hIPPYlib: AN EXTENSIBLE SOFTWARE FRAMEWORK FOR LARGE-SCALE INVERSE PROBLEMS GOVERNED BY PDES; PART I: DETERMINISTIC INVERSION AND LINEARIZED BAYESIAN INFERENCE ∗

UMBERTO VILLA∗, NOEMI PETRA†, AND OMAR Ghattas‡

Abstract. We present an extensible software framework, hIPPYlib, for solution of large-scale deterministic and Bayesian inverse problems governed by partial differential equations (PDEs) with (possibly) infinite-dimensional parameter fields (which are high-dimensional after discretization). hIPPYlib overcomes the prohibitive nature of Bayesian inversion for this class of problems by implementing state-of-the-art scalable algorithms for PDE-based inverse problems that exploit the structure of the underlying operators, notably the Hessian of the log-posterior. The key property of the algorithms implemented in hIPPYlib is that the solution of the deterministic and linearized Bayesian inverse problem is computed at a cost, measured in linearized forward PDE solves, that is independent of the parameter dimension. The mean of the posterior is approximated by the MAP point, which is found by minimizing the negative log-posterior. This deterministic nonlinear least-squares optimization problem is solved with an inexact matrix-free Newton-CG method. The posterior covariance is approximated by the inverse of the Hessian of the negative log posterior evaluated at the MAP point. This Gaussian approximation is exact when the parameter-to-observable map is linear; otherwise, its logarithm agrees to two derivatives with the log-posterior at the MAP point, and thus it can serve as a proposal for Hessian-based MCMC methods. The construction of the posterior covariance is made tractable by invoking a low-rank approximation of the Hessian of the log-likelihood. Scalable tools for sample generation are also implemented. hIPPYlib makes all of these advanced algorithms easily accessible to domain scientists and provides an environment that expedites the development of new algorithms. hIPPYlib is also a teaching tool that can be used to educate researchers and practitioners who are new to inverse problems and the Bayesian inference framework.

Key words. Infinite-dimensional inverse problems, adjoint-based methods, inexact Newton-CG method, low-rank approximation, Bayesian inference, uncertainty quantification, sampling, generic PDE toolkit

AMS subject classifications. 35Q62, 62F15, 35R30, 35Q93, 65C60, 65K10, 49M15, 49M37, 68N99

1. Introduction. Recent years have seen tremendous growth in the volumes of observational and experimental data that are being collected, stored, processed, and analyzed. The central question that has emerged is how do we extract knowledge and insight from all of this data? When the data correspond to observations of (natural or engineered) systems, and these systems can be represented by mathematical models, this knowledge-from-data problem is fundamentally a mathematical inverse problem.

∗Electrical & Systems Engineering, Washington University in St. Louis, St. Louis, MO, USA. Email: uvilla@wustl.edu.
†Applied Mathematics, School of Natural Sciences, University of California, Merced, CA, USA. Email: npetra@ucmerced.edu.
‡Oden Institute for Computational Engineering & Sciences, Department of Mechanical Engineering, and Department of Geological Sciences, The University of Texas at Austin, Austin, TX, USA. Email: omar@ices.utexas.edu.

This work was supported by the U.S. National Science Foundation, Software Infrastructure for Sustained Innovation (SI2: SSE & SSI) Program under grants ACI-1550593, and ACI-1550547; the Defense Advanced Research Projects Agency, Enabling Quantification of Uncertainty in Physical Systems Program under grant W911NF-15-2-0121; the US National Science Foundation, Division Of Chemical, Bioengineering, Environmental, & Transport Systems under grant CBET-1508713; and the Air Force Office of Scientific Research, Computational Mathematics program under grant FA9550-12-1-81243.
That is, given (possibly noisy) data and (a possibly uncertain) model, infer parameters that characterize the model. Inverse problems abound in all areas of science, engineering, technology, and medicine. As just a few examples of model-based inverse problems, we may infer: the initial condition in a time-dependent partial differential equation (PDE) model, a coefficient field in a subsurface flow model, the ice sheet basal friction field from satellite observations of surface flow, the earth structure from reflected seismic waves, subsurface contaminant plume spread from crosswell electromagnetic measurements, internal structural defects from measurements of structural vibrations, ocean state from surface temperature observations, and so on.

Typically, inverse problems are *ill-posed* and suffer from non-unique solutions; simply put, the data—even when they are large-scale—do not provide sufficient information to fully determine the model parameters. This is the usual case with PDE models that have parameters representing fields such as boundary conditions, initial conditions, source terms, or heterogeneous coefficients. Non-uniqueness can stem from noise in the data or model, from sparsity of the data, from smoothing properties of the map from input model parameters to output observables or from its nonlinearity, or from intrinsic redundancy in the data. In such cases, *uncertainty is a fundamental feature of the inverse problem*. Therefore, not only do we wish to infer the parameters, but we must also quantify the uncertainty associated with this inference, reflecting the degree of “confidence” we have in the solution.

Methods that facilitate the solution of Bayesian inverse problems governed by complex PDE models require a diverse and advanced background in applied mathematics, scientific computing, and statistics to understand and implement, e.g., Bayesian inverse theory, computational statistics, inverse problems in function space, adjoint-based first- and second-order sensitivity analysis, and variational discretization methods. In addition, to be efficient these methods generally require first and second derivative (of output observables with respect to input parameters) information from the underlying forward PDE model, which can be cumbersome to derive. In this paper, we present hIPPYlib, an Inverse Problems Python library (hIPPYlib), an extensible software framework aimed at overcoming these challenges and providing capabilities for additional algorithmic developments for large-scale deterministic and Bayesian inversion.

hIPPYlib builds on FEniCS (a parallel finite element element library) [46] for the discretization of the PDEs, and on PETSc [5] for scalable and efficient linear algebra operations and solvers. Hence, it is easily applicable to medium to large-scale problems. One of the main features of this library is that it exposes specific aspects from the model setup to the inverse solution, which can be useful not only for research purposes but also for learning and teaching. In the hIPPYlib examples, we show how to handle various PDE models and boundary conditions, and illustrate how to implement prior and log-likelihood terms for the Bayesian inference. hIPPYlib is implemented in a mixture of C++ and Python and has been released under the GNU General Public License version 2 (GPL). The source codes can be downloaded from https://hippylib.github.io. Below we summarize the main algorithmic and software contributions of hIPPYlib.

*Algorithmic contributions.*

1. A single pass randomized eigensolver for generalized symmetric eigenproblems that is more robust than the one proposed in [60].
2. A new scalable sampling algorithm for Gaussian random fields that exploits the structure of the given covariance operator. This extends the approach
proposed in [23] to covariance operators defined as the inverse of second order differential operators as opposed to the identity operator.

3. A scalable algorithm to estimate the pointwise variance of Gaussian random fields using randomized eigensolvers. For the same computational cost this algorithm allows for more accurate estimates than the stochastic estimator proposed in [9]. Our method drastically reduces the variance of the estimator at a cost of introducing a small bias.

Software contributions.

1. A modular approach to define complex inverse problems governed by (possibly nonlinear or time-dependent) PDEs. hIPPYlib automates the computation of higher order derivatives of the parameter-to-observable map for forward models and observation processes defined by the user using FEniCS.

2. Implementation of adjoints and Hessian actions needed to solve the deterministic inverse problem and to compute the maximum a posteriori (MAP) point of the Bayesian inverse problem. In addition, to test gradients and the Hessian action, hIPPYlib incorporates finite difference tests, which is an essential component of the verification process.

3. A robust implementation of the inexact Newton-conjugate gradient (Newton-CG) algorithm together with line search algorithms to guarantee global convergence of the optimizer.

4. Implementation of randomized algorithms to compute the low-rank factorization of the misfit part of the Hessian.

5. Scalable algorithms to construct and evaluate the Laplace approximation of the posterior.

6. Sampling capabilities to generate realizations of Gaussian random fields with a prescribed covariance operator.

7. An estimation of the pointwise variance of the prior distribution and Laplace approximation to the posterior.

Numerous toolkits and libraries for finite element computations based on variational forms are available, for instance COMSOL Multiphysics [22], deal.II [6], dune [7], FEniCS [46, 44], and Sundance, a package from Trilinos [38]. While these toolkits are usually tailored towards the solution of PDEs and systems of PDEs, they cannot be used straightforwardly for the solution of inverse problems with PDEs. However, several of them are sufficiently flexible to be extended for the solution of inverse problems governed by PDEs. Nevertheless, some knowledge of the structure underlying these packages is required since the optimality systems arising in inverse problems with PDEs often cannot be solved using generic PDE solvers, which do not exploit the optimization structure of the inverse problems. In [33] the authors present dolfin-adjoint, a project that also builds on FEniCS and derives discrete adjoints from a forward model written in the Python interface to dolfin using a combination of symbolic and automatic differentiation. While dolfin-adjoint could be used to solve deterministic inverse problems, it lacks the framework for Bayesian inversion. In addition, we avoid using the adjoint capabilities of dolfin-adjoint since this does not allow the user to have full control over the construction of derivatives. In [59] the authors present jInv, a flexible parallel software for parameter estimation with PDE forward models. The main limitations of this software are that it is restricted to deterministic inversion and that the user needs to provide the discretization for both the forward and adjoint problems. Finally, the Rapid Optimization Library, ROL [43], is a flexible and robust optimization package in Trilinos for the solution of
optimal design, optimal control and deterministic inverse problems in large-scale engineering applications. ROL implements state-of-the-art algorithms for unconstrained optimization, constrained optimization and optimization under uncertainty, and exposes an interface specific for optimization problems with PDE constraints. The main limitation is that the user has to interface with other software packages for the definition and implementation of the forward and adjoint problems. There also exist several general purpose libraries addressing uncertainty quantification (UQ) and Bayesian inverse problems. Among the most prominent we mention QUESO [52, 58], DAKOTA [30, 1], PSUADE [70]. All of these libraries provide Bayesian inversion capabilities, but the underlying methods do not fully exploit the structure of the problem or make use of derivatives and as such are not intended for high-dimensional problems. Finally, MUQ [54] provides powerful Bayesian inversion models and algorithms, but expects forward models to come equipped with gradients/Hessians to permit large-scale solution.

In summary, to the best of our knowledge, there is no available software (open-source or otherwise) that provides all the discretization, optimization and statistical tools to enable scalable and efficient solution of deterministic and Bayesian inverse problems governed by complex PDE forward models. hIPPYlib is the first software framework that allows to tackle this specific class of inverse problems by facilitating the construction of forward PDE models equipped with adjoint/derivative information, providing state-of-the-art scalable optimization algorithms for the solution of the deterministic inverse problem and/or MAP point computation, and integrating tools for characterizing the posterior distribution.

The paper is structured as follows. Section 2 gives a brief overview of the deterministic and Bayesian formulation of inverse problems in an infinite-dimensional Hilbert space setting, and addresses the discretization of the underlying PDEs using the finite element method. Section 3 contains an overview of the design of the hIPPYlib software and of its components. Section 4 provides a detailed description of the algorithms implemented in hIPPYlib to solve the deterministic and linearized Bayesian inverse problem, namely the inexact Newton-CG algorithm, the single and double pass randomized algorithms for the solution of generalized hermitian eigenproblems, scalable sampling techniques for Gaussian random fields, and stochastic algorithms to approximate the pointwise variance of the prior and posterior distributions. Section 5 demonstrates hIPPYlib’s capabilities for deterministic and linearized Bayesian inversion by solving two representative inverse problems: inversion for the coefficient field in an elliptic PDE model and for the initial condition in an advection-diffusion PDE model. Last, Section 6 contains our concluding remarks.

2. Infinite-dimensional deterministic and Bayesian inverse problems in hIPPYlib. In what follows, we provide a brief account of the deterministic [31, 72] and Bayesian formulation [67, 42] of inverse problems. Specifically, we adopt infinite-dimensional Bayesian inference framework [65], and we refer to [55, 3, 4, 15] for elaborations associated with discretization issues.

2.1. Deterministic inverse problems governed by PDEs. The inverse problem consists of using available observations \(d\) to infer the values of the unknown parameter field\(^3\) \(m\) that characterize a physical process modeled by PDEs. Mathematically

\(^3\) hIPPYlib also supports deterministic and Bayesian inversion for a finite-dimensional set of parameters, however, for ease of notation, in the present work we only present the infinite-dimensional case.
this inverse relationship is expressed as
\[ d = \mathcal{F}(m) + \eta, \]  
(2.1)
where the map \( \mathcal{F} : \mathcal{M} \rightarrow \mathbb{R}^q \) is the so-called parameter-to-observable map. This mapping can be linear or nonlinear. In the applications targeted in hIPPYlib, \( \mathcal{M} \subseteq L^2(\mathcal{D}) \), where \( \mathcal{D} \subseteq \mathbb{R}^d \) is a bounded domain, and evaluations of \( \mathcal{F} \) involve the solution of a PDE given \( m \), followed by the application of an observation operator to extract the observations from the state. That is, introducing the state variable \( u \in \mathcal{V} \) for a suitable Hilbert space \( \mathcal{V} \) of functions defined on \( \mathcal{D} \), the map \( \mathcal{F} \) is defined as
\[ \mathcal{F}(m) = B(u), \) s.t. \( r(u, m) = 0, \]  
(2.2)
where \( B : \mathcal{V} \rightarrow \mathbb{R}^q \) is a (possibly nonlinear) observation operator, and \( r : \mathcal{V} \times \mathcal{M} \rightarrow \mathcal{V}^\ast \)—referred as the forward problem from now on—represents the PDE problem. The observations \( d \) contain noise due to measurement uncertainties and model errors [67]. In (2.1), this is captured by the additive noise \( \eta \), which in hIPPYlib is modeled as \( \eta \sim \mathcal{N}(0, \Gamma_{\text{noise}}) \), i.e., a centered Gaussian at \( 0 \) with covariance \( \Gamma_{\text{noise}} \). A significant difficulty when solving infinite-dimensional inverse problems is that typically these are not well-posed (in the sense of Hadamard [69]). To overcome the difficulties due to ill-posedness, we regularize the problem, i.e., we include additional assumptions on the solution, such as smoothness. The deterministic inverse problems in hIPPYlib are regularized via Tikhonov regularization, which penalizes oscillatory components of the parameter \( m \), thus restricting the solution to smoothly varying fields [31, 72].

A deterministic inverse problem is therefore formulated as follows: given finite-dimensional noisy observations \( d \in \mathbb{R}^q \), one seeks to find the unknown parameter field \( m \) that best reproduces the observations. Mathematically this translates into the following nonlinear least-squares minimization problem
\[
\min_{m \in \mathcal{M}} J(m) := \frac{1}{2} \left\| \mathcal{F}(m) - d \right\|_{\Gamma_{\text{noise}}^{-1}}^2 + R(m),
\]
(2.3)
where the first term in the cost functional, \( J(m) \), represents the misfit between the observations, \( d \), and that predicted by the parameter-to-observable map \( \mathcal{F}(m) \), weighted by the inverse noise covariance \( \Gamma_{\text{noise}}^{-1} \). The regularization term, \( R(m) \), imposes regularity on the inversion field \( m \), such as smoothness. As explained above, in the absence of such a term, the inverse problem is ill-posed, i.e., its solution is not unique and is highly sensitive to errors in the observations [31, 72].

As we will explain in Section 4.1, to efficiently solve the nonlinear least-squares problem (2.3) with parameter-to-observable-map \( \mathcal{F} \) implicitly defined as in (2.2), first and second derivative information are needed. Using the Lagragian formalism [71], abstract expressions for the gradient and Hessian action are obtained below, and we refer to Section 5 for concrete examples. To this aim, we introduce an auxiliary variable \( p \in \mathcal{V} \)—from here on referred as the adjoint—and write the Lagragian functional
\[ \mathcal{L}^G(u, m, p) := \frac{1}{2} \left\| B(u) - d \right\|_{\Gamma_{\text{noise}}^{-1}}^2 + R(m) + \langle p, r(u, m) \rangle_{\mathcal{V}^\ast}, \]
where \( \langle \cdot, \cdot \rangle_{\mathcal{V}^\ast} \) denotes the duality pair between \( \mathcal{V} \) and its adjoint. The gradient for the cost functional (2.3) in an arbitrary direction \( \tilde{m} \in \mathcal{M} \) evaluated at \( m = m_0 \in \mathcal{M} \) is the Gâteaux derivative of \( \mathcal{L} \) with respect to \( m \), and reads
\[ (\mathcal{G}(m_0), \tilde{m}) = (R_m(m_0), \tilde{m}) + \langle p_0, [r_m(u_0, m_0)] \tilde{m} \rangle_{\mathcal{V}^\ast}, \]  
(2.4)
for all \( \tilde{m} \in \mathcal{M} \).
where \( \mathcal{R}_m(m_0), \hat{m} \) denotes the Gâteaux derivative of \( \mathcal{R} \) with respect to \( m \) in the direction \( \hat{m} \) evaluated at \( m = m_0 \), and \( [r_m(u_0, m_0)] \hat{m} \in \mathcal{V}^\ast \) the Gâteaux derivative of \( r \) with respect to \( m \) in the direction \( \hat{m} \) evaluated at \( u = u_0, m = m_0 \). Here \( u_0, p_0 \) are obtained by setting to zero the derivatives of \( \mathcal{L} \) with respect to \( p \) and \( u \); specifically, \( u_0 \) solves the forward problem
\[
\mathcal{V}(\hat{p}, r(u_0, m_0))_{\mathcal{V}^\ast} = 0 \quad \forall \hat{p} \in \mathcal{V},
\]
and \( p_0 \) solves the adjoint problem
\[
\mathcal{V}(p, [r_u(u_0, m_0)]\hat{u})_{\mathcal{V}^\ast} + \langle [\mathcal{B}_u(u_0)]\hat{u}, \mathcal{B}(u) - d \rangle_{\mathcal{V}^\ast} = 0, \quad \forall \hat{u} \in \mathcal{V}.
\]

In a similar way, to derive the expression for the Hessian action in an arbitrary direction \( \hat{m} \in \mathcal{M} \) we introduce the second order Lagrangian functional
\[
\mathcal{L}^H(u, m, p; \hat{u}, \hat{m}, \hat{p}) := \langle \mathcal{G}(m), \hat{m} \rangle + \mathcal{V}(\hat{p}, r(u, m))_{\mathcal{V}^\ast} + \langle [\mathcal{B}_u(u)]\hat{u}, \mathcal{B}(u) - d \rangle_{\mathcal{V}^\ast},
\]
where the first term is the gradient expression, the second term stems from the forward problem, and the last two terms represent the adjoint problem. Then, the action of the Hessian in a direction \( \hat{m} \in \mathcal{M} \) evaluated at \( m = m_0 \) is the variation of \( \mathcal{L}^H \) with respect to \( m \) and reads
\[
(\hat{m}, \mathcal{H}(m_0)\hat{m}) = \langle \hat{m}, [\mathcal{R}_{mm}(m_0)]\hat{m} \rangle + \langle \hat{m}, [p_0 r_{mm}(u_0, m_0)]\hat{m} \rangle + \mathcal{V}(\hat{p}, [r_u(u_0, m_0)]\hat{u})_{\mathcal{V}^\ast} + \mathcal{V}(p_0, [r_u(u_0, m_0)]\hat{u}, \hat{m})_{\mathcal{V}^\ast}, \quad \forall m \in \mathcal{M}.
\]
Here \( u_0, p_0 \) are the solution of the forward and adjoint problems (2.5) and (2.6), respectively. The incremental state \( \hat{u} \) and incremental adjoint \( \hat{p} \) solve the so-called incremental forward and incremental adjoint problems, which are obtained by setting to zero variations of \( \mathcal{L}^H \) with respect to \( p \) and \( u \), respectively. In Section 5.1.2, we present a Newton-type algorithm to minimize (2.3) that uses the expression for the gradient (2.4) and Hessian action (2.8) derived here.

Finally, we note that the solution of a deterministic inverse problem based on regularization is a point estimate of \( m \), which solves (2.1) in a least-squares sense. A systematic integration of the prior information on the model parameters and uncertainties associated with the observations can be achieved using a probabilistic point of view, where the prior information and noise model are represented by probability distributions. In the following section, we describe the probabilistic formulation of the inverse problem via a Bayesian framework, whose solution is a posterior probability distribution for \( m \).

### 2.2. Bayesian inversion in infinite dimensions.

In the Bayesian formulation in infinite dimensions, we state the inverse problem as a problem of statistical inference over the space of uncertain parameters, which are to be inferred from data and a physical model. In this setup, in contrast to the finite-dimensional case, there is no Lebesgue measure on \( \mathcal{M} \), the infinite-dimensional Bayes formula is given by
\[
\frac{d\mu_{\text{post}}}{d\mu_{\text{prior}}} \propto \pi_{\text{like}}(d|m).
\]
Here, \( d\mu_{\text{post}}/d\mu_{\text{prior}} \) denotes the Radon-Nikodym derivative of the posterior measure \( \mu_{\text{post}} \) with respect to \( \mu_{\text{prior}} \), and \( \pi_{\text{like}}(d|m) \) denotes the data likelihood. Conditions
The noise model and the likelihood. In our hIPPYlib framework, we assume an additive noise model, \( \mathbf{d} = \mathcal{F}(\mathbf{m}) + \mathbf{\eta} \), where \( \mathbf{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma}_{\text{noise}}) \) is a centered Gaussian on \( \mathbb{R}^q \). This implies

\[
\pi_{\text{like}}(\mathbf{d}|\mathbf{m}) \propto \exp\left\{ -\Phi(\mathbf{m}) \right\},
\]

where \( \Phi(\mathbf{m}) = \frac{1}{2} \| \mathcal{F}(\mathbf{m}) - \mathbf{d} \|^2_{\mathbf{\Gamma}_{\text{noise}}^{-1}} \) denotes the negative log-likelihood.

The prior. For many problems, it is reasonable to choose the prior to be Gaussian, i.e., \( \mathbf{m} \sim \mathcal{N}(\mathbf{m}_{\text{pr}}, \mathbf{C}_{\text{prior}}) \). This implies

\[
d\mu_{\text{prior}}(\mathbf{m}) \propto \exp\left\{ -\frac{1}{2} \| \mathbf{m} - \mathbf{m}_{\text{pr}} \|^2_{\mathbf{C}_{\text{prior}}^{-1}} \right\}.
\]

If the parameter represents a spatially correlated field defined on \( \mathcal{D} \subseteq \mathbb{R}^d \), the prior covariance operator \( \mathbf{C}_{\text{prior}} \) usually imposes smoothness on the parameter. This is because rough components of the parameter field are typically cannot be inferred from the data, and must be determined by the prior to result in a well-posed Bayesian inverse problem.

In hIPPYlib we use elliptic PDE operators to construct the prior covariance, which allows us to capitalize on fast, optimal complexity solvers. More precisely, the prior covariance operator is the inverse of the \( n \)-th power of a Laplacian-like operator, namely \( \mathbf{C}_{\text{prior}} := \mathcal{A}^{-n} = (-\gamma \Delta + \delta I)^{-n} \), where \( \gamma \), and \( \delta > 0 \) control the correlation length \( \rho \) and the pointwise variance \( \sigma^2 \) of the prior operator. Specifically, \( \rho \)–empirically defined as the distance \( \rho \) for which the two-points correlation coefficient is 0.1–is proportional to \( \sqrt{\gamma/\delta} \), and \( \sigma^2 \) is proportional to \( \delta^{-n} \rho^{-d} \) (see e.g. [45] where exact expressions for \( \rho \) and \( \sigma^2 \) as functions of \( \gamma \) and \( \delta \) are derived under the assumption of unbounded domain \( \mathcal{D} \) and constant coefficients \( \gamma \) and \( \delta \)). The coefficients \( \gamma \) and \( \delta \) can be constant (in which case the prior is stationary) or spatially varying. In addition, one can consider an anisotropic diffusion operator \( \mathcal{A} = -\gamma \nabla \cdot (\mathbf{\Theta} \nabla) + \delta I \), with \( \mathbf{\Theta} \) a symmetric positive definite (s.p.d.) tensor that models, for instance, stronger correlations in a specific direction. These choices of prior ensure that \( \mathbf{C}_{\text{prior}} \) is a trace-class operator, guaranteeing bounded pointwise variance and a well-posed infinite-dimensional Bayesian inverse problem [65, 15].

The posterior. Using the expression for the likelihood function (2.10) and prior distribution (2.11), the posterior distribution in (2.9) reads

\[
d\mu_{\text{post}} \propto \exp\left\{ -\frac{1}{2} \| \mathcal{F}(\mathbf{m}) - \mathbf{d} \|^2_{\mathbf{\Gamma}_{\text{noise}}^{-1}} - \frac{1}{2} \| \mathbf{m} - \mathbf{m}_{\text{pr}} \|^2_{\mathbf{C}_{\text{prior}}^{-1}} \right\}.
\]

The maximum a posteriori (MAP) point \( \mathbf{m}_{\text{MAP}} \) is defined as the parameter field that maximizes the posterior distribution. It can be obtained by solving the following deterministic optimization problem

\[
\mathbf{m}_{\text{MAP}} := \arg\min_{\mathbf{m} \in \mathcal{M}} \left( -\log d\mu_{\text{post}}(\mathbf{m}) \right) = \arg\min_{\mathbf{m} \in \mathcal{M}} \frac{1}{2} \| \mathcal{F}(\mathbf{m}) - \mathbf{d} \|^2_{\mathbf{\Gamma}_{\text{noise}}^{-1}} + \frac{1}{2} \| \mathbf{m} - \mathbf{m}_{\text{pr}} \|^2_{\mathbf{C}_{\text{prior}}^{-1}}.
\]

We note that, the prior information plays the role of Tikhonov regularization in (2.3); in fact the deterministic optimization problem (2.3) is the same as (2.13) for the
choice $\mathbb{R}(m) = \frac{1}{2} \| m - m_{\text{pr}} \|^2_{C^{-1}_{\text{prior}}}$ . The Hessian $\mathcal{H}(m_{\text{MAP}})$ of the negative log-posterior evaluated at $m_{\text{MAP}}$ plays a fundamental role in quantifying the uncertainty in the inferred parameter. In particular, this indicates which directions in the parameter space are most informed by the data [15]. We note that when $\mathcal{F}$ is linear, due to the particular choice of prior and noise model, the posterior measure is Gaussian, $\mathcal{N}(m_{\text{MAP}}, C_{\text{post}})$ with [65, Section 6.4],

$$C_{\text{post}} = \mathcal{H}^{-1} = (\mathcal{F}^* \Gamma_{\text{noise}}^{-1} \mathcal{F} + C_{\text{prior}}^{-1})^{-1}, \quad m_{\text{MAP}} = C_{\text{post}} (\mathcal{F}^* \Gamma_{\text{noise}}^{-1} d + C_{\text{prior}}^{-1} m_{\text{pr}}), \quad (2.14)$$

where $\mathcal{F}^* : \mathbb{R}^d \rightarrow \mathcal{M}$ is the adjoint of $\mathcal{F}$.

In the general case of nonlinear parameter-to-observable map $\mathcal{F}$ the posterior distribution is not Gaussian. However, under certain assumptions on the noise covariance $\Gamma_{\text{noise}}$, the number $q$ of observations, and the regularity of the parameter-to-observable map $\mathcal{F}$, the Laplace approximation [63, 74, 32, 57, 68] can be invoked to estimate posterior expectations of functionals of the parameter $m$. Specifically, assuming that the negative log-likelihood $\Phi(m)$ is strictly convex in a neighborhood of $m_{\text{MAP}}$\footnote{To guarantee a positive definite posterior covariance operator also in the case of non-locally convex negative log-likelihood $\Phi(m)$, the inverse of the Gauss-Newton Hessian of the negative log-posterior can be used instead. This corresponds to linearizing the parameter-to-observable map $\mathcal{F}$ around $m_{\text{MAP}}$.}, the Laplace approximation to the posterior constructs a Gaussian distribution $\hat{\mu}_{\text{post}}$,

$$\hat{\mu}_{\text{post}} \sim \mathcal{N}(m_{\text{MAP}}, \mathcal{H}(m_{\text{MAP}})^{-1}), \quad (2.15)$$

centered at $m_{\text{MAP}}$ and with covariance operator

$$\mathcal{H}(m_{\text{MAP}})^{-1} = (\mathcal{H}_{\text{misfit}}(m_{\text{MAP}}) + C_{\text{prior}}^{-1})^{-1}. \quad (2.16)$$

Here $\mathcal{H}_{\text{misfit}}$ denotes the Hessian of the negative log-likelihood evaluated at $m_{\text{MAP}}$ (see Section 5 for examples of the derivation of the action of $\mathcal{H}_{\text{misfit}}$ using variational calculus and Lagragian formalism).

The Laplace approximation above is an important tool in designing scalable and efficient methods for Bayesian inference and UQ implemented in hIPPYlib. It has been studied in the context of PDE-based inverse problems to draw approximate samples and compute approximate statistics (such as the pointwise variance) in [15]. Likewise, it has been exploited in [55] to efficiently explore the true posterior distribution by generating high quality proposals for Markov chain Monte Carlo algorithms, in [25] to construct likelihood informed subspaces that allows for optimal dimension reduction in Bayesian inference problems, and in [61, 19] to construct a dimension independent sparse grid to evaluate posterior expectations. It has also been invoked in [41] for scalable approximation of the predictive posterior distribution of a scalar quantity of interest. Finally, its use was advocated in [48, 49, 47, 4] to approximate the solution of Bayesian optimal experimental design problems.

2.3. Discretization of the Bayesian inverse problem. We present a brief discussion of the finite-dimensional approximations of the prior and the posterior distributions; a lengthier discussion can be found in [15]. We start with a finite-dimensional subspace $\mathcal{M}_h$ of $\mathcal{M} \subseteq L^2(\mathcal{D})$ originating from a finite element discretization with continuous Lagrange basis functions $\{\phi_j\}_{j=1}^n$ [8, 64]. The approximation of the inversion parameter function $m \in \mathcal{M}$ is then $m_h = \sum_{j=1}^n m_j \phi_j \in \mathcal{M}_h$, and, in
what follows, \( \mathbf{m} = (m_1, \ldots, m_n)^T \in \mathbb{R}^n \) denotes the vector of the coefficients in the finite element expansion of \( m_h \).

The finite-dimensional space \( M_h \) inherits the \( L^2 \)-inner product. Thus, inner products between nodal coefficient vectors must be weighted by a mass matrix \( M \in \mathbb{R}^{n \times n} \) to approximate the infinite-dimensional \( L^2 \)-inner product. This \( M \)-weighted inner product is denoted by \( \langle \cdot, \cdot \rangle_M \), where \( \langle \mathbf{y}, \mathbf{z} \rangle_M = \mathbf{y}^T M \mathbf{z} \) and \( M \) is the (symmetric positive definite) mass matrix

\[
M_{ij} = \int_D \phi_i(x) \phi_j(x) \, dx, \quad i, j = 1, \ldots, n.
\]

To distinguish \( \mathbb{R}^n \) equipped with the \( M \)-weighted inner product with the usual Euclidean space \( \mathbb{R}^n \), we denote it by \( \mathbb{R}^n_M \). For an operator \( B : \mathbb{R}^n_M \to \mathbb{R}^n_M \), we denote the matrix transpose by \( B^T \) with entries \((B^T)_{ij} = B_{ji}\). In contrast, the \( M \)-weighted inner product adjoint \( B^* \) satisfies, for \( \mathbf{y}, \mathbf{z} \in \mathbb{R}^n \),

\[
\langle B \mathbf{y}, \mathbf{z} \rangle_M = \langle \mathbf{y}, B^* \mathbf{z} \rangle_M,
\]

which implies that \( B^* \) is given by

\[
B^* = M^{-1}B^T M. \tag{2.17}
\]

With these definitions, the matrix representation of the bilinear form involving the elliptic PDE operator \( A^n \) defined in Section 2.2 is given by \( R \) whose components are

\[
R_{ij} = \int_D \phi_i(x) A^n \phi_j(x) \, dx, \quad i, j \in \{1, \ldots, n\}. \tag{2.18}
\]

Finally, restating Bayes’ theorem with Gaussian noise and prior in finite dimensions, we obtain:

\[
\pi_{\text{post}}(\mathbf{m}) \propto \exp\left(-\frac{1}{2} \| F(\mathbf{m}) - d_{\text{obs}} \|_{\Gamma^{-1}_{\text{noise}}}^2 - \frac{1}{2} \| \mathbf{m} - \mathbf{m}_{\text{pr}} \|_{\Gamma^{-1}_{\text{prior}}}^2 \right), \tag{2.19}
\]

where \( \mathbf{m}_{\text{pr}} \) is the mean of the prior distribution, \( \Gamma_{\text{prior}} := \mathbb{R}^{-1}_{M} \in \mathbb{R}^{n \times n} \) is the covariance matrix for the prior that arises upon discretization of \( \mathcal{C}_{\text{prior}} \), and \( \Gamma_{\text{noise}} \in \mathbb{R}^{q \times q} \) is the covariance matrix for the noise. The method of choice to explore the full posterior is Markov chain Monte Carlo (MCMC), which samples the posterior so that sample statistics can be computed. MCMC for large-scale inverse problems is still prohibitive for expensive forward problems and high-dimensional parameter spaces; here we make a quadratic approximation of the negative log of the posterior (2.19), which results—as discussed in Section 2.2 in the continuous setting—in the Laplace approximation of the posterior given by

\[
\pi_{\text{post}}(\mathbf{m}) \propto \mathcal{N}(\mathbf{m}_{\text{MAP}}, \Gamma_{\text{post}}) \tag{2.20}
\]

The mean of this approximate posterior distribution, \( \mathbf{m}_{\text{MAP}} \), is the parameter vector maximizing the posterior (2.19), and is known as the maximum a posteriori (MAP) point. It can be found by minimizing the negative log-posterior\(^3\), which amounts to solving the following optimization problem:

\[
\mathbf{m}_{\text{MAP}} := \underset{\mathbf{m}}{\text{argmin}} \left( -\log \pi_{\text{post}}(\mathbf{m}) \right) = \underset{\mathbf{m}}{\text{argmin}} \frac{1}{2} \| F(\mathbf{m}) - d_{\text{obs}} \|_{\Gamma^{-1}_{\text{noise}}}^2 + \frac{1}{2} \| \mathbf{m} - \mathbf{m}_{\text{pr}} \|_{\Gamma^{-1}_{\text{prior}}}^2, \tag{2.21}
\]

\(^3\)For simplicity, we assume that the negative log-posterior has a unique minimum. In general, the negative log-posterior is not guaranteed to be convex and may admit multiple minima; in this case domain specific techniques should be exploited to locate the global minimum.
which is the discrete counterpart of problem (2.13). Denoting with \( \tilde{H}(m_{MAP}) \) and \( \tilde{H}_{\text{misfit}}(m_{MAP}) \) the matrix representations of, respectively, the second derivative of negative log-posterior \( H \) and log-likelihood \( H_{\text{misfit}} \) (i.e., the data misfit component of the Hessian) in the \( M \)-weighted inner product, and assuming that \( \tilde{H}_{\text{misfit}}(m_{MAP}) \) is positive definite, the covariance matrix \( \Gamma_{\text{post}} \) in the Laplace approximation is given by

\[
\Gamma_{\text{post}} = \tilde{H}^{-1}(m_{MAP}) = \left( \tilde{H}_{\text{misfit}}(m_{MAP}) + \Gamma_{\text{prior}}^{-1} \right)^{-1}.
\]

(2.22)

For simplicity of the presentation, in the following we will let \( H = M\tilde{H} \) and \( H_{\text{misfit}} = M\tilde{H}_{\text{misfit}} \) be the matrix representation of the Hessian with respect to the standard Euclidean inner product. Using this notation, and recalling that \( \Gamma_{\text{prior}} = R^{-1}M \), we rewrite (2.22) as

\[
\Gamma_{\text{post}} = H^{-1}(m_{MAP}) M = (H_{\text{misfit}}(m_{MAP}) + R)^{-1} M.
\]

(2.23)

Equation (2.21) and (2.23) define the mean and covariance matrix of the Laplace approximation to the posterior in the discrete setting. In Section 4, we present scalable (with respect to the parameter dimension) algorithms to compute the discrete MAP point \( m_{MAP} \) and to efficiently manipulate the covariance matrix \( \Gamma_{\text{post}} \).

3. Design and software components of \texttt{hIPPYlib}. \texttt{hIPPYlib} implements state-of-the-art scalable algorithms for PDE-based deterministic and Bayesian inverse problems. It builds on \texttt{FEniCS} (a parallel finite element library) [46, 44] for discretization of PDEs and on \texttt{PETSc} [5] for scalable and efficient linear algebra operations and solvers. In \texttt{hIPPYlib} the user can express the forward PDE and the likelihood in variational form using the friendly, compact, near-mathematical notation of \texttt{FEniCS}, which will then automatically generate efficient code for the discretization. Linear and nonlinear, stationary and time-dependent PDEs are supported in \texttt{hIPPYlib}. For stationary problems, gradient and Hessian information can be automatically generated by \texttt{hIPPYlib} using \texttt{FEniCS} symbolic differentiation of the relevant variational forms. For time-dependent problems, instead, symbolic differentiation can only be used for the spatial terms, and the contribution to gradients and Hessians arising from the time dynamics needs to be provided by the user. Noise and prior covariance operators are modeled as inverses of elliptic differential operators allowing us to build on fast multigrid solvers for elliptic operators without explicitly constructing the dense covariance operator. The components of \texttt{hIPPYlib} are (also summarized in Figure 3.1):

1. The \texttt{hIPPYlib model} component describes the inverse problem, i.e., the data misfit functional (negative log-likelihood), the prior information, and the forward problem. More specifically, the user can select from among a library of data misfit functionals—such as pointwise observations or continuous observations in the domain or on the boundary—or implement new ones using the prescribed interface. \texttt{hIPPYlib} offers a library of priors the user can choose from and allows for user-provided priors as well. Finally, the user needs to provide the forward problem either in the form of a \texttt{FEniCS} variational form or (for more complicated or time dependent problem) as a user-defined object. When using \texttt{FEniCS} variational forms, \texttt{hIPPYlib} is able to derive expressions for the gradient and Hessian automatically using \texttt{FEniCS} symbolic differentiation. Otherwise, the user is required to include routines for the construction of the adjoint and incremental problems.
2. The **hIPPYlib algorithms** component contains the numerical methods needed for solving the deterministic and linearized Bayesian inverse problems, i.e., the globalized inexact Newton-CG algorithm, randomized generalized eigensolvers, scalable sampling of Gaussian fields, and trace/diagonal estimators for large-scale not-explicitly-available covariance matrices. These algorithms are described in detail in Section 4.

3. The **hIPPYlib outputs** component includes the parameter-to-observable map (and its linear approximation), gradient evaluation and Hessian action, and Laplace approximation of the posterior distribution (MAP point and low-rank based representation of the posterior covariance operator). The **hIPPYlib** outputs can be utilized as input to other UQ software, e.g., the MIT Uncertainty Quantification Library (MUQ), to perform a full characterization of the posterior distribution using advanced dimension-independent Markov chain Monte Carlo simulation, requiring derivative information.

4. **hIPPYlib algorithms.** In this section we describe the main algorithms implemented in **hIPPYlib** for solution of deterministic and linearized Bayesian inverse problems. Specifically, we focus on computation of the MAP point and various operations on prior and posterior covariance matrices. In the linear case (and under the assumption of Gaussian noise and Gaussian prior) the posterior distribution is also Gaussian, and therefore is fully characterized once the MAP point and posterior covariance matrix are computed. In the nonlinear case, efficient exploration of the posterior distribution for large-scale PDE problems will require the use of a Markov chain Monte Carlo (MCMC) sampling method enchanted by Hessian information (see e.g. [51, 24, 26, 16, 10]). In this case **hIPPYlib** provides the tools to generate proposals for MCMC.

4.1. **Deterministic inversion and MAP point computation via inexact Newton-CG.** **hIPPYlib** provides a robust implementation of the inexact Newton-conjugate gradient (Newton-CG) algorithm (e.g.,[2, 11]) to solve the deterministic inverse problem and, in the Bayesian framework, to compute the maximum a posterior (MAP) point (see Algorithm 1). The gradient and Hessian actions—whose expressions are given in (2.4) and (2.8), respectively—are automatically computed via their variational form specification in **FEniCS** by constraining the state and adjoint
Algorithm 1 The inexact Newton-CG algorithm to find the MAP point

\[ i \leftarrow 0 \]

Given \( m_0 \) solve the forward problem (2.5) to obtain \( u_0 \)

Given \( m_0 \) and \( u_0 \) compute the cost functional \( J_0 \) using (2.3)

while \( i < \text{max} \_ \text{iter} \) do

Given \( m_i \) and \( u_i \) solve the adjoint problem (2.6) to obtain \( p_i \)

Given \( m_i, u_i \) and \( p_i \) evaluate the gradient \( g_i \) using (2.4)

if \( \| g_i \| \leq \tau \) then

break

end if

Given \( m_i, u_i \) and \( p_i \) define a linear operator \( H_i \) that implements the Hessian action (2.8)

Using conjugate gradients, find a search direction \( \hat{m}_i \) such that

\[ \| H_i \hat{m}_i + g_i \| \leq \eta_i \| g_i \|, \text{ with } \eta_i = \left( \frac{\| g_i \|}{\| g_0 \|} \right)^\frac{1}{2} \]

\[ j \leftarrow 0, \alpha^{(0)} \leftarrow 1 \]

while \( j < \text{max} \_ \text{backtracking} \_ \text{iter} \) do

Set \( m^{(j)} = m_i + \alpha^{(j)} \hat{m}_i \)

Given \( m^{(j)} \) solve the forward problem (2.5) to obtain \( u^{(j)} \)

Given \( m^{(j)} \) and \( u^{(j)} \) compute the cost \( J^{(j)} \) using (2.3)

if \( J^{(j)} < J_i + \alpha^{(j)} c_{\text{armijo}} g_i^T \hat{m}_i \) then

\( m_{i+1} \leftarrow m^{(j)}, J_{i+1} \leftarrow J^{(j)} \)

break

end if

\( \alpha^{(j+1)} \leftarrow \alpha^{(j)}/2, \quad j \leftarrow j + 1 \)

end while

\[ i \leftarrow i + 1 \]

end while

variables to satisfy the forward and adjoint problem in (2.5) and (2.6) respectively. The Newton system is solved inexactly using early termination of CG iterations using Eisenstat–Walker [29] (to prevent oversolving) and Steihaug [62] (to avoid negative curvature) criteria. Specifically, the choice of the tolerance \( \eta_i \) in Algorithm 1 leads to superlinear convergence of Newton’s method, and represents a good compromise between the number of Newton iterations and the computational effort to compute the search direction. Globalization is achieved with an Armijo backtracking line search; we choose the Armijo constant \( c_{\text{armijo}} \) in the interval \([10^{-5}, 10^{-4}]\). For a wide class of nonlinear inverse problems, the number of outer Newton iterations and inner CG iterations is independent of the mesh size and hence parameter dimension [37]. This is a consequence of using Newton’s method, the compactness of the Hessian (of the data misfit term), and preconditioning with the inverse regularization operator. We note that the resulting preconditioned Hessian is a compact perturbation of the identity, for which Krylov subspace methods exhibit mesh-independent iterations [17].

4.2. Low-rank approximation of the Hessian. The Hessian (of the negative log-posterior) plays a critical role in inverse problems. First, its spectral properties characterize the degree of ill-posedness. Second, the Hessian is the underlining operator for Newton-type optimization algorithms for solving the inverse problem, which
are highly desirable due to their dimension-independent convergence. Third, the
inverse of the Hessian locally characterizes the uncertainty in the solution of the inverse
problem: under the Laplace approximation, it is precisely the posterior covariance
matrix. Unfortunately, after discretization, the Hessian is formally a large, dense
matrix; forming each column requires an incremental forward and adjoint solves (see
Section 2.1). Thus, construction of the Hessian is prohibitive for large-scale problems
since its dimension is equal to the dimension of the parameter. To make operations
with the Hessian tractable, we exploit the fact that, in many cases, the eigenval-
ues collapse to zero rapidly, since the data contain limited information about the
(infinite-dimensional) parameter field. Thus a low-rank approximation of the data
misfit component of the Hessian, $H_{\text{misfit}}$, can be constructed. This can be proven
analytically for certain linear forward PDE problems (e.g. advection-diffusion [35],
Poisson [34], Stokes [75], acoustics [13, 14], electromagnetics [14]), and demonstrated
numerically for more complex PDE problems (e.g. seismic wave propagation [15, 12],
mantle convection [76], ice sheet flow [41, 55], poroelasticity [39], and turbulent flow
[20]). The end result is that manipulations with the Hessian require a number of
forward PDE solves that is independent of the parameter and data dimensions.

More specifically, to compute the low-rank factorization of the data misfit com-
ponent of the Hessian we consider the following generalized symmetric eigenproblem:

$$H_{\text{misfit}} v_i = \lambda_i R v_i, \quad \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n,$$

where $R$ stems from the discretization (with respect to the Euclidean inner product)
of the inverse of the prior covariance (i.e., the regularization operator). We then
choose $r \ll n$ such that $\lambda_{i+1}, 0 < i \leq n - r$, is small relative to 1, and we define

$$V_r = [v_1, v_2, \ldots, v_r] \quad \text{and} \quad A_r = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_r),$$

where the matrix $V_r$ is $R$-orthogonal, that is $V_r^T R V_r = I_r$. As in [41], by using the
Sherman-Morrison-Woodbury formula, we write

$$H^{-1} = (R + H_{\text{misfit}})^{-1} = R^{-1} - V_r D_r V_r^T + O \left( \sum_{i=r+1}^{n} \frac{\lambda_i}{1 + \lambda_i} \right),$$

where $D_r = \text{diag}(\lambda_1/(\lambda_1 + 1), \ldots, \lambda_r/(\lambda_r + 1)) \in \mathbb{R}^{r \times r}$. As can be seen from the form
of the remainder term above, to obtain an accurate low-rank approximation of $H^{-1}$,
we can neglect eigenvectors corresponding to eigenvalues that are small compared
to 1. This result is used to efficiently apply the inverse and square-root inverse of
the Hessian to a vector, as needed for computing the pointwise variance and when
drawing samples from a Gaussian distribution with covariance $H^{-1}$, as will be shown
in Sections 4.3.2 and 4.4.2, respectively. Efficient algorithms implemented in hIPPYlib
for solving eigenproblems using randomized linear algebra methods are described next.

**4.2.1. Randomized algorithm for the generalized eigenvalue problem.**
Randomized algorithms for eigenvalue computations have proven to be extremely ef-
ective for matrices with rapidly decaying eigenvalues [36]. For this class of matrices,
in fact, randomized algorithms present several advantages compared to Krylov sub-
space methods. Krylov subspace methods require sophisticated algorithms to monitor
restart, orthogonality, and loss of precision. On the contrary, randomized algorithm
are easy to implement, can be made numerically robust, and expose more opportuni-
ties for parallelism since matrix-vector products can be done asynchronously across all
vectors. The flexibility in reordering the computation makes randomized algorithms particularly well suited for modern parallel architectures with many cores per node and deep memory hierarchies.

In hIPPYlib we apply randomized algorithms to compute the low-rank factorization of the misfit part of the Hessian $H_{\text{misfit}}$. With a change of notation, we write the generalized eigenvalue problem (4.1) as

$$Av = \lambda Bv \quad (4.3)$$

where $A \in \mathbb{R}^{n \times n}$ is symmetric, $B \in \mathbb{R}^{n \times n}$ is symmetric positive definite, and $v \in \mathbb{R}^n$. Here we present an extension of the randomized eigensolvers in [36] to the solution of the generalized symmetric eigenproblem (4.3). Randomized algorithms for generalized symmetric eigenproblems were first introduced in [60], and are revisited here with some modifications.

The main idea behind randomized algorithms is to construct a $B$-orthogonal matrix $Q \in \mathbb{R}^{n \times (r+l)}$ that approximates the range of $B^{-1}A$. Here, $r$ represents the number of eigenpairs we wish to compute, and $l$ is an oversampling factor. More specifically, we have

$$\|I - QQ^T\|_B \leq \epsilon, \quad (4.4)$$

where $\epsilon$ is a random variable whose distribution depends on the generalized eigenvalues of (4.3) with index greater than $r+l$. To construct $Q$, we let $\Omega \in \mathbb{R}^{n \times (r+l)}$ be a Gaussian random matrix—whose entries are independent identically distributed (i.i.d.) standard Gaussian random variables—and we compute a $B$-orthogonal basis for the range of $Y = B^{-1}A\Omega$ using the so called PreCholQR algorithm ([50], see Algorithm 4). The main computational cost is the construction of $Y$, which requires $(r+l)$ applications of the operator $A$, and $(r+l)$ linear solves to apply $B^{-1}$. In contrast, the computation of the $B$-orthogonal matrix $Q$ using PreCholQR requires only an additional $(r+l)$ applications of $B$ and $O(n(r+l)^2)$ dense linear algebra operations for the QR factorization. Using (4.4) it can be shown (see [36]) that

$$A \approx (BQ)(Q^T\Omega)(BQ)^T = (BQ)T(BQ)^T, \quad (4.6)$$

where we have defined $T := Q^T AQ$. Then we compute the eigendecomposition $T = SAS^T$ ($S^T S = I_{r+l}$), and we approximate the $r$ dominant eigenpairs $(A_r, V_r)$ of (4.3) by:

$$A_r = A(1 : r, 1 : r), \quad V_r = QS(:, 1 : r). \quad (4.5)$$

Algorithms 2 and 3 summarize the implementation of the double pass and single pass randomized algorithms [36]. The main difference between these two algorithms is how the small matrix $T$ is computed. In the double pass algorithm, $T$ is computed directly by performing a second round of multiplication $AQ$ with the operator $A$. In the single pass algorithm, $T$ is approximated from the information contained in $\Omega$ and $Y$. In particular, generalizing the single pass algorithm in [36] to (4.3), we approximate $T$ as the least-squares solution of

$$T = \arg\min_{X \in \mathbb{R}^{(r+l) \times (r+l)}, s.p.d} \|X(Q^T B\Omega) - Q^T BY\|_2^2. \quad (4.6)$$

For this reason, the single pass algorithm has a lower computational cost compared to the double pass algorithm; however the resulting approximation is less accurate. We remark that Algorithm 3 presents some improvements relative to the single pass algorithm presented in [60]. In [60], the authors define $T = (Q^T B\Omega)^{-1}(Q^T BY)(Q^T B\Omega)^{-1}$. 
however the pre and post multiplication by the inverse of \((Q^T B \Omega)\) may lead to loss of accuracy due to ill-conditioning of \((Q^T B \Omega)\). In contrast, definition of \(T\) as the least-squares solution of (4.6) is more numerically stable.

**Algorithm 2** The double pass randomized algorithm for the solution of the generalized symmetric eigenproblem.

1. Let \(r\) be the number of eigenpairs to compute and \(l\) an oversampling factor
2. Let \(\Omega \in \mathbb{R}^{n \times (r+l)}\) be a Gaussian random matrix
3. \(Y \leftarrow A \Omega, Y = B^{-1} Y\)
4. Use PreCholQR to factorize \(Y = QR\) such that \(Q^T B Q = I_{r+l}\)
5. \(T \leftarrow Q^T A Q\)
6. Compute the eigenvalue decomposition \(T = S \Lambda S^T\)
7. Keep the \(r\) largest eigenmodes and let \(S_r \leftarrow S(:,1:r), \Lambda_r \leftarrow \Lambda(1:r,1:r)\)

**Return**: \(V_r \leftarrow QS_r\), and \(\Lambda_r\)

**Algorithm 3** The single pass randomized algorithm for the solution of the generalized symmetric eigenproblem.

1. Let \(r\) be the number of eigenpairs to compute and \(l\) an oversampling factor
2. Let \(\Omega \in \mathbb{R}^{n \times (r+l)}\) be a Gaussian random matrix
3. \(Y \leftarrow A \Omega, Y = B^{-1} Y\)
4. Use PreCholQR to factorize \(Y = QR\) such that \(Q^T B Q = I_{r+l}\) and \(\bar{Q}\)
5. Find \(T\) s.t. \(\|T(Q^T \Omega) - Q^T Y\|_2^2 \rightarrow \min\)
6. Compute the eigenvalue decomposition \(T = S \Lambda S^T\)
7. Keep the \(r\) largest eigenmodes and let \(S_r \leftarrow S(:,1:r), \Lambda_r \leftarrow \Lambda(1:r,1:r)\)

**Return**: \(V_r \leftarrow QS_r\), and \(\Lambda_r\)

**Algorithm 4** PreCholQR

1. **Require**: \(Y \in \mathbb{R}^{n \times (r+l)}\), and \(B \in \mathbb{R}^{n \times n}\)
2. \([Z, R_Y] \leftarrow qr(Y)\)
3. \(Z \leftarrow B Z\)
4. \(R_Z = \text{chol}(Z^T Z)\)
5. \(Q = Z R_Z^{-1}, \bar{Q} = Z R_Z^{-1}, \text{and } R = R_Z R_Y\)

**4.3. Sampling from large-scale Gaussian random fields.** Sampling techniques play a fundamental role in exploring the posterior distribution and in quantifying the uncertainty in the inferred parameter; for example, Markov chain Monte Carlo methods often use the prior distribution (assumed Gaussian in our settings) or some Gaussian approximation to the posterior to generate proposals for the Metropolis-Hastings algorithm. In this section, we describe the sampling capabilities implemented in hIPPYlib to generate realizations of Gaussian random fields with a prescribed covariance operator \(C\). Then we describe how the low-rank factorization of the data misfit part of the Hessian in Section 4.2 can be exploited to generate samples from the Laplace approximation of the posterior distribution in (2.15). In what follows, we will denote the expected value (mean) of a random vector \(x\) with the symbol \(E[x]\), and its covariance with the symbol \(\text{cov}(x) := E[(x - E[x])(x - E[x])^T]\). We will also
denote with $\Gamma$ the matrix representation of the covariance operator $\mathcal{C}$ with respect to the standard Euclidean inner product.

To sample from a small-scale multivariate Gaussian distribution, it is common to resort to a Cholesky factorization of the covariance matrix $\Gamma = \mathbf{C}\mathbf{C}^T$. In fact, if $\eta$ is a vector of independent identically distributed Gaussian variables $\eta_i$ with zero mean ($\mathbb{E}[\eta] = 0$) and unit variance ($\text{cov}(\eta) = I$), then $\mathbf{x} = \mathbf{C}\eta$ is such that $\text{cov}(\mathbf{x}) = \mathbb{E}[\mathbf{x}\mathbf{x}^T] = \mathbb{E}[\mathbf{C}\eta\eta^T\mathbf{C}^T] = \mathbf{C}\mathbb{E}[\eta\eta^T]\mathbf{C}^T = \mathbf{C}\mathbf{C}^T = \Gamma$.

Since an affine transformation of a Gaussian vector is still Gaussian, we have that $\mathbf{x} \sim \mathcal{N}(0, \Gamma)$.

This approach is not feasible for large-scale problems since it requires a Cholesky factorization of the covariance matrix. However, note that the factorization $\Gamma = \mathbf{C}\mathbf{C}^T$ is not unique, and the matrix $\mathbf{C}$ need not to be a square matrix. In Appendix A, we exploit this observation and show a scalable sampling technique based on a rectangular decomposition of $\mathbf{R} = \Gamma^{-1}$, for the case when $\mathbf{R}$ is a finite element discretization of a differential operator.

4.3.1. Sampling from the prior. Sampling from a Gaussian distribution with a prescribed covariance matrix $\Gamma$ is a difficult task for large-scale problems. Different approaches have been investigated, but how to make these algorithms scalable is still an active area of research. In [53], the authors introduce a conjugate gradient sampler that is a simple extension of the conjugate gradient method for solving linear systems. However, loss of orthogonality in finite arithmetic and the need of a factorized preconditioner limit the efficiency of this sampler for large-scale applications. In [21], the authors consider a preconditioned Krylov subspace method to approximate the action of $\Gamma^{-1} z$ on a generic vector $z$. In [18], the authors discuss a method to compute $f(\Gamma)b$ via least-squares polynomial approximations for a generic matrix function $f(x) = \sqrt{x}$. To this aim the authors approximate the function by a spline of a desired accuracy on the spectrum of $\Gamma$ and introduce a weighted inner product to simplify the computation.

hIPPYlib implements a new sampling algorithm that strongly relies on the structure of the covariance matrix and on the assembly procedure of finite element matrices. In particular, we restrict ourselves to the class of priors described in Section 2.2. For this class of priors, the inverse of the covariance matrix admits a sparse representation as a finite element matrix $\mathbf{R}$ stemming from the finite element discretization of a coercive symmetric differential operator. To draw a sample $\mathbf{x}$ from the distribution $\mathcal{N}(0, \mathbf{R}^{-1})$, we solve the linear system

$$\mathbf{R}\mathbf{x} = \mathbf{C}\eta, \quad \text{where } \eta \sim \mathcal{N}(0, \mathbf{I}_q),$$

where $\mathbf{C} \in \mathbb{R}^{n \times q} (q \geq n)$ is the rectangular factor of $\mathbf{R}$ described in Appendix A. In particular, we have

$$\mathbb{E}[xx^T] = \mathbf{R}^{-1}\mathbf{C}\mathbb{E}[\eta\eta^T]\mathbf{C}^T\mathbf{R}^{-1} = \mathbf{R}^{-1}\mathbf{R}\mathbf{R}^{-1} = \mathbf{R}^{-1},$$

where we exploited the fact that $\mathbb{E}[\eta\eta^T] = \mathbf{I}_q$ and $\mathbf{C}\mathbf{C}^T = \mathbf{R}$. This method is particularly efficient at large-scale since: i) the matrix $\mathbf{C}$ is sparse and can be computed efficiently by exploiting the finite element assembly routine; ii) the dominant cost

\footnote{Note that $\Gamma$ differs from $\Gamma_{\text{prior}}$ and $\Gamma_{\text{post}}$, which are defined in terms of the $\mathbf{M}$-weighted inner product.}
is the solution of a linear system with coefficient matrix $R$, for which efficient and scalable methods are available (e.g., conjugate gradients with algebraic multigrid preconditioner); and iii) the stochastic dimension of $\eta$ also scales linearly with the size of the problem.

### 4.3.2. Sampling from the Laplace approximation of the posterior.

To sample from the Laplace approximation of the posterior we assume that the posterior covariance operator can be expressed as a low-rank update of the prior covariance, i.e., in the form of equation (4.2). This assumption is often verified for many inverse problems as we discussed in Section 4.2. Then given a sample from the prior distribution $\mathcal{N}(0,R^{-1})$, a sample from the Laplace approximation of the posterior $\mathcal{N}(0,(H_{\text{misfit}} + R)^{-1})$ can be computed as

$$y = (I_n - V_rS_rV_r^T R)x,$$

where $S_r = I_r - (A_r + I_r)^{-\frac{1}{2}} = \text{diag}(1 - 1/\sqrt{\lambda_1 + 1}, \ldots, 1 - 1/\sqrt{\lambda_r + 1}) \in \mathbb{R}^{r \times r}$. This can be verified by the following calculation,

$$\text{cov}(y) = \mathbb{E}[yy^T] = \mathbb{E}[(I_n - V_rS_rV_r^T R)xx^T(1 - V_rS_rV_r^T R)^T] = (I_n - V_rS_rV_r^T R) \mathbb{E}[xx^T](I_n - V_rS_rV_r^T R)^T = (I_n - V_rS_rV_r^T R) R^{-1} (I_n - V_rS_rV_r^T R)^T = R^{-1} - V_r(2S_r - S_r^2)V_r^T = R^{-1} - V_rD_rV_r^T \approx H^{-1},$$

where we have used the definition of $D_r$, the $R$-orthogonality of the eigenvectors matrix $V_r$ (i.e., $V_r^T RV_r = I_r$), and the fact that

$$2 \left( 1 - \frac{1}{\sqrt{1 + \lambda_i}} \right) - \left( 1 - \frac{1}{\sqrt{1 + \lambda_i}} \right)^2 = 1 - \left[ \left( 1 - \frac{1}{\sqrt{1 + \lambda_i}} \right) - 1 \right]^2 = \frac{\lambda_i}{1 + \lambda_i}.$$

#### 4.4. Pointwise variance of Gaussian random fields.

Consider a Gaussian random field $m \sim \mathcal{N}(0,C)$ and its discrete counterpart $m \sim \mathcal{N}(0,Q^{-1})$, where $Q$ is the precision matrix. Here $C = C_{\text{pr}}$ and $Q = R$ if we are interested in the prior distribution; $C = C_{\text{post}}$ and $Q = H$ for the posterior distribution. Then we define the pointwise variance of $m$ as the field $\sigma^2(x)$ such that

$$\sigma^2(x) = \text{Var}[m(x)], \quad \forall x \in D.$$

In this section, we present an efficient numerical method to compute a finite element approximation $\sigma^2_h$ of $\sigma^2(x)$. More specifically, we let $\sigma^2 \in \mathbb{R}^n$ be the vector collecting the coefficients of the expansion of $\sigma^2_h$ in the finite element basis used to discretize the parameter space. Then it is possible to show that $\sigma^2 = \text{diag}(Q^{-1})$ (see e.g. [15]). A naïve approach would require solution of $n$ linear systems with $Q$. This is not feasible for large-scale problems: even for the case when an optimal (i.e., $O(n)$) solver for $Q$ is available, the complexity is at least $O(n^2)$ operations. In what follows we discuss stochastic estimators and probing methods to efficiently estimate the pointwise variance of the prior distribution and we explore how the low-rank representation of the data misfit component of the Hessian can be efficiently exploited to compute the pointwise variance of the Laplace approximation of posterior distribution.
4.4.1. Pointwise variance of the prior. Estimating the pointwise variance of the prior reduces to the well studied problem of estimating the diagonal of the inverse of a matrix $R$. Recall that in our case, $R$ arises from finite element discretization of an elliptic differential operator. Two commonly used methods to solve this task are the stochastic estimator in [9] and the probing method in [66]. Specifically, the unbiased stochastic estimator for the diagonal of the inverse of $R$ in [9] reads

$$\text{diag}(R^{-1}) \approx \left[ \sum_{j=1}^{s} z_j \odot w_j \right] \odot \left[ \sum_{j=1}^{s} z_j \odot z_j \right], \quad (4.9)$$

where $w_j$ solves $Rw_j = z_j$ and $z_j$ are random i.i.d. vectors. Here $\odot$ and $\odot$ represent the element-wise multiplication and division operators of vectors, respectively. The convergence of the method is independent of the size of the problem, but convergence is in general slow. The probing method in [66], on the other hand, leads to faster convergence in the common situation in which $R^{-1}$ exhibits a decay property, i.e., the entries far away from the diagonal are small. Probing vectors are determined by applying some coloring algorithm to the graph $G$ whose adjacency matrix is the sparsity pattern of some power $k$ of $R$. More specifically, there are as many probing vectors as the number of colors in the graph $G$ and the probing vector associated to color $i$ is the binary vector whose non-zero entries correspond to the nodes of $G$ colored with color $i$. The higher the power $k$, the more accurate will be the estimation, but also the more expensive due to the increased number of colors (and, therefore, of probing vectors to solve for). The main shortcoming of this approach is that it is not mesh independent, i.e., as we refine the mesh we need to increase the power $k$ and therefore the number of probing vectors.

To overcome these difficulties, hIPPYlib implements, in addition to the methods mentioned above, a novel approach based on a randomized eigendecomposition of $R^{-1}$, taking advantage of the fact that $R$ is the discretization of an elliptic differential operator. Specifically, we write

$$\text{diag}(R^{-1}) \approx \left[ \sum_{i=1}^{r} \mu_i v_i \odot v_i \right], \quad (4.10)$$

where $\{(\mu_i, v_i)\}_1^r$ denote the approximation of the $r$ dominant eigenpairs of the matrix $R^{-1}$ obtained by using the double pass randomized eigensolver (Algorithm 2) with $l = 0$, $A = R^{-1}$, and $B$ the identity matrix. The main advantage of this approach is that, thanks to the rapid decay of the eigenvalues $\mu_i$ ($R^{-1}$ is compact), $r$ is much smaller than the number of samples $s$ necessary for the stochastic estimator in (4.9) to achieve a given accuracy and that, in contrast to the probing algorithm, it is independent of the mesh size.

4.4.2. Pointwise variance of the posterior. We resort to the low-rank representation of the data misfit component of the Hessian and the Woodbury formula to obtain the approximation

$$\text{diag}(H^{-1}) \approx \text{diag}(R^{-1} - V_r D_r V_r^T) = \text{diag}(R^{-1}) - \text{diag}(V_r D_r V_r^T). \quad (4.11)$$

In hIPPYlib, the first term is approximated using (4.10), while the data-informed correction $\text{diag}(V_r D_r V_r^T)$ can be explicitly computed in $O(n)$ operations as follows

$$\text{diag}(V_r D_r V_r^T) = \sum_{i=1}^{r} \left[ \left( \frac{\lambda_i}{1 + \lambda_i} \right) v_i \right].$$
5. Model problems. In this section we apply the inversion methods discussed in previous sections to two model problems: inversion for the log coefficient field in an elliptic partial differential equation and inversion for the initial condition in a time-dependent advection-diffusion equation. The main goal of this section is to illustrate the deterministic inversion and linearized Bayesian analysis capabilities of hIPPYlib for the solution of these two representative types of inverse problems. The numerical results showed below were obtained using hIPPYlib version 2.3.0 and FEniCS 2017.2. For a line-by-line explanation of the source code for these two model problems, we refer the reader to the Python Jupyter notebooks included in the on-line supplemental material.

5.1. Coefficient field inversion in a Poisson PDE problem. In this section, we study the inference of the log coefficient field $m$ in a Poisson partial differential equation from pointwise state observations. In what follows we describe the forward and inverse problems setup, present the prior and the likelihood distributions for the Bayesian inverse problem, and derive the expressions for the gradient and Hessian action using the standard Lagrangian approach as described in Section 2. The forward model is formulated as follows

\[ -(e^m \nabla u) \cdot n = h \quad \text{on } \Gamma_N, \]

where $D \subset \mathbb{R}^d$ ($d = 2, 3$) is an open bounded domain with sufficiently smooth boundary $\Gamma = \Gamma_D \cup \Gamma_N, \Gamma_D \cap \Gamma_N = \emptyset$. Here, $u$ is the state variable, $f \in L^2(D)$ is a source term, and $g \in H^{1/2}(\Gamma_D)$ and $h \in L^2(\Gamma_N)$ are Dirichlet and Neumann boundary data, respectively. We define the spaces,

\[ V_g = \{ v \in H^1(D) : v|_{\Gamma_D} = g \}, \quad V_0 = \{ v \in H^1(D) : v|_{\Gamma_D} = 0 \}, \]

where $H^1(D)$ is the Sobolev space of functions whose derivatives are in $L^2(D)$. Then, the weak form of (5.1) reads as follows: find $u \in V_g$ such that

\[ \langle e^m \nabla u, \nabla p \rangle = \langle f, p \rangle + \langle h, p \rangle|_{\Gamma_N}, \quad \forall p \in V_0. \]

Here $\langle \cdot, \cdot \rangle$ and $\langle \cdot, \cdot \rangle|_{\Gamma_N}$ denote the standard inner products in $L^2(D)$ and $L^2(\Gamma_N)$, respectively.

5.1.1. Prior and noise models. We take the prior as a Gaussian distribution $\mathcal{N}(m_{pr}, C_{prior})$, with mean $m_{pr}$ and covariance $C_{prior} = \mathcal{A}^{-2}$ following [65]. $\mathcal{A}$ is a differential operator with domain $\mathcal{M} := H^1(D)$ and action

\[ \mathcal{A}m = \begin{cases} -\gamma \nabla \cdot (\Theta \nabla m) + \delta m & \text{in } D, \\ \Theta \nabla m \cdot n + \beta m & \text{on } \partial D, \end{cases} \]

where $\beta \propto \sqrt{\gamma} \delta$ is the optimal Robin coefficient derived in [27] to minimize boundary artifacts, and $\Theta$ is an s.p.d. anisotropic tensor of the form

\[ \Theta = \begin{bmatrix} \theta_1 \sin(\alpha)^2 & (\theta_1 - \theta_2) \sin(\alpha) \cos \alpha \\ (\theta_1 - \theta_2) \sin(\alpha) \cos \alpha & \theta_2 \cos(\alpha)^2 \end{bmatrix}. \]

In Figure 5.1 (left), we show the prior mean $m_{pr}$ and three random draws from the prior distribution with $\gamma = 0.1$, $\delta = 0.5$, $\alpha = \frac{\pi}{4}$, $\theta_1 = 2$, $\theta_2 = 0.5$. 

Next, we specify the log-likelihood (data misfit) functional. We denote with \( d \in \mathbb{R}^q \) the vector of (noisy) pointwise observations of the state \( u \) at \( q = 50 \) random locations uniformly distributed in \( D_{\text{obs}} := [0.1, 0.9] \times [0.1, 0.5] \) (\( D_{\text{obs}} \subset D \)). That is,

\[
d = B u + \eta,
\]

where \( B : V_q \mapsto \mathbb{R}^q \) is a linear observation operator, a sum of delta functions to be specific, that extracts measurements from \( u \). The measurement noise vector \( \eta \) is a multivariate Gaussian variable with mean \( 0 \) and covariance \( \Gamma_{\text{noise}} = \sigma^2 I \), where \( \sigma = 0.01 \), and \( I \in \mathbb{R}^{q \times q} \).

### 5.1.2. Gradient and Hessian actions computation.

To find the MAP point we solve the following variational nonlinear least-squares optimization problem

\[
\min_{m \in M} J(m) := \frac{1}{2} \| Bu(m) - d \|_{\Gamma_{\text{noise}}^{-1}}^2 + \frac{1}{2} \| m - m_{\text{pr}} \|_{C_{\text{prior}}^{-1}}^2 ,
\]

where the state variable \( u \) is the solution to (5.1), \( m_{\text{pr}} \) is the prior mean of the log coefficient field \( m \), and \( d \in \mathbb{R}^q \) is a given data vector. To solve this optimization problem we use the inexact Newton-CG algorithm in Algorithm 1, which requires gradient and Hessian information. We note that the use of CG to solve the resulting Newton system does not require computing the Hessian operator by itself but only its action in a given direction.

In what follows, we apply the technique outlined in Section 2.1 and derive expressions for the gradient and Hessian actions of the cost functional \( J(m) \) defined in (5.3).

For the above optimization problem, the Lagrangian functional for the gradient computation is given by

\[
\mathcal{L}(u, m, p) := J(m) + \langle e^m \nabla u, \nabla p \rangle - \langle f, p \rangle - \langle p, h \rangle_{\Gamma_N},
\]

where the last three terms stem from the variational form (5.2) of the forward problem (5.1). The formal Lagrange multiplier method [71] requires that, at a minimizer of (5.3), variations of the Lagrangian functional with respect to \( p \) and \( u \) vanish, which yields to solving the forward and adjoint problems

\[
\langle e^m \nabla u, \nabla \tilde{p} \rangle - \langle f, \tilde{p} \rangle - \langle \tilde{p}, h \rangle_{\Gamma_N} = 0, \quad \forall \tilde{p} \in V_0;
\]

\[
\langle e^m \nabla \tilde{u}, \nabla p \rangle + \langle B^* \Gamma_{\text{noise}}^{-1} (Bu - d), \tilde{u} \rangle = 0, \quad \forall \tilde{u} \in V_0.
\]
The strong form of the forward problem is given in (5.1), while the strong form of the
adjoint problem reads
\[-\nabla \cdot (e^n \nabla p) = B^* \Gamma_{\text{noise}}^{-1} (Bu - d) \quad \text{in } D,\]
\[p = 0 \quad \text{on } \Gamma_D,\]
\[e^n \nabla p \cdot n = 0 \quad \text{on } \Gamma_N.\]

Finally, the gradient of the cost functional (5.3) is given in weak form by
\[\langle G(m), \hat{m} \rangle = \langle m - m_{\text{pr}}, \hat{m} \rangle_{\text{prior}} + \langle \hat{m} e^n \nabla u, \nabla p \rangle, \quad \forall \hat{m} \in \mathcal{M},\]
where \(u\) and \(p\) are solutions to the forward and adjoint problems (5.5a)-(5.5b), re-
spectively [71, 11]. In strong form this reads
\[G(m) = \left\{ \begin{array}{ll}
\mathcal{C} \Gamma_{\text{prior}}^{-1} (m - m_{\text{pr}}) + e^n (\nabla u \cdot \nabla p) & \text{in } D, \\
\gamma (\Theta \nabla m) \cdot n + \beta m & \text{on } \partial D.
\end{array} \right.\]

We note that to evaluate the gradient for a given parameter \(m\), one needs to solve
the forward problem (5.5a) for \(u\), and then given \(m\) and \(u\) solve the adjoint problem
for \(p\). This evaluation of the gradient costs one forward and one adjoint PDE solve.

Next, we derive the expression of the Hessian action following Section 2.1. The
second order Lagrangian functional in this case reads
\[\mathcal{L}^H(u, m, p; \hat{u}, \hat{m}, \hat{p}) := (G(m), \hat{m}) \]
\[+ \langle e^n \nabla u, \nabla \hat{p} \rangle - \langle f, \hat{p} \rangle - \langle h, \hat{p} \rangle_{\Gamma_N} \]
\[+ \langle e^n \nabla \hat{u}, \nabla p \rangle + \langle B^* \Gamma_{\text{noise}}^{-1} (Bu - d), \hat{u} \rangle.\]

To obtain the action of the Hessian in a direction \(\hat{m}\) we take the variation of \(\mathcal{L}^H\) with
respect to \(m\), namely
\[\langle \hat{m}, \mathcal{H}(m) \hat{m} \rangle = \langle \hat{m} e^n \nabla \hat{u}, \nabla \hat{p} \rangle + \langle \hat{m}, \hat{m} \rangle_{\text{prior}} \]
\[+ \langle \hat{m} e^n \nabla u, \nabla p \rangle + \langle \hat{m} e^n \nabla u, \nabla \hat{p} \rangle, \quad \forall \hat{m} \in \mathcal{M},\]
where as before \(u\) and \(p\) are the solutions of the forward and adjoint problems in
(5.5a) and (5.5b), respectively, and the \(\hat{u}\) and \(\hat{p}\) are the solutions of the incremental
forward and adjoint problems, respectively. These equations are given by
\[\langle e^n \nabla \hat{u}, \nabla \hat{p} \rangle + \langle \hat{m} e^n \nabla u, \nabla \hat{p} \rangle = 0, \quad \forall \hat{p} \in \mathcal{V}_0,\]
and
\[\langle B^* \Gamma_{\text{noise}}^{-1} B \hat{u}, \hat{u} \rangle + \langle \hat{m} e^n \nabla \hat{u}, \nabla \hat{p} \rangle + \langle e^n \nabla \hat{u}, \nabla \hat{p} \rangle = 0, \quad \forall \hat{u} \in \mathcal{V}_0.\]

Once we have the gradient and Hessian action expressions, we can apply Algo-

\subsection{Numerical results}
For the forward Poisson problem (5.1), no source
term, (i.e., \(f = 0\)) and no normal flux on \(\Gamma_N := \{0, 1\} \times (0, 1)\) (i.e., the homogeneous
Neumann condition \(e^n \nabla u \cdot n = 0\) on \(\Gamma_N\)) are imposed. Dirichlet conditions are
prescribed on the top and bottom boundaries, in particular \(u = 1\) on \((0, 1) \times \{1\}\) and
\(u = 0\) on \((0, 1) \times \{0\}\). This Dirichlet part of the boundary is denoted by \(\Gamma_D :=\)
In Figure 5.2, we show the true parameter field \( m_{\text{true}} \) (left) and the state \( u \) obtained by solving the forward PDE with \( m_{\text{true}} \) (right). The squares in (b) represent locations of the \( q = 50 \) randomly chosen observation points and their color corresponds to the observed noisy data \( d \).

\((0, 1) \times \{0, 1\}\). In Figure 5.2, we show the true parameter field used in our numerical tests, and the corresponding state field. We used quadratic finite elements to discretize the state and adjoint variables and linear elements for the parameter. The degrees of freedom for the state and parameter were 16641 and 4225, respectively.

Next we study the spectrum of the data misfit Hessian evaluated at the MAP point. Figure 5.3 shows a logarithmic plot of the eigenvalues of the generalized symmetric eigenproblem

\[
H_{\text{misfit}} v_i = \lambda_i R v_i, \quad \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n,
\]

where \( H_{\text{misfit}} \) and \( R \) stems from the discretization (with respect to the Euclidean inner product of the data misfit Hessian and prior precision (cf Eq. (4.1)). This plot shows that the spectrum decays rapidly. As seen in (4.2), an accurate low-rank based approximation of the inverse Hessian can be obtained by neglecting eigenvalues that are small compared to 1. Thus, retaining around \( r = 30 \) eigenvectors out of 4225 (i.e., the dimension of parameter space) appears to be sufficient. We stress that \( r \) is strictly less than the number \( q = 50 \) of observation, reflecting redundancy in the data. We note that the cost of obtaining this low-rank based approximation, measured in the number of forward and adjoint PDE solves, is \( 2(r + l) \), where \( r + l \) is the number of random vectors. Here, \( l = 20 \) is an oversampling parameter used to ensure the accurate computation of the most significant eigenvalues/eigenvectors. The corresponding retained eigenvectors are those modes in parameter space that are simultaneously well-informed by the data and assigned high probability by the prior. Figure 5.4 displays several of these eigenvectors.

Figure 5.5 depicts the prior and posterior pointwise variances computed using (4.10) and (4.11) with \( r = 300 \) and \( r = 50 \), respectively. One observes that the uncertainty is vastly reduced in the bottom half of the domain, which is expected given that observations are present only on the lower half of \( D \). In Figure 5.6 we show the MAP point (a) and samples from the Laplace approximation (2.20) of the posterior probability density function (b)-(d). These samples were obtained by first computing samples from the prior distribution—shown in Figure 5.1—according to (4.7), and then applying (4.8). The variance reduction between posterior samples in Figure 5.6 and prior samples in Figure 5.1 reflects the information gained from the data in solving the inverse problem. In addition, we note that the MAP point resembles the
Fig. 5.3. Log-linear plot of first 30 out of 4225 eigenvalues of the data misfit Hessian for the generalized eigenproblem (4.1). The low-rank based approximation captures the dominant, data-informed portion of the spectrum. The eigenvalues are truncated at 0.07.

Fig. 5.4. Prior-orthogonal eigenvectors of the data misfit Hessian corresponding (from left to right) to the 1st, 3rd, 8th, and 27th eigenvalues. Note that eigenvectors corresponding to smaller eigenvalues are increasingly more oscillatory (and thus inform smaller length scales of the parameter field) but are also increasingly less informed by the data.

truth better in the lower half of the domain where data are available. The presence (or absence) of data also affects the posterior samples, in fact, we observe higher variability in the upper half of the domain, where there is no data.

5.2. Inversion for the initial condition in an advection-diffusion PDE. Here we consider a time-dependent advection-diffusion equation for which we seek to infer an unknown initial contaminant field from pointwise measurements of its concentration. The problem description below closely follows the one in [56]. The PDE in the parameter-to-observable map models diffusive transport in a domain \( \mathcal{D} \subset \mathbb{R}^d \), which is depicted in Figure 5.7. The domain boundaries \( \partial \mathcal{D} \) include the outer boundaries as well as the internal boundaries of the rectangles, which represent buildings. The parameter-to-observable map \( \mathcal{F} \) maps an initial condition \( m \in L^2(\mathcal{D}) \) to pointwise spatiotemporal observations of the concentration field \( u(\mathbf{x}, t) \) through solution of the advection-diffusion equation given by

\[
\begin{align*}
    u_t - \kappa \Delta u + \mathbf{v} \cdot \nabla u = 0 & \quad \text{in } \mathcal{D} \times (0, T), \\
    u(\cdot, 0) = m & \quad \text{in } \mathcal{D}, \\
    \kappa \nabla u \cdot \mathbf{n} = 0 & \quad \text{on } \partial \mathcal{D} \times (0, T).
\end{align*}
\]
Fig. 5.5. Pointwise variance of the prior distribution (left) and the Laplace approximation of the posterior distribution (right). Note that uncertainty is mostly reduced in the lower half of the domain where data is measured.

Fig. 5.6. The MAP point (a) and samples drawn from the Laplace approximation of posterior distribution (b)–(d).

Here, $\kappa > 0$ is a diffusivity coefficient, and $T > 0$ is the final time of observations. The velocity field $v$, shown in Figure 5.7 (right), is computed by solving the steady-state Navier-Stokes equations for a two dimensional flow with Reynolds number 50 and boundary conditions as in Figure 5.7 (left); see [56] for details. The time evolution of the state variable $u$ from a given initial condition $m$ is illustrated in Figure 5.8.

To derive the weak formulation of (5.13), we define the spaces

$$V := \{ v \in H^1(D), \text{ for each } t \in (0,T) \}, \text{ and } M := H^1(D).$$

Then, the weak form of (5.13) reads as follows: Find $u \in V$ such that

$$\int_0^T \int_D (u_t + v \cdot \nabla u)p \, dx \, dt + \int_0^T \int_D \kappa \nabla u \cdot \nabla p \, dx \, dt + \int_D (u(x,0) - m)p_0 \, dx = 0, \quad (5.14)$$

$\forall p \in V, p_0 \in M$. Above, the initial condition $u(x,0) = m$ is imposed weakly by means of the test function $p_0 \in M$.

5.2.1. Gradient computation for the MAP point. In this section we derive the expression for the gradient for the problem of inferring the initial condition $m$ in (5.13) from pointwise noisy observations $d_i \in \mathbb{R}^{n_t}$ ($i = 1, \ldots, n_s$) of the state $u$ at $n_s$ discrete time samples $t_i$ in interval $[T_1, T] \subset [0, T]$, and $n_t$ locations in space. We assume that the observations $d_i$ are perturbed with $i.i.d.$ Gaussian additive
noise with variance $\sigma^2$. To construct the prior, we assume a constant mean and define $C_{\text{prior}} := \mathcal{A}^{-2} = (-\gamma \Delta + \delta I)^{-2}$, equipped with Robin boundary conditions $\gamma \nabla m \cdot n + \beta m$ on $\partial D$. The parameters $\gamma, \delta > 0$ control the correlation length and variance of the prior operator; here we take $\gamma = 1$ and $\delta = 8$. The Robin coefficient $\beta$ is chosen as in [27] to reduce boundary artifacts.

To compute the MAP point we minimize the negative log-posterior, defined in general in (2.13), which—for Gaussian prior and noise—is analogous to the least-squares functional minimized in the solution of a deterministic inverse problem. For this particular problem, this reads

$$J(m) := \frac{1}{\sigma^2} \sum_{i=1}^{n_s} \int_{T_i} (Bu - d_i)^2 \delta_{t_i} \, dt + \frac{1}{2} \| \mathcal{A}(m - m_{pr}) \|_{L^2(D)}^2,$$

(5.15)

where $B : \mathcal{V}_i \mapsto \mathbb{R}^{n_t}$ is the interpolation operators at the observation locations, $\delta_{t_i}$ is the Dirac delta functions at the observation time sample $t = t_i (i = 1, \ldots, n_s)$, and $\sigma^2$ represents the noise level in the observations $d_i$, here taken $2.45 \times 10^{-7}$, and $m_{pr} = 0$ is the prior mean.

To derive an expression for the gradient of $J(m)$ in (5.15), we define the Lagrangian functional

$$\mathcal{L}(u, m, p, p_0) := J(m) + \int_0^T \int_D (u_t + \mathbf{v} \cdot \nabla u)p \, dx \, dt$$

$$+ \int_0^T \int_D \kappa \nabla u \cdot \nabla p \, dx \, dt + \int_D (u(x, 0) - m)p_0 \, dx,$$

where $p \in \mathcal{V}$ and $p_0 \in \mathcal{M}$ are the Lagrangian multiplier, i.e., the adjoint variables, for, respectively, the advection-diffusion PDE and initial condition in the forward problem (5.13). Expressions needed to compute the gradient of (5.15) are obtained by setting variations of the Lagrangian $\mathcal{L}$ with respect to $p$, $p_0$ and $u$ to zero. Variations with respect to $p$ and $p_0$ recover the variational form (5.14) of the forward problem. The
Thus, the adjoint of the parameter-to-observable map

\[ \frac{1}{\sigma^2} \sum_{i=1}^{n_s} \int_{T_i} (Bu - d_i) \tilde{u}_i \, dt + \int_0^T \int_D (\tilde{u}_i + v \cdot \nabla \tilde{u}) p \, dx \, dt + \int_0^T \int_D \kappa \nabla \tilde{u} \cdot \nabla p \, dx \, dt + \int_D \tilde{u}(x, 0) p_0 \, dx = 0 \quad \forall \tilde{u} \in \mathcal{V}. \]

Integration by parts in time for the term \( \tilde{u}_i p \) and in space for \((v \cdot \nabla \tilde{u}) p \) and \( \kappa \nabla \tilde{u} \cdot \nabla p \) results in

\[ \frac{1}{\sigma^2} \sum_{i=1}^{n_s} \int_{T_i} (Bu - d_i) \tilde{u}_i \, dt - \int_0^T \int_D (\tilde{p}_i + \nabla \cdot (vp) + \kappa \Delta p) \tilde{u} \, dx \, dt + \int_D \tilde{u}(x, T) p(x, T) \]

\[ - \tilde{u}(x, 0) p(x, 0) + \tilde{u}(x, 0) p_0 \, dx + \int_0^T \int_{\partial D} (vp + \kappa \nabla p) \cdot n \tilde{u} \, dx \, dt = 0, \]

\( \tilde{u} \in \mathcal{V} \). This implies \( p_0 = p(x, 0) \) and leads to the strong form of the adjoint problem,

\[ -p_t - \nabla \cdot (pv) - \kappa \Delta p = -\frac{1}{\sigma^2} B^* \sum_{i=1}^{n_s} (Bu - d_i) \delta_{m_i} \quad \text{in} \ D \times (0, T), \]

\[ p(\cdot, T) = 0 \quad \text{in} \ D, \]

\[ (vp + \kappa \nabla p) \cdot n = 0 \quad \text{on} \ \partial D \times (0, T). \]

Note that (5.16) is a final value problem, since \( p \) is specified at \( t = T \) rather than at \( t = 0 \). Thus, (5.16) is solved backwards in time, which amounts to the solution of an advection-diffusion equation with velocity \(-v\).

Finally, the variation of the Lagrangian with respect to the initial condition \( m \) in a direction \( \tilde{m} \) gives the weak form of the gradient of the cost functional \( J(m) \),

\[ (G(m), \tilde{m}) = \int_D (A(m - m_pr))(A\tilde{m}) - p(x, 0)\tilde{m} \, dx, \quad \forall \tilde{m} \in \mathcal{M}, \]

(5.17)

where we used \( p_0 = p(x, 0) \). The strong form of the gradient expression then reads

\[ G(m) = \begin{cases} A^2(m - m_pr) - p(x, 0) & \text{in} \ D \\ \gamma \nabla m \cdot n + \beta m & \text{in} \ \partial D. \end{cases} \]

(5.18)

Thus, the adjoint of the parameter-to-observable map \( F^* \) is defined by setting \( F^* d = p(x, 0) \). We note that the gradient expression in (5.18) is linear in \( m \), since \( p \) depends linearly on \( u \) through the solution of the adjoint problem (5.16), and \( u \) depends linearly on \( m \) through the solution of the forward problem (5.13). Elimination of the forward and adjoint equations for a given \( m \) gives the action of the linear operator in (5.18) in the direction of that \( m \). Thus, to compute the MAP point, we use CG to set \( G(m) = 0 \) with the inverse of the regularization operator \( A^2 \) as preconditioner.

5.2.2. Numerical results. Next, we present numerical results for the initial condition inverse problem. The discretization of the forward and adjoint problems uses an unstructured triangular mesh. Galerkin finite elements with piecewise-quadratic globally-continuous polynomials, and an implicit Euler method for the time discretization. Galerkin Least-Squares stabilization of the convective term [40] is added to ensure stability of the discretization. The space-time dimension of the state variable is
433880 (10847 spatial degree of freedom times 40 time steps), and the dimension of the parameter space is 10847. The data dimension $q$ is 1200 with $n_t = 80$ measurement locations and $n_s = 15$ time samples.

To illustrate properties of the forward problem, Figures 5.8 and 5.9 show three snapshots in time of the field $u$, using the advective velocity $v$ from Figure 5.7 with the “true” initial condition and its MAP point estimate, respectively. Next we study the numerical rank of the prior-preconditioned data misfit Hessian. Note that due to linearity of the parameter-to-observable map $F$, the prior-preconditioned data misfit Hessian is independent of $m$. Figure 5.10 shows a logarithmic plot of the truncated spectra of the prior-preconditioned data misfit Hessians for several observation time horizons. This plot shows that the spectrum decays rapidly and, as expected, the decay is faster when the observation time horizon is shorter (i.e., there are fewer observations). As seen in (4.2), an accurate low-rank based approximation of the inverse Hessian can be obtained by neglecting eigenvalues that are small compared to 1. Thus, retaining around 70 eigenvectors out of 10847 appears to be sufficient for the target problem with spatial and temporal observation points in the interval $[1, 4]$. These eigenvalues and the corresponding prior-orthogonal eigenvectors (see Figure 5.11) were computed using the double pass algorithm Algorithm 2 with $r = 50$ and oversampling parameter $l = 10$.

Due to the linearity of the parameter-to-observable map and the choice of a Gaussian prior and noise model, the posterior distribution is also Gaussian whose mean coincides with the MAP point and the covariance with the inverse of the Hessian evaluated at the MAP point. Thus, for this problem, the Laplace approximation is the posterior distribution. Figure 5.12 depicts the prior and posterior pointwise variances. This figure shows that the uncertainty is reduced everywhere in the domain, and that the reduction is greatest near to the observations (at the boundaries of the interior rectangles). In Figure 5.13 we show samples (of the initial condition) from the prior and from the posterior, respectively. The difference between the two sets of samples reflects the information gained from the data in solving the inverse problem. The small differences in the parameter field $m$ across the posterior samples (other than near the external boundaries) demonstrate that there is small variability in the inferred parameters, reflecting large uncertainty reduction.

6. Conclusions. We have presented an extensible software framework for large-scale deterministic and linearized Bayesian inverse problems governed by partial dif-
Fig. 5.9. Forward advective-diffusive transport estimate of the inverse solution at initial time $t = 0$ (a), at $t = 1$ (b), $t = 2$ (c) and at final time $t = 4$ (d) with the MAP as initial condition.

Fig. 5.10. Log-linear plot of the truncated spectrum of prior-preconditioned data misfit Hessian for observation times (sampled every 0.2 time units) in the intervals $[1, 4]$ (blue), $[2, 4]$ (red), and $[3, 4]$ (green). The low-rank approximation captures the dominant, data-informed portion of the spectrum. The eigenvalues are truncated at around 0.06.

The contributions of our work are as follows. On the algorithm side, our framework incorporates modifications of state-of-the-art algorithms to ensure consistency with infinite-dimensional settings and a novel square-root-free implementation of the low-rank approximation of the Hessian, sampling strategies, and pointwise variance field computation. On the software side, we created a library for the solution of de-
terministic and linearized Bayesian inverse problems that allows researchers who are familiar with variational methods to solve inverse problems under uncertainty even without possessing expertise in all of the necessary numerical optimization and statistical aspects. Our framework provides dimension-independent algorithms for finding the maximum a posteriori (MAP) point, constructing a low-rank based approximation of the Hessian and its inverse at the MAP, sampling from the prior and posterior distributions, and computing pointwise variance fields. hIPPYlib is easily extensible, that is, if a user can express the forward problem in variational form using FEniCS, hIPPYlib effortlessly allows solving the inverse problem, exploring and testing various priors, observation operators, noise covariance models, etc.

The framework presented here relies on a second order Taylor expansion of the negative log-likelihood with respect to the uncertain parameter centered at the MAP point, which leads to the Laplace approximation of the posterior distribution. Ultimately one would like to relax this approximation and fully explore the resulting non-Gaussian distributions. Ongoing work includes the implementation of scalable, robust, Hessian-based MCMC methods capitalizing on hIPPYlib’s capabilities to build local Laplace approximations of the posterior based on gradient and Hessian information as described here.
Fig. 5.13. Top: Prior mean initial concentration $m_{pr}$ (a), and samples drawn from the prior distribution (b)–(d). Bottom: The MAP point (e), and samples drawn from the posterior distribution (f)–(h).

Acknowledgement. We thank Georg Stadler for providing very helpful comments on an earlier version of this paper.

REFERENCES

[1] Brian M Adams, WJ Bohnhoff, KR Dalbey, JP Eddy, MS Eldred, DM Gay, K Haskell, Patricia D Hough, and LP Swiler, DAKOTA, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: version 5.0 user’s manual, tech. report, 2009.

[2] Volkan Arçelik, George Biros, Omar Ghattas, Judith Hill, David Keyes, and Bart van Bloemen Waanders, Parallel PDE-constrained optimization, in Parallel Processing for Scientific Computing, M. Heroux, P. Raghaven, and H. Simon, eds., SIAM, 2006.

[3] Alen Alexanderian, Noemi Petra, Georg Stadler, and Omar Ghattas, A-optimal design of experiments for infinite-dimensional Bayesian linear inverse problems with regularized $\ell_0$-sparsification, SIAM Journal on Scientific Computing, 36 (2014), pp. A2122–A2148.

[4] Alen Alexanderian, Noemi Petra, Georg Stadler, and Omar Ghattas, A fast and scalable method for A-optimal design of experiments for infinite-dimensional Bayesian nonlinear inverse problems, SIAM Journal on Scientific Computing, 38 (2016), pp. A243–A272.

[5] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Karl Rupp, Barry F. Smith, and Hong Zhang, PETSc Web page. http://www.mcs.anl.gov/petsc, 2014.

[6] Wolfgang Bangerth, Ralf Hartmann, and Guido Kanschat, deal.II – a general-purpose object-oriented finite element library, ACM Transactions on Mathematical Software, 33 (2007), p. 24.

[7] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, M. Ohlberger, and O. Sander, A generic grid interface for parallel and adaptive scientific computing. Part I: Abstract framework, Computing, 82 (2008), pp. 103–119.

[8] E. B. Becker, G. F. Carey, and J. T. Oden, Finite Elements: An Introduction, Vol I, Prentice Hall, Englewoods Cliffs, New Jersey, 1981.

[9] Costas Bekas, Effrosyni Kokkolarou, and Yousef Saad, An estimator for the diagonal of a matrix, Applied Numerical Mathematics, 57 (2007), pp. 1214–1229.

[10] Alexandros Beskos, Mark Girolami, Shiwei Lan, Patrick E Farrell, and Andrew M Stuart, Geometric MCMC for infinite-dimensional inverse problems, Journal of Computational Physics, 335 (2017), pp. 327–351.
Alfio Borzì and Volker Schulz, *Computational Optimization of Systems Governed by Partial Differential Equations*, SIAM, 2012.

Tan Bui-Thanh, Carsten Burstedde, Omar Ghattas, James Martin, Georg Stadler, and Lucas C. Wilcox, *Extreme-scale UQ for Bayesian inverse problems governed by PDEs*, in SC12: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2012. Gordon Bell Prize finalist.

Tan Bui-Thanh and Omar Ghattas, *Analysis of the Hessian for inverse scattering problems. Part II: Inverse medium scattering of acoustic waves*, Inverse Problems, 28 (2012), p. 055002.

Tan Bui-Thanh, Carsten Burstedde, Omar Ghattas, James Martin, Georg Stadler, and Lucas C. Wilcox, *Extreme-scale UQ for Bayesian inverse problems governed by PDEs*, in SC12: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2012. Gordon Bell Prize finalist.

Tan Bui-Thanh and Omar Ghattas, *Analysis of the Hessian for inverse scattering problems. Part II: Inverse medium scattering of acoustic waves*, Inverse Problems, 28 (2012), p. 055002.

Tan Bui-Thanh and Omar Ghattas, *Analysis of the Hessian for inverse scattering problems. Part II: Inverse medium scattering of acoustic waves*, Inverse Problems, 28 (2012), p. 055002.

Tan Bui-Thanh and Omar Ghattas, *Analysis of the Hessian for inverse scattering problems. Part II: Inverse medium scattering of acoustic waves*, Inverse Problems, 28 (2012), p. 055002.

Tan Bui-Thanh and Omar Ghattas, *Analysis of the Hessian for inverse scattering problems. Part II: Inverse medium scattering of acoustic waves*, Inverse Problems, 28 (2012), p. 055002.

Tan Bui-Thanh and Omar Ghattas, *Analysis of the Hessian for inverse scattering problems. Part II: Inverse medium scattering of acoustic waves*, Inverse Problems, 28 (2012), p. 055002.

Tan Bui-Thanh and Omar Ghattas, *Analysis of the Hessian for inverse scattering problems. Part II: Inverse medium scattering of acoustic waves*, Inverse Problems, 28 (2012), p. 055002.

Tan Bui-Thanh and Omar Ghattas, *Analysis of the Hessian for inverse scattering problems. Part II: Inverse medium scattering of acoustic waves*, Inverse Problems, 28 (2012), p. 055002.

Tan Bui-Thanh and Omar Ghattas, *Analysis of the Hessian for inverse scattering problems. Part II: Inverse medium scattering of acoustic waves*, Inverse Problems, 28 (2012), p. 055002.

Tan Bui-Thanh and Omar Ghattas, *Analysis of the Hessian for inverse scattering problems. Part II: Inverse medium scattering of acoustic waves*, Inverse Problems, 28 (2012), p. 055002.

Tan Bui-Thanh and Omar Ghattas, *Analysis of the Hessian for inverse scattering problems. Part II: Inverse medium scattering of acoustic waves*, Inverse Problems, 28 (2012), p. 055002.

Tan Bui-Thanh and Omar Ghattas, *Analysis of the Hessian for inverse scattering problems. Part II: Inverse medium scattering of acoustic waves*, Inverse Problems, 28 (2012), p. 055002.

Tan Bui-Thanh and Omar Ghattas, *Analysis of the Hessian for inverse scattering problems. Part II: Inverse medium scattering of acoustic waves*, Inverse Problems, 28 (2012), p. 055002.

Tan Bui-Thanh and Omar Ghattas, *Analysis of the Hessian for inverse scattering problems. Part II: Inverse medium scattering of acoustic waves*, Inverse Problems, 28 (2012), p. 055002.

Tan Bui-Thanh and Omar Ghattas, *Analysis of the Hessian for inverse scattering problems. Part II: Inverse medium scattering of acoustic waves*, Inverse Problems, 28 (2012), p. 055002.

Tan Bui-Thanh and Omar Ghattas, *Analysis of the Hessian for inverse scattering problems. Part II: Inverse medium scattering of acoustic waves*, Inverse Problems, 28 (2012), p. 055002.

Tan Bui-Thanh and Omar Ghattas, *Analysis of the Hessian for inverse scattering problems. Part II: Inverse medium scattering of acoustic waves*, Inverse Problems, 28 (2012), p. 055002.
[34] Pearl H. Flath, Hessians-based response surface approximations for uncertainty quantification in large-scale statistical inverse problems, with applications to groundwater flow, PhD thesis, The University of Texas at Austin, 2013.

[35] Pearl H. Flath, Lucas C. Wilcox, Volkan Akcøelik, Judy Hill, Bart van Bloemen Waanders, and Omar Ghattas, Fast algorithms for Bayesian uncertainty quantification in large-scale linear inverse problems based on low-rank partial Hessian approximations, SIAM Journal on Scientific Computing, 33 (2011), pp. 407–432.

[36] Nathan Halko, Per Gunnar Martinsson, and Joel A. Tropp, Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, SIAM Review, 53 (2011), pp. 217–288.

[37] Matthias Heinonen, Mesh independence for nonlinear least squares problems with norm constraints, SIAM Journal on Optimization, 3 (1993), pp. 81–117.

[38] Michael A. Heroux, Roscoe A. Bartlett, Vicki E. Howle, Robert J. Hoekstra, Jonathan J. Hu, Tamara G. Kolda, Richard B. Lehoucq, Kevin R. Long, Roger P. Pawlowski, Eric T. Phipps, Andrew G. Salinger, Heidi K. Thornquist, Ray S. Tuminaro, James M. Willenbring, Alan Williams, and Kendall S. Stanley, An overview of the Trilinos project, ACM Transactions on Mathematical Software, 31 (2005), pp. 397–423.

[39] James Martin, Lucas C. Wilcox, Carsten Burstedde, and Omar Ghattas, Scalable and efficient algorithms for the propagation of uncertainty from data through inference to prediction for large-scale problems, with application to flow of the Antarctic ice sheet, Journal of Computational Physics, 296 (2015), pp. 348–368.

[40] Jari Kaipio and Erkki Somersalo, Statistical and Computational Inverse Problems, vol. 160 of Applied Mathematical Sciences, Springer-Verlag, New York, 2005.

[41] D. Kouri, D. Ridzal, and G. von Winckel, Rapid Optimization Library (ROL). https://trilinos.org/packages/rol/, 2018.

[42] Hans Petter Langtangen and Anders Logg, Solving PDEs in Python, Springer, 2017.

[43] Finn Lindgren, Håvard Rue, and Johan Lindström, An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 73 (2011), pp. 423–498.

[44] Anders Logg, Kent-Andre Mardal, and Garth N. Wells, Automated Solution of Differential Equations by the Finite Element Method, vol. 84 of Lecture Notes in Computational Science and Engineering, Springer, 2012.

[45] Quan Long, Mohammad Motamed, and Raúl Tempone, Fast Bayesian optimal experimental design for seismic source inversion, Computer Methods in Applied Mechanics and Engineering, 291 (2015), pp. 849 – 876.

[46] Quan Long, Marco Scavino, Raúl Tempone, and Suojin Wang, Fast estimation of expected information gains for Bayesian experimental designs based on Laplace approximations, Computer Methods in Applied Mechanics and Engineering, 259 (2013), pp. 24–39.

[47] Quan Long, Marco Scavino, Raúl Tempone, and Suojin Wang, A Laplace method for under-determined Bayesian optimal experimental designs, Computer Methods in Applied Mechanics and Engineering, 285 (2015), pp. 849 – 876.

[48] R. Lowery and J. Langou, Stability Analysis of QR factorization in an Oblique Inner Product, ArXiv e-prints, (2014).

[49] James Martin, Lucas C. Wilcox, Carsten Burstedde, and Omar Ghattas, A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion, SIAM Journal on Scientific Computing, 34 (2012), pp. A1460–A1487.

[50] Albert Parker and Colin Fox, Sampling Gaussian distributions in Krylov spaces with conjugate gradients, SIAM Journal on Scientific Computing, 34 (2012), pp. B312–B334.

[51] Noemi Petra, James Martin, Georg Stadler, and Omar Ghattas, A computational frame-
work for infinite-dimensional Bayesian inverse problems: Part II. Stochastic Newton MCMC with application to ice sheet inverse problems, SIAM Journal on Scientific Computing, 36 (2014), pp. A1525–A1555.

[56] Noemi Petra and Georg Stadler, Model variational inverse problems governed by partial differential equations, Tech. Report 11-05, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, 2011.

[57] S. J. Press, Subjective and Objective Bayesian Statistics: Principles, Methods and Applications, Wiley, New York, 2003.

[58] Ernesto E. Prudencio and Karl W. Schulz, The Parallel C++ Statistical Library ‘QUESO’: Quantification of Uncertainty for Estimation, Simulation and Optimization, in Euro-Par 2011: Parallel Processing Workshops, Michael Alexander, Pasqua D’Ambra, Adam Belloum, George Bosilca, Mario Cannataro, Marco Danelutto, Beniamino Di Martino, Michael Gerndt, Emmanuel Jeannot, Raymond Namyst, Jean Roman, Stephen L. Scott, Jesper Larsson Tøraa, Geoffrey Vallee, and Josef Weidendorfer, eds., vol. 7155 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 398–407.

[59] L. Ruthotto, E. Treister, and E. Haber, jInv – a flexible Julia package for PDE parameter estimation, SIAM Journal on Scientific Computing, 39 (2017), pp. S702–S722.

[60] A. K. Saibaba, J. Lee, and P. K. Kitanidis, Randomized algorithms for generalized Hermitian eigenvalue problems with application to computing Karhunen-Loève expansion, Numerical Linear Algebra with Applications, 23 (2016), pp. 314–339. nla.2026.

[61] Schillings, Claudia and Schwab, Christoph, Scaling limits in computational Bayesian inversion, ESAIM: M2AN, 50 (2016), pp. 1825–1856.

[62] T. Steihaug, Local and superlinear convergence for truncated iterated projections methods, Mathematical Programming, 27 (1983), pp. 176–190.

[63] Stigler, S. M., Laplace’s 1774 Memoir on Inverse Probability, Statist. Sci., 1 (1986), pp. 359–363.

[64] G. Strang and G. J. Fix, An Analysis of the Finite Element Method, Wellesley-Cambridge Press, Wellesley, MA, 1988.

[65] Andrew M. Stuart, Inverse problems: A Bayesian perspective, Acta Numerica, 19 (2010), pp. 451–559.

[66] Jok M Tang and Yousef Saad, A probing method for computing the diagonal of a matrix inverse, Numerical Linear Algebra with Applications, 19 (2012), pp. 485–501.

[67] Albert Tarantola, Inverse Problem Theory and Methods for Model Parameter Estimation, SIAM, Philadelphia, PA, 2005.

[68] L. Tierney and J. B. Kadane, Accurate Approximations for Posterior Moments and Marginal Densities, Journal of the American Statistical Association, 81 (1986), pp. 82–86.

[69] A. N. Tikhonov and V. A. Arsenin, Solution of Ill-posed Problems, Winston & Sons, 1977.

[70] C. Tong, Problem Solving Environment for Uncertainty Analysis and Design Exploration, Springer International Publishing, 2017, pp. 1695–1731.

[71] Frederic Tröltzsch, Optimal Control of Partial Differential Equations: Theory, Methods and Applications, vol. 112 of Graduate Studies in Mathematics, American Mathematical Society, 2010.

[72] Curt R. Vogel, Computational Methods for Inverse Problems, Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002.

[73] David Williams, Probability with Martingales, Cambridge University Press, 1991.

[74] R. Wong, Asymptotic Approximations of Integrals, Society for Industrial and Applied Mathematics, 2001.

[75] Jennifer Worthen, Inverse Problems in Mantle Convection: Models, Algorithms, and Applications, PhD thesis, The University of Texas at Austin, 2012.

[76] Jennifer Worthen, Georg Stadler, Noemi Petra, Michael Gurnis, and Omar Ghattas, Towards adjoint-based inversion for rheological parameters in nonlinear viscous mantle flow, Physics of the Earth and Planetary Interiors, 234 (2014), pp. 23–34.

Appendix A. Finite element assembly and rectangular decompositions.

In this section, we describe a technique based on rectangular decompositions of finite element matrices to efficiently generate large scale samples from priors of the form described in Section 2.2. More specifically, we present a finite element assembly procedure to compute a rectangular decomposition of the form

$$A = CC^T,$$  \hspace{1cm} (A.1)

for any symmetric positive definite finite element matrix $A$. More specifically, consider
the finite element assembly procedure for a generic symmetric positive definite bilinear form \( a(u_h, v_h) \) on \( \mathcal{M}_h \), a finite-dimensional subspace of \( \mathcal{M} \subseteq L^2(\mathcal{D}) \). The entry \((i, j)\) of the matrix \( A \), which stems from finite element discretization of the bilinear form \( a(u_h, v_h) \), are given by

\[
A_{i,j} = a(\phi_i, \phi_j), \quad i, j = 1, \ldots, n,
\]

where \( \{\phi_i\}_{i=1}^n \) is the finite element basis of the space \( \mathcal{M}_h \). In the finite element assembly procedure we first compute the element matrices \( A_e \), which correspond to the restriction of the bilinear form \( a \) to each element \( e \) in the mesh. Then, using the global-to-local mapping of the degrees of freedom (dof) \( G_e \), the global matrix \( A \) is computed by summing all of the local contributions as follows:

\[
A = \sum_e G_e^T A_e G_e = \sum_e G_e^T B^T D_e B G_e. \quad (A.2)
\]

Here we have written the element matrix \( A_e = B^T D_e B \) as the product of the element-independent dof-to-quadrature point basis evaluation matrix \( B \) and the (block) diagonal matrix \( D_e \in \mathbb{R}^{q \times q} \) at the quadrature points, where \( q \) denotes the total number of quadrature nodes over all elements, which scales linearly with the number of elements in the mesh.

A rectangular decomposition of \( A \) can then be explicitly constructed from the matrices \( G_e \), \( B \), and \( D_e \) as follows. For each element \( e \) of the mesh we define the matrix \( C_e = G_e^T B^T D_e B G_e \). Since, for any two elements \( e_i \) and \( e_j \) \((i \neq j)\) in the mesh, the sets of quadrature nodes relative to the elements \( e_i \) and \( e_j \) are disjoint, we have that

\[
C_{e_i} C_{e_j}^T = \delta_{ij} A_{e_i}. \quad (A.3)
\]

Then the rectangular matrix \( C \in \mathbb{R}^{n \times q} \) defined as

\[
C = \sum_e G_e^T B^T D_e^\frac{1}{2} = \sum_e C_e \quad (A.4)
\]

satisfies \((A.1)\). In fact, thanks to \((A.3)\) we have

\[
CC^T = \left( \sum_e C_e \right) \left( \sum_e C_e \right)^T = \sum_e C_e C_e^T = \sum_e G_e^T B^T D_e B G_e = A.
\]