hIPPYlib: An Extensible Software Framework for Large-Scale Inverse Problems Governed by PDEs: Part I: Deterministic Inversion and Linearized Bayesian Inference

UMBERTO VILLA, Washington University in St. Louis, USA
NOEMI PETRA, University of California, Merced, USA
OMAR GHATTAS, The University of Texas at Austin, USA

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 prohibitively expensive 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 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 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. 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 discussed. hIPPYlib makes all of these advanced algorithms easily accessible to domain scientists and provides an environment that expedites the development of new algorithms.

CCS Concepts: • **Mathematics of computing** → **Bayesian computation**; **Mathematical optimization**; **Partial differential equations**; *Computations on matrices*; *Discretization*; *Solvers*; • **Computing methodologies** → **Uncertainty quantification**; • **Applied computing** → *Physical sciences and engineering*;

Additional Key Words and Phrases: Infinite-dimensional inverse problems, adjoint-based methods, inexact Newton-CG method, low-rank approximation, Bayesian inference, uncertainty quantification, sampling, generic PDE toolkit

16

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. That is, given (possibly noisy) data and (a possibly uncertain) model, the goal becomes to 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 nonunique 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. Nonuniqueness 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 also we must 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 article, we present hIPPYlib, an **I**nverse **P**roblems **Py**thon **lib**rary (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) [Logg et al. 2012] for the discretization of the PDEs, and on PETSc [Balay et al. 2014] 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 clearly displays and utilizes 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.[1] 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 accurate than the one proposed in Saibaba et al. [2016].

(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 Croci et al. [2018] 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 Bekas et al. [2007]. 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 through `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` [COMSOL AB 2009], `deal.II` [Bangerth et al. 2007], `dune` [Bastian et al. 2008], `FEniCS` [Langtangen and Logg 2017; Logg et al. 2012], and

---

[1]`hIPPYlib` is currently used to teach several graduate-level classes on inverse problems at various universities, including the University of Texas at Austin, University of California, Merced, Washington University in St. Louis, New York University, and North Carolina State. `hIPPYlib` has also been demonstrated with hands-on interactive sessions at workshops and summer schools, such as the 2015 ICERM IdeaLab, the 2016 SAMSI Optimization Program Summer School, and the 2018 Gene Golub SIAM Summer School.

Sundance, a package from `Trilinos` [Heroux et al. 2005]. 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. Farrell et al. [2013] 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. Ruthotto et al. [2017] 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) [Kouri et al. 2018] 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 [McDougall et al. 2017; Prudencio and Schulz 2012], DAKOTA [Adams et al. 2009; Eldred et al. 2002], PSUADE [Tong 2017], and UQTk [Debusschere et al. 2017]. 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 [Parno et al. 2015] 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 article 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 [Engl et al. 1996; Vogel 2002] and Bayesian formulation [Kaipio and Somersalo 2005; Tarantola 2005] of inverse problems. Specifically, we adopt the infinite-dimensional Bayesian inference framework [Stuart 2010], and we refer to [Alexanderian et al. 2014, 2016; Bui-Thanh et al. 2013; Petra et al. 2014] 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[2] m that characterize a physical process modeled by PDEs. Mathematically this inverse relationship is expressed as

$$d = \mathcal{F}(m) + \eta, \tag{1}$$

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 $\mathcal{D} \subset \mathbb{R}^d$ 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

$$\mathcal{F}(m) = \mathcal{B}(u), \text{ s.t. } r(u, m) = 0, \tag{2}$$

where $\mathcal{B} : \mathcal{V} \rightarrow \mathbb{R}^q$ is a (possibly nonlinear) observation operator, and $r : \mathcal{V} \times \mathcal{M} \rightarrow \mathcal{V}^*-$ referred as the forward problem from now on—represents the PDE problem. The observations d contain noise due to measurement uncertainties and model errors [Tarantola 2005]. In Equation (1), this is captured by the additive noise η, which in hIPPYlib is modeled as $\eta \sim \mathcal{N}(0, \Gamma_{\text{noise}})$, i.e., a centered Gaussian at 0 with covariance Γ_{noise}. A significant difficulty when solving infinite-dimensional inverse problems is that typically these are not well posed (in the sense of Hadamard [Tikhonov and Arsenin 1977]). 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 [Engl et al. 1996; Vogel 2002].

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:

$$\min_{m \in \mathcal{M}} \mathcal{J}(m) := \frac{1}{2} \|\mathcal{F}(m) - d\|^2_{\Gamma_{\text{noise}}^{-1}} + \mathcal{R}(m), \tag{3}$$

where the first term in the cost functional, $\mathcal{J}(m)$, 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, $\mathcal{R}(m)$, imposes regularity on the inversion field m, such

---

[2]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.

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 [Engl et al. 1996; Vogel 2002].

As we will explain in Section 4.1, to efficiently solve the nonlinear least-squares problem (Equation (3)) with parameter-to-observable-map $\mathcal{F}$ implicitly defined as in Equation (2), first and second derivative information are needed. Using the Lagrangian formalism [Tröltzsch 2010], 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 to as the adjoint, and write the Lagrangian functional

$$\mathcal{L}^{\mathcal{G}}(u, m, p) := \frac{1}{2}\|\mathcal{B}(u) - d\|^2_{\Gamma^{-1}_{\text{noise}}} + \mathcal{R}(m) + {}_{\mathcal{V}}\langle p, r(u, m)\rangle_{\mathcal{V}^*},$$

where ${}_{\mathcal{V}}\langle \cdot, \cdot \rangle_{\mathcal{V}^*}$ denotes the duality pair between $\mathcal{V}$ and its adjoint. The gradient for the cost functional (Equation (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

$$(\mathcal{G}(m_0), \tilde{m}) = (\mathcal{R}_m(m_0), \tilde{m}) + {}_{\mathcal{V}}\langle p_0, r_m(u_0, m_0)[\tilde{m}]\rangle_{\mathcal{V}^*}, \quad \forall \tilde{m} \in \mathcal{M}, \tag{4}$$

where $(\mathcal{R}_m(m_0), \tilde{m}) \in \mathbb{R}$ denotes the Gâteaux derivative of $\mathcal{R}$ with respect to m in the direction $\tilde{m}$ evaluated at $m = m_0$, and $r_m(u_0, m_0)[\tilde{m}] \in \mathcal{V}^*$ the Gâteaux derivative of r with respect to m in the direction $\tilde{m}$ 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

$$_{\mathcal{V}}\langle \tilde{p}, r(u_0, m_0)\rangle_{\mathcal{V}^*} = 0 \quad \forall \tilde{p} \in \mathcal{V}, \tag{5}$$

and p_0 solves the adjoint problem

$$_{\mathcal{V}}\langle p_0, r_u(u_0, m_0)[\tilde{u}]\rangle_{\mathcal{V}^*} + \langle \mathcal{B}_u(u_0)[\tilde{u}], \mathcal{B}(u) - d\rangle_{\mathbb{R}^q} = 0, \quad \forall \tilde{u} \in \mathcal{V}. \tag{6}$$

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

$$\begin{aligned}\mathcal{L}^{\mathcal{H}}(u, m, p; \hat{u}, \hat{m}, \hat{p}) := {}&(\mathcal{G}(m), \hat{m}) \\ &+ {}_{\mathcal{V}}\langle \hat{p}, r(u, m)\rangle_{\mathcal{V}^*} \\ &+ {}_{\mathcal{V}}\langle p, r_u(u, m)[\hat{u}]\rangle_{\mathcal{V}^*} + \langle \mathcal{B}_u(u)[\hat{u}], \mathcal{B}(u) - d\rangle_{\mathbb{R}^q},\end{aligned} \tag{7}$$

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 $\mathcal{L}^{\mathcal{H}}$ with respect to m and reads

$$\begin{aligned}(\tilde{m}, \mathcal{H}(m_0)\hat{m}) = {}&(\tilde{m}, \mathcal{R}_{mm}(m_0)[\hat{m}]) + (p_0, r_{mm}(u_0, m_0)[\tilde{m}, \hat{m}]) \\ &+ {}_{\mathcal{V}}\langle \hat{p}, r_m(u_0, m_0)[\tilde{m}]\rangle_{\mathcal{V}^*} + {}_{\mathcal{V}}\langle p_0, r_{um}(u_0, m_0)[\hat{u}, \tilde{m}]\rangle_{\mathcal{V}^*}, \quad \forall m \in \mathcal{M}.\end{aligned} \tag{8}$$

Here u_0, p_0 are the solution of the forward and adjoint problems (Equations (5) and (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 $\mathcal{L}^{\mathcal{H}}$ with respect to p and u, respectively. In Appendix A, we present a Newton-type algorithm to minimize Equation (3) that uses the expression for the gradient (Equation (4)) and Hessian action (Equation (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 Equation (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

$$\frac{d\mu_{\text{post}}}{d\mu_{\text{prior}}} \propto \pi_{\text{like}}(\boldsymbol{d}|m). \tag{9}$$

Here, $d\mu_{\text{post}}/d\mu_{\text{prior}}$ denotes the Radon-Nikodym derivative [Williams 1991] of the posterior measure μ_{post} with respect to μ_{prior}, and $\pi_{\text{like}}(\boldsymbol{d}|m)$ denotes the data likelihood. Conditions under which the posterior measure is well defined and Equation (9) holds are given in detail in Stuart [2010].

The noise model and the likelihood. In our hIPPYlib framework, we assume an additive noise model, $\boldsymbol{d} = \mathcal{F}(m) + \boldsymbol{\eta}$, where $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{0}, \Gamma_{\text{noise}})$ is a centered Gaussian on $\mathbb{R}^q$. This implies

$$\pi_{\text{like}}(\boldsymbol{d}|m) \propto \exp\{-\Phi(m)\}, \tag{10}$$

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

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

$$d\mu_{\text{prior}}(m) \propto \exp\left\{-\frac{1}{2}\|m - m_{\text{pr}}\|^2_{C_{\text{prior}}^{-1}}\right\}. \tag{11}$$

If the parameter represents a spatially correlated field defined on $\mathcal{D} \in \mathbb{R}^d$, the prior covariance operator C_{prior} usually imposes smoothness on the parameter. This is because rough components of the parameter field 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 νth power ($\nu > \frac{d}{2}$) of a Laplacian-like operator, namely $C_{\text{prior}} := \mathcal{A}^{-\nu} = (-\gamma\,\Delta + \delta I)^{-\nu}$, where γ and $\delta > 0$ control the correlation length ρ and the pointwise variance σ^2 of the prior operator. Specifically, ρ—empirically defined as the distance ρ for which the two-point correlation coefficient is 0.1—is proportional to $\sqrt{\gamma/\delta}$, and σ^2 is proportional to $\delta^{-\nu}\rho^{-d}$ (see, e.g.. Lindgren et al. [2011], where exact expressions for ρ and σ^2 as functions of γ and δ are derived under the assumption of unbounded domain $\mathcal{D}$ and constant coefficients γ and δ). The coefficients γ and δ can be constant (in which case the prior is stationary) or spatially varying. In addition, one can consider an anisotropic diffusion operator $\mathcal{A} = -\gamma\,\nabla \cdot (\Theta\,\nabla) + \delta I$, with Θ a symmetric positive definite (s.p.d.) tensor that models, for instance, stronger correlations in a specific direction. These choices of prior ensure that C_{prior} is a trace-class operator, guaranteeing bounded pointwise variance and a well-posed infinite-dimensional Bayesian inverse problem [Bui-Thanh et al. 2013; Stuart 2010].

The posterior. Using the expression for the likelihood function (Equation (10)) and prior distribution (Equation (11)) , the posterior distribution in Equation (9) reads

$$d\mu_{\text{post}} \propto \exp\left\{-\frac{1}{2}\|\mathcal{F}(m) - \boldsymbol{d}\|^2_{\Gamma_{\text{noise}}^{-1}} - \frac{1}{2}\|m - m_{\text{pr}}\|^2_{C_{\text{prior}}^{-1}}\right\}. \tag{12}$$

The MAP point m_{MAP} is defined as the parameter field that maximizes the posterior distribution. It can be obtained by solving the following deterministic optimization problem:

$$m_{\text{MAP}} := \underset{m \in \mathcal{M}}{\text{argmin}}(-\log d\mu_{\text{post}}(m)) = \underset{m \in \mathcal{M}}{\text{argmin}} \frac{1}{2}\|\mathcal{F}(m) - \boldsymbol{d}\|^2_{\Gamma^{-1}_{\text{noise}}} + \frac{1}{2}\|m - m_{\text{pr}}\|^2_{C^{-1}_{\text{prior}}}. \quad (13)$$

We note that the prior information plays the role of Tikhonov regularization in Equation (3); in fact, the deterministic optimization problem (Equation (3)) is the same as Equation (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_{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 [Bui-Thanh et al. 2013]. 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 [Stuart 2010, Section 6.4],

$$C_{\text{post}} = \mathcal{H}^{-1} = (\mathcal{F}^*\Gamma^{-1}_{\text{noise}}\mathcal{F} + C^{-1}_{\text{prior}})^{-1}, \qquad m_{\text{MAP}} = C_{\text{post}}(\mathcal{F}^*\Gamma^{-1}_{\text{noise}}\boldsymbol{d} + C^{-1}_{\text{prior}}m_{\text{pr}}), \quad (14)$$

where $\mathcal{F}^* : \mathbb{R}^q \to \mathcal{M}$ 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 Γ_{noise}, the number q of observations, and the regularity of the parameter-to-observable map $\mathcal{F}$, the *Laplace approximation* [Evans and Swartz 2000; Press 2003; Stigler, S. M. 1986; Tierney and Kadane 1986; Wong 2001] 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_{MAP}[3], the Laplace approximation to the posterior constructs a Gaussian distribution $\hat{\mu}_{\text{post}}$,

$$\hat{\mu}_{\text{post}} \sim \mathcal{N}\left(m_{\text{MAP}}, \mathcal{H}(m_{\text{MAP}})^{-1}\right), \quad (15)$$

centered at m_{MAP} and with covariance operator

$$\mathcal{H}(m_{\text{MAP}})^{-1} = (\mathcal{H}_{\text{misfit}}(m_{\text{MAP}}) + C^{-1}_{\text{prior}})^{-1}. \quad (16)$$

Here $\mathcal{H}_{\text{misfit}}$ denotes the Hessian of the negative log-likelihood evaluated at m_{MAP} (see Section 5 for examples of the derivation of the action of $\mathcal{H}_{\text{misfit}}$ using variational calculus and Lagrangian 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 Bui-Thanh et al. [2013]. Likewise, it has been exploited in Petra et al. [2014] to efficiently explore the true posterior distribution by generating high-quality proposals for Markov chain Monte Carlo algorithms, in Cui et al. [2014] to construct likelihood informed subspaces that allow for optimal dimension reduction in Bayesian inference problems, and in Chen et al. [2017] and Schillings et al. [2016] to construct a dimension-independent sparse grid to evaluate posterior expectations. It has also been invoked in Isaac et al. [2015] for scalable approximation of the predictive posterior distribution of a scalar quantity of interest. Finally, its use was advocated in Alexanderian et al. [2016] and Long et al. [2013, 2015a, 2015b] to approximate the solution of Bayesian optimal experimental design problems.

---

[3]To guarantee a positive definite posterior covariance operator also in the case of nonlocally 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_{MAP}.

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 Bui-Thanh et al. [2013]. 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 $\{\phi_j\}_{j=1}^n$ [Becker et al. 1981; Strang and Fix 1988]. 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, $\boldsymbol{m} = (m_1, \ldots, m_n)^T \in \mathbb{R}^n$ denotes the vector of the coefficients in the finite element expansion of m_h.

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

$$M_{ij} = \int_{\mathcal{D}} \phi_i(\boldsymbol{x}) \phi_j(\boldsymbol{x}) \, d\boldsymbol{x}, \quad i, j = 1, \ldots, n.$$

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

$$\langle \mathbf{B}\boldsymbol{y}, \boldsymbol{z} \rangle_{\mathrm{M}} = \langle \boldsymbol{y}, \mathbf{B}^* \boldsymbol{z} \rangle_{\mathrm{M}},$$

which implies that $\mathbf{B}^*$ is given by

$$\mathbf{B}^* = \mathbf{M}^{-1} \mathbf{B}^T \mathbf{M}. \tag{17}$$

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

$$R_{ij} = \int_{\mathcal{D}} \phi_i(\boldsymbol{x}) \mathcal{A}^v \phi_j(\boldsymbol{x}) \, d\boldsymbol{x}, \quad i, j \in \{1, \ldots, n\}. \tag{18}$$

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

$$\pi_{\mathrm{post}}(\boldsymbol{m}) \propto \exp\left(-\frac{1}{2} \|\mathbf{F}(\boldsymbol{m}) - \boldsymbol{d}_{\mathrm{obs}}\|^2_{\Gamma_{\mathrm{noise}}^{-1}} - \frac{1}{2} \|\boldsymbol{m} - \boldsymbol{m}_{\mathrm{pr}}\|^2_{\Gamma_{\mathrm{prior}}^{-1}} \right), \tag{19}$$

where $\boldsymbol{m}_{\mathrm{pr}}$ is the mean of the prior distribution, $\Gamma_{\mathrm{prior}} := \mathbf{R}^{-1}\mathbf{M} \in \mathbb{R}^{n \times n}$ is the covariance matrix for the prior that arises upon discretization of $\mathcal{C}_{\mathrm{prior}}$, and $\Gamma_{\mathrm{noise}} \in \mathbb{R}^{q \times q}$ 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 (Equation (19)), which results—as discussed in Section 2.2 in the continuous setting—in the Laplace approximation of the posterior given by

$$\pi_{\mathrm{post}}(\boldsymbol{m}) \propto \mathcal{N}(\boldsymbol{m}_{\mathrm{MAP}}, \Gamma_{\mathrm{post}}). \tag{20}$$

The mean of this approximate posterior distribution, $\boldsymbol{m}_{\mathbf{MAP}}$, is the parameter vector maximizing the posterior (Equation (19)) and is known as the MAP point. It can be found by minimizing the

negative log-posterior,[4] which amounts to solving the following optimization problem:

$$m_{MAP} := \underset{m}{\text{argmin}}(-\log \pi_{post}(m)) = \underset{m}{\text{argmin}} \frac{1}{2}\|F(m) - d_{obs}\|^2_{\Gamma^{-1}_{noise}} + \frac{1}{2}\|m - m_{pr}\|^2_{\Gamma^{-1}_{prior}}, \quad (21)$$

which is the discrete counterpart of Equation (13). Denoting with $\tilde{H}(m_{MAP})$ and $\tilde{H}_{misfit}(m_{MAP})$ the matrix representations of, respectively, the second derivative of negative log-posterior $\mathcal{H}$ and log-likelihood $\mathcal{H}_{misfit}$ (i.e., the data misfit component of the Hessian) in the M-weighted inner product, and assuming that $\tilde{H}_{misfit}(m_{MAP})$ is positive definite, the covariance matrix Γ_{post} in the Laplace approximation is given by

$$\Gamma_{post} = \tilde{H}^{-1}(m_{MAP}) = \left(\tilde{H}_{misfit}(m_{MAP}) + \Gamma^{-1}_{prior}\right)^{-1}. \quad (22)$$

For simplicity of the presentation, in the following we will let $H = M\tilde{H}$ and $H_{misfit} = M\tilde{H}_{misfit}$ be the matrix representation of the Hessian with respect to the standard Euclidean inner product. Using this notation, and recalling that $\Gamma_{prior} = R^{-1}M$, we rewrite Equation (22) as

$$\Gamma_{post} = H^{-1}(m_{MAP}) M = (H_{misfit}(m_{MAP}) + R)^{-1} M. \quad (23)$$

Equations (21) and (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 Γ_{post}.

3 DESIGN AND SOFTWARE COMPONENTS OF hIPPYlib

hIPPYlib implements state-of-the-art scalable algorithms for PDE-based deterministic and Bayesian inverse problems. It builds on FEniCS (a parallel finite element library) [Langtangen and Logg 2017; Logg et al. 2012] for discretization of PDEs and on PETSc [Balay et al. 2014] for scalable and efficient linear algebra operations and solvers. In hIPPYlib the user can express the forward PDE and the likelihood in variational form using the friendly, compact, near-mathematical notation of FEniCS, which will then automatically generate efficient code for the discretization. Linear and nonlinear, stationary and time-dependent PDEs are supported in hIPPYlib. For stationary problems, gradient and Hessian information can be automatically generated by hIPPYlib using 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 main components, classes, and functionalities of hIPPYlib are summarized in Figure 1. These include:

(1) The **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. 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 FEniCS variational form or

---

[4]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.
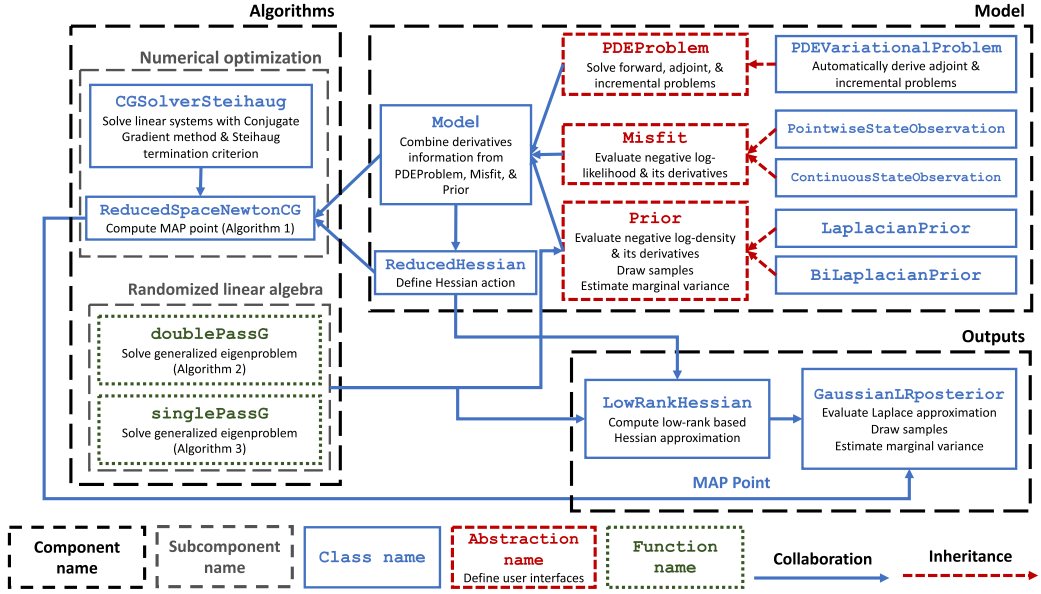
Fig. 1. Design of the `hIPPYlib` framework. Red dotted connectors represent inheritance and blue solid connectors represent collaborations among the main components (black dashed boxes), classes (blue solid boxes and red dashed boxes), and functionalities (green dotted boxes) of `hIPPYlib`. The legend is shown on the bottom.

> (for more complicated or time-dependent problems) as a user-defined object. When using `FEniCS` variational forms, `hIPPYlib` is able to derive expressions for the gradient and Hessian-apply automatically using `FEniCS`' symbolic differentiation. This means that for stationary problems the user will only have to provide the variational form of the forward problem. For more complex problems (e.g., time-dependent problems), `hIPPYlib` allows the user to implement their own derivatives.

(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 inputs 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.

We refer to Villa et al. [2020] for a detailed description of the modules, classes, and functions implemented in `hIPPYlib`.

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 an MCMC sampling method enchanted by Hessian information (see, e.g., [Beskos et al. 2017; Bui-Thanh and Girolami 2014; Cui et al. 2016, 2014c; Martin et al. 2012]). 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-CG algorithm (e.g., [Akçelik et al. 2006; Borzì and Schulz 2012]) to solve the deterministic inverse problem and, in the Bayesian framework, to compute the MAP point (see Algorithm 1). The gradient and Hessian actions—whose expressions are given in Equations (4) and (8), respectively—are automatically computed via their variational form specification in FEniCS by constraining the state and adjoint variables to satisfy the forward and adjoint problem in Equations (5) and (6), respectively. The Newton system is solved inexactly using early termination of CG iterations using Eisenstat–Walker [Eisenstat and Walker 1996] (to prevent oversolving) and Steihaug [Steihaug 1983] (to avoid negative curvature) criteria. Specifically, the choice of the tolerance η_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_{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 [Heinkenschloss 1993]. 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 [Campbell et al. 1996].

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, which are highly desirable when solving inverse problems due to their dimension-independent convergence property. 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 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 eigenvalues 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, $\mathbf{H}_{\text{misfit}}$, can be constructed. This can be proven analytically for certain linear forward PDE problems (e.g., advection-diffusion [Flath et al. 2011], Poisson [Flath 2013], Stokes [Worthen 2012], acoustics [Bui-Thanh and Ghattas 2012, 2013], electromagnetics [Bui-Thanh and Ghattas 2013]) and demonstrated numerically for more complex PDE problems (e.g., seismic wave propagation [Bui-Thanh et al. 2012, 2013], mantle convection [Worthen et al. 2014], ice sheet flow [Isaac et al. 2015; Petra et al. 2014], poroelasticity [Hesse and Stadler 2014], and turbulent flow [Chen et al. 2019]).

ALGORITHM 1: The inexact Newton-CG algorithm to find the MAP point

---

$i \leftarrow 0$
Given $\boldsymbol{m}_0$ solve the forward problem (5) to obtain $\boldsymbol{u}_0$
Given $\boldsymbol{m}_0$ and $\boldsymbol{u}_0$ compute the cost functional $\mathcal{J}_0$ using (3)
while $i <$ max_iter **do**
    Given $\boldsymbol{m}_i$ and $\boldsymbol{u}_i$ solve the adjoint problem (6) to obtain $\boldsymbol{p}_i$
    Given $\boldsymbol{m}_i$, $\boldsymbol{u}_i$ and $\boldsymbol{p}_i$ evaluate the gradient $\boldsymbol{g}_i$ using (4)
    if $\|\boldsymbol{g}_i\| \leq \tau$ **then**
        break
    end if
    Given $\boldsymbol{m}_i$, $\boldsymbol{u}_i$, and $\boldsymbol{p}_i$ define a linear operator $\mathbf{H}_i$ that implements the Hessian action (8)
    Using conjugate gradients, find a search direction $\widehat{\boldsymbol{m}}_i$ such that

$$\|\mathbf{H}_i\widehat{\boldsymbol{m}}_i + \boldsymbol{g}_i\| \leq \eta_i\|\boldsymbol{g}_i\|, \text{ with } \eta_i = \left(\frac{\|\boldsymbol{g}_i\|}{\|\boldsymbol{g}_0\|}\right)^{\frac{1}{2}}$$

    $j \leftarrow 0, \alpha^{(0)} \leftarrow 1$
    while $j <$ max_backtracking_iter **do**
        Set $\boldsymbol{m}^{(j)} = \boldsymbol{m}_i + \alpha^{(j)}\widehat{\boldsymbol{m}}_i$
        Given $\boldsymbol{m}^{(j)}$ solve the forward problem (5) to obtain $\boldsymbol{u}^{(j)}$
        Given $\boldsymbol{m}^{(j)}$ and $\boldsymbol{u}^{(j)}$ compute the cost $\mathcal{J}^{(j)}$ using (3)
        if $\mathcal{J}^{(j)} < \mathcal{J}_i + \alpha^{(j)}c_{\text{armijo}}\,\boldsymbol{g}_i^T\widehat{\boldsymbol{m}}_i$ **then**
            $\boldsymbol{m}_{i+1} \leftarrow \boldsymbol{m}^{(j)}, \mathcal{J}_{i+1} \leftarrow \mathcal{J}^{(j)}$
            break
        end if
        $\alpha^{(j+1)} \leftarrow \alpha^{(j)}/2, \quad j \leftarrow j + 1$
    end while
    if $j =$ max_backtracking_iter **then**
        break
    end if
    $i \leftarrow i + 1$
end while

The end result is that manipulations with the Hessian require a number of forward PDE solves that are independent of the parameter and data dimensions.

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

$$\mathbf{H}_{\text{misfit}}\boldsymbol{v}_i = \lambda_i\mathbf{R}\boldsymbol{v}_i, \quad \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n, \tag{24}$$

where $\mathbf{R}$ 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 λ_{r+i}, $0 < i \leq n - r$, is *small* relative to 1, and we define

$$\mathbf{V}_r = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots \boldsymbol{v}_r] \text{ and } \Lambda_r = \text{diag}([\lambda_1, \lambda_2, \ldots, \lambda_r]),$$

where the matrix $\mathbf{V}_r$ has $\mathbf{R}$-orthonormal columns, that is, $\mathbf{V}_r^T\mathbf{R}\mathbf{V}_r = \mathbf{I}_r$. As in Isaac et al. [2015], by using the Sherman-Morrison-Woodbury formula, we write

$$\mathbf{H}^{-1} = (\mathbf{R} + \mathbf{H}_{\text{misfit}})^{-1} = \mathbf{R}^{-1} - \mathbf{V}_r\mathbf{D}_r\mathbf{V}_r^T + O\left(\sum_{i=r+1}^{n}\frac{\lambda_i}{1 + \lambda_i}\right), \tag{25}$$

where $\mathbf{D}_r = \text{diag}(\lambda_1/(\lambda_1 + 1), \ldots, \lambda_r/(\lambda_r + 1)) \in \mathbb{R}^{r \times r}$. As can be seen from the form of the remainder term above, to obtain an accurate low-rank approximation of $\mathbf{H}^{-1}$, 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 $\mathbf{H}^{-1}$, 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 effective for matrices with rapidly decaying eigenvalues [Halko et al. 2011]. For this class of matrices, in fact, randomized algorithms present several advantages compared to Krylov subspace methods. Krylov subspace methods require sophisticated algorithms to monitor restart, orthogonality, and loss of precision. On the contrary, randomized algorithms are easy to implement, can be made numerically robust, and expose more opportunities 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 $\mathbf{H}_{\mathrm{misfit}}$. With a change of notation, we write the generalized eigenvalue problem (Equation (24)) as

$$\mathbf{A}\boldsymbol{v} = \lambda \mathbf{B}\boldsymbol{v}, \tag{26}$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is symmetric, $\mathbf{B} \in \mathbb{R}^{n \times n}$ is symmetric positive definite, and $\boldsymbol{v} \in \mathbb{R}^n$. Here we present an extension of the randomized eigensolvers in Halko et al. [2011] to the solution of the generalized symmetric eigenproblem (Equation (26)). Randomized algorithms for generalized symmetric eigenproblems were first introduced in Saibaba et al. [2016] and are revisited here with some modifications.

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

$$\|(\mathbf{I} - \mathbf{Q}\mathbf{Q}^T)\mathbf{A}\|_{\mathbf{B}} \le \epsilon, \tag{27}$$

where ϵ is a random variable whose distribution depends on the generalized eigenvalues of Equation (26) with index greater than $r + l$. To construct $\mathbf{Q}$, 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 $\mathbf{B}$-orthogonal basis for the range of $\mathbf{Y} = \mathbf{B}^{-1}\mathbf{A}\Omega$ using the so-called PreCholQR algorithm (Lowery and Langou [2014], see Algorithm 4). The main computational cost is the construction of $\mathbf{Y}$, which requires $(r + l)$ applications of the operator $\mathbf{A}$ and $(r + l)$ linear solves to apply $\mathbf{B}^{-1}$. In contrast, the computation of the matrix $\mathbf{Q}$ with $\mathbf{B}$-orthonormal columns using PreCholQR requires only an additional $(r + l)$ applications of $\mathbf{B}$ and $O(n(r + l)^2)$ dense linear algebra operations for the QR factorization. Using Equation (27), it can be shown (see Halko et al. [2011]) that $\mathbf{A} \approx (\mathbf{B}\mathbf{Q})(\mathbf{Q}^T\mathbf{A}\mathbf{Q})(\mathbf{B}\mathbf{Q})^T = (\mathbf{B}\mathbf{Q})\mathbf{T}(\mathbf{B}\mathbf{Q})^T$, where we have defined $\mathbf{T} := \mathbf{Q}^T\mathbf{A}\mathbf{Q}$. Then we compute the eigendecomposition $\mathbf{T} = \mathbf{S}\Lambda\mathbf{S}^T$ ($\mathbf{S}^T\mathbf{S} = \mathbf{I}_{r+l}$), and we approximate the r dominant eigenpairs $(\Lambda_r, \mathbf{V}_r)$ of Equation (26) by

$$\Lambda_r = \Lambda(1:r, 1:r), \quad \mathbf{V}_r = \mathbf{Q}\mathbf{S}(:, 1:r). \tag{28}$$

Algorithms 2 and 3 summarize the implementation of the double-pass and single-pass randomized algorithms [Halko et al. 2011]. The main difference between these two algorithms is how the *small* matrix $\mathbf{T}$ is computed. In the double-pass algorithm, $\mathbf{T}$ is computed directly by performing a second round of multiplication $\mathbf{A}\mathbf{Q}$ with the operator $\mathbf{A}$. In the single-pass algorithm, $\mathbf{T}$ is approximated from the information contained in Ω and $\mathbf{Y}$. In particular, generalizing the single-pass
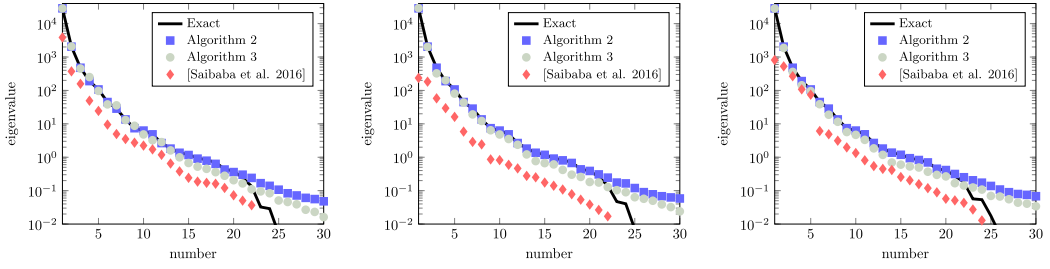
Fig. 2. Log-linear plot of first 30 out of 4,225 generalized eigenvalues of the data misfit Hessian in Section 5.1. A deterministic eigensolver (Exact), Algorithm 2, Algorithm 3, and the single-pass algorithm in Saibaba et al. [2016] are compared for different choices of the oversampling parameter l (left $l = 5$, center $l = 10$, right $l = 20$). Note that Algorithm 3 is more accurate for the single-pass algorithm in Saibaba et al. [2016] for all choices of l.

ALGORITHM 2: The double-pass randomized algorithm for the solution of the generalized symmetric eigenproblem.

Let r be the number of eigenpairs to compute and l an oversampling factor
Let $\Omega \in \mathbb{R}^{n \times (r+l)}$ be a Gaussian random matrix
$\bar{\mathbf{Y}} \leftarrow \mathbf{A}\Omega, \mathbf{Y} = \mathbf{B}^{-1}\bar{\mathbf{Y}}$
Use PreCholQR to factorize $\mathbf{Y} = \mathbf{QR}$ such that $\mathbf{Q}^T\mathbf{BQ} = \mathbf{I}_{r+l}$
$\mathbf{T} \leftarrow \mathbf{Q}^T\mathbf{AQ}$
Compute the eigenvalue decomposition $\mathbf{T} = \mathbf{S}\Lambda\mathbf{S}^T$
Keep the r largest eigenmodes and let $\mathbf{S}_r \leftarrow \mathbf{S}(:, 1:r), \Lambda_r \leftarrow \Lambda(1:r, 1:r)$
Return: $\mathbf{V}_r \leftarrow \mathbf{QS}_r$, and Λ_r

---

algorithm in Halko et al. [2011] to Equation (26), we approximate $\mathbf{T}$ as the least-squares solution of

$$\mathbf{T} = \underset{\mathbf{X} \in \mathbb{R}^{(r+l) \times (r+l)}, s.s.p.d}{\operatorname{argmin}} \|\mathbf{X}(\mathbf{Q}^T\mathbf{B}\Omega) - \mathbf{Q}^T\mathbf{BY}\|_2^2. \tag{29}$$

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 is more accurate than the single-pass algorithm presented in Saibaba et al. [2016]. The key difference between the two algorithms is in the definition of $\mathbf{T}$. Saibaba et al. [2016] define $\mathbf{T} = (\mathbf{Q}^T\mathbf{B}\Omega)^{-1}(\mathbf{Q}^T\mathbf{BY})(\mathbf{Q}^T\mathbf{B}\Omega)^{-1}$, while in Algorithm 3 we define $\mathbf{T}$ as the least-squares solution of Equation (29). Figure 2 numerically illustrates the higher accuracy of the proposed approach when computing the first 30 eigenvalues of the data misfit Hessian discussed in Section 5.1.

ALGORITHM 3: The single-pass randomized algorithm for the solution of the generalized symmetric eigenproblem.

Let r be the number of eigenpairs to compute and l an oversampling factor
Let $\Omega \in \mathbb{R}^{n \times (r+l)}$ be a Gaussian random matrix
$\bar{\mathbf{Y}} \leftarrow \mathbf{A}\Omega, \mathbf{Y} = \mathbf{B}^{-1}\bar{\mathbf{Y}}$
Use PreCholQR to factorize $\mathbf{Y} = \mathbf{QR}$ such that $\mathbf{Q}^T\mathbf{BQ} = \mathbf{I}_{r+l}$ and $\bar{\mathbf{Q}}$ such that $\bar{\mathbf{Q}}^T\mathbf{B}^{-1}\bar{\mathbf{Q}} = \mathbf{I}_{r+l}$
Find $\mathbf{T}$ s.s.p.d such that $\|\mathbf{T}(\bar{\mathbf{Q}}^T\Omega) - \bar{\mathbf{Q}}^T\mathbf{Y}\|_2^2 \rightarrow \min$
Compute the eigenvalue decomposition $\mathbf{T} = \mathbf{S}\Lambda\mathbf{S}^T$
Keep the r largest eigenmodes and let $\mathbf{S}_r \leftarrow \mathbf{S}(:, 1:r), \Lambda_r \leftarrow \Lambda(1:r, 1:r)$
Return: $\mathbf{V}_r \leftarrow \mathbf{QS}_r$, and Λ_r

ALGORITHM 4: PreCholQR

Require: $Y \in \mathbb{R}^{n \times (r+l)}$, and $B \in \mathbb{R}^{n \times n}$
$[Z, R_Y] \leftarrow qr(Y)$
$\bar{Z} \leftarrow BZ$
$R_Z = chol(Z^T \bar{Z})$
Return: $Q = ZR_Z^{-1}$, $\bar{Q} = \bar{Z}R_Z^{-1}$, 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 Equation (15). In what follows, we will denote the expected value (mean) of a random vector x with the symbol $\mathsf{E}[x]$, and its covariance with the symbol $cov(x) := \mathsf{E}[(x - \mathsf{E}[x])(x - \mathsf{E}[x])^T]$. We will also denote with Γ the matrix representation of the covariance operator C with respect to the standard Euclidean inner product.[5]

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

$$cov(x) = \mathsf{E}[xx^T] = \mathsf{E}[C\eta\eta^T C^T] = C\mathsf{E}[\eta\eta^T]C^T = CC^T = \Gamma.$$

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

This approach is not feasible for large-scale problems since it requires computing the Cholesky factorization of the covariance matrix. However, note that a decomposition of the form $\Gamma = CC^T$ can be obtained using a matrix C other than the Cholesky factor. In particular, the matrix C need not be a triangular, or even square, matrix. In Appendix C, we exploit this observation and show a scalable sampling technique based on a rectangular decomposition of $R = \Gamma^{-1}$, for the case when R is a finite element discretization of a differential operator. We note that a similar approach was, independently, investigated in [Croci et al. 2018] to sample realizations of white noise by exploiting a rectangular decomposition of the finite element mass matrix. Our approach is more general as it allows for decomposing matrices stemming from finite element discretization of operators other than identity.

4.3.1 Sampling from the Prior. Sampling from a Gaussian distribution with a prescribed covariance matrix Γ 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. Parker and Fox [2012] 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 for a factorized preconditioner limit the efficiency of this sampler for large-scale applications. Chow and Saad [2014] consider a preconditioned Krylov subspace method to approximate the action of $\Gamma^{\frac{1}{2}}$ on a generic vector z. Chen et al. [2011] discuss a method to compute $f(\Gamma)b$ via least-squares polynomial approximations for a generic matrix function $f(x) = \sqrt{x}$. To

---

[5]Note that Γ differs from Γ_{prior} and Γ_{post}, which are defined in terms of the M-weighted inner product.

this aim the authors approximate the function by a spline of a desired accuracy on the spectrum of Γ 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 x from the distribution $\mathcal{N}(\mathbf{0}, \mathbf{R}^{-1})$, we solve the linear system

$$\mathbf{R}x = \mathbf{C}\boldsymbol{\eta}, \text{ where } \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_q), \tag{30}$$

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

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

where we exploited the fact that $\mathsf{E}[\boldsymbol{\eta}\boldsymbol{\eta}^T] = \mathbf{I}_q$ and $\mathbf{C}\mathbf{C}^T = \mathbf{R}$. This method is particularly efficient at large scale since: (1) the matrix $\mathbf{C}$ is sparse and can be computed efficiently by exploiting the finite element assembly routine; (2) the dominant cost is the solution of a linear system with coefficient matrix $\mathbf{R}$, for which efficient and scalable methods are available (e.g., conjugate gradients with algebraic multigrid preconditioner); and (3) the stochastic dimension of $\boldsymbol{\eta}$ 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 (25). This assumption is often verified for many inverse problems as we discussed in Section 4.2. Then, given a sample from the prior distribution $x \sim \mathcal{N}(\mathbf{0}, \mathbf{R}^{-1})$, a sample from the Laplace approximation of the posterior $\mathcal{N}(\mathbf{0}, (\mathbf{H}_{\text{misfit}} + \mathbf{R})^{-1})$ can be computed as

$$y = \left(\mathbf{I}_n - \mathbf{V}_r \mathbf{S}_r \mathbf{V}_r^T \mathbf{R} \right) x, \tag{31}$$

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

$$\begin{aligned}
\text{cov}(y) = \mathsf{E}[yy^T] &= \mathsf{E}\left[\left(\mathbf{I}_n - \mathbf{V}_r \mathbf{S}_r \mathbf{V}_r^T \mathbf{R} \right) xx^T \left(\mathbf{I}_n - \mathbf{V}_r \mathbf{S}_r \mathbf{V}_r^T \mathbf{R} \right)^T \right] \\
&= \left(\mathbf{I}_n - \mathbf{V}_r \mathbf{S}_r \mathbf{V}_r^T \mathbf{R} \right) \mathsf{E}[xx^T] \left(\mathbf{I}_n - \mathbf{V}_r \mathbf{S}_r \mathbf{V}_r^T \mathbf{R} \right)^T \\
&= \left(\mathbf{I}_n - \mathbf{V}_r \mathbf{S}_r \mathbf{V}_r^T \mathbf{R} \right) \mathbf{R}^{-1} \left(\mathbf{I}_n - \mathbf{V}_r \mathbf{S}_r \mathbf{V}_r^T \mathbf{R} \right)^T \\
&= \mathbf{R}^{-1} - \mathbf{V}_r (2\mathbf{S}_r - \mathbf{S}_r^2)\mathbf{V}_r^T \\
&= \mathbf{R}^{-1} - \mathbf{V}_r \mathbf{D}_r \mathbf{V}_r^T \approx \mathbf{H}^{-1},
\end{aligned}$$

where we have used the definition of $\mathbf{D}_r$, the $\mathbf{R}$-orthogonality of the eigenvectors matrix $\mathbf{V}_r$ (i.e., $\mathbf{V}_r^T \mathbf{R} \mathbf{V}_r = \mathbf{I}_r$), 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 $\boldsymbol{m} \sim \mathcal{N}(0, \mathbf{Q}^{-1})$, where $\mathbf{Q}$ is the precision matrix. Here $C = C_{\mathrm{pr}}$ and $\mathbf{Q} = \mathbf{R}$ if we are interested in the prior distribution; $C = C_{\mathrm{post}}$ and $\mathbf{Q} = \mathbf{H}$ for the posterior distribution. Then we define the pointwise variance of m as the field $\sigma^2(x)$ such that

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

In this section, we present an efficient numerical method to compute a finite element approximation σ_h^2 of $\sigma^2(x)$. As shown in Bui-Thanh et al. [2013], the diagonal of Q^{-1} (i.e., $\mathrm{diag}(\mathbf{Q}^{-1})$) is the vector corresponding to the coefficients of the expansion of σ_h^2 in the finite element basis. A naïve approach would require solution of n linear systems with $\mathbf{Q}$. This is not feasible for large-scale problems: even for the case when an optimal (i.e., $O(n)$) solver for $\mathbf{Q}$ 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 $\mathbf{R}$. Recall that in our case, $\mathbf{R}$ arises from finite element discretization of an elliptic differential operator. Two commonly used methods to solve this task are the stochastic estimator in Bekas et al. [2007] and the probing method in Tang and Saad [2012]. Specifically, the unbiased stochastic estimator for the diagonal of the inverse of $\mathbf{R}$ in Bekas et al. [2007] reads

$$\mathrm{diag}(\mathbf{R}^{-1}) \approx \left[\sum_{j=1}^{s} \boldsymbol{z}_j \odot \boldsymbol{w}_j \right] \oslash \left[\sum_{j=1}^{s} \boldsymbol{z}_j \odot \boldsymbol{z}_j \right], \tag{32}$$

where $\boldsymbol{w}_j$ solves $\mathbf{R}\boldsymbol{w}_j = \boldsymbol{z}_j$ and $\boldsymbol{z}_j$ are random *i.i.d.* vectors. Here $\odot$ and $\oslash$ 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 Tang and Saad [2012], on the other hand, leads to faster convergence in the common situation in which $\mathbf{R}^{-1}$ 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 $\mathcal{G}$ whose adjacency matrix is the sparsity pattern of some power k of $\mathbf{R}$. More specifically, there are as many probing vectors as the number of colors in the graph $\mathcal{G}$ and the probing vector associated to color i is the binary vector whose nonzero entries correspond to the nodes of $\mathcal{G}$ 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 $\mathbf{R}^{-1}$, taking advantage of the fact that $\mathbf{R}$ is the discretization of an elliptic differential operator. Specifically, we write

$$\mathrm{diag}(\mathbf{R}^{-1}) \approx \left[\sum_{i=1}^{\bar{r}} \mu_i \overline{\boldsymbol{v}}_i \odot \overline{\boldsymbol{v}}_i \right], \tag{33}$$

where $\{(\mu_i, \overline{\boldsymbol{v}}_i)\}_1^{\bar{r}}$ denote the approximation of the $\bar{r}$ dominant eigenpairs of the matrix $\mathbf{R}^{-1}$ obtained by using the double-pass randomized eigensolver (Algorithm 2) with $l = 0$, $\mathbf{A} = \mathbf{R}^{-1}$, and $\mathbf{B}$ the identity matrix. The main advantage of this approach is that, thanks to the rapid decay of the
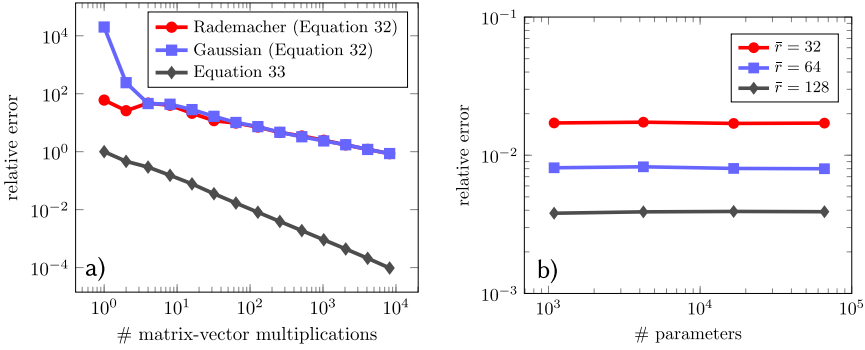
Fig. 3. The $L^2(\Omega)$ relative error for the estimation of the marginal variance for the prior distribution in Section 5.1 as a function of number of covariance operator applied (a) and number of parameters (b).

eigenvalues μ_i ($\mathbf{R}^{-1}$ is compact), $\bar{r}$ is much smaller than the number of samples s necessary for the stochastic estimator in Equation (32) to achieve a given accuracy and that, in contrast to the probing algorithm, it is independent of the mesh size.

To illustrate the convergence properties of the proposed method, we estimate the marginal variance of the Gaussian prior distribution in Equation (37) for the model problem in Section 5.1. Figure 3(a) shows the superior accuracy of our method when compared to the stochastic estimator in Bekas et al. [2007] for a given number of covariance operators applied. Figure 3(b) numerically demonstrates the mesh independence of the proposed method.

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

$$\text{diag}(\mathbf{H}^{-1}) \approx \text{diag}(\mathbf{R}^{-1} - \mathbf{V}_r \mathbf{D}_r \mathbf{V}_r^T) = \text{diag}(\mathbf{R}^{-1}) - \text{diag}(\mathbf{V}_r \mathbf{D}_r \mathbf{V}_r^T). \tag{34}$$

In `hIPPYlib`, the first term is approximated using Equation (33), while the data-informed correction $\text{diag}(\mathbf{V}_r \mathbf{D}_r \mathbf{V}_r^T)$ can be explicitly computed in $O(n)$ operations as follows

$$\text{diag}(\mathbf{V}_r \mathbf{D}_r \mathbf{V}_r^T) = \sum_{i=1}^{r} \left[\left(\frac{\lambda_i}{1 + \lambda_i} \boldsymbol{v}_i \right) \odot \boldsymbol{v}_i \right].$$

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 shown below were obtained using `hIPPYlib` version 2.3.0 and FEniCS 2017.2. A Docker image [Merkel 2014] containing the preinstalled software and examples can be downloaded at https://hub.docker.com/r/hippylib/toms. For a line-by-line explanation of the source code for these two model problems, we refer the reader to the Python Jupyter notebooks available at https://hippylib.github.io/tutorial_v2.3.0/.

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
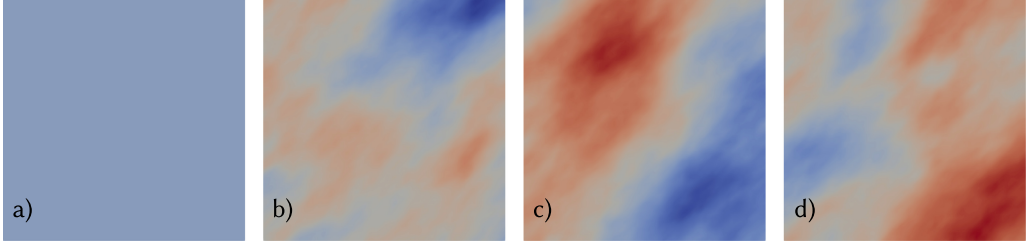
Fig. 4. Prior mean m_{pr} (a), and samples drawn from the prior distribution (b)–(d).

problem 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:

$$
\begin{aligned}
-\nabla \cdot (e^m \nabla u) &= f &&\text{in } \mathcal{D}, \\
u &= g &&\text{on } \Gamma_D, \\
e^m \nabla u \cdot \boldsymbol{n} &= h &&\text{on } \Gamma_N,
\end{aligned}
\tag{35}
$$

where $\mathcal{D} \subset \mathbb{R}^d$ ($d = 2, 3$) 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, $f \in L^2(\mathcal{D})$ 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,

$$
\mathcal{V}_g = \{v \in H^1(\mathcal{D}) : v\big|_{\Gamma_D} = g\}, \quad \mathcal{V}_0 = \{v \in H^1(\mathcal{D}) : v\big|_{\Gamma_D} = 0\},
$$

where $H^1(\mathcal{D})$ is the Sobolev space of functions whose derivatives are in $L^2(\mathcal{D})$. Then, the weak form of Equation (35) reads as follows: find $u \in \mathcal{V}_g$ such that

$$
\langle e^m \nabla u, \nabla p \rangle = \langle f, p \rangle + \langle h, p \rangle_{\Gamma_N}, \quad \forall p \in \mathcal{V}_0.
\tag{36}
$$

Here $\langle \cdot, \cdot \rangle$ and $\langle \cdot, \cdot \rangle_{\Gamma_N}$ denote the standard inner products in $L^2(\mathcal{D})$ 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 Stuart [2010]. $\mathcal{A}$ is a differential operator with domain $\mathcal{M} := H^1(\mathcal{D})$ and action

$$
\mathcal{A}m = \begin{cases} -\gamma \nabla \cdot (\Theta \nabla m) + \delta m & \text{in } \mathcal{D} \\ \Theta \nabla m \cdot \boldsymbol{n} + \beta m & \text{on } \partial \mathcal{D}, \end{cases}
\tag{37}
$$

where $\beta \propto \sqrt{\gamma \delta}$ is the optimal Robin coefficient derived in Daon and Stadler [2018] and Roininen et al. [2014] to minimize boundary artifacts, and Θ 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 4, 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 $\boldsymbol{d} \in \mathbb{R}^q$ the vector of (noisy) pointwise observations of the state u at $q = 50$ random locations uniformly distributed in $\mathcal{D}_{obs} := [0.1, 0.9] \times [0.1, 0.5]$ ($\mathcal{D}_{obs} \subset \mathcal{D}$). That is,
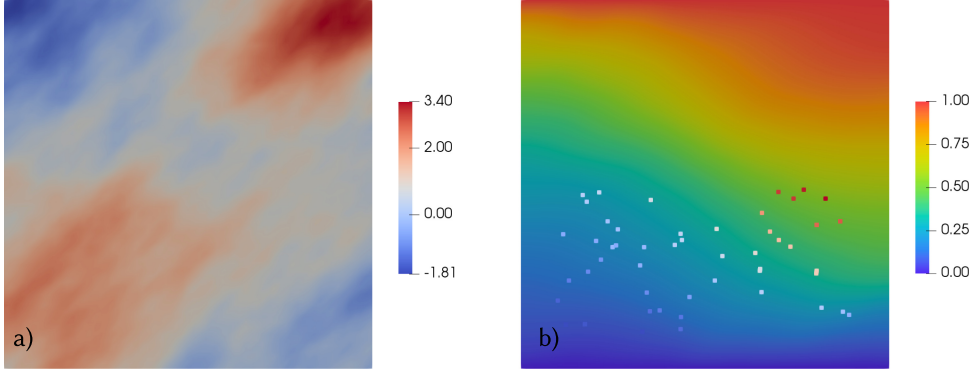
$$
\boldsymbol{d} = \mathcal{B}u + \boldsymbol{\eta},
$$

Fig. 5. The true parameter field m_{true} (a) and the state u obtained by solving the forward PDE with m_{true} (b). The squares in (b) represent locations of the $q = 50$ randomly chosen observation points and their color corresponds to the observed noisy data $\boldsymbol{d}$.

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

5.1.2   The MAP Point. To find the MAP point we solve the following variational nonlinear least-squares optimization problem:

$$\min_{m \in \mathcal{M}} \mathcal{J}(m) := \frac{1}{2} \left\| \mathcal{B}u(m) - \boldsymbol{d} \right\|_{\Gamma_{\text{noise}}^{-1}}^2 + \frac{1}{2} \left\| m - m_{\text{pr}} \right\|_{C_{\text{prior}}^{-1}}^2 , \tag{38}$$

where the state variable u is the solution to Equation (35), m_{pr} is the prior mean of the log coefficient field m, and $\boldsymbol{d} \in \mathbb{R}^q$ is a given data vector. To solve this optimization problem we use the inexact Newton-CG algorithm in Algorthim 1, which requires gradient and Hessian information. These are automatically computed by hIPPYlib applying symbolic differentiation to the variational form of the forward problem (Equation (36)); we also refer to Appendix A where the gradient and Hessian-apply expressions are derived using the Lagrangian formalism. 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.

5.1.3   Numerical Results. For the forward Poisson problem (Equation (35)), 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 $e^m \nabla u \cdot \boldsymbol{n} = 0$ on Γ_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 $\Gamma_D := (0, 1) \times \{0, 1\}$. In Figure 5, 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 16,641 and 4,225, respectively.

Next we study the spectrum of the data misfit Hessian evaluated at the MAP point. Figure 6 (left) shows a logarithmic plot of the eigenvalues of the generalized symmetric eigenproblem

$$\mathbf{H}_{\text{misfit}} \boldsymbol{v}_i = \lambda_i \mathbf{R} \boldsymbol{v}_i, \quad \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n,$$

where $\mathbf{H}_{\text{misfit}}$ and $\mathbf{R}$ stem from the discretization (with respect to the Euclidean inner product of the data misfit Hessian and prior precision (cf. Equation (24)). This plot shows that the spectrum decays rapidly. As seen in Equation (25), an accurate low-rank-based approximation of the inverse Hessian
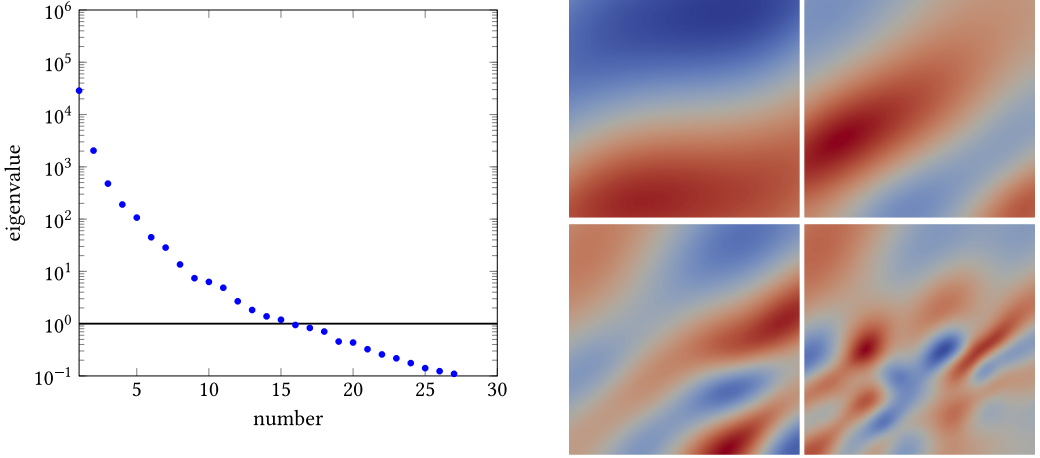
Fig. 6. Left: Log-linear plot of first 30 out of 4,225 eigenvalues of the data misfit Hessian for the generalized eigenproblem (Equation (24)). The low-rank based approximation captures the dominant, data-informed portion of the spectrum. The eigenvalues are truncated at 0.07. Right: Prior-orthogonal 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.

can be obtained by neglecting eigenvalues that are small compared to 1. Thus, retaining around $r = 30$ eigenvectors out of 4,225 (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 6 (right) displays several of these eigenvectors.

Figure 7 depicts the prior and posterior pointwise variances computed using Equations (33) and (34) with $\bar{r} = 300$ 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 $\mathcal{D}$. In Figure 8 we show the MAP point (a) and samples from the Laplace approximation (Equation (20)) of the posterior probability density function (b)–(d). These samples were obtained by first computing samples from the prior distribution—shown in Figure 4—according to Equation (30), and then applying Equation (31). The variance reduction between posterior samples in Figure 8 and prior samples in Figure 4 reflects the information gained from the data in solving the inverse problem. In addition, we note that the MAP point resembles the 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 Petra and Stadler [2011]. The PDE in the parameter-
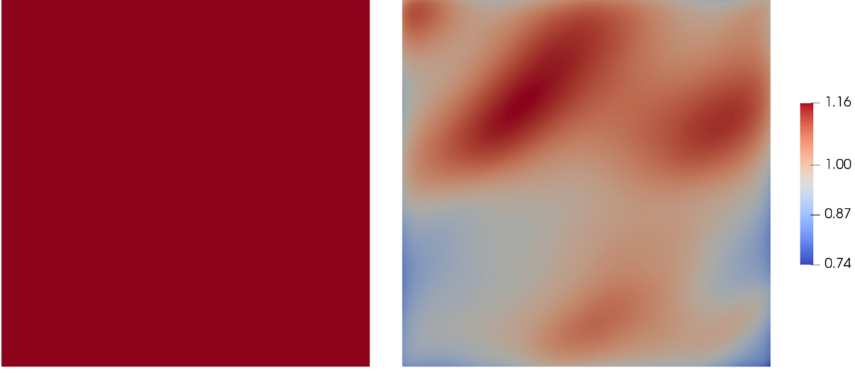
Fig. 7. Pointwise variance of the prior distribution (left) and the Laplace approximation of the posterior distribution (right). Note that uncertainty is mostly reduced in the lower half of the domain where data is measured.
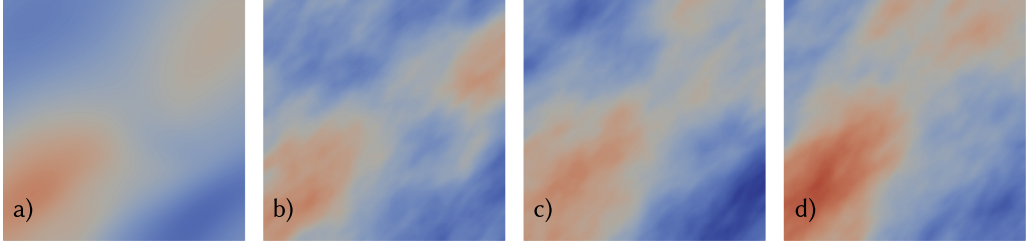

Fig. 8. The MAP point (a) and samples drawn from the Laplace approximation of posterior distribution (b)–(d).

to-observable map models diffusive transport in a domain $\mathcal{D} \subset \mathbb{R}^d$, which is depicted in Figure 9. The domain boundaries $\mathcal{D}$ 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 $u(\boldsymbol{x}, t)$ through solution of the advection-diffusion equation given by

$$
\begin{aligned}
u_t - \kappa \Delta u + \boldsymbol{v} \cdot \nabla u &= 0 && \text{in } \mathcal{D} \times (0, T), \\
u(\cdot, 0) &= m && \text{in } \mathcal{D}, \\
\kappa \nabla u \cdot \boldsymbol{n} &= 0 && \text{on } \partial \mathcal{D} \times (0, T).
\end{aligned}
\tag{39}
$$

Here, $\kappa > 0$ is a diffusivity coefficient, and $T > 0$ is the final time of observations. The velocity field $\boldsymbol{v}$, shown in Figure 9 (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 9 (left); see Petra and Stadler [2011] for details. The time evolution of the state variable u from a given initial condition m is illustrated in Figure 10 (top).

To derive the weak formulation of Equation (39), we define the spaces

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

Then, the weak form of Equation (39) reads as follows: Find $u \in \mathcal{V}$ such that

$$
\int_0^T \int_{\mathcal{D}} (u_t + \boldsymbol{v} \cdot \nabla u) p \, d\boldsymbol{x} \, dt + \int_0^T \int_{\mathcal{D}} \kappa \nabla u \cdot \nabla p \, d\boldsymbol{x} \, dt + \int_{\mathcal{D}} (u(\boldsymbol{x}, 0) - m) p_0 \, d\boldsymbol{x} = 0, \tag{40}
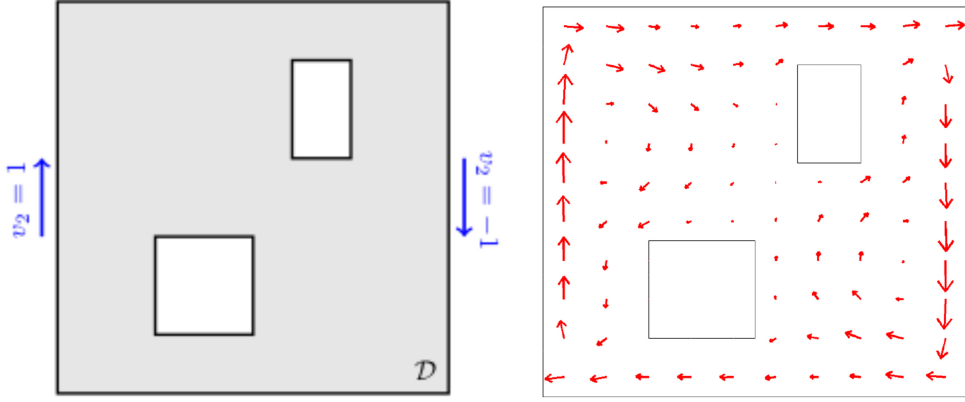$$

Fig. 9. Left: Sketch of domain for the advective-diffusive inverse transport problem (Equation (39)) showing imposed velocities used to compute the velocity field $\boldsymbol{v}$. Right: The velocity field computed from the solution of the Navier-Stokes equations and subsampled on a coarser grid for visualization purposed.
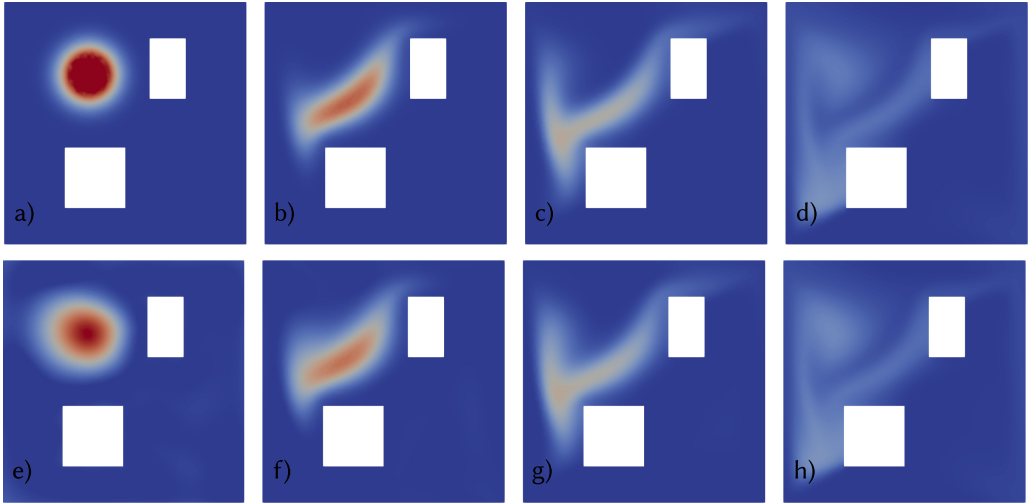

Fig. 10. Forward advective-diffusive transport estimate of the inverse solution at initial time t = 0 (a, d), at t = 1 (b, e) and t = 2 (c, f), and at final time t = 4 (d) with the "true" (top) and MAP (bottom) as initial conditions.

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

5.2.1   The Noise and Prior Models. We consider the problem of inferring the initial condition m in Equation (39) from pointwise noisy observations $\boldsymbol{d}_i \in \mathbb{R}^{n_t}$ ($i = 1, \ldots, n_s$) 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 $\boldsymbol{d}_i$ are perturbed with *i.i.d.* Gaussian additive noise with variance σ^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 $\gamma \nabla m \cdot \boldsymbol{n} + \beta m$ on $\partial \mathcal{D}$. 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 β is chosen as in [Daon and Stadler 2018; Roininen et al. 2014] to reduce boundary artifacts.
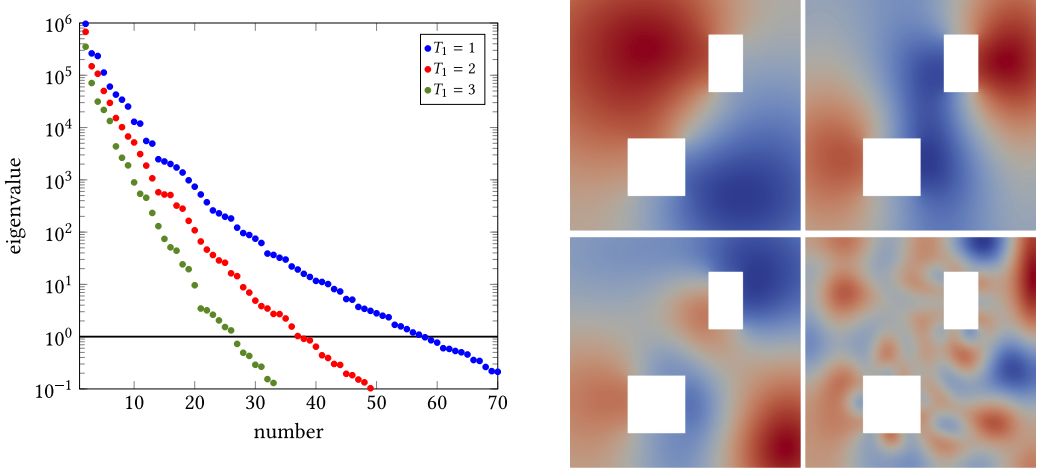
Fig. 11. Left: Log-linear plot of the truncated spectrum of prior-preconditioned data misfit Hessian for observation times (sampled every 0.2 time units) in the intervals $[1, 4]$ (blue), $[2, 4]$ (red), and $[3, 4]$ (green). The low-rank approximation captures the dominant, data-informed portion of the spectrum. The eigenvalues are truncated at around 0.06. Right: Prior-orthogonal eigenvectors of the prior-preconditioned data misfit Hessian corresponding (from left to right) to the 1st, 4th, 7th, and 60th eigenvalues. Eigenvectors corresponding to smaller eigenvalues are increasingly more oscillatory (and thus inform smaller-length scales of the initial concentration) but are also increasingly less informed by the data.

5.2.2 The MAP Point. To compute the MAP point, we minimize the negative log-posterior, defined in general in Equation (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

$$\mathcal{J}(m) := \frac{1}{2\sigma^2} \sum_{i=1}^{n_s} \int_{T_1}^{T} (\mathcal{B}u - \boldsymbol{d}_i)^2 \delta_{t_i} \, dt + \frac{1}{2} \left\| \mathcal{A}(m - m_{\text{pr}}) \right\|_{L^2(\mathcal{D})}^2, \tag{41}$$

where $\mathcal{B} : \mathcal{V}_0 \mapsto \mathbb{R}^{n_t}$ is the interpolation operators at the observation locations, δ_{t_i} is the Dirac delta functions at the observation time sample $t = t_i$ ($i = 1, \ldots, n_s$), and σ^2 represents the noise level in the observations $\boldsymbol{d}_i$, here taken 2.45×10^{-7}, and $m_{\text{pr}} = 0$ is the prior mean. We use the conjugate gradient method to solve this (linear) inverse problem. The derivation of the gradient and Hessian-apply is given in Appendix B.

5.2.3 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 [Hughes et al. 1989] is added to ensure stability of the discretization. The space-time dimension of the state variable is 433,880 (10,847 spatial degree of freedom times 40 time steps), and the dimension of the parameter space is 10,847. The data dimension q is 1,200 with $n_t = 80$ measurement locations and $n_s = 15$ time samples.

To illustrate properties of the forward problem, Figure 10 shows three snapshots in time of the field u, using the advective velocity $\boldsymbol{v}$ from Figure 9 with the "true" initial condition (top row) and its MAP point estimate (bottom row), 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 $\mathcal{F}$, the prior-preconditioned data misfit Hessian is independent of m. Figure 11
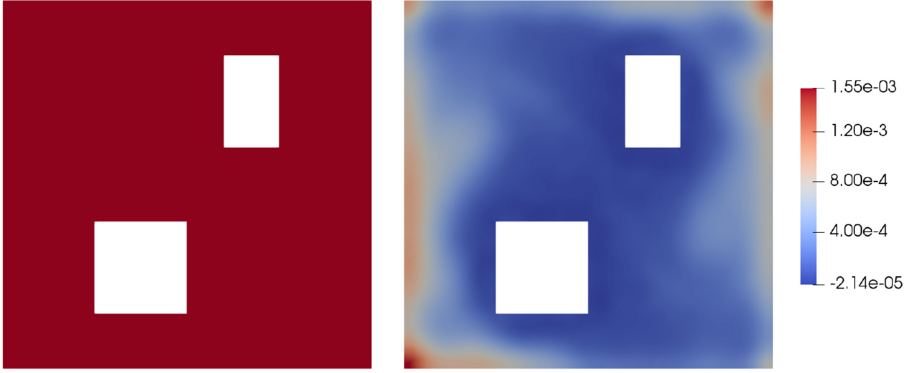
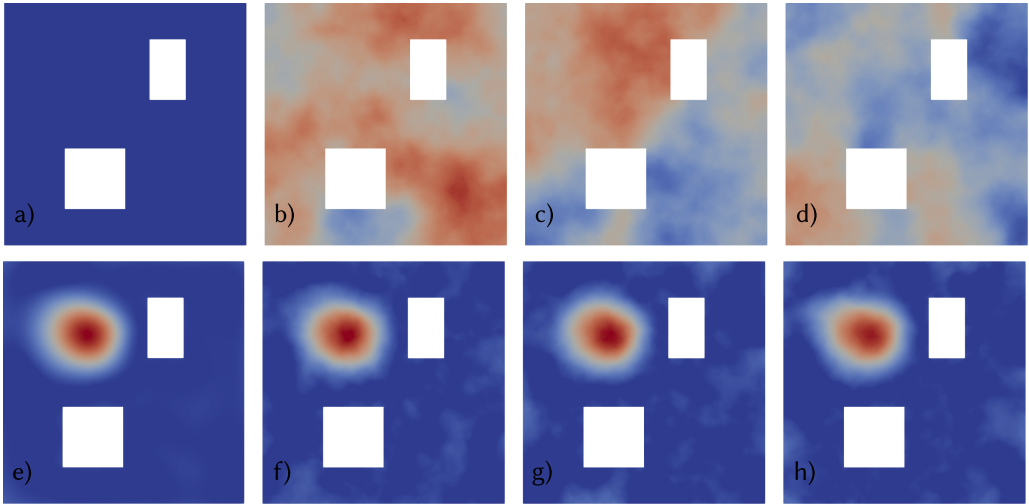Fig. 12. This figure shows the pointwise variance of the prior (left) and posterior (right) distributions.

Fig. 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).

(left) 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 Equation (25), 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 10,847 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 the right panel in Figure 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 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 the observations (at the boundaries of the interior rectangles). In Figure 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 differential equations. The main advantage of this framework is that it exploits the structure of the underlying infinite-dimensional PDE-based parameter-to-observable map, in particular the low effective dimensionality, which leads to scalable algorithms for carrying out the solution of deterministic and linearized Bayesian inverse problems. By scalable, we mean that the cost—measured in number of (linearized) forward (and adjoint) solves—is independent of the state, parameter, and data dimensions. The cost depends only on the number of modes in parameter space that are informed by the data. We have described the main algorithms implemented in `hIPPYlib`, namely the inexact Newton-CG method to compute the MAP point (Section 4.1), randomized eigensolvers to compute the low-rank approximation of the Hessian evaluated at the MAP point (Section 4.2), and algorithms for sampling and computing the pointwise variance from large-scale Gaussian random fields (Sections 4.3 and 4.4). To illustrate their use, we applied these methods to two model problems: inversion for the log coefficient field in a Poisson equation and inversion for the initial condition in a time-dependent advection-diffusion equation.

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 deterministic 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 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, and so forth.

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.

APPENDICES

A GRADIENT AND HESSIAN ACTIONS COMPUTATION FOR THE INVERSE PROBLEM GOVERNED BY THE POISSON PDE

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 $\mathcal{J}(m)$ defined in Equation (38). The Lagrangian

functional for this optimization problem is given by

$$\mathcal{L}^{\mathcal{G}}(u, m, p) := \mathcal{J}(m) + \langle e^m \nabla u, \nabla p \rangle - \langle f, p \rangle - \langle p, h \rangle_{\Gamma_N}, \tag{42}$$

where the last three terms stem from the variational form (Equation (36)) of the forward problem (Equation (35)). The formal Lagrange multiplier method [Tröltzsch 2010] requires that, at a minimizer of Equation (38), variations of the Lagrangian functional with respect to p and u vanish, which yields to solving the forward and adjoint problems

$$\langle e^m \nabla u, \nabla \tilde{p} \rangle - \langle f, \tilde{p} \rangle - \langle \tilde{p}, h \rangle_{\Gamma_N} = 0, \quad \forall \tilde{p} \in \mathcal{V}_0; \tag{43a}$$

$$\langle e^m \nabla \tilde{u}, \nabla p \rangle + \langle \mathcal{B}^* \Gamma_{\text{noise}}^{-1}(\mathcal{B}u - d), \tilde{u} \rangle = 0, \quad \forall \tilde{u} \in \mathcal{V}_0. \tag{43b}$$

The strong form of the forward problem is given in Equation (35), while the strong form of the adjoint problem reads

$$\begin{aligned} -\nabla \cdot (e^m \nabla p) &= \mathcal{B}^* \Gamma_{\text{noise}}^{-1}(\mathcal{B}u - d) \quad \text{in } \mathcal{D}, \\ p &= 0 \quad \text{on } \Gamma_D, \\ e^m \nabla p \cdot \boldsymbol{n} &= 0 \quad \text{on } \Gamma_N. \end{aligned} \tag{44}$$

Finally, the gradient of the cost functional (Equation (38)) is given in weak form by

$$(\mathcal{G}(m), \tilde{m}) = \langle m - m_{\text{pr}}, \tilde{m} \rangle_{C_{\text{prior}}^{-1}} + \langle \tilde{m} e^m \nabla u, \nabla p \rangle, \quad \forall \tilde{m} \in \mathcal{M}, \tag{45}$$

where u and p are solutions to the forward and adjoint problems (Equations (43a) and (43b)), respectively [Borzì and Schulz 2012; Tröltzsch 2010]. In strong form this reads

$$\mathcal{G}(m) = \begin{cases} C_{\text{prior}}^{-1}(m - m_{\text{pr}}) + e^m (\nabla u \cdot \nabla p) & \text{in } \mathcal{D}, \\ \gamma (\Theta \nabla m) \cdot \boldsymbol{n} + \beta m & \text{on } \partial \mathcal{D}. \end{cases} \tag{46}$$

We note that to evaluate the gradient for a given parameter m, one needs to solve the forward problem (Equation (43a)) 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

$$\begin{aligned} \mathcal{L}^{\mathcal{H}}(u, m, p; \hat{u}, \hat{m}, \hat{p}) := {} & (\mathcal{G}(m), \hat{m}) \\ & + \langle e^m \nabla u, \nabla \hat{p} \rangle - \langle f, \hat{p} \rangle - \langle \hat{p}, h \rangle_{\Gamma_N} \\ & + \langle e^m \nabla \hat{u}, \nabla p \rangle + \langle \mathcal{B}^* \Gamma_{\text{noise}}^{-1}(\mathcal{B}u - d), \hat{u} \rangle. \end{aligned}$$

To obtain the action of the Hessian in a direction $\hat{m}$ we take the variation of $\mathcal{L}^{\mathcal{H}}$ with respect to m, namely:

$$(\tilde{m}, \mathcal{H}(m)\hat{m}) = \langle \tilde{m} e^m \nabla \hat{u}, \nabla p \rangle + \langle \hat{m}, \tilde{m} \rangle_{C_{\text{prior}}^{-1}} \tag{47}$$

$$+ \langle \tilde{m} \hat{m} e^m \nabla u, \nabla p \rangle + \langle \tilde{m} e^m \nabla u, \nabla \hat{p} \rangle, \quad \forall \tilde{m} \in \mathcal{M}, \tag{48}$$

where as before u and p are the solutions of the forward and adjoint problems in Equations (43a) and (43b)), respectively, and $\hat{u}$ and $\hat{p}$ are the solutions of the incremental forward and adjoint problems, respectively. These equations are given by

$$\langle e^m \nabla \hat{u}, \nabla \tilde{p} \rangle + \langle \hat{m} e^m \nabla u, \nabla \tilde{p} \rangle = 0, \quad \forall \tilde{p} \in \mathcal{V}_0, \tag{49}$$

and

$$\langle \mathcal{B}^* \Gamma_{\text{noise}}^{-1} \mathcal{B}\hat{u}, \tilde{u} \rangle + \langle \hat{m} e^m \nabla \tilde{u}, \nabla p \rangle + \langle e^m \nabla \tilde{u}, \nabla \hat{p} \rangle = 0, \quad \forall \tilde{u} \in \mathcal{V}_0. \tag{50}$$

Once we have the gradient and Hessian action expressions, we can apply Algorithm 1 to solve the optimization problem given by Equation (38).

B   GRADIENT COMPUTATION FOR INVERSE PROBLEM GOVERNED BY THE ADVECTION-DIFFUSION PDE

To derive an expression for the gradient of $\mathcal{J}(m)$ in Equation (41), we define the Lagrangian functional

$$\mathcal{L}^{\mathcal{G}}(u, m, p, p_0) := \mathcal{J}(m) + \int_0^T \int_{\mathcal{D}} (u_t + \boldsymbol{v} \cdot \nabla u) p \, d\boldsymbol{x} \, dt$$
$$+ \int_0^T \int_{\mathcal{D}} \kappa \nabla u \cdot \nabla p \, d\boldsymbol{x} \, dt + \int_{\mathcal{D}} (u(\boldsymbol{x}, 0) - m) p_0 \, d\boldsymbol{x},$$

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 (Equation (39)). Expressions needed to compute the gradient of Equation (41) are obtained by setting variations of the Lagrangian $\mathcal{L}^{\mathcal{G}}$ with respect to p, p_0, and u to zero. Variations with respect to p and p_0 recover the variational form (Equation (40)) of the forward problem. The variation with respect to u in an arbitrary direction $\tilde{u}$ yields

$$\frac{1}{\sigma^2} \sum_{i=1}^{n_s} \int_{T_1}^T (\mathcal{B}u - \boldsymbol{d}_i) \tilde{u} \, \delta_{t_i} \, dt + \int_0^T \int_{\mathcal{D}} (\tilde{u}_t + \boldsymbol{v} \cdot \nabla \tilde{u}) p \, d\boldsymbol{x} \, dt$$
$$+ \int_0^T \int_{\mathcal{D}} \kappa \nabla \tilde{u} \cdot \nabla p \, d\boldsymbol{x} \, dt + \int_{\mathcal{D}} \tilde{u}(\boldsymbol{x}, 0) p_0 \, d\boldsymbol{x} = 0 \quad \forall \tilde{u} \in \mathcal{V}.$$

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

$$\frac{1}{\sigma^2} \sum_{i=1}^{n_s} \int_{T_1}^T (\mathcal{B}u - \boldsymbol{d}_i) \tilde{u} \, \delta_{t_i} \, dt - \int_0^T \int_{\mathcal{D}} (\tilde{p}_t + \nabla \cdot (\boldsymbol{v} p) + \kappa \Delta p) \tilde{u} \, d\boldsymbol{x} \, dt + \int_{\mathcal{D}} \tilde{u}(\boldsymbol{x}, T) p(\boldsymbol{x}, T)$$
$$- \tilde{u}(\boldsymbol{x}, 0) p(\boldsymbol{x}, 0) + \tilde{u}(\boldsymbol{x}, 0) p_0 \, d\boldsymbol{x} + \int_0^T \int_{\partial \mathcal{D}} (\boldsymbol{v} p + \kappa \nabla p) \cdot \boldsymbol{n} \tilde{u} \, d\boldsymbol{x} \, dt = 0,$$

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

$$-p_t - \nabla \cdot (p\boldsymbol{v}) - \kappa \Delta p = -\frac{1}{\sigma^2} \mathcal{B}^* \sum_{i=1}^{n_s} (\mathcal{B}u - \boldsymbol{d}_i) \delta_{t_i} \qquad \text{in } \mathcal{D} \times (0, T),$$
$$p(\cdot, T) = 0 \qquad \text{in } \mathcal{D}, \qquad\qquad\qquad (51)$$
$$(\boldsymbol{v} p + \kappa \nabla p) \cdot \boldsymbol{n} = 0 \qquad \text{on } \partial \mathcal{D} \times (0, T).$$

Note that Equation (51) is a final value problem, since p is specified at $t = T$ rather than at $t = 0$. Thus, Equation (51) is solved backwards in time, which amounts to the solution of an advection-diffusion equation with velocity $-\boldsymbol{v}$.

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 $\mathcal{J}(m)$:

$$(\mathcal{G}(m), \tilde{m}) = \int_{\mathcal{D}} (\mathcal{A}(m - m_{\text{pr}}))(\mathcal{A}\tilde{m}) - p(\boldsymbol{x}, 0) \tilde{m} \, d\boldsymbol{x}, \quad \forall \tilde{m} \in \mathcal{M}, \qquad (52)$$

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

$$\mathcal{G}(m) = \begin{cases} \mathcal{A}^2(m - m_{\text{pr}}) - p(\boldsymbol{x}, 0) & \text{in } \mathcal{D} \\ \gamma \nabla m \cdot \boldsymbol{n} + \beta m & \text{in } \partial \mathcal{D}. \end{cases} \qquad (53)$$

Thus, the adjoint of the parameter-to-observable map $\mathcal{F}^*$ is defined by setting $\mathcal{F}^* \boldsymbol{d} = p(\boldsymbol{x}, 0)$. We note that the gradient expression in Equation (53) is linear in m, since p depends linearly on u through the solution of the adjoint problem (Equation (51)), and u depends linearly on m

through the solution of the forward problem (Equation (39)). Elimination of the forward and adjoint equations for a given m gives the action of the linear operator in Equation (53) in the direction of that m. Thus, to compute the MAP point, we use CG to set $\mathcal{G}(m) = 0$ with the inverse of the regularization operator $\mathcal{A}^2$ as preconditioner.

C   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. A similar approach has been also proposed in Croci et al. [2018] to generate samples of white noise. The method described here is more general as it can be applied to matrices stemming from finite element discretization of any differential operator and not only mass matrices. More specifically, we present a finite element assembly procedure to compute a rectangular decomposition of the form

$$\mathbf{A} = \mathbf{C}\mathbf{C}^T, \tag{54}$$

for any symmetric positive definite finite element matrix $\mathbf{A}$. 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 entries (i, j) of the matrix $\mathbf{A}$, 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 $\mathbf{A}_e$, 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) $\mathbf{G}_e$, the global matrix $\mathbf{A}$ is computed by summing all of the local contributions as follows:

$$\mathbf{A} = \sum_e \mathbf{G}_e^T \mathbf{A}_e \mathbf{G}_e = \sum_e \mathbf{G}_e^T \mathbf{B}^T \mathbf{D}_e \mathbf{B} \mathbf{G}_e. \tag{55}$$

Here we have written the element matrix $\mathbf{A}_e = \mathbf{B}^T \mathbf{D}_e \mathbf{B}$ as the product of the element-independent dof-to-quadrature point basis evaluation matrix $\mathbf{B}$ and the (block) diagonal matrix $\mathbf{D}_e \in \mathbb{R}^{q \times q}$ 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 $\mathbf{A}$ can then be explicitly constructed from the matrices $\mathbf{G}_e$, $\mathbf{B}$, and $\mathbf{D}_e$ as follows. For each element e of the mesh we define the matrix $\mathbf{C}_e = \mathbf{G}_e^T \mathbf{B}^T \mathbf{D}_e^{\frac{1}{2}}$. 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

$$\mathbf{C}_{e_i} \mathbf{C}_{e_j}^T = \delta_{ij} \mathbf{A}_{e_i}. \tag{56}$$

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

$$\mathbf{C} = \sum_e \mathbf{G}_e^T \mathbf{B}^T \mathbf{D}_e^{\frac{1}{2}} = \sum_e \mathbf{C}_e \tag{57}$$

satisfies Equation (54). In fact, thanks to Equation (56), we have

$$\mathbf{C}\mathbf{C}^T = \left(\sum_e \mathbf{C}_e\right)\left(\sum_e \mathbf{C}_e\right)^T = \sum_e \mathbf{C}_e \mathbf{C}_e^T = \sum_e \mathbf{G}_e^T \mathbf{B}^T \mathbf{D}_e \mathbf{B} \mathbf{G}_e = \mathbf{A}.$$

REFERENCES

Brian M. Adams, W. J. Bohnhoff, K. R. Dalbey, J. P. Eddy, M. S. Eldred, D. M. Gay, K. Haskell, Patricia D. Hough, and L. P. Swiler. 2009. *DAKOTA, A Multilevel Parallel Object-oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 5.0 User's Manual.* Technical Report.

Volkan Akçelik, George Biros, Omar Ghattas, Judith Hill, David Keyes, and Bart van Bloeman Waanders. 2006. Parallel PDE-constrained optimization. In *Parallel Processing for Scientific Computing*, M. Heroux, P. Raghaven, and H. Simon (Eds.). SIAM.

Alen Alexanderian, Noemi Petra, Georg Stadler, and Omar Ghattas. 2014. A-optimal design of experiments for infinite-dimensional Bayesian linear inverse problems with regularized ℓ_0-sparsification. *SIAM Journal on Scientific Computing* 36, 5 (2014), A2122–A2148. DOI : https://doi.org/10.1137/130933381

Alen Alexanderian, Noemi Petra, Georg Stadler, and Omar Ghattas. 2016. A fast and scalable method for A-optimal design of experiments for infinite-dimensional Bayesian nonlinear inverse problems. *SIAM Journal on Scientific Computing* 38, 1 (2016), A243–A272. DOI : https://doi.org/10.1137/140992564

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. 2014. PETSc Web page. http://www.mcs.anl.gov/petsc.

Wolfgang Bangerth, Ralf Hartmann, and Guido Kanschat. 2007. deal.II – A general-purpose object-oriented finite element library. *ACM Transactions on Mathematical Software* 33, 4 (2007), 24. DOI : https://doi.org/10.1145/1268776.1268779

P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, M. Ohlberger, and O. Sander. 2008. A generic grid interface for parallel and adaptive scientific computing. Part I: Abstract framework. *Computing* 82, 2 (2008), 103–119.

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

Costas Bekas, Effrosyni Kokiopoulou, and Yousef Saad. 2007. An estimator for the diagonal of a matrix. *Applied Numerical Mathematics* 57, 11 (2007), 1214–1229.

Alexandros Beskos, Mark Girolami, Shiwei Lan, Patrick E. Farrell, and Andrew M. Stuart. 2017. Geometric MCMC for infinite-dimensional inverse problems. *Journal of Computational Physics* 335 (2017), 327–351.

Alfio Borzì and Volker Schulz. 2012. *Computational Optimization of Systems Governed by Partial Differential Equations.* SIAM.

Tan Bui-Thanh, Carsten Burstedde, Omar Ghattas, James Martin, Georg Stadler, and Lucas C. Wilcox. 2012. Extreme-scale UQ for Bayesian inverse problems governed by PDEs. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC'12)*. Gordon Bell Prize finalist.

Tan Bui-Thanh and Omar Ghattas. 2012. Analysis of the Hessian for inverse scattering problems. Part II: Inverse medium scattering of acoustic waves. *Inverse Problems* 28, 5 (2012), 055002. DOI : https://doi.org/10.1088/0266-5611/28/5/055002

Tan Bui-Thanh, Omar Ghattas, James Martin, and Georg Stadler. 2013. A computational framework for infinite-dimensional Bayesian inverse problems Part I: The linearized case, with application to global seismic inversion. *SIAM Journal on Scientific Computing* 35, 6 (2013), A2494–A2523. DOI : https://doi.org/10.1137/12089586X

Tan Bui-Thanh and Mark Andrew Girolami. 2014. Solving large-scale PDE-constrained bayesian inverse problems with Riemann manifold Hamiltonian Monte Carlo. *Inverse Problems* 30 (2014), 114014.

S. L. Campbell, I. C. F. Ipsen, C. T. Kelley, C. D. Meyer, and Z. Q. Xue. 1996. Convergence estimates for solution of integral equations with GMRES. *Journal of Integral Equations and Applications* 8 (1996), 19–34.

Jie Chen, Mihai Anitescu, and Yousef Saad. 2011. Computing $f(A)b$ via least squares polynomial approximations. *SIAM Journal on Scientific Computing* 33, 1 (Feb. 2011), 195–222. DOI : https://doi.org/10.1137/090778250

P. Chen, U. Villa, and O. Ghattas. 2017. Hessian-based adaptive sparse quadrature for infinite-dimensional Bayesian inverse problems. *Computer Methods in Applied Mechanics and Engineering* 327 (2017), 147–172. https://doi.org/10.1016/j.cma.2017.08.016

Peng Chen, Umberto Villa, and Omar Ghattas. 2019. Taylor approximation and variance reduction for PDE-constrained optimal control under uncertainty. *Journal of Computational Physics* 385 (2019), 163–186. https://arxiv.org/abs/1804.04301

Edmond Chow and Yousef Saad. 2014. Preconditioned Krylov subspace methods for sampling multivariate gaussian distributions. *SIAM Journal on Scientific Computing* 36, 2 (2014), A588–A608.

COMSOL AB. 2009. COMSOL Multiphysics Reference Guide. http://www.comsol.com.

Matteo Croci, Michael B. Giles, Marie E. Rognes, and Patrick E. Farrell. 2018. Efficient white noise sampling and coupling for multilevel monte carlo with nonnested meshes. *SIAM/ASA Journal on Uncertainty Quantification* 6, 4 (2018), 1630–1655.

T. Cui, K. J. H. Law, and Y. M. Marzouk. 2016. Dimension-independent likelihood-informed MCMC. *Journal of Computational Physics* 304 (2016), 109–137.

Tiangang Cui, James Martin, Youssef M. Marzouk, Antti Solonen, and Alessio Spantini. 2014. Likelihood-informed dimension reduction for nonlinear inverse problems. *Inverse Problems* 30, 11 (2014), 114015.

Tiangang Cui, James Martin, Youssef M. Marzouk, Antti Solonen, and Alessio Spantini. 2014c. Likelihood-informed dimension reduction for nonlinear inverse problems. *Inverse Problems* 30, 11 (2014), 114015. http://stacks.iop.org/0266-5611/30/i=11/a=114015

Yair Daon and Georg Stadler. 2018. Mitigating the influence of boundary conditions on covariance operators derived from elliptic PDEs. *Inverse Problems and Imaging* 12, 5 (2018), 1083–1102. arXiv:1610.05280

B. Debusschere, K. Sargsyan, C. Safta, and K. Chowdhary. 2017. *Uncertainty Quantification Toolkit (UQTk)*. Springer International Publishing, 1807–1827. DOI:https://doi.org/10.1007/978-3-319-12385-1_56

Stanley C. Eisenstat and Homer F. Walker. 1996. Choosing the forcing terms in an inexact Newton method. *SIAM Journal on Scientific Computing* 17, 1 (1996), 16–32. DOI:https://doi.org/10.1137/0917003

M. S. Eldred, A. A. Giunta, B. G. van Bloemen Waanders, S. F. Wojkiewicz, W. E. Hart Jr., and M. P. Alleva. 2002. *DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis. Version 3.0 Reference Manual*. Technical Report SAND2001-3796. Sandia National Laboratories.

Heinz W. Engl, Martin Hanke, and Andreas Neubauer. 1996. *Regularization of Inverse Problems*. Springer Netherlands.

M. Evans and T. Swartz. 2000. *Approximating Integrals Via MOnte Carlo and Deterministic Methods*. Vol. 20. OUP Oxford.

Patrick E. Farrell, David A. Ham, Simon W. Funke, and Marie E. Rognes. 2013. Automated derivation of the adjoint of high-level transient finite element programs. *SIAM Journal on Scientific Computing* 35, 4 (2013), C369–C393.

Pearl H. Flath. 2013. *Hessian-based Response Surface Approximations for Uncertainty Quantification in Large-scale Statistical Inverse Problems, with Applications to Groundwater Flow*. Ph.D. Dissertation. The University of Texas at Austin.

Pearl H. Flath, Lucas C. Wilcox, Volkan Akçelik, Judy Hill, Bart van Bloemen Waanders, and Omar Ghattas. 2011. 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, 1 (2011), 407–432. DOI:https://doi.org/10.1137/090780717

Nathan Halko, Per Gunnar Martinsson, and Joel A. Tropp. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review* 53, 2 (2011), 217–288.

Matthias Heinkenschloss. 1993. Mesh independence for nonlinear least squares problems with norm constraints. *SIAM Journal on Optimization* 3 (1993), 81–117.

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. 2005. An overview of the Trilinos project. *ACM Transactions on Mathematical Software* 31, 3 (Sept. 2005), 397–423. DOI:https://doi.org/10.1145/1089014.1089021

Marc Hesse and Georg Stadler. 2014. Joint inversion in coupled quasistatic poroelasticity. *Journal of Geophysical Research: Solid Earth* 119, 2 (2014), 1425–1445.

T. J. R. Hughes, L. P. Franca, and G. M. Hulbert. 1989. A new finite element formulation for computational fluid dynamics: VIII the Galerkin/least-squares method for advection-diffusion equations. *Computer Methods in Applied Mechanics and Engineering* 73 (1989), 173–189.

Tobin Isaac, Noemi Petra, Georg Stadler, and Omar Ghattas. 2015. 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 (September 2015), 348–368. DOI:https://doi.org/10.1016/j.jcp.2015.04.047

Jari Kaipio and Erkki Somersalo. 2005. *Statistical and Computational Inverse Problems*. Applied Mathematical Sciences, Vol. 160. Springer-Verlag, New York. xvi+339 pages.

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

Hans Petter Langtangen and Anders Logg. 2017. *Solving PDEs in Python*. Springer. DOI:https://doi.org/10.1007/978-3-319-52462-7

Finn Lindgren, H*r*avard Rue, and Johan Lindström. 2011. 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, 4 (2011), 423–498. DOI:https://doi.org/10.1111/j.1467-9868.2011.00777.x

Anders Logg, Kent-Andre Mardal, and Garth N. Wells (Eds.). 2012. *Automated Solution of Differential Equations by the Finite Element Method*. Lecture Notes in Computational Science and Engineering, Vol. 84. Springer. DOI:https://doi.org/10.1007/978-3-642-23099-8

Quan Long, Mohammad Motamed, and Raúl Tempone. 2015a. Fast bayesian optimal experimental design for seismic source inversion. *Computer Methods in Applied Mechanics and Engineering* 291 (2015), 123–145. DOI:https://doi.org/10.1016/j.cma.2015.03.021

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

Quan Long, Marco Scavino, Raúl Tempone, and Suojin Wang. 2015b. A Laplace method for under-determined Bayesian optimal experimental designs. *Computer Methods in Applied Mechanics and Engineering* 285 (2015), 849-876. DOI : https://doi.org/10.1016/j.cma.2014.12.008

B. R. Lowery and J. Langou. 2014. Stability analysis of QR factorization in an oblique inner product. *ArXiv E-prints* (jan 2014). arxiv:math.NA/1401.5171

James Martin, Lucas C. Wilcox, Carsten Burstedde, and Omar Ghattas. 2012. A stochastic newton MCMC method for large-scale statistical inverse problems with application to seismic inversion. *SIAM Journal on Scientific Computing* 34, 3 (2012), A1460–A1487. DOI : https://doi.org/10.1137/110845598

D. McDougall, N. Malaya, and R. D. Moser. 2017. *The Parallel C++ Statistical Library for Bayesian Inference: QUESO*. Springer International Publishing, 1829–1865. DOI : https://doi.org/10.1007/978-3-319-12385-1_57

Dirk Merkel. 2014. Docker: Lightweight Linux containers for consistent development and deployment. *Linux Journal* 2014, 239, Article 2 (2014). http://dl.acm.org/citation.cfm?id=2600239.2600241

Albert Parker and Colin Fox. 2012. Sampling Gaussian distributions in Krylov spaces with conjugate gradients. *SIAM Journal on Scientific Computing* 34, 3 (2012), B312–B334.

M. Parno, A. Davis, P. Conrad, and Y. M. Marzouk. 2012–2015. MIT Uncertainty Quantification (MUQ) library. https://bitbucket.org/mituq/muq.

Noemi Petra, James Martin, Georg Stadler, and Omar Ghattas. 2014. A computational framework for infinite-dimensional Bayesian inverse problems: Part II. stochastic Newton MCMC with application to ice sheet inverse problems. *SIAM Journal on Scientific Computing* 36, 4 (2014), A1525–A1555.

Noemi Petra and Georg Stadler. 2011. *Model Variational Inverse Problems Governed by Partial Differential Equations*. Technical Report 11-05. The Institute for Computational Engineering and Sciences, The University of Texas at Austin.

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

Ernesto E. Prudencio and Karl W. Schulz. 2012. 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 Traff, Geoffroy Vallee, and Josef Weidendorfer (Eds.). Lecture Notes in Computer Science, Vol. 7155. Springer, Berlin, 398–407. DOI : https://doi.org/10.1007/978-3-642-29737-3_44

Lassi Roininen, Janne M. J. Huttunen, and Sari Lasanen. 2014. Whittle-Matérn priors for Bayesian statistical inversion with applications in electrical impedance tomography. *Inverse Problems & Imaging* 8, 2 (2014), 561. DOI : https://doi.org/10.3934/ipi.2014.8.561

L. Ruthotto, E. Treister, and E. Haber. 2017. jInv – A flexible julia package for PDE parameter estimation. *SIAM Journal on Scientific Computing* 39, 5 (2017), S702–S722.

A. K. Saibaba, J. Lee, and P. K. Kitanidis. 2016. Randomized algorithms for generalized hermitian eigenvalue problems with application to computing Karhunen-Loéve expansion. *Numerical Linear Algebra with Applications* 23, 2 (2016), 314–339. DOI : https://doi.org/10.1002/nla.2026 nla.2026.

Claudia Schillings and Christoph Schwab. 2016. Scaling limits in computational Bayesian inversion. *ESAIM: M2AN* 50, 6 (2016), 1825–1856. DOI : https://doi.org/10.1051/m2an/2016005

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

S. M. Stigler.1986. Laplace's 1774 memoir on inverse probability. *Statistical Science* 1, 3 (08 1986), 359–363. DOI : https://doi.org/10.1214/ss/1177013620

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

Andrew M. Stuart. 2010. Inverse problems: A Bayesian perspective. *Acta Numerica* 19 (2010), 451–559. DOI : https://doi.org/doi:10.1017/S0962492910000061

Jok M. Tang and Yousef Saad. 2012. A probing method for computing the diagonal of a matrix inverse. *Numerical Linear Algebra with Applications* 19, 3 (2012), 485–501.

Albert Tarantola. 2005. *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, Philadelphia, PA. xii+342 pages.

L. Tierney and J. B. Kadane. 1986. Accurate approximations for posterior moments and marginal densities. *Journal of American Statistical Association* 81, 393 (1986), 82–86. DOI : https://doi.org/10.1080/01621459.1986.10478240

A. N. Tikhonov and V. A. Arsenin. 1977. *Solution of Ill-posed Problems*. Winston & Sons.

C. Tong. 2017. *Problem Solving Environment for Uncertainty Analysis and Design Exploration*. Springer International Publishing, 1695–1731. DOI : https://doi.org/10.1007/978-3-319-12385-1_53

Fredi Tröltzsch. 2010. *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*. Graduate Studies in Mathematics, Vol. 112. American Mathematical Society.

Umberto Villa, Noemi Petra, and Omar Ghattas. 2020. hIPPYlib User Manual: Version 2. DOI: https://doi.org/10.6084/m9.figshare.12510578

Curt R. Vogel. 2002. *Computational Methods for Inverse Problems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA. xvi+183 pages.

K. Wang, T. Bui-Thanh, and O. Ghattas. 2018. A randomized maximum a posteriori method for posterior sampling of high dimensional nonlinear bayesian inverse problems. *SIAM Journal on Scientific Computing* 40, 1 (2018), A142–A171.

David Williams. 1991. *Probability with Martingales*. Cambridge University Press.

R. Wong. 2001. *Asymptotic Approximations of Integrals*. Society for Industrial and Applied Mathematics. DOI: https://doi.org/10.1137/1.9780898719260 arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9780898719260

Jennifer Worthen. 2012. *Inverse Problems in Mantle Convection: Models, Algorithms, and Applications*. Ph.D. Dissertation. The University of Texas at Austin.

Jennifer Worthen, Georg Stadler, Noemi Petra, Michael Gurnis, and Omar Ghattas. 2014. Towards adjoint-based inversion for rheological parameters in nonlinear viscous Mante flow. *Physics of the Earth and Planetary Interiors* 234 (2014), 23–34. DOI: https://doi.org/10.1016/j.pepi.2014.06.006