solves, and (ii) to characterize uncertainty in the optimal solution. Figure 1 depicts our proposed approach. In this section, we detail our contributions to enable this workflow, which include:

1. defining a prior distribution on \( \theta \) to reflect knowledge of the discrepancy,
2. Formulating and solving (in closed form) a Bayesian inverse problem using the data \( \{ z_\ell, y_\ell \}_{\ell=1}^N \) to determine a distribution for \( \theta \), and

3. Computing posterior samples of discrepancy and propagating them through the sensitivity operator to compute a distribution on the optimal solution.

Due to the high dimensionality of the model discrepancy parameterization, \( \theta \in \mathbb{R}^p \), it is critical that sampling from the posterior is performed efficiently. We make judicious choices in the problem formulation and manipulate the linear algebra to derive a scalable algorithm. In particular, the complexity of the algorithm scales with \( N \), the number of high-fidelity forward solves, which we assume is small because of computational costs. The limited data from high-fidelity solves results in an undetermined problem which emphasizes the importance of our proposed uncertainty quantification framework.

**Discrepancy Parameterization**

\[
\delta(z, \theta) \approx S(z) - \tilde{S}(z)
\]

**Low-fidelity Optimal Solution**

\[
\min_z J(S(z) + \delta(z, \theta), z)
\]

**Post-Optimality Sensitivity**

\[
F^{\theta}(0)
\]

**Optimal Solution Posterior**

\[
z_{\text{post}} \approx F^{\theta}(0) \theta_{\text{post}}
\]

**High-fidelity Data**

\[
S(z_t) - \tilde{S}(z_t)
\]

**Physics knowledge**

\[
\pi_{\text{post}} \propto \pi_{\text{likelihood}} \pi_{\text{prior}}
\]

Figure 1: Diagram of the proposed analysis pipeline. We use \( \pi \) to denote probability distributions arising in the Bayesian inverse problem, and \( \theta_{\text{post}} \) and \( z_{\text{post}} \) to denote the random vectors distributed according to the discrepancy and optimal solution posteriors, respectively.

### 4.1. Defining the prior

To facilitate efficient computations we use a Gaussian prior for \( \theta \) with mean \( 0 \) and a covariance matrix to incorporate prior knowledge. Such priors are common in infinite dimensional Bayesian inverse problems. Since \( \theta \in \mathbb{R}^p \) does not have a physical interpretation, we focus on defining priors in \( \mathbb{R}^m \) and \( \mathbb{R}^n \) (the state and parameter spaces).

Let \( L \in \mathbb{R}^{m \times m} \) be a symmetric positive definite matrix that encodes known characteristics of the discrepancy. This may embed assumptions about the smoothness of \( \delta \), boundary conditions, or penalize deviations from known conservation properties. A common approach is to define \( L \) as the square of an elliptic differential operator (defined on the state space) so that \( L^{-1} \) corresponds to the discretization of a well-defined prior covariance in the infinite dimensional state space [13, 38, 40].

The matrix \( L \) imposes knowledge on the discrepancy independent of \( z \) and is only defined on \( \mathbb{R}^m \) rather than the product space \( \mathbb{R}^p \). To impose prior knowledge on \( z \), we assume a Gaussian distribution with mean \( \bar{z} \) and covariance \( \Gamma \in \mathbb{R}^{n \times n} \) to model the space of possible parameters. This focuses our analysis around the nominal solution \( \bar{z} \) with deviation from it characterized by \( \Gamma \). There may be less prior information on \( z \) since it is being determined through the optimization problem; however, \( \Gamma \) plays an important role defining characteristic length scales for \( z \).
Given \( L \) and \( \Gamma \), along with the Kronecker structure of \( \delta \) (5), we compute the expected value (with respect to \( z \)) of the \( L \)-weighted inner product of \( \delta(z, \theta) \) with itself. This yields the symmetric positive definite matrix

\[
M_\theta = \begin{pmatrix}
L & L \otimes M_z \bar{z} \\
L \otimes M_z \bar{z} & L \otimes (\Gamma + \bar{z} \bar{z}^T) M_z
\end{pmatrix} \in \mathbb{R}^{p \times p},
\]

for which \( \theta^T M_\theta \theta = \mathbb{E}_z [(\delta(z, \theta), \delta(z, \theta))_L] \).

Hence \( M_\theta \) defines an inner product for \( \theta \) to measure the size of the model discrepancy \( \delta(z, \theta) \) according to our prior knowledge imposed in \( L \) and \( \Gamma \). This provides a mechanism to impose prior knowledge onto the state and parameter spaces and form the product space via the Kronecker structure of \( \delta \). Letting \( M_\theta^{-1} \) be the prior covariance for \( \theta \) defines a prior that is rooted in established theory relating known physical properties to covariance matrices which can be efficiently manipulated [13, 38, 40].

As a building block for our subsequent analysis, note that we can decompose \( M_\theta = CC^T \) where

\[
C = \begin{pmatrix}
L \frac{1}{2} & 0 \\
L \frac{1}{2} \otimes M_z \bar{z} & L \frac{1}{2} \otimes M_z \Gamma \frac{1}{2}
\end{pmatrix}
\]

and

\[
C^{-1} = \begin{pmatrix}
L^{-\frac{1}{2}} & 0 \\
L^{-\frac{1}{2}} \otimes (-\Gamma^{-\frac{1}{2}} \bar{z}) & L^{-\frac{1}{2}} \otimes \Gamma^{-\frac{1}{2}} M_z^{-1}
\end{pmatrix}.
\]

This factorization enables efficient manipulation and sampling of the posterior discrepancy distribution.

4.2. Bayesian inverse problem

We formulate a Bayesian inverse problem to estimate \( \delta \) using the data pairs \( \{z_\ell, y_\ell\}_{\ell=1}^N \). To facilitate the analysis we assume:

**Assumption 1.** \( \{z_\ell\}_{\ell=1}^N \) is a linearly independent set of vectors in \( \mathbb{R}^n \).

This is usually satisfied in practice as one explores the model space in different directions since the number of model evaluations \( N \) is small relative to the dimension of \( z \). For notational simplicity, we define

\[
A_\ell = (I_m \quad I_m \otimes z_\ell^T M_z) \in \mathbb{R}^{m \times p}, \quad \ell = 1, 2, \ldots, N,
\]

so that \( \delta(z, \theta) = A_\ell \theta \), and the concatenation of these matrices

\[
A = \begin{pmatrix}
A_1 \\
A_2 \\
\vdots \\
A_N
\end{pmatrix} \in \mathbb{R}^{mN \times p}.
\]

\( A \theta \in \mathbb{R}^{mN} \) corresponds to the evaluation of \( \delta(z, \theta) \) for all of the input data. In an analogous fashion, define \( b \in \mathbb{R}^{mN} \) by stacking \( y_\ell, \ell = 1, 2, \ldots, N \), into a vector. Then we seek \( A \theta = b \).

To enable a closed form expression for the posterior we consider an additive Gaussian noise model with mean 0 and covariance \( \alpha I \), where \( \alpha > 0 \) is specified by the user. Typically the noise covariance in a Bayesian inverse problem is defined using knowledge of the data collection process. In cases where data is collected from simulation we do not have a clear definition of noise. However, \( \delta \) is defined as a linear function of \( z \), which is an approximation since \( \hat{S}(z) - \tilde{S}(z) \) is, in general,
a nonlinear function. Hence the noise may be interpreted as the approximation error due to the linearization of the discrepancy. Nonetheless, one can show that the range space of $A$ equals $\mathbb{R}^{mN}$, so there exists infinitely many $\theta \in \mathbb{R}^p$ such that $A\theta = b$. The users choice of $\alpha$ dictates the weight given to the data misfit relative to the prior. Taking a small $\alpha$ will drive the inverse problem toward interpolation with little regard for the prior. We will demonstrate that the user may experiment with different $\alpha$’s at a modest computational cost.

Given our formulation with Gaussian prior and noise models, and the linearity of $\delta(z, \theta)$, the posterior is Gaussian with a (unnormalized) negative log probability density function

$$\frac{1}{2\alpha} (A\theta - b)^T (A\theta - b) + \frac{1}{2} \theta^T M_\theta \theta. \tag{11}$$

The posterior mean and covariance is given by

$$\overline{\theta} = \frac{1}{\alpha} \Sigma A^T b \quad \text{and} \quad \Sigma = \left( M_\theta + \frac{1}{\alpha} A^T A \right)^{-1} \tag{12}.$$ 

Samples from a Gaussian distribution may be generated by multiplying a factor of the covariance matrix with a standard normal random vector, and adding the mean. Next we derive explicit and computationally efficient expressions for $\overline{\theta}$ and a factorization of $\Sigma$ to facilitate sampling.

4.3. Expressions for posterior sampling

We introduce a series of matrices and their factorizations to enable analysis of the posterior. Some of these expressions will not be computed but rather are intermediate steps toward our final algorithm. This section presents the derivations from first principles, followed by a description of the computational requirements in Section 5. Emphasis is given to preserving the Kronecker product structure throughout our analysis so that computation in $\mathbb{R}^p$ is ultimately not needed. The derivation of posterior quantities are described in four steps in the following subsections:

1. factorize $A$ in a $M_\theta$ orthogonal basis to rewrite $\Sigma^{-1}$,
2. invert $\Sigma^{-1}$ by exploiting orthogonality to invert a sum,
3. factorize $\Sigma$ by determining an eigenvalue decomposition,
4. sample the posterior distribution through judicious matrix-vector multiplication.

Factorizing $A$

To rewrite the inverse posterior covariance, $M_\theta + \frac{1}{\alpha} A^T A$, in a form amenable for inversion, we decompose $A$ in a chosen inner product. In particular, the generalized singular value decomposition (GSVD) of $A$ is used with the $M_\theta$ weighted inner product on the column space and is given by $A = \Xi \Phi \Psi^T M_\theta$. $\Phi$ is the diagonal matrix of singular values, and $\Xi$ and $\Psi$ are matrices containing the singular vectors which satisfy $\Xi^T \Xi = I$ and $\Psi^T M_\theta \Psi = I$. To determine the singular vectors, note that $A M_\theta^{-1} A^T = \Xi \Phi^2 \Xi^T$. Using the expressions (10) for $A$ and (9) for $M_\theta^{-1} = C^{-T} C^{-1}$, we can write

$$A M_\theta^{-1} A^T = G \otimes L^{-1}$$

where

$$G = ee^T + (Z - \Xi e^T)^T \Gamma^{-1} (Z - \Xi e^T) \in \mathbb{R}^{N \times N},$$

$Z = (z_1, z_2, \ldots, z_N)$ is the matrix of parameter data, and $e \in \mathbb{R}^N$ is the vector of ones. Hence the left singular vectors $\Xi$ correspond to the eigenvectors of $G \otimes L^{-1}$ and the squared singular values $\Phi^2$ correspond to the eigenvalues.

Denoting the eigenvectors and eigenvalues of $G$ and $L$ with $(g_i, \lambda_i)$ and $(l_j, \rho_j)$, respectively, and recalling properties of the eigenvalue decomposition of a Kronecker product, we observe that the
squared generalized singular values of $A$ are given by $\frac{\lambda_i}{\rho_j}$ and are associated with the left singular vectors $\xi_{i,j} = g_i \otimes l_j$. Rewriting the GSVD to solve for $\Psi$, the right singular vector associated with $\xi_{i,j}$ is given by

$$\psi_{i,j} = \frac{1}{\sqrt{\lambda_i \rho_j}} \left( l_j \otimes M^{-1}_z \Gamma^{-1} w_i \right), \quad i = 1, 2, \ldots, N \quad j = 1, 2, \ldots, m,$$

(13)

where

$$w_i = Z g_i - (e^T g_i) \bar{z} \quad \text{and} \quad s_i = (e^T g_i) - w_i^T \Gamma^{-1} \bar{z}.$$  

(14)

Given this decomposition of $A$ and our previously noted decomposition $M_\theta = CC^T$, we express $\Sigma^{-1}$ as

$$\Sigma^{-1} = \frac{1}{\alpha} CXC^T$$

(15)

where

$$X = \alpha I + C^T \Psi \Phi^2 \Psi^T C.$$  

(16)

**Inverting $\Sigma^{-1}$**

This factorization of $\Sigma^{-1}$ implies that $\Sigma = \alpha C^{-T} X^{-1} C^{-1}$ and facilitates easy manipulation because $X$ is a combination of diagonal and orthogonal matrices. Applying the Sherman-Morrison-Woodbury formula to (16), we have

$$X^{-1} = \frac{1}{\alpha} (I - C^T \Psi D \Psi^T C)$$

where $D \in \mathbb{R}^{mN \times mN}$ is a diagonal matrix whose entries are given by $\frac{\lambda_i}{\lambda_i + \alpha \rho_j}$. Algebraic simplifications yields

$$\Sigma = M_\theta^{-1} - \Psi D \Psi^T.$$  

(17)

The posterior covariance may be interpreted as the prior covariance $M_\theta^{-1}$ with its uncertainty reduced in the directions of the columns of $\Psi$ (13) with weight $\frac{\lambda_i}{\lambda_i + \alpha \rho_j}$, a combination of the eigenvalues from $L$ and $G$, along with the noise covariance $\alpha$.

**Factorizing $\Sigma$**

The expression (17) provides an interpretation of the posterior covariance and a way to compute the posterior mean $\bar{\theta} = \frac{1}{\alpha} \Sigma A^T b$. However, posterior sampling require a factorization of $\Sigma$ which is not easily attained from (17). Accordingly, we compute the eigenvalue decomposition of $X$.

Note that $(C^T \Psi)^T (C^T \Psi) = \Psi D \Psi^T = I$, and hence the columns of $C^T \Psi$ are orthonormal in the Euclidean inner product. This implies that $C^T \psi_{i,j}$ is an eigenvector of $X$ with eigenvalue $\alpha + \frac{\lambda_i}{\rho_j}$. By multiplying (8) and (13), we express these eigenvectors as

$$C^T \psi_{i,j} = \frac{1}{\sqrt{\lambda_i}} \left( l_j \otimes M^{-1}_z \Gamma^{-1} w_i \right).$$

(18)

This gives a total of $mN$ eigenpairs of $X \in \mathbb{R}^{p \times p}$. These have repeated eigenvalue $\alpha$ and eigenvectors from a set of orthonormal vectors that are orthogonal to the columns of $C^T \Psi \in \mathbb{R}^{p \times mN}$. To determine these remaining eigenpairs we assume:

**Assumption 2.** $z_1 = \bar{z}$.

We require that data is collected at the nominal optimal solution $\bar{z}$; a reasonable assumption given that our goal is to use high-fidelity forward solves to improve it. Recalling (14) and using $z_1 = \bar{z}$, we
observe that \( \text{span}\{w_i\}_{i=1}^N = \text{span}\{\tilde{z}_\ell - \bar{z}\}_{\ell=2}^N \) and let \( \{\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_{n-N+1}\} \) in \( \mathbb{R}^n \) be an orthonormal set of vectors which are orthogonal to \( \{\Gamma^{-\frac{1}{2}}(z_\ell - \bar{z})\}_{\ell=2}^N \). Then remaining eigenvectors of \( X \) are given by

\[
\begin{pmatrix}
    0 \\
    l_j \otimes \tilde{z}_k
\end{pmatrix}
\quad j = 1, 2, \ldots, m, \quad k = 1, 2, \ldots, n - N + 1.
\]

(19)

Collecting this set of \( p \) eigenvectors, through the union of (18) and (19), in an orthogonal matrix \( Q \) and the corresponding eigenvalues into a diagonal matrix \( \Upsilon \), the eigenvalue decomposition is given by

\[
X = Q\Upsilon Q^T.
\]

Hence, recalling (15), we decompose the covariance matrix as

\[
\Sigma = \alpha C^{-T}X^{-1}C^{-1} = TT^T
\]

(20)

where

\[
T = \sqrt{\alpha}C^{-T}Q\Upsilon^{-\frac{1}{2}}.
\]

(21)

The expression (21) facilitates efficient sampling, as we show below. Note that \( T \) depends on \( \{\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_{n-N+1}\} \), which is difficult to compute if \( n \) is large. However, as shown below, existence of the vectors \( \{\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_{n-N+1}\} \) enables further derivations which culminate in expressions that do not require their explicit computation.

**Computing posterior samples**

Given the expressions (17) and (20), we write the posterior samples in terms of linear solves in \( \mathbb{R}^m \) and \( \mathbb{R}^n \). Our analysis thus far involves summing over all eigenpairs of \( L \), the size of which scales with the mesh resolution of the state. To address the computational challenges we will compute matrix-vector products and write the resulting vectors in terms of linear solves with \( L \) rather than sums over its eigenpairs.

Using the definition of the mean (12), covariance (17) and discrepancy matrix (10), the posterior mean is derived as

\[
\bar{\theta} = \frac{1}{\alpha} \sum_{\ell=1}^N \left[ \left( u_\ell \otimes M_\zeta \Gamma^{-1}(z_\ell - \bar{z}) \right) - \sum_{i=1}^N b_{i,\ell} \left( u_{i,\ell} \otimes M_\zeta \Gamma^{-1}w_i \right) \right]
\]

with constants

\[
a_\ell = 1 - \bar{z}^T \Gamma^{-1}(z_\ell - \bar{z}) \quad \text{and} \quad b_{i,\ell} = (z_\ell - \bar{z})^T \Gamma^{-1}Zg_i + (e^T g_i) a_\ell
\]

and vectors

\[
u_\ell = L^{-1}y_\ell \quad \text{and} \quad u_{i,\ell} = (\alpha L + \lambda_i I)^{-1} u_\ell.
\]

We observe that it depends on linear solves \(^1\) involving \( L, \Gamma \), and \( M_\zeta \) where the right hand sides arise from the data \( \{z_\ell, y_\ell\}_{\ell=1}^N \). The posterior mean can be interpreted as a linear combination of the discrepancy data preconditioned by the prior weighted by the noise covariance \( \alpha \) and the eigenvalues \( \lambda_i \) from the \( \{z_\ell\}_{\ell=1}^N \) data informed matrix \( G \).

To quantify uncertainty, we draw samples from the posterior distribution which takes the form \( \bar{\theta} + T\omega \), where \( \omega \sim \mathcal{N}(0, I_p) \) follows a standard normal distribution in \( \mathbb{R}^p \). A direct calculation

\(^1\)In Section 6, we demonstrate how to efficiently invert \( L \) and \( (\alpha L + \lambda_i I) \) using a generalized singular value decomposition when \( L \) is defined as the square of an elliptic operator.
of \( T\omega \) poses problems because we cannot write the product in terms of linear solves with \( L \). The factorization \( \Sigma = TT^T \) does not preserve the eigendecomposition of \( L \). We use the property that Gaussian random vectors are invariant under orthogonal transformations, introduce the orthogonal matrix \((I_{n+1} \otimes V^L)^T \in \mathbb{R}^{p \times p} \), where \( V^L \in \mathbb{R}^{m \times m} \) is the matrix whose columns are eigenvectors of \( L \), and sample \( \omega = (I_{n+1} \otimes V^L)^T \nu \), where \( \nu \in \mathcal{N}(\mathbf{0}, I_p) \). Combining this transformation with (21) the posterior samples can be computed as

\[
T\omega = T(I_{n+1} \otimes V^L)^T \nu = \hat{\theta} + \tilde{\theta}
\]

where

\[
\hat{\theta} = \sqrt{\alpha} \sum_{i=1}^{N} \frac{1}{\sqrt{\lambda_i}} \left( \hat{u}_i \otimes M^{-1}_z \Gamma^{-1} w_i \right) \quad \text{and} \quad \tilde{\theta} = \sum_{k=1}^{n-N+1} \left( \tilde{u}_k \otimes \tilde{w}_k \right).
\]

These sample components \( \hat{\theta} \) and \( \tilde{\theta} \) depend on

\[
\hat{u}_i = (\alpha L + \lambda_i I)^{-\frac{1}{2}} \nu_i \quad \quad \quad \tilde{s}_k = -\tilde{z}^T \Gamma^{-\frac{1}{2}} \tilde{z}_k
\]

\[
\tilde{u}_k = L^{-\frac{1}{2}} \nu_{N+k} \quad \quad \quad \tilde{w}_k = M^{-1}_z \Gamma^{-\frac{1}{2}} \tilde{z}_k
\]

where \( \nu = (\nu_1^T, \nu_2^T, \ldots, \nu_{n+1}^T)^T \) and \( \nu_i \in \mathbb{R}^m \), \( i = 1, 2, \ldots, n+1 \). As in the expression for the mean, \( \hat{\theta} \) and \( \tilde{\theta} \) are written in terms of linear solves with \( L \) rather than its eigenpairs. This is made possible because of our formulation to apply \( T \) to the vector \((I_{n+1} \otimes V^L)^T \nu \) allows manipulation of the eigenvectors.

Combining these results, posterior samples for \( \theta \) may be decomposed as

\[
\bar{\theta} + \hat{\theta} + \tilde{\theta}
\]

where \( \bar{\theta} \) is the mean (a deterministic quantity), \( \hat{\theta} \) is a random vector modeling uncertainty in directions informed by the data, and \( \tilde{\theta} \) is a random vector modeling uncertainty in directions which are not informed by the data.

4.4. Posterior model discrepancy

Substituting the expressions for \( \bar{\theta}, \hat{\theta}, \) and \( \tilde{\theta} \) into (5), samples of the posterior model discrepancy are given by

\[
\delta(z, \bar{\theta} + \hat{\theta} + \tilde{\theta}) = \tilde{\delta}(z) + \hat{\delta}(z) + \bar{\delta}(z)
\]  

(22)

where

\[
\bar{\delta}(z) = \frac{1}{\alpha} \sum_{i=1}^{N} \left( 1 + (z_i - \bar{z})^T \Gamma^{-1} (z_i - \bar{z}) \right) u_i - \sum_{i=1}^{N} b_{i,\ell} (e^T g_i + w_i^T \Gamma^{-1} (z_i - \bar{z})) u_{i,\ell}
\]

(23)

corresponds to evaluating the mean model discrepancy,

\[
\hat{\delta}(z) = \sqrt{\alpha} \sum_{i=1}^{N} \frac{1}{\sqrt{\lambda_i}} (e^T g_i + w_i^T \Gamma^{-1} (z_i - \bar{z})) \hat{u}_i
\]

(24)

represents discrepancy uncertainty in the directions defined by the data \( (z_i)_{i=1}^{N} \), and

\[
\tilde{\delta}(z) = \sum_{k=1}^{n-N+1} \left( \tilde{z}_k^T \Gamma^{-\frac{1}{2}} (z - \bar{z}) \right) \tilde{u}_k
\]
represents discrepancy uncertainty in directions informed exclusively by the prior. Observe that $\delta$ vanishes in data-informed directions, $\delta(z_\ell) = 0$, $\ell = 1, 2, \ldots, N$, since $\{\tilde{z}_k\}_{k=1}^{n-N+1}$ are orthogonal to $\{(\Gamma^{-\frac{1}{2}}(z_\ell - \bar{z}))\}_{\ell=1}^{N}$. Furthermore, observe that
\[
\delta(\bar{z} + \Gamma^{\frac{1}{2}}\tilde{z}_k) = \delta(\bar{z}) \quad \text{and} \quad \delta(\bar{z} + \Gamma^{\frac{1}{2}}\tilde{z}_k) = \delta(\bar{z}) \quad k = 1, 2, \ldots, n - N + 1
\]
do not vary in directions orthogonal to the data. In addition, as $\alpha \to 0$, $\delta(\bar{z})$ interpolates the observed data since $\bar{\theta}$ is a minimizer of (11), while $\delta(\bar{z}) \to 0$ as $\alpha \to 0$, indicates full confidence in the data interpolation. $\delta(\bar{z})$ does not depend on $\alpha$ since it acts in directions orthogonal to the data and thereby captures uncertainty from the prior in directions which are not informed by data.

4.5. Applying the sensitivity operator to posterior samples

To characterize uncertainty in the solution of the optimization problem, we propagate posterior samples $\bar{\theta} + \hat{\theta} + \tilde{\theta}$ through the post-optimality sensitivity operator $F_\theta'(0)$. Given the Kronecker product structure, we compute the action of $B$ on samples and then apply $-H^{-1}$ to the resulting vectors. Observe that
\[
B\bar{\theta} = \frac{1}{\alpha} S_{\bar{z}}^T \nabla_{u,u} J \left[ \sum_{\ell=1}^{N} \left( u_\ell - \sum_{i=1}^{N} b_{i,\ell} (e^T g_i) u_{i,\ell} \right) \right]
\]
\[+ \frac{1}{\alpha} \sum_{\ell=1}^{N} (\nabla_u J u_\ell) \Gamma^{-1}(z_\ell - \bar{z}) \]
\[= \frac{1}{\alpha} \sum_{\ell=1}^{N} \sum_{i=1}^{N} b_{i,\ell} (\nabla_u J u_{i,\ell}) \Gamma^{-1} w_i
\]
\[
B\theta = \sqrt{\alpha} S_{\bar{z}}^T \nabla_u J \left( \sum_{i=1}^{N} \frac{e^T g_i}{\sqrt{\lambda_i}} \hat{u}_i \right) + \sqrt{\alpha} \sum_{i=1}^{N} \frac{\nabla_u J \hat{u}_i}{\sqrt{\lambda_i}} \Gamma^{-1} w_i.
\]
and
\[
B\tilde{\theta} = \sum_{k=1}^{n-N+1} (\nabla_u J \tilde{u}_k) \Gamma^{-\frac{1}{2}} \tilde{z}_k.
\]
Hence, we may compute samples of the optimal solution as $\bar{z} - H^{-1}B(\bar{\theta} + \hat{\theta} + \tilde{\theta})$ without requiring computation in $R^p$. Equations (25)-(27) provides a systematic and interpretable combination of high-fidelity data defining $u_\ell$ and $u_{i,\ell}$, the low-fidelity model in $S_{\bar{z}}^T$, prior information via $L$ and $\Gamma$ (which appear in several terms in the expressions), and the optimization objective in $\nabla_u J$ and $\nabla_{u,u} J$.

5. Algorithmic overview and computational cost analysis

This section presents an algorithmic overview to detail how quantities derived in the previous section - the prior discrepancy, the posterior discrepancy, and the posterior optimal solution - can be computed efficiently. We provide pseudo-code to summarize the algorithm and computational cost by counting the total number of linear system solves (in $R^m$ or $R^n$). Since the number of high-fidelity forward solves $N$ is small we consider computation in $R^n$ to be negligible.

5.1. Prior discrepancy samples

Using the factorization $M_\theta^{-1} = C^{-T} C^{-1}$, we derive an algorithm to draw prior samples of the model discrepancy. In particular, a prior sample of $\theta$ is given by $C^{-T} \omega_p$, where $\omega_p \sim N(0, I_p)$. 

11
Evaluating the model discrepancy at such samples yields

$$\delta(z, C^{-T}\omega_p) = L^{-\frac{1}{2}}\omega_0 + L^{-\frac{1}{2}}\Omega^{-\frac{1}{2}}(z - \bar{z})$$

where $$\omega_0 \in \mathbb{R}^m$$ and $$\Omega \in \mathbb{R}^{m \times n}$$ have entries that are independent identically distributed (i.i.d) samples from a standard normal distribution. To avoid sampling and forming the dense matrix $$\Omega \in \mathbb{R}^{m \times n}$$, we evaluate $$\delta(z, C^{-T}\omega)$$ at particular $$z$$’s sampled as $$z \sim \mathcal{N}(-\hat{z}, \Gamma)$$. Such samples take the form $$z = \bar{z} + \Gamma^{\frac{1}{2}}\omega_n$$, where $$\omega_n \sim \mathcal{N}(0, I_n)$$, and hence a prior discrepancy sample evaluated at $$z$$ takes the form

$$\delta(\bar{z} + \Gamma^{\frac{1}{2}}\omega_n, C^{-T}\omega_p) = L^{-\frac{1}{2}}(\omega_0 + \Omega\omega_n)$$.

Given a fixed $$\omega_n$$, we sample discrepancies as $$\delta(\bar{z} + \Gamma^{\frac{1}{2}}\omega_n, C^{-T}\omega_p) = L^{-\frac{1}{2}}\omega_m$$, where $$\omega_m \sim \mathcal{N}(0, (1 + \omega_n^T\omega_n)I_m)$$ follows from taking sums and linear transformations of Gaussian random vectors. Thus, to compute $$S$$ samples of $$z$$ and $$\delta(z, \theta)$$ we apply $$L^{-\frac{1}{2}}$$ and $$\Gamma^{\frac{1}{2}}$$ to $$S$$ standard normal random vectors in $$\mathbb{R}^m$$ and $$\mathbb{R}^n$$, respectively. The resulting images of $$L^{-\frac{1}{2}}$$ are scaled by $$\sqrt{1 + \omega_n^T\omega_n}$$. This is summarized in Algorithm 1.

**Algorithm 1** Compute samples of the prior $$z$$ and discrepancy $$\delta(z, \theta)$$.

1: Input: Number of desired samples $$S$$
2: for $$s = 1, 2, \ldots, S$$ do
3: Sample $$\omega_m \in \mathbb{R}^m$$ and $$\omega_n \in \mathbb{R}^n$$ with i.i.d. standard normal entries
4: Compute: $$z_{s}^{\text{prior}} = \bar{z} + \Gamma^{\frac{1}{2}}\omega_n \in \mathbb{R}^n$$
5: $$\delta_{s}^{\text{prior}}(z_{s}^{\text{prior}}) = \sqrt{1 + \omega_n^T\omega_n}L^{-\frac{1}{2}}\omega_m$$
6: end for
7: Number of linear solves: $$\Gamma^{\frac{1}{2}}L^{-\frac{1}{2}}$$
8: $$S$$ $$S$$

5.2. Posterior discrepancy samples

To generate posterior samples, we compute $$\Gamma^{-\frac{1}{2}}z_{\ell}, L^{-\frac{1}{2}}y_{\ell},$$ and $$(\alpha L + \lambda I)^{-\frac{1}{2}}L^{-\frac{1}{2}}y_{\ell}, \ell = 1, 2, \ldots, N,$$ i = 1, 2, . . . , N, as a preprocessing step and store the resulting vectors. Similar to the prior, we simplify our analysis by specifically evaluating at the input data $$\{z_{\ell}\}_{\ell=1}^{N}$$. Recalling the decomposition (22) into $$\delta, \tilde{\delta},$$ and $$\hat{\delta}$$, and the observation that $$\hat{\delta}(z_{\ell}) = 0$$, we ignore the computation of $$\hat{\delta}$$. Realizations of $$\tilde{\delta}$$ at other inputs will be the scaled versions of the prior discrepancy.

We compute $$\delta(z_{\ell})$$ in closed form using precomputed data. Computing a sample of $$\tilde{\delta}(z_{\ell})$$ requires $$N$$ linear solves to apply $$(\alpha L + \lambda I)^{-\frac{1}{2}}, i = 1, 2, \ldots, N,$$ to random vectors, as summarized in Algorithm 2.

Note that the posterior discrepancy samples reveal the effect of $$\alpha$$ on our analysis. If the user computes $$S$$ posterior samples and observes that the data is interpolated with little uncertainty then $$\alpha$$ may be increased to put less weight on the misfit. If the samples appear more like the prior samples than the data, $$\alpha$$ can be increased to put more weight on the misfit. To generate $$S$$ samples for a new $$\alpha$$ will require recomputing $$N^2$$ solves with $$(\alpha L + \lambda I)^{-\frac{1}{2}}$$ and $$NS$$ solves with $$(\alpha L + \lambda I)^{-\frac{1}{2}}$$.

5.3. Posterior optimal solution samples

Given the computation of posterior discrepancy samples, this subsection details their propagation through $$F_p(0)$$ to get optimal solution posterior samples. We propagate posterior samples through $$B$$ and then apply $$-H^{-1}$$ to the resulting vectors in $$\mathbb{R}^n$$. Using the decomposition $$\overline{\delta} + \tilde{\delta} + \hat{\delta}$$ of the samples, we demonstrate how to efficiently apply $$B$$ to each term.

$$B\overline{\delta} \in \mathbb{R}^n$$ (25) can be formed using data that was precomputed in the posterior discrepancy sampling, along with matrix-vector products consisting of $$\nabla_{u,\bar{z}}J$$ and $$S_\ell^T$$. This cost is dominated by the matrix-vector product with $$S_\ell^T$$ which involves a linear system solve in $$\mathbb{R}^m$$. 


12
To compute $B\hat{\theta}$ (26), we leverage the data from computing posterior discrepancy samples and observe that the additional cost to compute $B\hat{\theta}$ are matrix-vector products with $\nabla_u u J$ and $\bar{S}_m^T$ per sample.

Lastly, computing $B\hat{\theta}$ using (27) involves linear combinations of the vectors $\Gamma^{-\frac{1}{2}}\tilde{z}_k$ with coefficients

$$\nabla_u J\tilde{u}_k = \nabla_u JL^{-\frac{1}{2}}\nu_{N+k}.$$ 

With a naive implementation, this may be computationally intensive. However, the coefficients are normally distributed (scalar) random variables with mean 0 and variance $\nabla_u JL^{-1}\nabla_u J^T$. Accordingly, we consider the equivalent sample

$$\sqrt{\nabla_u JL^{-1}\nabla_u J^T}\Gamma^{-\frac{1}{2}}\sum_{k=1}^{n-N+1} c_k\tilde{z}_k$$

where $c_k, k = 1, 2, \ldots, n-N+1$, are i.i.d samples from a standard normal distribution. Recall that $\{\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_{n-N+1}\}$ is an orthonormal set of vectors which are orthogonal to $\mathcal{M} = \text{span}\{\Gamma^{-\frac{1}{2}}(z_{\ell} - \bar{z})\}_{\ell=2}^N$, and let $\{\tilde{z}_{1}^M, \tilde{z}_{2}^M, \ldots, \tilde{z}_{N-1}^M\}$ be an orthonormal basis for $\mathcal{M}$. Letting $P_{\mathcal{M}^\perp}$ be a projector onto $\mathcal{M}^\perp$, the orthogonal complement of $\mathcal{M}$, i.e. $P_{\mathcal{M}^\perp}\tilde{z}_k = \tilde{z}_k$ and $P_{\mathcal{M}^\perp}\tilde{z}_k^M = 0$, we sample $B\hat{\theta}$ as

$$\sqrt{\nabla_u JL^{-1}\nabla_u J^T}\Gamma^{-\frac{1}{2}}P_{\mathcal{M}^\perp}\left(\sum_{k=1}^{n-N+1} c_k\tilde{z}_k + \sum_{k=1}^{N-1} c_{n-N+1+k}\tilde{z}_k^M\right),$$

where $c_{n-N+1+k}$ are independent standard normal random variables. Since

$$\{\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_{n-N+1}\} \cup \{\tilde{z}_{1}^M, \tilde{z}_{2}^M, \ldots, \tilde{z}_{N-1}^M\}$$

is an orthonormal basis for $\mathbb{R}^n$, the sum over an orthonormal basis with i.i.d standard normal coefficients corresponds to rotating a Gaussian and does not change its distribution. Hence we may equivalently compute samples as

$$\sqrt{\nabla_u JL^{-1}\nabla_u J^T}\Gamma^{-\frac{1}{2}}P_{\mathcal{M}^\perp}\omega_n.$$
and the inversion of $\hat{\omega}_n$ is computing $\Gamma\omega_n$. Since $\{\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_{N+1}\}$ can be written in terms of $\{\Gamma^{-\frac{1}{2}}(z_t - \mathcal{Z})\}_{t=0}^{N}$, we use properties of orthogonal projectors to arrive at the expression to sample $B\omega$ as

$$\sqrt{\nabla_u JL^{-1}\nabla_u J^T} \left( I_n - \Gamma^{-1}\hat{Z} \left( \hat{Z}^T \Gamma^{-1}\hat{Z} \right)^{-1} \hat{Z}^T \right) \Gamma^{-\frac{1}{2}}\omega_n$$  \hspace{1cm} (28)

where

$$\hat{Z} = ((z_2 - \mathcal{Z}) \quad (z_3 - \mathcal{Z}) \ldots (z_N - \mathcal{Z})) \in \mathbb{R}^{n \times (N-1)}.$$

Since $\Gamma^{-1}z_t$, $t = 1, 2, \ldots, N$ has already been computed, we form $\Gamma^{-1}\hat{Z}$ with no additional solves, and the inversion of $\hat{Z}^T \Gamma^{-1}\hat{Z}$ is done on a $N \times N$ matrix with negligible cost. The only additional cost is computing $\Gamma^{-\frac{1}{2}}\omega_n$ for each sample. For problems where $n$ is large and $N$ is small, this provides a considerable computational savings relative to explicitly forming $\{\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_{N+1}\}$.

Sampling from the optimal solution posterior is summarized in Algorithm 3. The reduced cost is a result of considerable precomputed data from the prior and posterior discrepancy sampling.

**Algorithm 3** Compute optimal solution posterior samples.

1: Input: Number of desired samples $S$
2: Compute: $L^{-1}\nabla_u J^T$
3: $B\hat{\theta}$ using (25) by applying $\hat{S}_2^T \nabla_u J$ to precomputed data
4: $-H^{-1}B\theta$
5: for $s = 1, 2, \ldots, S$ do
6: Compute: $B\hat{\theta}$ using (26) by applying $\hat{S}_2^T \nabla_u J$ to precomputed data
7: $\Gamma^{-\frac{1}{2}}\omega_n$ for a sample $\Omega_n \sim \mathcal{N}(0, I_n)$
8: $B\theta$ using (28) with precomputed data
9: $-H^{-1}(B\theta + B\hat{\theta})$
10: end for
11: Number of linear solves: $L^{-1}$ $\Gamma^{-\frac{1}{2}}$ $\hat{S}_2^T$ $H^{-1}$
12: $S$ $S + 1$ $S + 1$

6. Numerical results

We demonstrate our proposed method on an illustrative 1D control example, where model discrepancy arises from a failure to include advection, and on an application of fluid flow in 2D, where Stokes is used as the low-fidelity approximation of Navier-Stokes.

6.1. Illustrative example

A 1D steady state control problem is formulated that seeks an optimal source to achieve a target state profile. The problem is formulated as

$$\min z \; \frac{1}{2} \int_0^1 (\hat{S}(z)(x) - T(x))^2 dx + \frac{\beta_1}{2} \int_0^1 z(x)^2 dx + \frac{\beta_2}{2} \int_0^1 z'(x)^2 dx$$

where $T(x) = 50 - 30(x - 0.5)^2$ is the target, $z: [0, 1] \to \mathbb{R}$ is a source controller, $\beta_1 = 10$ and $\beta_2 = 10^{-2}$ are regularization coefficients, and $\hat{S}(z)$ is the solution operator for the diffusion equation

$$-\kappa u'' = z \quad \text{on } (0, 1)$$

$$\kappa u' = hu \quad \text{on } \{0, 1\}$$
with Robin boundary condition. Our assumption is that the diffusion model is missing physics, specifically it lacks advection. We consider the high-fidelity model to be

\[-\kappa u'' + vu' = z \quad \text{on } (0, 1)\]
\[\kappa u' = hu \quad \text{on } \{0, 1\}\]

represented by solution operator \(S(z)\). We take parameter values \(\kappa = 1, v = 0.5, h = 2\).

**Optimization**

After discretizing with linear finite elements on a uniform mesh of 200 nodes, i.e. \(m = n = 200\), we solve both high and low-fidelity optimization problems to illustrate the effect of model error, which is attributed to the missing advection term. We observe a considerable difference in the optimal sources, shown in Figure 3. Since advection moves the state from left to right, including advection in the model results in a source skewed to the left.

**Prior model discrepancy**

To define the discrepancy prior, the \(z\) covariance matrix is taken as \(\Gamma = (D_z M^{-1} D_z)^{-1}\), where \(D_z\) is the discretization of the elliptic operator \(s_z(-\epsilon_z \Delta + I)\), \(s_z = 1\) and \(\epsilon_z = 10^{-5}\), equipped with zero Neumann boundary conditions, and \(M\) is the mass matrix arising from the finite element discretization. In a similar fashion, the prior state discrepancy is taken as a zero mean Gaussian with covariance \(L^{-1} = (D_u M^{-1} D_u)^{-1}\), where \(D_u\) is the discretization of the elliptic operator \(s_u(-\epsilon_u \Delta + I)\), \(s_u = 15\) and \(\epsilon_u = 5 \times 10^{-3}\), equipped with zero Neumann boundary conditions. Our choice of prior covariances are common in PDE-constrained Bayesian inverse problems because of the interpretability of their parameters, theoretical properties of the prior samples, and computational convenience of efficient linear solves and matrix factorizations for sampling. Executing Algorithm 1, we generate prior samples which are displayed in the left and center panels of Figure 2. These samples provide a visual interpretation of the prior to assist in choosing \(s_z, \epsilon_z, s_u, \) and \(\epsilon_u\), which play an important role defining length scales in the analysis.

**Posterior model discrepancy**

We evaluate the high-fidelity model twice to generate \(N = 2\) data points \(\{z_\ell, y_\ell\}_{\ell=1}^2\). Algorithm 2 is executed, with the noise covariance \(\alpha = 0.01\), to fit the model discrepancy. The right panel of Figure 2 displays posterior samples of \(\delta\) evaluated at \(z = z_1\). Its mean is given by the black line, samples are denoted using grey lines, and the discrepancy data \(y_1\) is shown by the broken red lines.

![Figure 2: Five samples from the prior source (left panel) and prior model discrepancy (center panel). The right panel shows the posterior distribution of \(\delta\) evaluated at \(z = z_1\). Its mean is given by the black line, samples are denoted using grey lines, and the discrepancy data \(y_1\) is shown by the broken red lines.](image)

**Posterior model discrepancy**

We evaluate the high-fidelity model twice to generate \(N = 2\) data points \(\{z_\ell, y_\ell\}_{\ell=1}^2\). Algorithm 2 is executed, with the noise covariance \(\alpha = 0.01\), to fit the model discrepancy. The right panel of Figure 2 displays posterior samples of \(\delta\) evaluated at \(z_1\). We observe that the mean matches the data well and the posterior samples are concentrated around the mean. This is unsurprising given that \(S(z) - \tilde{S}(z)\) is a linear function of \(z\) and \(\alpha\) was taken small enough to encourage trust in the data. As with the prior, plotting the posterior discrepancy enables visual analysis of the user specified noise covariance \(\alpha\). If \(\alpha\) is taken too small we would see negligible uncertainty in the posterior, whereas if \(\alpha\) is taken too large we would see the posterior discrepancy mean being close to zero as the inversion puts more weight on the prior than the data. Visualizing the posterior discrepancy
provides an efficient way to tune $\alpha$ before the more computationally intensive generation of optimal solution posterior samples.

**Posterior optimal solution**

Executing Algorithm 3 to propagate the posterior model discrepancy through the post-optimality sensitivity operator, we arrive at a posterior distribution for the optimal source as shown in Figure 3. We observe that the posterior mean for the optimal solution is much closer to the high-fidelity solution $z^*$ than our nominal low-fidelity solution $\tilde{z}$. Hence using the low-fidelity optimization problem along with two high-fidelity forward solves, we are able to find a good approximation of the high-fidelity optimal solution. Furthermore, the greatest errors (difference in the posterior mean and high-fidelity solution) occur near the boundaries and this error is captured by the wider range of posterior uncertainty in those regions.

![Figure 3: Posterior distribution of the optimal source $z$. The mean is given by the black line and samples are denoted using grey lines. The low and high-fidelity solutions $\tilde{z}$ and $z^*$ are shown by the blue and broken red lines, respectively.](image)

### 6.2. Control of viscous fluid flow

In this subsection the proposed approach is demonstrated on control of viscous fluid flow. In particular, we consider the high-fidelity model to be governed by the Navier-Stokes equation and the low-fidelity mode represented by the Stokes equation. The goal is find the optimal distributed controller $z = (z_x, z_y)$ that minimizes the vertical fluid flow. The low-fidelity optimization problem is

$$
\min_z \frac{1}{2} \int_\chi \tilde{v}_y(z)^2 + \frac{\beta_1}{2} \int_\Omega (z_x^2 + z_y^2) + \frac{\beta_2}{2} \int_\Omega (||\nabla z_x||^2 + ||\nabla z_y||^2)
$$

where $\tilde{v}(z) = (\tilde{v}_x(z), \tilde{v}_y(z))$ is the velocity solution operator for the Stokes equation

$$
- \mu \nabla v + \nabla p = g + z \quad \text{on } \Omega
$$

$$
\nabla \cdot v = 0 \quad \text{on } \Omega
$$

where $\Omega = (0,1)^2$, and $\chi = (0,1) \times (0,0.5)$. The flow is driven by the effect of gravity $g = (0, 9.81)$, viscosity $\mu = 0.5$, and the inflow boundary conditions

$$
v_x(0,y) = 6y(1-y) \quad \text{and} \quad v_y(x,1) = -2 \sin(2\pi x)^2.
$$

We solve the optimization problem with regularization coefficients $\beta_1 = \beta_2 = 10^{-5}$. The high-fidelity Navier-Stokes model

$$
- \mu \nabla v + (v \cdot \nabla)v + \nabla p = g + z \quad \text{on } \Omega
$$

$$
\nabla \cdot v = 0 \quad \text{on } \Omega
$$
includes the nonlinear convective term omitted in the Stokes equation.

**Optimization**

The three state variables $v_x$, $v_y$, and $p$ are discretized with Taylor-Hood elements yielding $m = 23003$ degrees of freedom. Figure 4 displays the uncontrolled and optimally controlled solutions of the Stokes equation. The inflow from the left and above, combined with the force of gravity, produce a flow field with high downward velocity in the lower region of the domain. The speed of this flow is minimized and the efficacy of the controller is apparent from the center column of Figure 4. However, the Navier-Stokes solution evaluated at the optimal controller exhibits faster flows as shown in the left panel of Figure 7.

![Figure 4: Uncontrolled (top) and controlled (bottom) state solution for the Stokes equation. The $x$-velocity $v_x$, $y$-velocity $v_y$, and pressure $p$ are plotted from left to right.](image)

**Prior model discrepancy**

Similar to the previous example, we define the prior covariances $\Gamma$ and $L^{-1}$ as the square of the inverse of elliptic differential operators. However, to mitigate the effect of high boundary variance in the prior we follow [14] by imposing a Robin boundary condition which is optimized to reduce the boundary effect. This condition is not sufficient to completely eliminate the inflated variance near the boundary, so we complement it with a rescaling of the covariance based on a point-wise variance estimate using 100 samples. The prior controller covariance is $\Gamma = s_z^2 \left( E^{-1}_z \left( D_z \left( D_z E^{-1}_z \right)^{-1} \right)^{-1} \right)$, where $D_z$ is the discretization of the elliptic operator $(-\epsilon_z \Delta + I)$, $s_z = 13$, and $\epsilon_z = 10^{-4}$, and $E_z$ is a diagonal matrix which approximates the point-wise variance of $(D_z E^{-1}_z D_z)^{-1}$, and $M_z$ is the mass matrix for the controller discretization. Similarly, $L^{-1} = s_u^2 \left( E_u^{-\frac{1}{2}} D_u M^{-1}_u D_u E_u^{-\frac{1}{2}} \right)^{-1}$, where $D_u$ is the discretization of the elliptic operator $(-\epsilon_u \Delta + I)$, $s_u = 1$, and $\epsilon_u = 0.2$, and $E_u$ is a diagonal matrix which approximates the point-wise variance of $(D_u M^{-1}_u D_u)^{-1}$, and $M_u$ is the mass matrix for the state discretization. We omit showing prior samples for conciseness, but emphasize that the interpretably of the elliptic operator allowed for easy specification of the variance and correlation lengths to ensure that the prior is reflective of the physical characteristics of $z$ and $\delta$.

To facilitate efficient computation, especially inversion of the shifted linear systems $\alpha L + \lambda_i I$ and its square root, we compute the generalized singular value decomposition of $D_u^{-1}$ in the $M_u$ and $E_u^{-1}$ inner product, which after algebraic manipulations gives the eigenvalue decomposition of $L^{-1}$. The subsequent linear solves involving $L$ may be efficiently approximated using the truncated
eigenvalue decomposition and the error in the approximation is controlled by the rank, which we take to be 1000, ensuring a relative truncation error of $O(10^{-3})$.

Posterior model discrepancy

We evaluate the Navier-Stokes equation at the optimal Stokes controller to generate the data point $(z_1, y_1)$ to fit the model discrepancy, i.e. $N = 1$. With a setting of $\alpha = 1.0$, we solve the Bayesian inverse problem to calibrate $\delta(z)$. Figure 5 displays the posterior mean of $\delta$ evaluated at the controller $z_1$. We observe that it matches the data well, as we expect. Posterior samples to characterize uncertainty were also generated, but are not presented for conciseness.

![Figure 5: Observed data (top) and mean fit (bottom) for the state discrepancy. The x-velocity $v_x$, y-velocity $v_y$, and pressure $p$ are plotted from left to right.](image)

Posterior optimal solution

Lastly, the mean and 100 samples from the posterior discrepancy are propagated through the post-optimality sensitivity operator to produce the posterior optimal solution. To illustrate the benefit of the controller update, Figure 6 displays the optimal controller generated by solving the Stokes problem, the mean of the optimal solution posterior, and the optimal controller generated by solving the control problem constrained by the Navier-Stokes equation. Although our assumption is that the latter controller is generally not available in practice, we compute and display it here for validation. By comparing across the rows of Figure 6, we see that calibrating the discrepancy with $N = 1$ high-fidelity forward solves significantly improves the controller, particularly in the case of the y-velocity controller in the left region of the domain.

To further quantify the benefit of the controller update, Figure 7 displays the vertical velocity $v_y$ of the Navier-Stokes solution when evaluated at the Stokes optimal controller (left), the updated optimal controller (center), and the Navier-Stokes optimal controller (right). The performance improvement is significant given that it was achieved using only $N = 1$ Navier-Stokes solve to update the Stokes solution. The comparison is further quantified in Table 1 which displays the value of the objective function, evaluated using the Navier-Stokes model, for each of the controllers. The updated controller decreased the value of the objective by two order of magnitude relative to the nominal Stokes control solution, while falling one order of magnitude short of the best possible value from the Navier-Stokes optimal control solution.

Lastly, with a recognition that computing the optimal controller update is underdetermined using only one high-fidelity solve, Figure 8 displays the posterior covariance of the optimal controller. Due to its high-dimensionality, we visualize the uncertainty by performing principle component analysis.
Figure 6: Controllers acting on the \(x\)-velocity component (top) and the \(y\)-velocity component (bottom). The left column is the nominal control solution, the center column is the mean of the posterior optimal solution, and the right column is the optimal solution computed by solving the control problem constrained by the Navier-Stokes model.

Figure 7: Vertical velocity state solutions corresponding to: Navier-Stokes solve with nominal control (left), Navier-Stokes solve with updated control (center), and Navier-Stokes with optimal control (right).

Table 1: Value of the objective function \(J(S(z), z)\), where \(S(z)\) is the Navier-Stokes solution operator.

| Controller | Objective Function Value |
|------------|--------------------------|
| \(\overline{z}\) | \(3.13 \times 10^7\) |
| \(\overline{z} + F_\theta'\theta(0)\overline{\theta}\) | \(1.39 \times 10^{-2}\) |
| \(z^\star\) | \(2.24 \times 10^{-3}\) |

(PCA) on the 100 posterior samples and plot the leading four principle components (or modes). Each row in Figure 8 corresponds to a principle component, with the left column showing a histogram of the coefficient corresponding to the given mode, and the center \((z_x)\) and right \((z_y)\) columns showing the modes. The black vertical line indicates the mean of the optimal solution posterior, which is always zero since the distribution is centered. The blue and red vertical lines indicate the coefficients of the nominal Stokes optimal controller and the Navier-Stokes optimal controller, respectively, when they are centered and projected onto the PCA mode. Ideally, the Navier-Stokes optimal controller (the red line) should be in a high probability region. This is true in the leading four modes shown in Figure 8, but is not true for all of the modes. Having the Navier-Stokes solution lie outside the high probably region for some modes is unsurprising given that the distribution was calibrated using only \(N = 1\) high-fidelity forward solve.
7. Conclusion

It is common that high-fidelity models are too computationally intensive and/or the code structures are not conducive to intrusive implementations of efficient optimization algorithms, for instance, it may be difficult to implement adjoints to enable efficient derivative computations. In such cases, current practice is to construct low-fidelity models through either simplifications of the equations from physics assumptions, or reduced order modeling techniques to approximate the high-fidelity model at a lower computational cost. The low-fidelity model is used to constrain an optimization problem and its solution is assessed by evaluating the high-fidelity model a small number of times. However, these high-fidelity solves may be used for more than assessing the quality of the optimal solution. This article provides a framework which systematically uses the high-fidelity data to improve the optimal solution, without requiring any additional access to the high-fidelity solver. We have demonstrated how the optimal solution can be improved with uncertainty characterization by coupling a Bayesian approach to model discrepancy estimation with post-optimality sensitivity analysis.

Our proposed approach is computationally scalable thanks to judicious manipulations of linear algebra to enable closed form expressions for posterior samples. These efficiencies were discovered by taking a wholistic perspective which considered every aspect of the problem formulation (discrepancy representation, Gaussian prior and likelihood, post-optimality sensitivity operator, and the resulting matrix factorizations) to develop a mathematically rigorous and computationally advantageous approach. The fundamental assumption is a linear approximation of the model discrepancy to optimal solution mapping. However, as we demonstrated, this can provide considerable improvements for nonlinear PDE-constrained optimization problems with high-dimensional optimization variables.

Since we consider a Bayesian inverse problem to characterize the model discrepancy in a limited data setting, the prior distribution plays an important role in our analysis. By leveraging developments in the infinite dimensional Bayesian inverse problems literature, we have proposed a prior on the discrepancy which is interpretable and computationally efficient. The algorithm requires some user specified parameters related to length scales, smoothness, and approximation errors; however, the analysis provides clear interpretations for how these choices influence the resulting posterior distribution.

There are many potential applications of our analysis including, but not limited too, (1) using models in three spatial dimensions to improve optimal solutions computed using models in two spatial dimensions, (2) using multi-scale models to improve solutions computed using homogenized models, (3) using coupled systems to improve optimal solutions computed using only a subset of the system, and (4) using high-fidelity data from controlled experiments in place of a high-fidelity model to improve design or control strategies. Other mathematical questions regarding how and when to query the high-fidelity model also remain and are the topic of ongoing research.

Acknowledgements

This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA-0003525. SAND2022-14144 O.

References

[1] Alexandrov, N., Jr., J.D., Lewis, R., Torczon, V., 1998. A trust-region framework for managing the use of approximation models in optimization. Structural Optimization 15, 16–23.

[2] Antil, H., Kouri, D.P., Lacasse, M., Ridzal, D. (Eds.), 2018. Frontiers in PDE-Constrained Optimization. Springer.
[3] Arendt, P.D., Apley, D.W., Chen, W., 2012. Quantification of model uncertainty: Calibration, model discrepancy, and identifiability. Journal of Mechanical Design 134.

[4] Ascher, U.M., Haber, E., 2001. Grid refinement and scaling for distributed parameter estimation problems. Inverse Problems 17, 571–590.

[5] Biegler, L., Biros, G., Ghattas, O., Heinkenschloss, M., Keyes, D., Mallick, B., Marzouk, Y., Tenorio, L., van Bloemen Waanders, B., Willcox, K. (Eds.), 2011. Large-Scale Inverse Problems and Quantification of Uncertainty. John Wiley and Sons.

[6] Biegler, L.T., Ghattas, O., Heinkenschloss, M., Keyes, D., van Bloemen Waanders, B. (Eds.), 2007. Real-Time PDE-Constrained Optimization. Volume 3. SIAM Computational Science and Engineering.

[7] Biros, G., Ghattas, O., 2005. Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Parts I-II. SIAM J. Sci. Comput. 27, 687–738.

[8] Bonnans, J.F., Shapiro, A., 1998. Optimization problems with perturbations: A guided tour. SIAM Review 40, 228–264.

[9] Borzi, A., 2007. High-order discretization and multigrid solution of elliptic nonlinear constrained optimal control problems. J. Comp. Applied Math 200, 67–85.

[10] Brandes, K., Griess, R., 2007. Quantitative stability analysis of optimal solutions in PDE-constrained optimization. Journal of Computational and Applied Mathematics 206, 908–926.

[11] Brynjarsdóttir, J., O’Hagan, A., 2014. Learning about physical parameters: the importance of model discrepancy. Inverse Problems 30.

[12] Bryson, D.E., Rumplkeil, M.P., 2018. Multifidelity quasi-newton method for design optimization. AIAA JOURNAL 56.

[13] Bui-Thanh, T., Ghattas, O., Martin, J., Stadler, G., 2013. A computational framework for infinite-dimensional Bayesian inverse problems. Part I: The linearized case, with applications to global seismic inversion. SIAM Journal on Scientific Computing 35, A2494–A2523.

[14] Daon, Y., Stadler, G., 2018. Mitigating the influence of the boundary on pde-based covariance operators. Inverse Problems and Imaging 12, 1083–1102.

[15] Fiacco, A.V., Ghaemi, A., 1982. Sensitivity analysis of a nonlinear structural design problem. Computers and Operations Research 9, 29–55.

[16] Gardner, P., Rogers, T., Lord, C., Bar thorpe, R., 2021. Learning model discrepancy: A gaussian process and sampling-based approach. Mechanical Systems and Signal Processing 152.

[17] Gorodetsky, A.A., Geraci, G., Eldred, M.S., Jakeman, J.D., 2020. A generalized approximate control variate framework for multifidelity uncertainty quantification. Journal of Computational Physics 408.

[18] Griess, R., 2004a. Parametric sensitivity analysis in optimal control of a reaction-diffusion system – part II: practical methods and examples. Optimization Methods and Software 19, 217–242.

[19] Griess, R., 2004b. Parametric sensitivity analysis in optimal control of a reaction diffusion system. I. solution differentiability. Numerical Functional Analysis and Optimization 25, 93–117.

[20] Haber, E., Ascher, U.M., 2001. Preconditioned all-at-once methods for large, sparse parameter estimation problems. Inverse Problems 17, 1847–1864.
[21] Hart, J., van Bloemen Waanders, B., 2021. Enabling hyper-differential sensitivity analysis for ill-posed inverse problems. arXiv:2106.11813.

[22] Hart, J., van Bloemen Waanders, B., 2022. Hyper-differential sensitivity analysis with respect to model discrepancy: Mathematics and computation. arXiv:2210.09037.

[23] Hart, J., van Bloemen Waanders, B., Hertzog, R., 2020. Hyper-differential sensitivity analysis of uncertain parameters in PDE-constrained optimization. International Journal for Uncertainty Quantification 10, 225–248.

[24] Hart, J., van Bloemen Waanders, B., Hood, L., Parish, J., 2022. Sensitivity driven experimental design to facilitate control of dynamical systems. arXiv:2202.03312.

[25] Hazra, S.B., Schulz, V., 2006. Simultaneous pseudo-timestepping for aerodynamic shape optimization problems with state constraints. SIAM J. Sci. Comput. 28, 1078–1099.

[26] Higdon, D., Gattiker, J., Williams, B., Rightley, M., 2008. Computer model calibration using high-dimensional output. Journal of the American Statistical Association 103, 570–583.

[27] Hintermuller, M., Vicente, L.N., 2005. Space mapping for optimal control of partial differential equations. SIAM J. Opt. 15, 1002–1025.

[28] Hinze, M., Pinnau, R., Ulbrich, M., Ulbrich, S., 2009. Optimization with PDE Constraints. Springer.

[29] Kennedy, M.C., O’Hagan, A., 2001. Bayesian calibration of computer models. Journal of the Royal Statistical Society 63, 425–464.

[30] L. T. Biegler, O. Ghattas, M.H., van Bloemen Waanders, B. (Eds.), 2003. Large-Scale PDE-Constrained Optimization. volume 30. Springer-Verlag Lecture Notes in Computational Science and Engineering.

[31] Laird, C.D., Biegler, L.T., van Bloemen Waanders, B., Bartlett, R.A., 2005. Time dependent contaminant source determination for municipal water networks using large scale optimization. ASCE J. Water Res. Mgt. Plan., 125–134.

[32] Ling, Y., Mullins, J., Mahadevan, S., 2014. Selection of model discrepancy priors in bayesian calibration. Journal of Computational Physics 276, 665–680.

[33] Maupin, K.A., Swiler, L.P., 2020. Model discrepancy calibration across experimental settings. Reliability Engineering & System Safety 200.

[34] Menhorn, F., Geraci, G., Seidl, D.T., Eldred, M.S., King, R., Bungartz, H.J., Marzouk, Y., 2020. Higher moment multilevel estimators for optimization under uncertainty applied to wind plant design, AIAA SciTech Forum.

[35] Ng, L.W., Willcox, K.E., 2014. Multifidelity approaches for optimization under uncertainty. International Journal for Numerical Methods in Engineering.

[36] Oakley, J., O’Hagan, A., 2002. Bayesian inference for the uncertainty distribution of computer model outputs. Biometrika 89, 769–784.

[37] Peherstorfer, B., Willcox, K., Gunzburger, M., 2018. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. SIAM Review 60, 550–591.

[38] Petra, N., Martin, J., Stadler, G., Ghattas, O., 2014. A computational framework for infinite-dimensional bayesian inverse problems. Part II: Stochastic newton mcmc with application to ice sheet flow inverse problems. SIAM Journal on Scientific Computing 36, A1525–A1555.
[39] Saibaba, A.K., Hart, J., van Bloemen Waanders, B., 2021. Randomized algorithms for generalized singular value decomposition with application to sensitivity analysis. Numerical Linear Algebra with Applications.

[40] Stuart, A.M., 2010. Inverse problems: a Bayesian perspective. Acta Numerica.

[41] Sunseri, I., Alexanderian, A., Hart, J., van Bloemen Waanders, B., 2022. Hyper-differential sensitivity analysis for nonlinear bayesian inverse problems. arXiv:2202.02219.

[42] Sunseri, I., Hart, J., van Bloemen Waanders, B., Alexanderian, A., 2020. Hyper-differential sensitivity analysis for inverse problems constrained by partial differential equations. Inverse Problems 36.

[43] Vogel, C.R., 1999. Sparse matrix computations arising in distributed parameter identification. SIAM J. Matrix Anal. Appl., 1027–1037.

[44] Vogel, C.R., 2002. Computational Methods for Inverse Problems. SIAM Frontiers in Applied Mathematics Series.
Figure 8: Representation of uncertainty in the optimal solution posterior. Principal component analysis was performed on 100 posterior optimal solution samples. The four leading principal components (or modes) are shown in the rows. The first four modes capture approximately 84% of the posterior variance. The left column is a histogram showing the magnitude of the projection of the samples (minus the posterior mean) onto the mode. The projection of the nominal, updated, and high-fidelity controllers (minus the posterior mean) are indicated by the vertical lines. The projection of the updated controller equals zero for all modes because the data was centered around it.