hIPPYlib-MUQ: A Bayesian Inference Software Framework for Integration of Data with Complex Predictive Models under Uncertainty

KI-TAE KIM, University of California, Merced, USA
UMBERTO VILLA, The University of Texas at Austin, USA
MATTHEW PARNO, Dartmouth College, USA
YOUSSEF MARZOUK, Massachusetts Institute of Technology, USA
OMAR GHATTAS, The University of Texas at Austin, USA
NOEMI PETRA, University of California, Merced, USA

Bayesian inference provides a systematic framework for integration of data with mathematical models to quantify the uncertainty in the solution of the inverse problem. However, the solution of Bayesian inverse problems governed by complex forward models described by **partial differential equations (PDEs)** remains prohibitive with black-box **Markov chain Monte Carlo (MCMC)** methods. We present hIPPYlib-MUQ, an extensible and scalable software framework that contains implementations of state-of-the art algorithms aimed to overcome the challenges of high-dimensional, PDE-constrained Bayesian inverse problems. These algorithms accelerate MCMC sampling by exploiting the geometry and intrinsic low-dimensionality of parameter space via derivative information and low rank approximation. The software integrates two complementary open-source software packages, hIPPYlib and MUQ. hIPPYlib solves PDE-constrained inverse problems using automatically-generated adjoint-based derivatives, but it lacks full Bayesian capabilities. MUQ provides a spectrum of powerful Bayesian inversion models and algorithms, but expects forward models to come equipped with gradients and Hessians to permit large-scale solution. By combining these two complementary libraries, we created a robust, scalable, and efficient software framework that realizes the benefits of each and allows us to tackle complex large-scale Bayesian inverse problems across a broad spectrum of scientific and engineering disciplines. To illustrate the capabilities of hIPPYlib-MUQ, we present a

This work was supported by the U.S. National Science Foundation, Software Infrastructure for Sustained Innovation (SI2: SSE & SSI) Program under grants ACI-1550593, ACI-1550547, and ACI-1550487 and the Division of Mathematical Sciences under the CAREER grant 1654311. MP and YM were also supported in part by Office of Naval Research MURI grant N00014-20-1-2595. OG was also supported in part by Department of Energy Advanced Scientific Computing Research grants DE-SC0021239 and DE-SC0019303. The authors gratefully acknowledge computing time on the Multi-Environment Computer for Exploration and Discovery (MERCED) cluster at UC Merced, which was funded by National Science Foundation Grant No. ACI-1429783.

Authors' addresses: K.-T. Kim and N. Petra, University of California, Merced, Department of Applied Mathematics, 5200 N. Lake Road, Merced, CA, 95343, USA; emails: {kkim107, npetra}@ucmerced.edu; U. Villa, The University of Texas at Austin, Oden Institute for Computational Engineering & Sciences, 201 E. 24th Street, Austin, TX, 78712-1229, USA; email: uvilla@oden.utexas.edu; M. Parno, Dartmouth College, Department of Mathematics, 27 N. Main Street , Hanover, NH, 03755-3551, USA; email: matthew.d.parno@dartmouth.edu; Y. Marzouk, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 77 Massachusetts Avenue, Cambridge, MA, 02139, USA; email: ymarz@mit.edu; O. Ghattas, The University of Texas at Austin, Oden Institute for Computational Engineering & Sciences, Department of Mechanical Engineering, 201 E. 24th Street, Austin, TX, 78712-1229, USA; email: omar@oden.utexas.edu.

0098-3500/2023/06-ART17 $15.00
https://doi.org/10.1145/3580278

comparison of a number of MCMC methods available in the integrated software on several high-dimensional Bayesian inverse problems. These include problems characterized by both linear and nonlinear PDEs, various noise models, and different parameter dimensions. The results demonstrate that large (~50×) speedups over conventional black box and gradient-based MCMC algorithms can be obtained by exploiting Hessian information (from the log-posterior), underscoring the power of the integrated hIPPYlib-MUQ framework.

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

ACM Reference format:
Ki-Tae Kim, Umberto Villa, Matthew Parno, Youssef Marzouk, Omar Ghattas, and Noemi Petra. 2023. hIPPYlib-MUQ: A Bayesian Inference Software Framework for Integration of Data with Complex Predictive Models under Uncertainty. *ACM Trans. Math. Softw.* 49, 2, Article 17 (June 2023), 31 pages.
https://doi.org/10.1145/3580278

1 INTRODUCTION

With the rapid explosion of observational and experimental data, a prominent challenge is how to derive knowledge and insight from this data to make better predictions and high-consequence decisions. This question arises in all areas of science, engineering, technology, and medicine, and in many cases, there are mathematical models available that represent the underlying physical systems of which the data is observed or measured. These models are often subject to considerable uncertainties stemming from unknown or uncertain input model parameters (e.g., coefficient fields, constitutive laws, source terms, geometries, initial and/or boundary conditions) as well as from noisy and limited observations. The goal is to infer these unknown model parameters from observations of model outputs through corresponding **partial differential equation (PDE)** models, and to quantify the uncertainty in the solution of such inverse problems.

Bayesian inversion provides a systematic framework for integration of data with complex PDE-based models to quantify uncertainties in model parameter inference [Kaipio and Somersalo 2005; Tarantola 2005]. In the Bayesian framework, noisy data and, possibly uncertain, mathematical models are integrated together with a prior information, yielding a posterior probability distribution of the model parameters. The **Markov chain Monte Carlo (MCMC)** method is a common way to explore the posterior distribution by use of sampling techniques. However, Bayesian inversion with complex forward models via conventional MCMC methods faces several computational challenges. First, characterizing the posterior distribution of the model parameters or subsequent predictions often requires repeated evaluations of expensive-to-solve large-scale PDE models. Second, the posterior distribution often has a complex structure stemming from the nonlinear mapping from model parameter to observed quantities. Third, the parameters often are fields, which, after discretization, lead to very high-dimensional posteriors. These difficulties make the solution of Bayesian inverse problems with complex large-scale PDE forward models computationally intractable.

Extensive research efforts have been devoted to overcome the prohibitiveness of Bayesian inverse problems governed by large-scale PDEs. With rapid progress in high-performance computing, and advances in scalable PDE solvers, repeated evaluations of forward PDE models for different input parameters [Balay et al. 2018; Trilinos Project Team 2020] are becoming tractable.

Furthermore, structure-exploiting MCMC methods have effectively facilitated the exploration of complex posterior distributions [Beskos et al. 2017; Bui-Thanh et al. 2012; Cotter et al. 2012; Petra et al. 2014]. Finally, dimension reduction methods have proven to significantly reduce the computational cost of MCMC simulations [Cui et al. 2016b; Zahm et al. 2022]. Applying and combining these advanced techniques can be extremely challenging. Therefore, a computational tool that will assist the computational and scientific community to apply, extend, and tailor these methods will be very beneficial.

In this paper, we present a software framework to tackle large-scale Bayesian inverse problems with PDE-based forward models, which has applications across a wide range of science and engineering fields. The software integrates two open-source software packages, an **Inverse Problems Python library (hIPPYlib)** [Villa et al. 2021] and the **MIT Uncertainty Quantification Library (MUQ)** [Parno et al. 2014], respecting their attractive complementary capabilities.

hIPPYlib is an extensible software framework for the solution of deterministic and linearized Bayesian inverse problems constrained by complex PDE models. Based on FEniCS [Logg et al. 2012] for the finite element approximation of PDEs and on PETSc [Balay et al. 2014] for high-performance linear algebra operations and solvers, hIPPYlib allows users to describe (and solve) the underlying PDE-based forward model (required by the inverse problem solver) in a relatively straightforward way. hIPPYlib also contains implementations of efficient numerical methods for the solution of deterministic and linearized Bayesian inverse problems. These include globalized inexact Newton-conjugate gradient [Akçelik et al. 2006; Borzì and Schulz 2012], adjoint-based computation of gradients and Hessian actions [Tröltzsch 2010], randomized linear algebra [Halko et al. 2011], and scalable sampling from large-scale Gaussian fields. The state-of-the-art algorithms implemented in hIPPYlib efficiently deliver the solution of the linearized Bayesian inverse problem. hIPPYlib is, however, mainly designed for deterministic and linearized Bayesian inverse problems, and lacks full Bayesian inversion capabilities.

MUQ complements hIPPYlib's capabilities with more support for the formulation and solution of Bayesian inference problems. MUQ is a modular software framework designed to address uncertainty quantification problems involving complex models. The software provides an abstract modeling interface for combining physical (e.g., PDEs) and statistical components (e.g., additive error models, Gaussian process priors, etc.) to define Bayesian posterior distributions in a flexible and semi-intrusive way. MUQ also contains a suite of powerful uncertainty quantification algorithms including Markov chain Monte Carlo (MCMC) methods [Parno and Marzouk 2018], transport maps [Marzouk et al. 2016], likelihood-informed subspaces, sparse adaptive **generalized polynomial chaos (gPC)** expansions [Conrad and Marzouk 2013], Karhunen-Loéve expansions, Gaussian process modeling [Hartikainen and Särkkä 2010; Rasmussen and Williams 2005], and prediction methods enabling global sensitivity analysis and optimal experimental design. To effectively apply these tools to Bayesian inverse problems, however, MUQ needs to be equipped with the type of gradient and/or Hessian information that hIPPYlib can provide.

By interfacing these two software libraries, we aim to create a robust, scalable, efficient, flexible, and easy-to-use software framework that overcomes the computational challenges inherent in complex large-scale Bayesian inverse problems. Representative features of the software are summarized as follows:

- The software combines the benefits of the two packages, hIPPYlib and MUQ, to enable scalable solution of Bayesian inverse problems governed by large-scale PDEs.
- Various advanced MCMC methods are available that can exploit problem structure (e.g., the derivative/Hessian information of the log-posterior).

- The software makes use of sparsity, low-dimensionality, and geometric structure of the log-posterior to achieve scalable and efficient MCMC methods.
- Convergence diagnostics are implemented to assess the quality of MCMC samples.

In the following sections, we first briefly review the Bayesian formulation of inverse problems governed by PDEs both in infinite-dimensional and in finite-dimensional spaces (Section 2). We then describe MCMC methods used to characterize the posterior (Section 3) and summarize convergence diagnostics available in the software (Section 3.2). Next, we present the design of hIPPYlib-MUQ (Section 4). Finally, we present numerous benchmark problems and a step-by-step implementation guide to illustrate the key aspect of the present software (Section 5). Section 6 provides concluding remarks.

2 THE BAYESIAN INFERENCE FRAMEWORK

In this section, we present a brief discussion of the Bayesian inference approach to solving inverse problems governed by PDEs. We begin by providing an overview of the framework for infinite-dimensional Bayesian inverse problems following Bui-Thanh et al. [2013]; Petra et al. [2014]; Stuart [2010]. Then 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]. Finally, we present the Laplace approximation to the posterior distribution, which requires the solution of a PDE-constrained optimization problem for the computation of the ***maximum a posteriori* (MAP)**.

2.1 Infinite-dimensional Bayesian Inverse Problems

The objective of the inverse problem is to determine an unknown input parameter field m that would give rise to given observational (or experimental) noisy data $\boldsymbol{d}$ by means of a (physics-based) mathematical model. In other words, given $\boldsymbol{d} \in \mathbb{R}^q$, we seek to infer $m \in \mathcal{M}$ such that

$$\boldsymbol{d} \approx \mathcal{F}(m), \tag{1}$$

where $\mathcal{F} : \mathcal{M} \to \mathbb{R}^q$ is the *parameter-to-observable* map that predicts observations from a given parameter m through a forward mathematical model, and $\mathcal{M}$ is an infinite-dimensional Hilbert space of functions defined on a domain $\mathcal{D} \subset \mathbb{R}^d$. Note that the evaluation of this map involves solving the forward PDE model given m, followed by extracting the observations from the solution of the forward problem. In what follows, we assume that the forward equation residual is continuously Fréchet-differentiable and its Jacobian a continuous linear operator with continuous inverse [Ghattas and Willcox 2021].

In the Bayesian approach, the inverse problem is framed as a statistical inference problem. The uncertain parameter m and the observational data $\boldsymbol{d}$ are deemed as random variables and the solution is a conditional probability distribution $\mu_{\text{post}}(m|\boldsymbol{d})$ that represents level of confidence in the estimation of the parameter given the data. The approach combines a *prior model* reflecting our prior knowledge or assumptions about the parameters before data are acquired, and a *likelihood model* representing the probability that a given set of parameters might give rise to the observed data.

Using the Radon-Nikodym derivative [Williams 1991] of the posterior measure $\mu_{\text{post}}(m)$ with respect to the prior measure $\mu_{\text{prior}}(m)$, Bayes' theorem in infinite dimensions is stated as

$$\frac{d\mu_{\text{post}}}{d\mu_{\text{prior}}} \propto \pi_{\text{like}}(\boldsymbol{d}|m), \tag{2}$$

where π_{like} denotes the likelihood function. For detailed conditions under which the posterior measure is well-defined, we refer the reader to Stuart [2010].

Without loss of generality, the probability density function of the likelihood can be expressed as

$$\pi_{\text{like}}(\boldsymbol{d}|m) \propto \exp\left\{-\Phi(m;\boldsymbol{d})\right\}. \tag{3}$$

The negative log-likelihood function $\Phi(m;\boldsymbol{d})$ has different forms depending on how one models the noise that stems from measurement uncertainties and/or modeling errors; for example, in the case of an additive Gaussian noise model $\boldsymbol{d} = \mathcal{F}(m) + \boldsymbol{\eta}$ with a Gaussian noise random variable $\boldsymbol{\eta} \in \mathbb{R}^q$ with zero mean and covariance matrix $\boldsymbol{\Gamma}_{\text{noise}} \in \mathbb{R}^{q\times q}$, it has the form $\Phi(m;\boldsymbol{d}) = \frac{1}{2}\|\mathcal{F}(m) - \boldsymbol{d}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}}$, where $\|\cdot\|_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}}$ denotes the L^2 norm weighted by the inverse noise covariance $\boldsymbol{\Gamma}^{-1}_{\text{noise}}$.

We take the prior to be a Gaussian measure,[1] i.e., $\mu_{\text{prior}} = \mathcal{N}(m_{\text{pr}}, \mathcal{C}_{\text{prior}})$, and assume that samples from the prior distribution are square-integrable functions in the domain $\mathcal{D}$, i.e., belong to $L^2(\mathcal{D})$. The prior covariance operator $\mathcal{C}_{\text{prior}}$ is constructed to be a trace-class operator to guarantee bounded variance of samples from the prior distribution and well-posedness of the Bayesian inverse problem [Bui-Thanh et al. 2013; Stuart 2010; Villa et al. 2021] for detailed explanation. Specifically, we take the prior covariance operator as the inverse of the υth power of a Laplacian-like operator, namely $\mathcal{C}_{\text{prior}} := \mathcal{A}^{-\upsilon} = (-\gamma\Delta + \delta I)^{-\upsilon};\ \upsilon > \frac{d}{2}$. Here γ and $\delta > 0$ control the correlation length and the pointwise variance of the prior operator [Lindgren et al. 2011; Villa et al. 2021]. These choices of prior ensure that $\mathcal{C}_{\text{prior}}$ 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].

2.2 Discretization of the Bayesian Formulation

Here, we briefly present the finite-dimensional approximation of the Bayesian formulation described in the previous section. We consider a finite-dimensional subspace $\mathcal{M}_h$ of $\mathcal{M}$, defined by the span of a set of basis functions $\{\phi_j\}_{j=1}^n$. For example, in hIPPYlib-MUQ, these basis functions are globally continuous piecewise polynomials on each element of a mesh discretization of the domain $\mathcal{D}$ [Becker et al. 1981; Strang and Fix 1988]. These are the natural choice for the discretization of the elliptic operator $\mathcal{A}$ used in the definition of the prior covariance. The parameter field m is then approximated as $m \approx m_h = \sum_{j=1}^n m_j\phi_j$, and, in what follows, $\boldsymbol{m} = (m_1, \ldots, m_n)^T \in \mathbb{R}^n$ denotes the vector of the finite element coefficients of m_h.

The finite-dimensional approximation of the prior measure μ_{prior} is now specified by the density

$$\pi_{\text{prior}}(\boldsymbol{m}) \propto \exp\left(-\frac{1}{2}\|\boldsymbol{m} - \boldsymbol{m}_{\text{pr}}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{prior}}}\right), \tag{4}$$

where $\boldsymbol{m}_{\text{pr}} \in \mathbb{R}^n$ and $\boldsymbol{\Gamma}_{\text{prior}} \in \mathbb{R}^{n\times n}$ are the mean vector and the covariance matrix that arise upon discretization of m_{pr} and $\mathcal{C}_{\text{prior}}$, respectively. We refer the reader to Bui-Thanh et al. [2013]; Villa et al. [2021] for the explicit expression of the prior covariance matrix $\boldsymbol{\Gamma}_{\text{prior}}$.

Then the Bayes' theorem for the density of the finite-dimensional approximation of the posterior measure μ_{post} is given by

$$\pi_{\text{post}}(\boldsymbol{m}) := \pi_{\text{post}}(\boldsymbol{m}|\boldsymbol{d}) \propto \pi_{\text{like}}(\boldsymbol{d}|\boldsymbol{m})\pi_{\text{prior}}(\boldsymbol{m}). \tag{5}$$

The finite-dimensional posterior probability density function can be expressed explicitly as

$$\pi_{\text{post}}(\boldsymbol{m}) \propto \exp\left(-\Phi(\boldsymbol{m};\boldsymbol{d}) - \frac{1}{2}\|\boldsymbol{m} - \boldsymbol{m}_{\text{pr}}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{prior}}}\right). \tag{6}$$

[1]The use of a Gaussian measure (often with Matérn covariance covariance kernel) is a common choice in infinite-dimensional inverse problems, where the inversion parameter is a spatially varying field. In hIPPYlib-MUQ, non-Gaussian prior probability distributions can be handled in a case-by-case basis by introducing a nonlinear *invertible* change of variables to map the desired prior distribution to a Gaussian one.

Here, evaluating $\Phi(\boldsymbol{m};\boldsymbol{d})$ requires constructing a finite dimensional discretization of the parameter-to-observable map, $\mathbf{F}(\boldsymbol{m})$.

2.3 The Laplace Approximation of the Posterior Distribution

In general, the posterior probability distribution (6) is not Gaussian due to the nonlinearity of the parameter-to-observable map. In this section, we discuss the solution to the so-called *linearized* Bayesian inverse problem by use of the Laplace approximation. The Laplace approximation amounts to constructing a Gaussian distribution centered at the *maximum a posteriori* (MAP) point. The MAP point represents the most probable value of the parameter vector over the posterior, i.e.,

$$\boldsymbol{m}_{\text{MAP}} := \underset{\boldsymbol{m}}{\text{argmin}}(-\log \pi_{\text{post}}(\boldsymbol{m})) = \underset{\boldsymbol{m}}{\text{argmin}} \left[\Phi(\boldsymbol{m};\boldsymbol{d}) + \frac{1}{2}\|\boldsymbol{m}-\boldsymbol{m}_{\text{pr}}\|^2_{\Gamma^{-1}_{\text{prior}}} \right]. \tag{7}$$

The covariance matrix of the Laplace approximation is the inverse of the Hessian of the negative log-posterior evaluated at the MAP point. Then under the Laplace approximation, the solution of the linearized Bayesian inverse problem is given by

$$\hat{\pi}_{\text{post}}(\boldsymbol{m}) \sim \mathcal{N}(\boldsymbol{m}_{\text{MAP}}, \Gamma_{\text{post}}), \tag{8}$$

with

$$\Gamma_{\text{post}} = \mathbf{H}^{-1}(\boldsymbol{m}_{\text{MAP}}) = \left(\mathbf{H}_{\text{misfit}}(\boldsymbol{m}_{\text{MAP}}) + \Gamma^{-1}_{\text{prior}}\right)^{-1}, \tag{9}$$

where $\mathbf{H}(\boldsymbol{m}_{\text{MAP}})$ and $\mathbf{H}_{\text{misfit}}(\boldsymbol{m}_{\text{MAP}})$ denote the Hessian matrices of, respectively, the negative log-posterior and the negative log-likelihood evaluated at the MAP point; see Villa et al. [2021] for a derivation of this Hessian using the adjoint-based method.

The quality of the Gaussian approximation of the posterior depends on the degree of nonlinearity in the parameter-to-observable map, the noise covariance matrix, and the number of observations [Bui-Thanh et al. 2013; Evans and Swartz 2000; Gelman et al. 2004; Isaac et al. 2015; Press 2003; Stigler 1986; Tarantola 2005; Tierney and Kadane 1986; Wong 2001]. When the parameter-to-observable map is linear and the additive noise and prior models are both Gaussian, the Laplace approximation is exact. Even if the parameter-to-observable map is significantly nonlinear, the Laplace approximation is a crucial ingredient to achieve scalable, efficient, and accurate posterior sampling with MCMC methods, as we will discuss in the following section.

Note that the Laplace approximation involves the Hessian of the negative log-likelihood (the data misfit part of the Hessian), which may not be explicitly constructed when the parameter dimension is large. However, the data typically provide only limited information about the parameter field, and thus the eigenspectrum of the Hessian matrix often decays very rapidly. We exploit this compact nature of the Hessian to overcome its prohibitive computational cost, and construct a low-rank approximation of the data misfit Hessian matrix using a matrix-free method (such as the randomized subspace iteration [Halko et al. 2011]).

Concretely, we construct a low-rank approximation of the data misfit Hessian, i.e., $\mathbf{H}_{\text{misfit}} \approx \Gamma^{-1}_{\text{prior}}\mathbf{V}_r\mathbf{\Lambda}_r\mathbf{V}_r^T\Gamma^{-1}_{\text{prior}}$, where $\mathbf{\Lambda}_r = \text{diag}(\lambda_1,\ldots,\lambda_r) \in \mathbb{R}^{r\times r}$ and $\mathbf{V}_r = [\boldsymbol{v}_1,\ldots,\boldsymbol{v}_r] \in \mathbb{R}^{n\times r}$ contain the r largest eigenvalues and corresponding eigenvectors, respectively, of the generalized symmetric eigenvalue problems

$$\mathbf{H}_{\text{misfit}}\boldsymbol{v}_i = \lambda_i \Gamma^{-1}_{\text{prior}}\boldsymbol{v}_i; \quad i = 1,\ldots,n. \tag{10}$$

Note that the eigenvectors $\boldsymbol{v}_i$ are orthonormal with respect to $\Gamma^{-1}_{\text{prior}}$, that is $\boldsymbol{v}_i^T\Gamma^{-1}_{\text{prior}}\boldsymbol{v}_j = \delta_{ij}$, where δ_{ij} is the Kronecker delta. With this row-rank approximation and using the Sherman-Morrison-Woodbury formula [Golub and Van Loan 1996], we obtain, for the inverse of the Hessian in (9),

$$\mathbf{H}^{-1} = \left(\mathbf{H}_{\text{misfit}} + \Gamma_{\text{prior}}^{-1}\right)^{-1} = \Gamma_{\text{prior}} - \mathbf{V}_r \mathbf{D}_r \mathbf{V}_r^T + \mathcal{O}\left(\sum_{i=r+1}^{n} \frac{\lambda_i}{1+\lambda_i}\right), \tag{11}$$

where $\mathbf{D}_r = \text{diag}(\lambda_1/(\lambda_1+1), \ldots, \lambda_r/(\lambda_r+1)) \in \mathbb{R}^{r\times r}$. We refer the reader to Villa et al. [2021] for a detailed description of how the randomized algorithm [Halko et al. 2011] proceeds to construct the low-rank approximation of the Hessian, including the associated computational complexity.

We can see from the last remainder term in (11) that to obtain an accurate low-rank approximation of $\mathbf{H}^{-1}$, we must keep eigenvectors corresponding to eigenvalues that are greater than 1. This approximation is used to efficiently perform various operations related to the Hessian, for example, applying the square-root inverse of the Hessian to a vector, which is needed to draw samples from the Gaussian approximation discussed in this section. We remark that the efficiency and scalability of our approach is based on the low rankness of the data misfit Hessian. We refer the reader to Ghattas and Willcox [2021], where this argument is made via model problems where low-rankness can be analytically shown, and for references to more complex problems where it can be shown empirically.

3 MCMC SAMPLING

As mentioned above, when the parameter-to-observable map is nonlinear, the Laplace approximation may be a poor approximation of the posterior. In this case, one needs to apply a sampling-based method to explore the full posterior. In this section, we focus on several advanced Markov chain Monte Carlo (MCMC) methods available in the present software. We outline the general structure of MCMC methods with a brief discussion of their key features. We then present various diagnostics to assess the convergence of MCMC simulations.

3.1 Markov Chain Monte Carlo

MCMC provides a flexible framework for exploring the posterior distribution. It generates samples from the posterior distribution that can be employed in Monte Carlo approximations of posterior expectations. For example, the posterior expectation of a quantity of interest $\mathcal{G}(m)$ can be approximated by

$$\int \mathcal{G}(m)\, d\mu_{\text{post}} \approx \frac{1}{N} \sum_{i=1}^{N} \mathcal{G}\left(m_i\right), \tag{12}$$

where each $m_i \sim \mu_{\text{post}}$ is a sample of the posterior distribution.

MCMC techniques construct ergodic Markov chains where the posterior distribution is the unique stationary distribution of the chain [Robert and Casella 2005]. Asymptotically, the states of the Markov chain are therefore exact samples of the posterior distribution and can be used in (12). Markov chains are defined in terms of a transition kernel, which is a position dependent probability distribution $K(\cdot|\boldsymbol{m}_i)$ over state $\boldsymbol{m}_{i+1}$ in the chain given the previous state $\boldsymbol{m}_i$, i.e., $\boldsymbol{m}_{i+1} \sim K(\cdot|\boldsymbol{m}_i)$. Note that chains of finite length must be employed in practice and the statistical accuracy of the Monte Carlo estimator is therefore highly dependent on the ability of the transition kernel to efficiently explore the parameter space.

There are several frameworks for constructing transition kernels that are appropriate for MCMC, including the well known **Metropolis-Hastings (MH)** rule [Hastings 1970; Metropolis et al. 1953], Gibbs sampler (e.g., [Casella and George 1992]), and **delayed rejection (DR)** [Mira et al. 2001]. MUQ provides implementations of these frameworks, as well as the **generalized Metropolis-Hastings (gMH)** kernel [Calderhead 2014] and multilevel MCMC framework of [Dodwell et al. 2019]. Most of these frameworks start by drawing samples from one or more proposal distributions $q_1(\cdot|\boldsymbol{m}_i)$, $\ldots$, $q_K(\cdot|\boldsymbol{m}_i)$ that are easy to sample from (e.g., Gaussian) and then "correct" the

proposed samples to obtain exact, but correlated, posterior samples. In the MH and DR kernels, corrections take the form of accepting or rejecting the proposed point. In the gMH kernel, the correction involves analytically sampling a finite state Markov chain over multiple proposed points. Intuitively, proposal distributions that capture the shape of the posterior, either locally around m_i or globally over the parameter space, tend to require fewer "corrections" and yield more efficient algorithms.

Proposal Distributions. Let $q(\cdot|\boldsymbol{m}_i)$ denote a proposal distribution that is "parameterized" by the current state of the chain $\boldsymbol{m}_i$. We require that the proposal distribution is easily sampled and that its density can be efficiently evaluated. The MH rule [Hastings 1970; Metropolis et al. 1953] defines a transition kernel $K_{MH}(\cdot|\boldsymbol{m}_i)$ through a two step process: first draw a random sample $\boldsymbol{m}' \sim q(\cdot|\boldsymbol{m}_i)$ from the proposal distribution, and then accept the proposed sample $\boldsymbol{m}'$ as the next step in the chain $\boldsymbol{m}_{i+1}$ with a probability $\alpha = \min\{1, \gamma\}$ where $\gamma = \frac{\pi_{\text{post}}(\boldsymbol{m}')}{\pi_{\text{post}}(\boldsymbol{m}_i)} \frac{q(\boldsymbol{m}_i|\boldsymbol{m}')}{q(\boldsymbol{m}'|\boldsymbol{m}_i)}$. If rejected, set $\boldsymbol{m}_{i+1} = \boldsymbol{m}_i$. Under mild technical conditions on the proposal distribution (see e.g., Roberts et al. [2004]), the MH rule defines a Markov chain that is ergodic and has μ_{post} as a stationary distribution, thus enabling states in the chain to be used in Monte Carlo estimators. Note that the detailed balance condition (see e.g., Owen [2013]) is commonly employed to verify that a Markov chain has μ_{post} as a stationary distribution, but this condition alone is not sufficient to guarantee that the chain will converge to the stationary distribution. See Roberts et al. [2004] for a detailed discussion of MH convergence and convergence rates.

While the MH rule will yield a valid MCMC kernel for a large class of proposal distributions, the dependence of the proposal on the previous state, combined with possible rejection of the proposed state, results in inter-sample correlations in the Markov chain. Because of these correlations, the error of the Monte Carlo approximation in (12) will be larger when using MCMC than in the classic Monte Carlo setting with independent samples. Markov chains with large correlations will result in larger estimator variance. To reduce correlation in the Markov chain, we seek proposal distributions that can take large steps with a high probability of acceptance. From the acceptance probability in the MH rule we see that this can occur when the proposal density $q(\boldsymbol{m}|\boldsymbol{m}_i)$ is a good approximation to $\pi_{\text{post}}(\boldsymbol{m})$, so that γ is close to 1.

We now turn to describing specific proposal distributions used in hIPPYlib-MUQ. First, we begin by describing common proposal mechanisms that exploit gradient and curvature information to accelerate sampling in finite-dimensional spaces. These algorithms comprise the left face of the cube in Figure 1. We then show how these ideas can be extended to construct proposals with performance that is independent of mesh-refinement (i.e., independent of dimension), thus "lifting" the derivative-accelerated proposals to an infinite-dimensional setting. This "lifting" operation transforms proposals on the left face in Figure 1 to their dimension-independent analogs on the right face of the proposal cube.

Exploiting Gradient and Curvature Information. Perhaps the simplest and most common, but not generally efficient, proposal distribution takes the form of a Gaussian distribution centered at the current state in the chain,

$$q_{\text{RW}}(\boldsymbol{m}|\boldsymbol{m}_i) = \mathcal{N}\left(\boldsymbol{m}_i, \boldsymbol{\Gamma}_{\text{prop}}\right), \tag{13}$$

where $\boldsymbol{\Gamma}_{\text{prop}} \in \mathbb{R}^{n\times n}$ is a user defined covariance matrix. When used with the MH rule, this ***random walk*** **(RW)** proposal yields an MCMC algorithm that is commonly called the random walk Metropolis algorithm. The **adaptive Metropolis (AM)** algorithm employs a variant of this proposal where the covariance $\boldsymbol{\Gamma}_{\text{prop}}$ is adapted based on previous samples [Haario et al. 2001]. A proposal covariance $\boldsymbol{\Gamma}_{\text{prop}}$ that matches posterior covariance increases efficiency, but the random walk proposal is still a poor approximation of the posterior density.

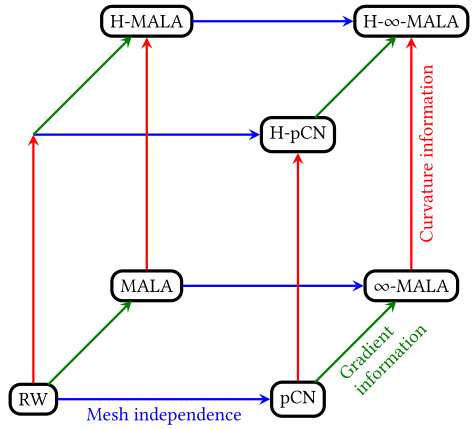


Fig. 1. The relationship of various MCMC proposal distributions with respect to mesh-refinement independence (blue arrow), gradient awareness (green arrow), and curvature awareness (red arrow). The abbreviations stand for the following MCMC proposals: RW for random walk, pCN for preconditioned Crank-Nicolson, MALA for Metropolis-adjusted Langevin algorithm, H-pCN for curvature-informed pCN, H-MALA for curvature-informed MALA, ∞-MALA for infinite-dimensional MALA, and H-∞-MALA for curvature-informed infinite-dimensional MALA.

A slightly more efficient proposal can be obtained through a one-step Euler-Maruyama discretization of the Langevin stochastic differential equations [Roberts and Stramer 2003]. The resulting Langevin proposal takes the form

$$q_{\text{MALA}}(\boldsymbol{m}|\boldsymbol{m}_i) = \mathcal{N}\left(\boldsymbol{m}_i + \tau\boldsymbol{\Gamma}_{\text{prop}}\nabla \log \pi_{\text{post}}(\boldsymbol{m}_i),\ 2\tau\boldsymbol{\Gamma}_{\text{prop}}\right), \tag{14}$$

where τ is the step size parameter. MH samplers with this proposal are called ***Metropolis-adjusted Langevin algorithms* (MALA)**. Like the AM algorithm, adapting the covariance of the MALA proposal can also improve performance [Atchadé 2006; Marshall and Roberts 2012]. It is also common to use an approximation of the posterior covariance, such as the inverse of the log-posterior Hessian, to help the MALA proposal capture the posterior correlation. In this work for example, we employ a low rank-based approximation of the log-posterior Hessian at the MAP point (c.f. Equation (11))

$$q_{\text{H-MALA}}(\boldsymbol{m}|\boldsymbol{m}_i) = \mathcal{N}\left(\boldsymbol{m}_i + \tau\mathbf{H}^{-1}\nabla \log \pi_{\text{post}}(\boldsymbol{m}_i), 2\tau\mathbf{H}^{-1}\right). \tag{15}$$

This metric is similar to the one used by Martin et al. [2012] and is equivalent to the preconditioned MALA proposal in (14) using the covariance of the Laplace approximation in (9).

Both (13) and (14) use a covariance that is constant across the parameter space. Allowing this covariance to adapt to the local correlation structure of the posterior density enables higher order approximations to be obtained, resulting in more efficient MCMC algorithms. In Girolami and Calderhead [2011], a differential geometric viewpoint was employed to define a family of proposal mechanisms on a Riemannian manifold. Adapting the MALA proposal in (14) to this manifold setting and ignoring the manifold's curvature, results in

$$q_{\text{sMMALA}}(\boldsymbol{m}|\boldsymbol{m}_i) = \mathcal{N}\left(\boldsymbol{m}_i + \tau\mathbf{G}^{-1}(\boldsymbol{m}_i)\nabla \log \pi_{\text{post}}(\boldsymbol{m}_i), 2\tau\mathbf{G}^{-1}(\boldsymbol{m}_i)\right), \tag{16}$$

where $\mathbf{G}(\boldsymbol{m})$ is a position-dependent metric tensor. This is known as the **simplified Manifold MALA (sMMALA)** proposal. Girolami and Calderhead [2011] defined the metric tensor $\mathbf{G}(\boldsymbol{m})$

using the expected Fisher information metric, which provides a positive definite approximation of the posterior covariance at the point $\boldsymbol{m}$.

Hamiltonian Monte Carlo techniques, including the **No-U-Turn Sampler (NUTS)** [Hoffman and Gelman 2014], are another important class of MCMC proposals. These techniques approximately solve a Hamiltonian system to take large jumps in parameter space. While efficient in many scenarios (see e.g., Neal [2010]), especially with purely statistical models, we have found that solving the Hamiltonian system involves an intractable number of posterior gradient evaluations on our PDE-based problems of interest. The transport map MCMC algorithms of Parno and Marzouk [2018] are also not considered here because of the challenge of building high-dimensional transformations.

Dimension-Independent Proposal Distributions. For finite-dimensional parameters, the random walk and MALA proposals defined above can be used with the MH rule for MCMC. However, their performance is not discretization invariant. As the discretization of the function m is refined, the performance of the samplers on the finite-dimensional posterior $\pi_{\text{post}}(\boldsymbol{m})$ will worsen. As the dimension increases, the difference between the largest two eigenvalues of the MCMC transition kernel (i.e., the spectral gap), goes to zero and the mixing times of the chains grows indefinitely; see Cotter et al. [2012]; Hairer et al. [2014] for details. Some modifications to the proposals are necessary to obtain "dimension-independent" performance. The works of Cotter et al. [2012], Beskos et al. [2017], and Bardsley et al. [2020], for example, modify existing finite-dimensional proposals to ensure the algorithm performance is independent of mesh refinement.

The dimension-independent analog of the RW proposal is the **preconditioned Crank-Nicolson (pCN)** proposal introduced in Cotter et al. [2012]. It takes the form

$$q_{\text{pCN}}(\boldsymbol{m}|\boldsymbol{m}_i) = \mathcal{N}\left(\boldsymbol{m}_{\text{pr}} + \sqrt{1-\beta^2}(\boldsymbol{m}_i - \boldsymbol{m}_{\text{pr}}),\ \beta^2\boldsymbol{\Gamma}_{\text{prior}}\right). \tag{17}$$

Notice that when $\beta = 1$, the pCN proposal is equal to the prior distribution. The MALA proposal was also adapted in Cotter et al. [2012] to obtain the infinite-dimensional MALA (∞-MALA) proposal

$$q^{\infty}_{\text{MALA}}(\boldsymbol{m}|\boldsymbol{m}_i) = \mathcal{N}\left(\sqrt{1-\beta^2}\boldsymbol{m}_i + \beta\frac{\sqrt{h}}{2}\left(\boldsymbol{m}_{\text{pr}} - \boldsymbol{\Gamma}_{\text{prior}}\nabla\Phi(\boldsymbol{m}_i)\right),\ \beta^2\boldsymbol{\Gamma}_{\text{prior}}\right), \tag{18}$$

where $\beta = 4\sqrt{h}/(4+h)$ and h is a parameter that can be tuned. While the pCN and ∞-MALA proposals result in discretization-invariant Metropolis-Hastings algorithms, they suffer from the same deficiencies as their finite-dimensional RW and MALA analogs, i.e., they do not capture the posterior geometry.

Several efforts have worked to minimize this deficiency, see for example Beskos et al. [2017]; Petra et al. [2014]; Pinski et al. [2015]; Rudolph and Sprungk [2018]. We consider a generalization of the pCN proposal described in Pinski et al. [2015]. It incorporates the MAP point and the posterior curvature information at that point into the pCN proposal, which is denoted by H-pCN and takes the form

$$q_{\text{H-pCN}}(\boldsymbol{m}|\boldsymbol{m}_i) = \mathcal{N}\left(\boldsymbol{m}_{\text{MAP}} + \sqrt{1-\beta^2}(\boldsymbol{m}_i - \boldsymbol{m}_{\text{MAP}}),\ \beta^2\mathbf{H}^{-1}\right). \tag{19}$$

Another method that can exploit the posterior geometry is an extension of the ∞-MALA proposal discussed in Beskos et al. [2017]:

$$q^{\infty}_{\text{sMMALA}}(\boldsymbol{m}|\boldsymbol{m}_i) = \mathcal{N}\left(\mu'(\boldsymbol{m}_i),\ \Gamma'(\boldsymbol{m}_i)\right), \tag{20}$$

where

$$\mu'(\boldsymbol{m}_i) = \sqrt{1-\beta^2}\boldsymbol{m}_i + \beta\frac{\sqrt{h}}{2}\left(\boldsymbol{m}_i - \mathbf{G}^{-1}\boldsymbol{\Gamma}_{\text{prior}}^{-1}(\boldsymbol{m}_i - \boldsymbol{m}_{\text{pr}}) - \mathbf{G}^{-1}\nabla\Phi(\boldsymbol{m}_i)\right) \tag{21}$$

$$\Gamma'(\boldsymbol{m}_i) = \beta^2\mathbf{G}^{-1}(\boldsymbol{m}_i). \tag{22}$$

This ∞-sMMALA proposal simplifies to ∞-MALA when $\mathbf{G}^{-1}(\boldsymbol{m}_i) = \boldsymbol{\Gamma}_{\text{prior}}$. When $\mathbf{G}(\boldsymbol{m}_i)$ is the Laplace approximation Hessian from (9), the ∞-sMMALA proposal simplifies to

$$q^{\infty}_{\text{H-MALA}}(\boldsymbol{m}|\boldsymbol{m}_i) = \mathcal{N}\left(\sqrt{1-\beta^2}\boldsymbol{m}_i + \beta\frac{\sqrt{h}}{2}\left(\boldsymbol{m}_i - \mathbf{H}^{-1}\boldsymbol{\Gamma}_{\text{prior}}^{-1}(\boldsymbol{m}_i - \boldsymbol{m}_{\text{pr}}) - \mathbf{H}^{-1}\nabla\Phi(\boldsymbol{m}_i)\right), \beta^2\mathbf{H}^{-1}\right), \tag{23}$$

which we denote by H-∞-MALA.

Alternative Transition Kernels. The proposal distributions above are classically considered in the context of a Metropolis-Hasting kernel. However, there are alternative transition kernels that also result in ergodic Markov chains. Here we consider transition kernels constructed from the delayed rejection approach of Mira et al. [2001] as well as Metropolis-within-Gibbs kernels, which repeatedly use the Metropolis-Hastings rule on different conditional slices of the posterior distribution to construct the Markov chain. In particular, we consider the family of **dimension-independent likelihood-informed (DILI)** approaches [Cui et al. 2016a; Cui and Zahm 2021], which define a Metropolis-within-Gibbs sampler that inherits dimension-independent properties from an appropriate dimension-independent proposal. By dimension-independence here we mean that the acceptance rate and mixing properties will not deteriorate when the dimension of the problem increases.

The delayed rejection kernel allows multiple proposals to be attempted in each step of the Markov chain. This can be advantageous when using multiple proposals with complementary properties. For example, it is possible to start with a proposal that attempts to make large ambitious jumps across the parameter space but may have low acceptance probability while falling back on a more conservative proposal that takes smaller steps with a larger probability of acceptance. Similarly, it is possible to start with a proposal that is more computationally efficient (e.g., does not require gradient information) but less likely to be accepted, while employing a more expensive proposal mechanism in a second stage to ensure the chain explores the space. In either case, if the first proposed move is rejected by the Metropolis-Hastings rule, another more expensive proposal that is more likely to be accepted can be tried with an adjusted acceptance probability. More than two stages can also be employed.

DILI divides the parameter space into a finite-dimensional subspace, which can be explored with standard proposal mechanisms, and a complementary infinite-dimensional space that can be explored with a dimension-independent approach, such as those described above. The resulting transition kernel is more complicated than the Metropolis-Hastings rule, but inherits the dimension-independent properties of the complementary space proposal. The likelihood-informed subspace is computed using the generalized eigenvalue problem (10). If the eigenvalue is larger than one, it indicates that the likelihood function dominates the prior density in that direction. The same low rank structure used to approximate the posterior Hessian can therefore be used to decompose the parameter space into a **likelihood-informed subspace (LIS)** spanned by the columns of $\mathbf{V}_r$ and an orthogonal **complementary space (CS)**. Within each subspace, a standard Metropolis-Hastings kernel is employed. As long as the kernel in the CS uses a dimension-independent proposal (typically pCN), then the DILI sampler will remain dimension-independent. Unlike the original implementation described in Cui et al. [2016a], the MUQ implementation does not use a whitening transform and thus avoids computing any symmetric decomposition

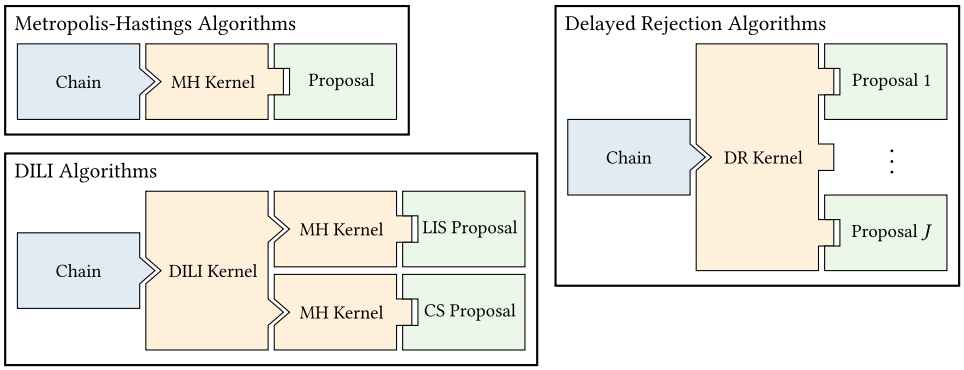


Fig. 2. The flexible framework of hIPPYlib-MUQ allows many different combinations of transition kernels and proposal distributions to be employed. Note that each kernel can interact with any proposal distribution, which enables many different MCMC algorithms to be constructed from the same basic components.

of the prior covariance. In general, the Hessian used in (10) can be adapted to capture more correlation structure. However, we did not find this necessary in the numerical experiments below.

Assembling an MCMC Algorithm. It is possible to combine nearly any of the proposals and kernels described above, resulting in myriad possible MCMC algorithms. As suggested in Figure 2, there are three fundamental building blocks to an MCMC algorithm. The chain keeps track of previous points and allows computing Monte Carlo estimates. The kernel defines a mechanism for sampling the next state $\boldsymbol{m}_{i+1}$ given the value of the current state $\boldsymbol{m}_i$ and one or more proposal distributions. The proposal defines a position specific probability distribution that can be easily sampled and has a density that can be efficiently evaluated. We mimic these abstract interfaces in our software design to define and test a large number of kernel-proposal combinations.

3.2 MCMC Diagnostics

Two questions naturally arise when analyzing a length I Markov chain $[\mathbf{m}_1, \ldots, \mathbf{m}_I]$ produced by MCMC. First, has the chain converged to the stationary distribution? Second, what is the statistical efficiency of the chain, that is, how many independent samples does the chain have that actually contribute to the accuracy of Monte Carlo estimators? Most theoretical guarantees are asymptotic, and it is important to quantitatively answer these questions when employing finite-length MCMC chains. Based on these considerations, this section describes the diagnostics implemented in hIPPYlib-MUQ to check the convergence and statistical efficiency of high-dimensional MCMC chains.

3.2.1 Assessing Convergence. To assess convergence, we compute two different asymptotically unbiased estimators of the posterior covariance: one that is an overestimate for finite I and one that is an underestimate for finite I. As the ratio of these two estimates approaches one, we can be confident that the MCMC chain has converged (see e.g., Brooks and Gelman [1998]; Gelman et al. [2004]; Vehtari et al. [2020]).

The estimates are based on running J independent chains starting from randomly chosen points that are more disperse than the posterior distribution μ_{post}, where we define a "disperse" distribution as one that has a larger covariance than μ_{post}. Each chain has the same length I.

Letting $\mathbf{m}_{ij}$ be the ith MCMC sample in chain j, we define the within-sequence covariance matrix $\mathbf{W}$ and the between-sequence covariance matrix $\mathbf{B}$ as

$$\mathbf{W} = \frac{1}{J(I-1)} \sum_{j=1}^{J} \sum_{i=1}^{I} (\mathbf{m}_{ij} - \bar{\mathbf{m}}_{\cdot j})(\mathbf{m}_{ij} - \bar{\mathbf{m}}_{\cdot j})^T; \quad \bar{\mathbf{m}}_{\cdot j} = \frac{1}{I} \sum_{i=1}^{I} \mathbf{m}_{ij}, \tag{24}$$

$$\mathbf{B} = \frac{I}{J-1} \sum_{j=1}^{J} (\bar{\mathbf{m}}_{\cdot j} - \bar{\mathbf{m}}_{\cdot\cdot})(\bar{\mathbf{m}}_{\cdot j} - \bar{\mathbf{m}}_{\cdot\cdot})^T; \qquad \bar{\mathbf{m}}_{\cdot\cdot} = \frac{1}{J} \sum_{j=1}^{J} \bar{\mathbf{m}}_{\cdot j}. \tag{25}$$

As pointed out in Brooks and Gelman [1998], $\mathbf{W}$ and $\mathbf{B}$ can be combined to produce an estimate $\widehat{\mathbf{V}}$ of the posterior covariance that takes the form

$$\widehat{\mathbf{V}} = \frac{I-1}{I}\mathbf{W} + \frac{J+1}{JI}\mathbf{B}. \tag{26}$$

The overdispersion of the initial points in each chain causes $\widehat{\mathbf{V}}$ to overestimate the posterior covariance for finite I. On the other hand, the average within-chain covariance $\mathbf{W}$ will tend to underestimate the covariance because the chains have not explored the entire parameter space. Comparing $\mathbf{W}$ and $\widehat{\mathbf{V}}$ thus provides a way of assessing convergence.

The $\hat{R}$ statistic of Gelman et al. [2004] and Vehtari et al. [2020] is a common way of comparing $\mathbf{W}$ and $\widehat{\mathbf{V}}$. It uses the ratio of the diagonal component of $\widehat{\mathbf{V}}$ and $\mathbf{W}$ to construct a componentwise convergence diagnostic. For high dimensional problems, it is more natural to consider a multivariate convergence diagnostic. We will therefore employ the **multivariate potential scale reduction factor (MPSRF)** of Brooks and Gelman [1998], which is a natural extension of the componentwise $\hat{R}$ statistic. The MPSRF is defined by

$$\text{MPSRF} = \sqrt{\max_a \frac{a^T \widehat{\mathbf{V}} a}{a^T \mathbf{W} a}} = \sqrt{\frac{I-1}{I} + \frac{J+1}{JI}\lambda_{\max}}, \tag{27}$$

where $\lambda_{\max}$ is the largest eigenvalue satisfying the generalized eigenvalue problem $\mathbf{B}\boldsymbol{v} = \lambda \mathbf{W}\boldsymbol{v}$.

Note that MPSRF ≥ 1 when $\lambda_{\max} > 1$, which occurs when the chains have overdispersed starting points that cause the inter-chain variance $\mathbf{B}$ to be larger than the within-chain variance $\mathbf{W}$. When the MPSRF approaches 1, the variance within each sequence approaches the variance across sequences, thus indicating that each individual chain has converged to the target distribution. The literature contains several recommendations for values of MPSRF that indicate convergence; for example, Gelman and Rubin [1992] suggest the commonly used value MPSRF < 1.1 while Vehtari et al. [2020] argues for the more conservative threshold MPSRF < 1.01.

3.2.2 Statistical Efficiency. The samples in an MCMC chain are generally correlated, which increases the variance of Monte Carlo estimators constructed with MCMC samples. For a quantity of interest $\mathcal{G}(\boldsymbol{m})$, the **effective sample size (ESS)** of a Markov chain is defined as the number of independent samples of the posterior that would be needed to estimate $\mathbb{E}[\mathcal{G}]$ with the same statistical accuracy as an estimate from the Markov chain. The ESS is therefore a measure of how much information is contained in the MCMC chain. In this work, it is commonly assumed that the ESS is derived for estimators of the posterior mean, i.e., $\mathbb{E}[\mathcal{G}] = \mathbb{E}[\boldsymbol{m}]$, and here we derive the ESS under this common assumption. Other ESS variants, like those described by Vehtari et al. [2020], are more suitable for problems involving tail probabilities, but the implementation of these methods in hIPPYlib-MUQ is left to future work.

There are several ways of estimating the ESS. For instance, spectral approaches use the integrated autocorrelation of the MCMC chain to estimate the effective sample size (see e.g., Gelman et al. [2004]; Wolff et al. [2004]). Other common methods use the statistics of small sample batches

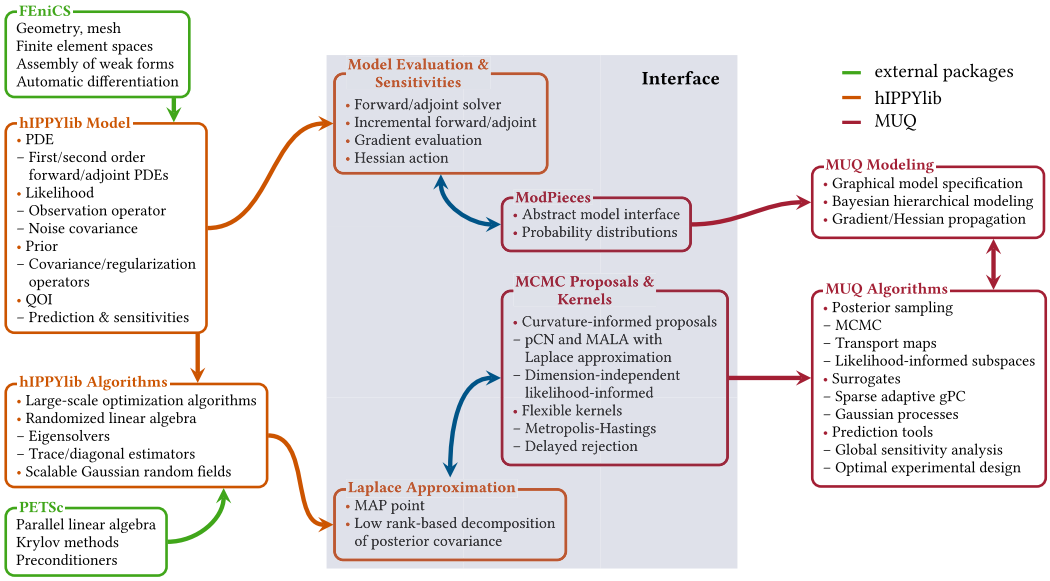


Fig. 3. Description of the functionalities of hIPPYlib and MUQ and their interface. Orange and red boxes represent hIPPYlib and MUQ functionalities, respectively. Green boxes indicate external software libraries, FEniCS and PETSc, that provide parallel implementation of finite element discretizations and solvers. Arrows represent one-way or reciprocal interactions.

(see e.g., Flegal and Jones [2010]; Vats et al. [2019]). MUQ provides implementations of both spectral and batch methods. Here we focus on the spectral formulation of ESS however, because it gives additional insight into the structure of MCMC chains. The ESS for component k of $\boldsymbol{m}$ is defined by

$$\mathrm{ESS}_k = \frac{JI}{1 + 2\sum_{t=1}^{\infty} \rho_{kt}}, \tag{28}$$

where ρ_{kt} is the autocorrelation function of component k in the MCMC chain at lag t. Here, the autocorrelation function ρ_{kt} is estimated by the following formula [Gelman et al. 2004]:

$$\rho_{kt} \approx \hat{\rho}_{kt} = 1 - \frac{v_{kt}}{2\hat{V}_{kk}}, \tag{29}$$

where $\hat{V}_{kk}$ is the kth diagonal component of the posterior covariance estimate defined in (26) and v_{kt} is the variogram defined by

$$v_{kt} = \frac{1}{J(I-t)} \sum_{j=1}^{J} \sum_{i=t+1}^{I} (m_{ij,k} - m_{(i-t)j,k})^2. \tag{30}$$

In practice, $\hat{\rho}_{kt}$ is noisy for large values of t and we truncate the summation (28) at some lag t'. Following common practice, we choose $t' \geq 0$ to be the lag for which the sum successive autocorrelation estimates $\hat{\rho}_{2t'} + \hat{\rho}_{2t'+1}$ is negative [Gelman et al. 2004].

4 SOFTWARE FRAMEWORK

hIPPYlib-MUQ is a Python interface that integrates two open source software libraries, hIPPYlib and MUQ, into a unique software framework, allowing the user to implement state-of-the-art Bayesian inversion algorithms in a seamless way. We outline in Figure 3 the main functionalities of hIPPYlib and MUQ and the integration of their complementary components.

hIPPYlib, built on FEniCS and PETSc for the discretization and solution of PDEs, provides Python implementations of scalable adjoint-based algorithms for solving large-scale deterministic and linearized Bayesian inverse problems governed by PDEs. hIPPYlib model component provides a collection of libraries by which users can describe the forward PDE, the prior model, and the likelihood model in the FEniCS form language [Logg et al. 2012]. hIPPYlib algorithms component incorporates optimization algorithms, randomized linear algebra, and scalable sampling of Gaussian fields that are required to efficiently solve the deterministic and linearized Bayesian inverse problems. The hiPPYlib user manual can be found at Villa et al. [2020], which includes details on the software installation, documentation, and tutorials.

MUQ is an easy-to-use software framework for defining and solving uncertainty quantification problems. MUQ modeling tools allow users to easily and flexibly construct complicated models, including Bayesian hierarchical models, in a semi-intrusive way that enables efficient gradient and Hessian evaluations. MUQ also implements a variety of advanced uncertainty quantification techniques including MCMC sampling methods, surrogate models (e.g., Gaussian processes), and prediction tools (e.g., global sensitivity analysis). We refer the reader to muq.mit.edu for a detailed description of the software with installation instructions and step-by-step tutorials.

We note that a significant synergistic effect can be obtained by making use of complementary aspects of these two software libraries: hIPPYlib outputs such as the gradient and Hessian evaluations and the Laplace approximation, and MUQ's advanced MCMC sampling modules and flexible modeling capabilities. In the integrated framework, hIPPYlib is used to define the forward model, the prior, and the likelihood, to compute the **maximum a posteriori (MAP)** point, and to construct a Laplace approximation of the posterior distribution based on approximations of the posterior covariance as a low-rank update of the prior [Bui-Thanh et al. 2013]. MUQ is employed to exploit advanced MCMC methods to fully characterize the posterior distribution in non-Gaussian/nonlinear settings. hIPPYlib-MUQ offers a set of wrappers that encapsulate the functionality of hIPPYlib in a way that various features of hIPPYlib can be accessed by MUQ. A key aspect of hIPPYlib-MUQ is that it enables the use of curvature-informed MCMC methods, which is crucial for efficient and scalable exploration of the posterior distribution for large-scale Bayesian inverse problems.

In the context of hIPPYlib-MUQ, hIPPYlib provides tools for (i) automatically implementing adjoint gradients and Hessian actions, (ii) efficiently sampling **Gaussian Markov Random fields (GMRF)**, and (iii) constructing Laplace approximations with low-rank Hessians. On the other hand, MUQ provides two important capabilities: (i) an abstract graphical modeling framework that provides an interface for implementing model components (e.g., prior distributions or forward models) and enables multiple components of a model or inverse problem to be easily composed and differentiated, and (ii) a suite of MCMC algorithms, including curvature-informed and discretization-invariant methods. The adjoint techniques of hIPPYlib enable MUQ to efficiently compute gradients and Hessian actions of a graphical model with PDE-based components. The efficient GMRF sampling and low-rank Laplace approximations accelerate MUQ's discretization-invariant MCMC techniques, which use these hIPPYlib tools within the MCMC proposal. The details of our object orientated approach for seamlessly blending these MUQ and hIPPYlib tools are provided below.

Figure 4 gives an overview of the Python classes implemented by the hIPPYlib-MUQ interface. Inherited from MUQ classes, the interface classes wrap the hIPPYlib functionalities needed to achieve curvature-informed MCMC sampling methods. These include:

(1) Prior Gaussian interface (`LaplaceGaussian` and `BiLaplaceGaussian`) to enable the use of hIPPYlib prior models (`LaplacianPrior` and `BiLaplacianPrior`) in MUQ probability distribution models (`GaussianBase`). The interface allows MUQ to use Gaussian prior with covariance as inverse of the v-th power of a Laplacian-like operator ($v = 1$ for `LaplaceGaussian`

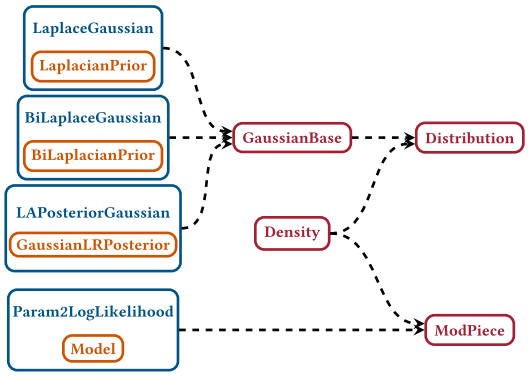


Fig. 4. Class hierarchy for hIPPYlib-MUQ framework. Classes of hIPPYlib, MUQ, and the interface are colored in orange, red, and blue, respectively. Dashed arrows represent inheritance relationship between two classes: the arrowhead attaches to the super-class and the other attaches to the sub-class.

and $v = 2$ for `BiLaplacianPrior`) and scalable sampling techniques for Gaussian random fields.

(2) Likelihood interface (`Param2LogLikelihood`) to incorporate hIPPYlib likelihood functions (`Model`) into MUQ Bayesian models (`ModPiece`). The interface enables the MUQ model to efficiently perform the model evaluation (the parameter-to-observable map) including forward PDE solves and the adjoint-based computation of gradient and Hessian actions.

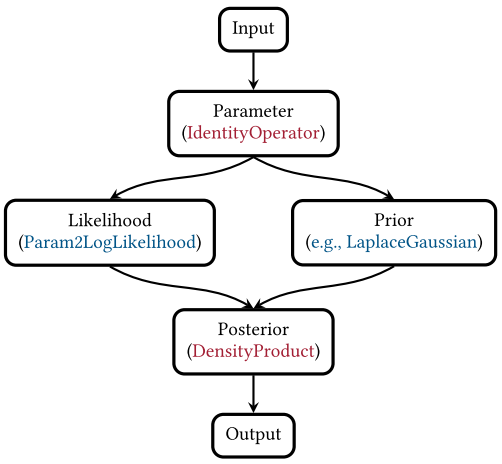
(3) Laplace approximation interface (`LAPosteriorGaussian`) to let MUQ MCMC modules get access to the Laplace approximation of the posterior distribution generated by hIPPYlib (`GaussianLRPosterior`). This interface provides the MAP point and/or the low-rank approximation of the Hessian at the MAP point for MUQ MCMC proposals, which leads to efficient and scalable sampling of the posterior.

These interface classes can then be used to form a Bayesian posterior model governed by PDEs using MUQ's graphical modeling interface (`WorkGraph`) as shown in Figure 5, as well as to construct MCMC proposals.

hIPPYlib-MUQ also implements the MCMC convergence diagnostics described in Section 3.2. These include the potential scale reduction factor and its extension to multivariate parameter cases [Brooks and Gelman 1998], the autocorrelation function, and the effective sample size. A detailed description of all classes and functionalities of hIPPYlib-MUQ can also be found at https://hippylib2muq.readthedocs.io/en/latest/modules.html.

5 NUMERICAL ILLUSTRATION

The objective of this section is to showcase applications of the integrated software framework discussed in previous sections via a step-by-step implementation procedure. We focus on comparing the performance of several MCMC methods available in the software framework. For the illustration we first revisit the model problem considered in Villa et al. [2021], an inverse problem of reconstructing the log-diffusion coefficient field in a two-dimensional elliptic partial differential equation. We then consider a nonlinear p-Poisson problem in three dimensions for which the parameter field in a Robin boundary condition is inferred. In this section, we summarize the Bayesian formulation of the example problems and present numerical results obtained using the proposed software framework, hIPPYlib-MUQ version 0.2.0 that builds on hIPPYlib version 3.1.0




```
# Example code snippet
import muq.Modeling as mm
import hippylib2muq as hm

#  ...  Use hIPPYlib  to  define  prior  and model variables

# Convert hiPPYlib components to MUQ components
prior_density = hm.BiLaplaceGaussian(prior).AsDensity()
likelihood = hm.Param2LogLikelihood(model)

# Add all of the components to the graph
graph = mm.WorkGraph()
graph.AddNode(mm.IdentityOperator(dim), 'Parameter')
graph.AddNode(prior_density, 'Prior')
graph.AddNode(likelihood, 'Likelihood')
graph.AddNode(mm.DensityProduct(2), 'Posterior')

# Define right branch: Parameter->Prior->Posterior
graph.AddEdge('Parameter', 0, 'Prior', 0)
graph.AddEdge('Prior', 0, 'Posterior', 0)

# Define left branch: Parameter->Likelihood->Posterior
graph.AddEdge('Parameter', 0, 'Likelihood', 0)
graph.AddEdge('Likelihood', 0, 'Posterior', 1)
```

Fig. 5. Graphical description of Bayesian posterior modeling using hIPPYlib-MUQ software framework (left) and an example code snippet (right). In the left figure, class names of MUQ and the interface are colored in red and blue, respectively. MUQ's `WorkGraph` class provides a way to combine all the Bayesian posterior model components by its member functions `AddNode` and `AddEdge`. MUQ's `IdentityOperator` class identifies input parameters and the input argument `dim` represents the parameter dimension. MUQ's `DensityProduct` class defines the product of prior and likelihood densities and the input argument `2` means the number of input densities. The second and fourth arguments of the function `AddEdge` mean the output index of the first argument and the input index of the third argument, respectively. For example, `graph.AddEdge('Parameter', 0, 'Prior', 0)` means that the output of 'Parameter' indexed 0 is connected to the input of 'Prior' indexed 0.

with FEniCS version 2019 and MUQ version 0.3.0; see https://hippylib2muq.readthedocs.io/en/latest/installation.html for the software installation instruction and its dependencies. The accompanying Jupyter notebook provides a detailed description of the hIPPYlib-MUQ implementations; see https://hippylib2muq.readthedocs.io/en/latest/tutorial.html. Additional examples, including some with hierarchical models, can also be found at muq.mit.edu.

5.1 Coefficient Field Inversion in a Two-dimensional Poisson Linear PDE

We first consider the coefficient field inversion in a Poisson partial differential equation given pointwise noisy state measurements. We begin by describing the forward model setup and quantity of interest, followed by the definition of the prior and the likelihood distributions. Next, we present the Laplace approximation of the posterior and apply several MCMC methods to characterize the posterior distribution, as well as the predictive posterior distribution of a scalar quantity of interest. Finally, the scalability of the proposed methods with respect to the parameter dimension is then assessed in a mesh refinement study.

5.1.1 Forward Model. Let $\Omega \in \mathbb{R}^d (d = 2, 3)$ be an open bounded domain with boundary $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N, \partial\Omega_D \cap \partial\Omega_N = \emptyset$. Given a realization of the uncertain parameter field m, the state

variable u is governed by

$$\begin{aligned} -\nabla \cdot (e^m \nabla u) &= f \quad \text{in } \Omega, \\ u &= g \quad \text{on } \partial\Omega_D, \\ e^m \nabla u \cdot \mathbf{n} &= h \quad \text{on } \partial\Omega_N, \end{aligned} \tag{31}$$

where f is a volume source term, g and h are the prescribed Dirichlet and Neumann boundary data, respectively, and $\mathbf{n}$ is the outward unit normal vector.

The weak form of (31) reads as follows: Find $u \in \mathcal{V}_g$ such that

$$\langle e^m \nabla u, \nabla p \rangle = \langle f, p \rangle + \langle h, p \rangle_{\partial\Omega_N} \quad \forall p \in \mathcal{V}_0, \tag{32}$$

where

$$\begin{aligned} \mathcal{V}_g &= \left\{ v \in H^1(\Omega) | v = g \text{ on } \partial\Omega_D \right\}, \\ \mathcal{V}_0 &= \left\{ v \in H^1(\Omega) | v = 0 \text{ on } \partial\Omega_D \right\}. \end{aligned} \tag{33}$$

Above, we denote the L^2-inner product over Ω by $\langle \cdot, \cdot \rangle$ and that over $\partial\Omega_N$ by $\langle \cdot, \cdot \rangle_{\partial\Omega_N}$.

As a quantity of interest, the logarithm of the normal flux through the bottom boundary $\partial\Omega_b \subset \partial\Omega_D$ is considered. Specifically, we define the quantity of interest $\mathcal{G}(m)$ as

$$\mathcal{G}(m) = \ln \left\{ - \int_{\partial\Omega_b} e^m \nabla u \cdot \mathbf{n} \, ds \right\}. \tag{34}$$

In this example, we consider a unit square domain in $\mathbb{R}^2$ with no source term ($f = 0$), no normal flux ($h = 0$) on the left and right boundaries, and the Dirichlet condition imposed on the top boundary ($g = 1$) and the bottom boundary ($g = 0$).

For the spatial discretization, we use quadratic finite elements for the state variable (also for the adjoint variable) and linear finite elements for the parameter variable. The computational domain is discretized using a regular mesh with 2,048 triangular elements. This leads to 4,225 and 1,089 degrees of freedom for the state and parameter variables, respectively. In the scalability results presented in Section 5.1.6, the mesh is then refined with up to four levels of uniform refinement leading to 263,169 and 66,049 degrees of freedom for the state and parameter variables, respectively, on the finest level.

5.1.2 Prior Model. As discussed in Section 2, we choose the prior to be a Gaussian distribution $\mathcal{N}(m_{\text{pr}}, C_{\text{prior}})$ with $C_{\text{prior}} = \mathcal{A}^{-2}$ where $\mathcal{A}$ is a Laplacian-like operator given as

$$\mathcal{A}m = \begin{cases} -\gamma \nabla \cdot (\Theta \nabla m) + \delta m & \text{in } \Omega, \\ \Theta \nabla m \cdot \mathbf{n} + \beta m & \text{on } \partial\Omega. \end{cases} \tag{35}$$

Here, $\beta \propto \sqrt{\gamma\delta}$ is the optimal Robin coefficient introduced to alleviate undesirable boundary effects [Daon and Stadler 2018], and an anisotropic tensor Θ is of the form

$$\Theta = \begin{bmatrix} \theta_1 \sin^2\alpha + \theta_2 \cos^2\alpha & (\theta_1 - \theta_2) \sin\alpha \cos\alpha \\ (\theta_1 - \theta_2) \sin\alpha \cos\alpha & \theta_1 \cos^2\alpha + \theta_2 \sin^2\alpha \end{bmatrix}. \tag{36}$$

For this example we take $\gamma = 0.1, \delta = 0.5, \beta = \sqrt{\gamma\delta}/1.42, \theta_1 = 2.0, \theta_2 = 0.5$ and $\alpha = \pi/4$. Figure 6 shows the prior mean m_{pr} and three samples from the prior distribution.

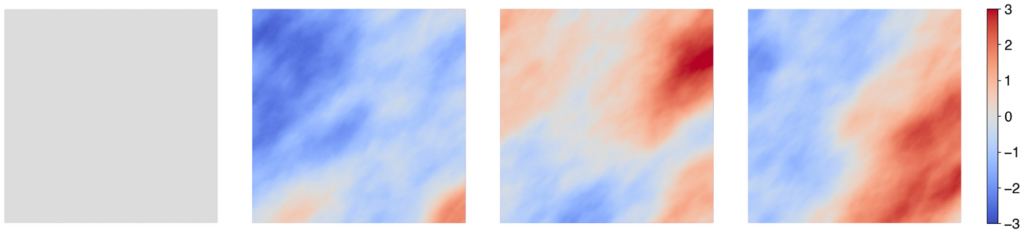


Fig. 6. Prior mean (leftmost) and three sample fields drawn from the prior distribution for the Poisson problem.

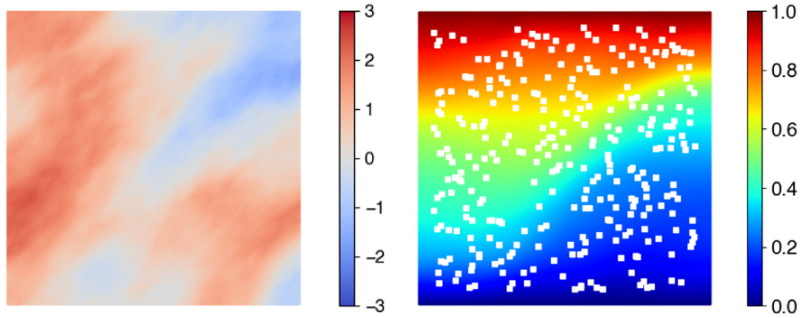


Fig. 7. True parameter field (left) and the corresponding state field (right) for the Poisson problem. The locations of the observation points are marked as white squares in the right figure.

5.1.3 Observations with Noise and the Likelihood. We generate state observations at l random locations uniformly distributed over $[0.05, 0.95]^2$ by solving the forward problem on the finest mesh with the true parameter field m_{true} (here a sample from the prior is used) and then adding a random Gaussian noise to the resulting state values; see Figure 7. The number of observations l is set to 300 for this example. The vector of synthetic observations is given by

$$\boldsymbol{d} = \mathcal{B}u + \boldsymbol{\eta}, \tag{37}$$

where $\mathcal{B}$ is a linear observation operator, restricting the state solution to the l observation points. The additive noise vector $\boldsymbol{\eta}$ has mutually independent components that are normally distributed with zero mean and standard deviation $\sigma = 0.005$. The likelihood function is then given by

$$\pi_{\text{like}}(\boldsymbol{d} \mid m) \propto \exp\left(-\frac{1}{2}\|\mathcal{B}\,u(m) - \boldsymbol{d}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}}\right), \tag{38}$$

where $\boldsymbol{\Gamma}_{\text{noise}} = \sigma^2\mathbf{I}$.

5.1.4 Laplace Approximation of the Posterior. We next construct the Laplace approximation of the posterior, a Gaussian distribution $\hat{\mu}_{\text{post}} \sim \mathcal{N}(m_{\text{MAP}}, \mathcal{H}(m_{\text{MAP}})^{-1})$ with mean equal to the MAP point and covariance given by the Hessian of the negative log-posterior evaluated at the MAP point. The MAP point is obtained by minimizing the negative log-posterior, i.e.,

$$\min_{m\in\mathcal{M}} \mathcal{J}(m) \;:=\; \frac{1}{2}\|\mathcal{B}\,u(m) - \boldsymbol{d}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}} + \frac{1}{2}\|m - m_{\text{pr}}\|^2_{\mathcal{C}^{-1}_{\text{prior}}}. \tag{39}$$

We employ the inexact Newton-CG algorithm implemented in `hIPPYlib` to solve the above optimization problem. We refer the reader to Villa et al. [2021] for a detailed description of the algorithm and the expressions for the gradient and Hessian actions of the negative log-posterior $\mathcal{J}(m)$.

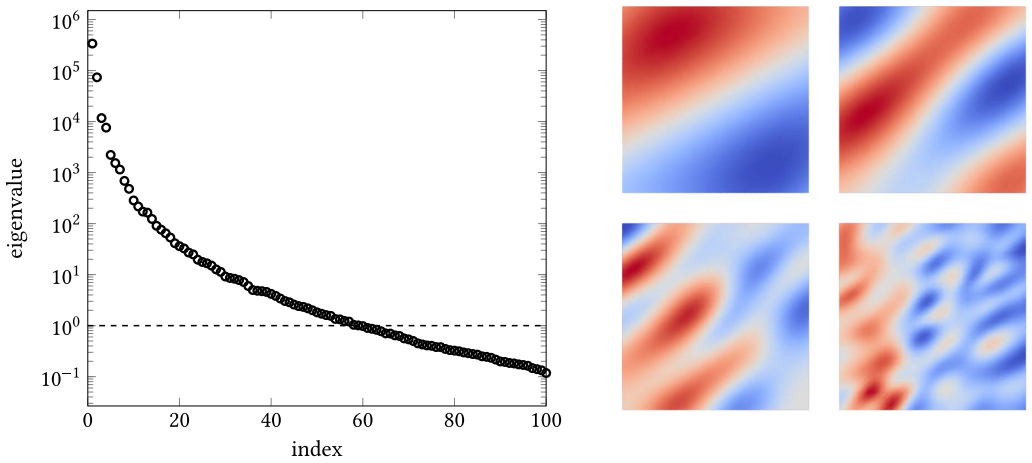


Fig. 8. Logarithmic plot of the $r = 100$ dominant eigenvalues of the prior-preconditioned data misfit Hessian and the eigenvectors corresponding to the $1st$, $4th$, $16th$, and $64th$ largest eigenvalues for the Poisson problem.

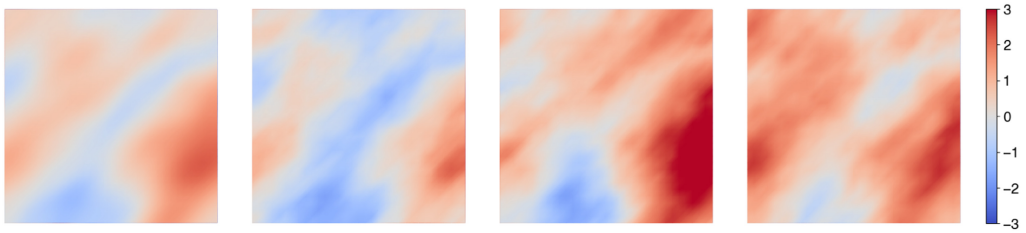


Fig. 9. The MAP point (leftmost) and three sample fields drawn from the Laplace approximation of the posterior distribution for the Poisson problem.

As pointed out in Section 2, explicitly computing the Hessian is prohibitive for large-scale problems, as this entails solving two forward-like PDEs as many times as the number of parameters. To make the operations with the Hessian scalable with respect to the parameter dimension, we invoke a low-rank approximation of the data misfit part of the Hessian, retaining only r eigenvectors that are the most significantly informed directions from the data [Villa et al. 2021].

Figure 8 shows the eigenspectrum of the prior-preconditioned data misfit Hessian. The double pass randomized algorithm provided by `hIPPYlib` with an oversampling factor of 20 is used to accurately compute the dominant eigenpairs. We see that eigenvalues are smaller than 1 after around the $60th$ eigenvalue, indicating that keeping 60 eigenpairs is sufficient for the low-rank approximation. Figure 8 also shows four eigenvectors, which, as expected, illustrate that eigenvectors corresponding to smaller eigenvalues display more fluctuations.

In Figure 9, we depict the MAP point and three samples drawn from the Laplace approximation of the posterior.

5.1.5 Exploring the Posterior using MCMC Methods. In this section, we implement the advanced MCMC algorithms discussed in the Section 3 to explore the posterior and compare their performance.

Table 1. Comparison of the Performance of Several MCMC Methods for the Poisson Problem: pCN, MALA, ∞-MALA, DR, DILI, and their Hessian-informed Versions

Method	AR (%)	MPSRF	Min. ESS (index)	Max. ESS (index)	Avg. ESS	NPS/ES
pCN (5.0E-3)	24	2.629	25 (24)	225 (8)	84	5,952
MALA (6.0E-6)	48	2.642	26 (22)	874 (5)	148	10,135
∞-MALA (1.0E-5)	57	2.943	25 (23)	1,102 (5)	160	9,375
H-pCN (4.0E-1)	27	1.192	64 (1)	3,598 (15)	2,314	216
H-MALA (6.0E-2)	60	1.014	545 (1)	8,868 (19)	6,459	232
H-∞-MALA (1.0E-1)	71	1.016	582 (1)	8,417 (18)	5,905	254
DR (H-pCN (1.0E0), H-MALA (6.0E-2))	(4, 61)	1.013	641 (1)	12,522 (17)	9,222	215
DR (H-pCN (1.0E0), H-∞-MALA (2.0E-1))	(4, 48)	1.011	613 (1)	12,812 (17)	9,141	213
DILI-PRIOR (0.8, 0.1)	(60, 33)	1.064	314 (1)	4,667 (13)	3,216	548
DILI-LA (0.8, 0.1)	(83, 36)	1.017	562 (1)	10,882 (17)	7,192	245
DILI-MAP (0.8, 0.1)	(77, 22)	1.006	1,675 (1)	10,271 (20)	8,692	202

Acceptance rate (AR), **multivariate potential scale reduction factor (MPSRF)**, and **effective sample sample size (ESS)** are reported for convergence diagnostics. MPSRF and ESS are computed with respect to the projection of parameter samples along the first 25 dominant eigenvectors of the prior-preconditioned data misfit Hessian at the MAP point. Two values of AR are listed in DR and DILI-MAP, which are for the first and the second proposal moves, respectively. We also provide the **number of forward and/or adjoint PDE solves per effective sample (NPS/ES)** for sampling efficiency. We use 20 MCMC chains, each with 25,000 iterations (500,000 samples in total). The numbers in parentheses in each method name represent the parameter values used (β for pCN, τ for MALA, h for ∞-MALA, and β and τ for and DILI). The numbers in parentheses of the minimum ESS and the maximum ESS indicate the corresponding eigenvector index.

In particular, we consider pCN, MALA, ∞-MALA, DR, DILI, and their Hessian-informed counterparts. For each method, we simulate 20 independent MCMC chains, each with 25,000 samples (after discarding 2,500 samples as burn-in), and hence draw a total of 500,000 samples from the posterior. A sample from the Laplace approximation of the posterior is chosen as a starting point for each chain.

For checking the convergence and statistical efficiency of MCMC chains, we consider the subspace spanned by the r dominant eigenvectors of the generalized eigensystem in (10), instead of all components of the parameter vector $\boldsymbol{m}$. Specifically, we compute the MPSRF, autocorrelation time, and ESS with respect to a coefficient vector $\mathbf{c} \in \mathbb{R}^r$ defined by

$$\mathbf{c} = \mathbf{V}_r^T \boldsymbol{\Gamma}_{\text{prior}}^{-1} \mathbf{m}. \tag{40}$$

Table 1 shows the convergence diagnostics and computational efficiency of the MCMC samples. MPSRF and ESS are computed with respect to the projection of parameter samples along the first 25 dominant eigenvectors of the prior-preconditioned data misfit Hessian at the MAP point. Table 1 reports the mininum, maximum, and average ESS over all the 25 projections.

The last column in Table 1 represents the number of forward and/or adjoint PDE solves required to draw a single independent sample (average ESS is used). This quantity can be used to measure the sampling efficiency and rank the methods in terms of computational efficiency. Under this metric, DILI-MAP is the most efficient method and requires only 202 PDE solves for an effective sample. DR (213 NPS/ES for H-∞-MALA and 215 NPS/ES for H-MALA) and H-pCN (216 NPS/ES) are close seconds.

We next assess the convergence of MCMC samples of the quantity of interest in (34) to the predictive posterior distribution of $\mathcal{G}(m)$: the autocorrelation function estimates of the quantity of interest (34) are shown in Figure 10 (here, we use the formula (29) to account for the use of multiple chains), and trace plots from three independent MCMC chains and histograms of all the MCMC samples for pCN and H-pCN are depicted in Figure 11.

Lastly, we compare estimates of moments of the quantity of interest for the different sampling strategies. For each MCMC chain, the kth ($k = 1, 2, 3$) moment of the quantity of interest computed

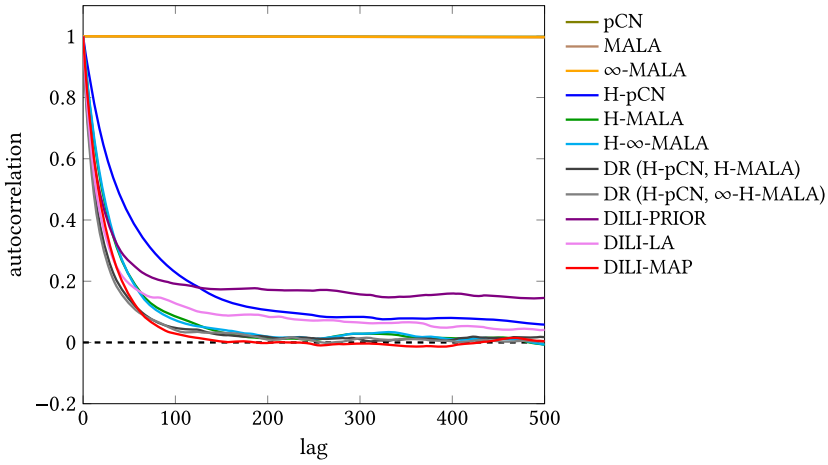


Fig. 10. Autocorrelation function estimate (29) of the quantity of interest $\mathcal{G}$ (34) for several MCMC methods. Note that the autocorrelation function plots for pCN, MALA, and ∞-MALA appear unchanged and overlap.

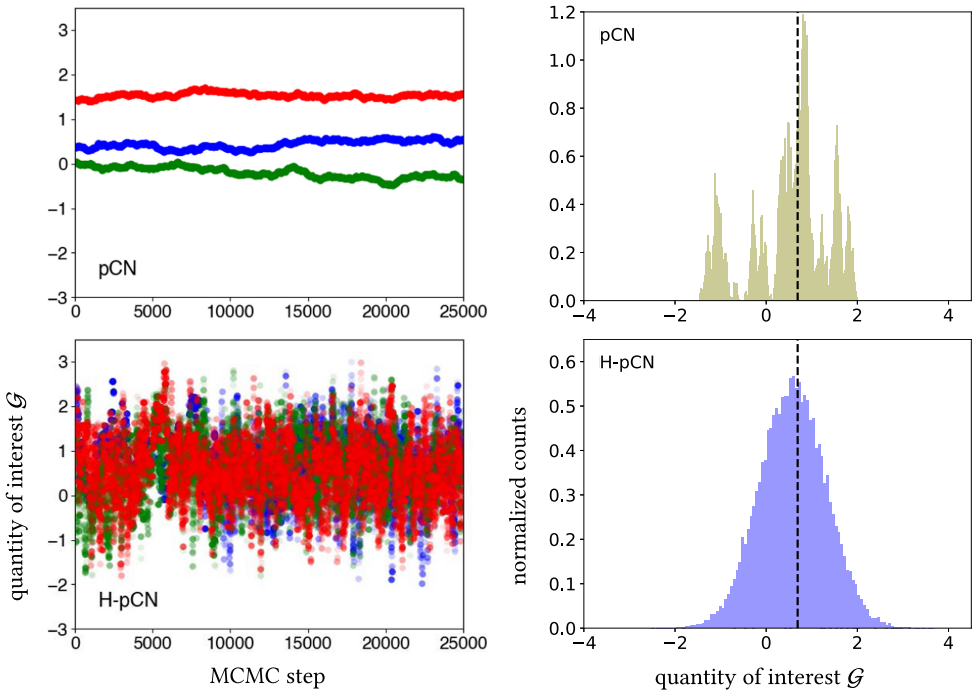


Fig. 11. Left: Trace plots of the quantity of interest $\mathcal{G}$ (34) from three MCMC chains (out of 20 independent chains) of pCN and H-pCN simulations; different colors (here blue, green, and red) represent the trace of each chain. Right: Probability density function estimate of the quantity of interest $\mathcal{G}$ (34) computed from pCN and H-pCN samples; all the 500,000 samples, 20 chains with 25,000 samples each, are pulled together in the histogram; the number of counts is normalized so that the plot represents a probability density function; the black dashed line represents the quantity of interest computed from the true parameter field m_{true}.

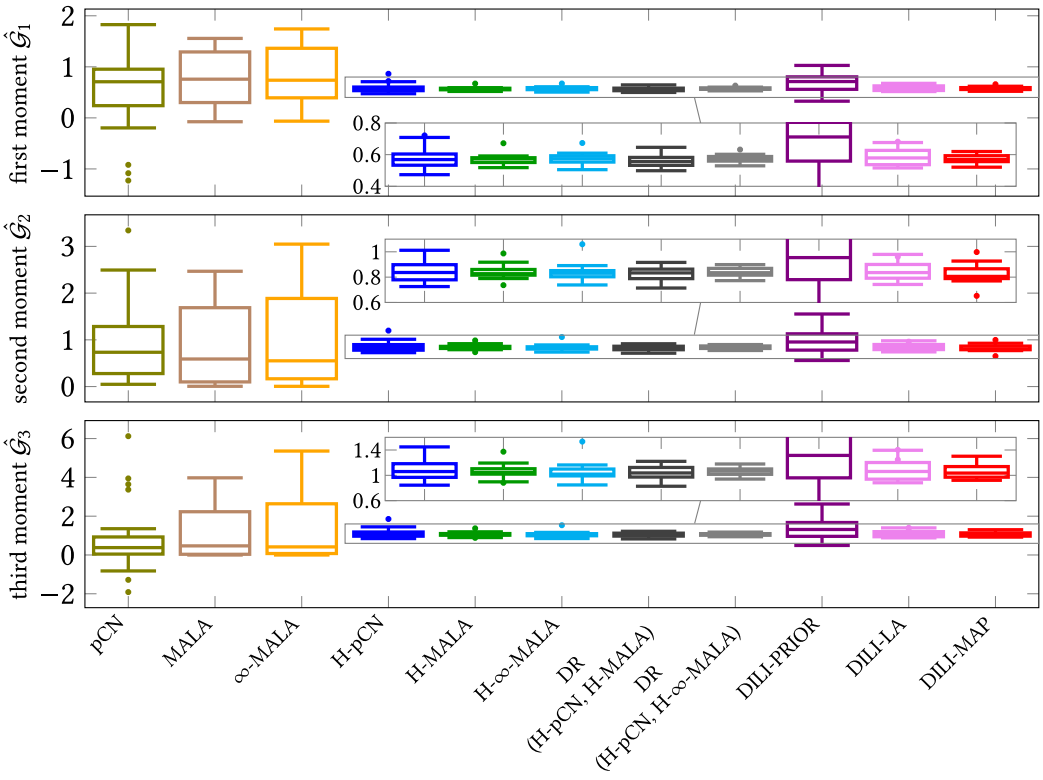


Fig. 12. Whisker plots of first, second, and third moment estimates ($\hat{\mathcal{G}}_k, k = 1, 2, 3$) of the quantity of interest (41) computed by using several MCMC methods. The central mark is the median; lower and upper quartiles represent 25th and 75th percentiles, respectively. Whiskers extend to the extreme data points that fall within the distance from the lower or upper quartiles to 1.5 times the interquartile range (the distance between the upper and lower quartiles); all the other data points are plotted as outliers. The number of data points for each method is 20, the number of independent MCMC chains.

from parameter samples $\mathbf{m}_i$ ($i = 1, 2, \ldots, N; N = 25,000$) is computed as

$$\hat{\mathcal{G}}_k = \frac{1}{N} \sum_{i=1}^{N} \mathcal{G}^k(\mathbf{m}_i). \tag{41}$$

The results are reported in Figure 12 as box-and-whisker plots.

From the results presented in this section, we draw the following conclusions:

- The Hessian information at the MAP point plays an important role in enhancing the sampling performance of the MCMC methods. In fact, MCMC chains without the Hessian information did not converge over the entire length of the chain and were localized around the starting point. The convergence was achieved or nearly achieved only when the MCMC proposal exploited the Laplace approximation of the posterior that incorporates the Hessian information.
- DILI-MAP shows the best sampling efficiency in terms of the number of forward and/or adjoint PDE solves per effective sample. Note that the parameter value used in the MCMC

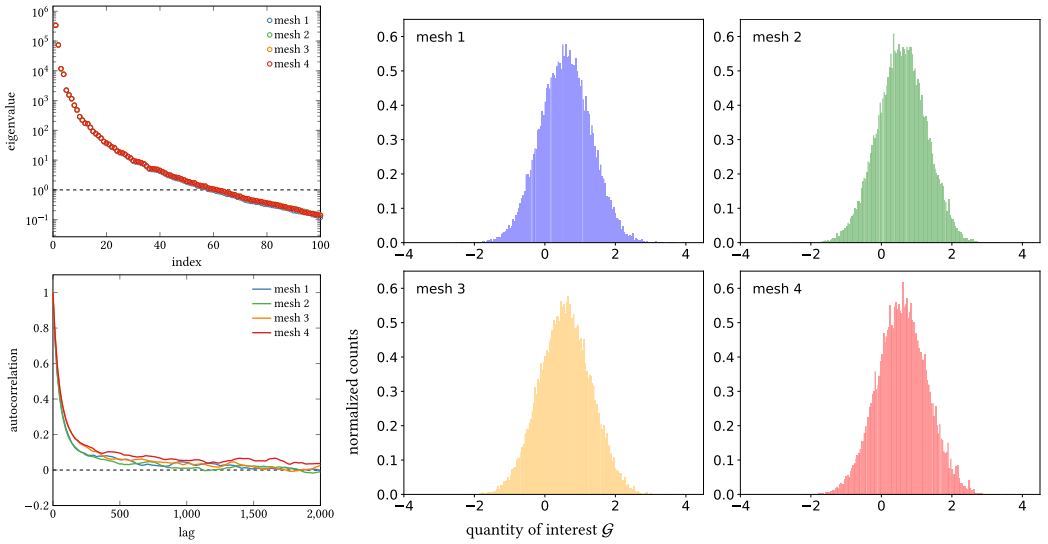


Fig. 13. Left: Logarithmic plot of the $r = 100$ dominant eigenvalues of the prior-preconditioned data misfit Hessian (top), and autocorrelation function estimate (29) of the quantity of interest $\mathcal{G}$ (34) (bottom). Right: Probability density function estimate of the quantity of interest $\mathcal{G}$ (34); all the samples, 20 chains with 25,000 samples each, so 500,000 in total, are pulled together in the histogram; the number of counts is normalized so that the plot represents a probability density function. We consider four different meshes which are increasingly refined from the coarsest (mesh 1) to the finest (mesh 4). We use the H-pCN method ($\beta = 0.4$) to draw samples.

methods (e.g., β and/or τ) was not the optimal and a different result may be obtained with different parameter values.

We further study the performance of MCMC methods under different problem settings to provide more insight into the practical use of the hIPPYlib-MUQ framework.

5.1.6 Scalability of Hessian-informed pCN. Here we investigate the effect of mesh resolution on the sampling performance. A curvature aware MCMC method, the H-pCN is selected with $\beta = 0.4$ for the test. The dimensions of the parameter and the state variables from a coarse mesh (mesh 1) to the finest mesh (mesh 4) are (1,089, 4,225), (4,225, 16,641), (16,641, 66,049), and (66,049, 263,169), respectively.

We follow the same problem setting as before, and use the same synthetic observations (obtained from the true parameter field generated from the finest mesh) for all levels. The top-left part of Figure 13 shows the $r = 100$ dominant eigenvalues of the prior-preconditioned data misfit Hessian. One observes that the eigenspectrum is virtually independent of mesh refinement.

To assess the convergence of the MCMC methods, in Table 2 we report the acceptance rate, MPSRF, and ESS of the posterior samples. The MPSRF and ESS are computed with respect to the projection of parameter samples along the first 25 dominant eigenvectors of the prior-preconditioned data misfit Hessian at the MAP point. We present the autocorrelation function estimates (29) in the bottom-left part of Figure 13, and show histograms for the quantity of interest $\mathcal{G}$ (34) in the right part of Figure 13. The results show that the convergence of samples is almost independent with respect to the MPSRF and the autocorrelation function.

Table 2. Acceptance Rate (AR), Multivariate Potential Scale Reduction Factor (MPSRF) and Effective Sample Size (ESS) of the Posterior Samples Generated by using the H-pCN Method with Different Dimensions

Dimension (state, parameter)	AR (%)	MPSRF	Min. ESS (index)	Max. ESS (index)	Avg. ESS
(4,225, 1,089)	27	1.192	64 (1)	3,598 (15)	2,314
(16,641, 4,225)	24	1.333	63 (1)	3,221 (18)	1,830
(66,049, 16,641)	23	1.075	209 (1)	3,073 (11)	1,940
(263,169, 66,049)	22	1.117	102 (2)	3,276 (15)	1,767

We use $\beta = 0.4$ for the H-pCN method and draw in total 500,000 samples (20 MCMC chains, each with 25,000 iterations). MPSRF and ESS are computed with respect to the projection of samples along the first 25 dominant eigenvectors of the prior-preconditioned data misfit Hessian at the MAP point. The numbers in parentheses of the minimum ESS and the maximum ESS indicate the corresponding eigenvector index.

5.2 Coefficient Field Inversion in a Robin Boundary Condition for a Three-dimensional p-Poisson Nonlinear PDE

So far, we restricted our attention to the additive Gaussian noise model. While this additive noise model is the most commonly used model, it would be inappropriate in some cases such as speckle noise found in **synthetic aperture radar (SAR)** images. In this example, we consider a different noise model where the noise is proportional to the value of the observations.

The example forward model is a nonlinear PDE in three space dimensions for which we seek to infer an unknown coefficient field in a Robin boundary condition. Specifically, the forward governing equations are given by

$$\begin{aligned} -\nabla \cdot \left(|\nabla u|_\epsilon^{p-2} \nabla u \right) &= f \quad \text{in } \Omega, \\ |\nabla u|_\epsilon^{p-2} \nabla u \cdot \mathbf{n} + e^m u &= 0 \quad \text{on } \partial\Omega_R, \\ |\nabla u|_\epsilon^{p-2} \nabla u \cdot \mathbf{n} &= 0 \quad \text{on } \partial\Omega \setminus \partial\Omega_R, \end{aligned} \tag{42}$$

with $1 \le p \le \infty$. Note that the p-Laplacian, $\nabla \cdot (|\nabla u|^{p-2}\nabla u)$, is singular when $p < 2$ and degenerates when $p > 2$ at points $\nabla u = 0$ [Brown 2010; Lindqvist 2017], so a regularization term ϵ (here we take $\epsilon = 1.0 \times 10^{-8}$) is introduced in the above equation as $|\nabla u|_\epsilon = \sqrt{|\nabla u|^2 + \epsilon}$. The p-Laplacian is a nonlinear counterpart of the Laplacian operator, and appears in many nonlinear diffusion problems (e.g., non-Newtonian fluids), where a nonlinear diffusion is modeled as a power law type.

We assume $p = 3$ and consider a thin brick domain $\Omega = [0, 1]^2 \times [0, 0.05]$ with a volume source term ($f = 1$) and a mixed boundary condition, e.g., we impose a Robin boundary condition on the bottom boundary surface and no normal flux on the remaining boundary surfaces.

The problem domain Ω is discretized using a regular tetrahedral grid and using linear finite elements for all the state, adjoint, and parameter variables. After discretization, the dimension is 66,564 for the state and adjoint variables, and 16,641 for the parameter variable.

For the prior, we use the same Gaussian distribution as the one employed in the previous example. We create synthetic state observations at $l = 300$ random locations uniformly distributed on the top boundary surface by first solving the forward problem with the true parameter field m_{true} obtained from a sample of the prior, and then multiplying a Gamma-distributed noise. Specifically, the vector of synthetic observations is of the form

$$\boldsymbol{d} = \boldsymbol{\eta} \odot \mathcal{B}u, \tag{43}$$

where $\odot$ denotes component-wise multiplication, and each component of $\boldsymbol{\eta}$ is independently and identically Gamma-distributed with shape κ and scale ν, i.e., $\eta_i = \text{Gamma}(\kappa, \nu), i = 1, \ldots, q$. We

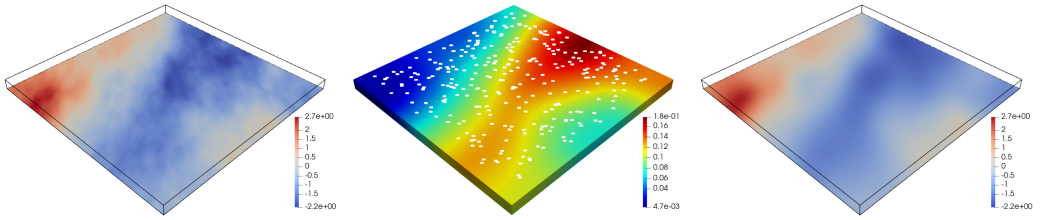


Fig. 14. Left: True parameter field on the bottom surface. Middle: Corresponding state field and $l = 300$ observation points (white square marks) on the top surface. Right: The MAP point on the bottom surface.

Table 3. Convergence Diagnostics for the p-Poisson Problem: Acceptance Rate (AR), Multivariate Potential Scale Reduction Factor (MPSRF), and Effective Sample Sample Size (ESS) of the Projection of the Parameter Samples along the First 25 Eigenvectors of the Prior-preconditioned Data Misfit Hessian at the MAP Point

AR (%)	MPSRF	Min. ESS (index)	Max. ESS (index)	Avg. ESS
23	1.041	243 (0)	2,891 (22)	1,586

We use H-pCN method ($\beta = 0.2$) with 40 chains, each with 25,000 iterations (1,000,000 samples in total). The numbers in parentheses of the minimum ESS and the maximum ESS indicate the corresponding eigenvector index.

take $\kappa = 1/\nu$ and in this case the negative log-likelihood function has the form

$$\Phi(m; \boldsymbol{d}) = \kappa \sum_{i=1}^{q} \left(\log(\mathcal{B}u)_i + \frac{d_i}{(\mathcal{B}u)_i} \right), \tag{44}$$

where the subscript i means ith component of the corresponding vector. In this example, we set $\kappa = 10^4$.

Figure 14 illustrates the true parameter field on the bottom boundary, the locations of the observations on the top surface, and the MAP point obtained by solving the optimization problem of minimizing the negative log-posterior. The Laplace approximation of the posterior is then constructed based on the low-rank factorization of the data misfit Hessian at the MAP point. The spectrum of the prior-preconditioned data misfit Hessian indicates that the number of dominant eigenvalues (larger than 1) is about 55.

5.2.1 MCMC Results for Characterizing the Posterior. Here we present MCMC simulation results for the uncertain boundary coefficient vector. In this example, we consider the H-pCN method with $\beta = 0.2$ and run 40 independent MCMC chains, each with 25,000 iterations after discarding 2,500 samples as burn-in (1,000,000 samples are generated in total). For each MCMC run, a sample from the Laplace approximation of the posterior is taken as the starting point.

Table 3 lists convergence diagnostics of the MCMC simulation. The parameter samples are projected onto the first 25 eigenvectors of the prior-preconditioned data misfit Hessian at the MAP point, and the MPSRF and the ESS are evaluated based on this projection.

Figure 15 shows marginal distributions of the posterior MCMC samples and of the Laplace approximation. As before, these marginals are computed with respect to the projection of the parameter samples onto the eigenvectors. We observe that there is a clear difference between the marginal distributions of the MCMC samples and those of the Laplace approximation, especially for the eigenvector corresponding to larger eigenvalues.

The numerical studies have been carried out on the **Multi-Environment Computer for Exploration and Discovery (MERCED)** cluster at UC Merced. While there are some performance

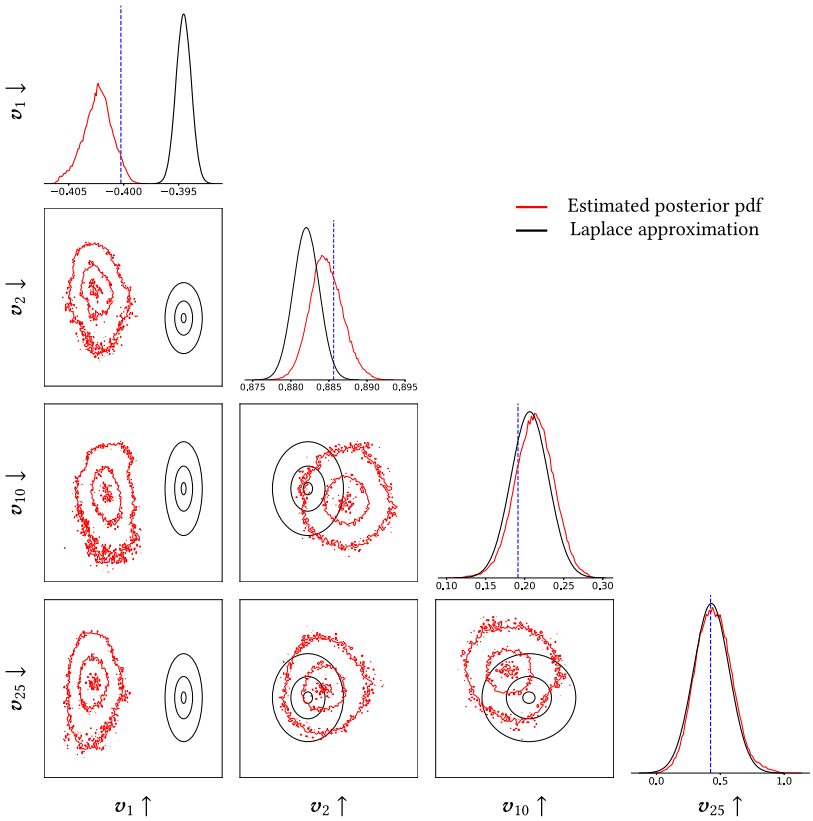


Fig. 15. Marginal distributions of the MCMC samples (red) and the Laplace approximation (black), both for the projection of parameter samples onto the eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_{10}$, and, $\boldsymbol{v}_{25}$. In the one-dimensional marginal plots, the blue dashed lines represent the projection of the true parameter vector onto the eigenvectors. For the two-dimensional marginal plots, we only show three contours that represent 5%, 50%, and 95% of the distribution, respectively. We note the difference in scaling and hence the appearance of a much larger difference between the estimated and the Laplace approximation of the posterior in the first column.

variations depending on the cluster CPU node that a job is assigned to, approximately, a single PDE solve required about 0.07 seconds for the first example, and about 4 seconds for the second three-dimensional example (both are the elapsed time).

6 CONCLUSION

We have presented a robust and scalable software framework for the solution of large-scale Bayesian inverse problems governed by PDEs. The software integrates two complementary open-source software libraries, hIPPYlib and MUQ, resulting in a unique software framework that addresses the prohibitive nature of Bayesian solution of inverse problems governed by PDEs. The main objectives of the proposed software framework are to

(1) provide to domain scientists a suite of sophisticated and computationally efficient MCMC methods that exploit Bayesian inverse problem structure; and
(2) allow researchers to easily implement new methods and compare against the state of the art.

The integration of the two libraries allows advanced MCMC methods to exploit the geometry and intrinsic low-dimensionality of parameter space, leading to efficient and scalable exploration of the posterior distribution. In particular, the Laplace approximation of the posterior is employed to generate high-quality MCMC proposals. This approximation is based on the inverse of the Hessian of the log-posterior, made tractable via low-rank approximation of the Hessian of the log-likelihood. Numerical experiments on linear and nonlinear PDE-based Bayesian inverse problems illustrate the ability of Laplace-based proposals to accelerate MCMC sampling by factors of ∼50×.

Despite the fast and dimension-independent convergence of these advanced structure-exploiting MCMC methods, many Bayesian inverse problems governed by expensive-to-solve PDEs remain out of reach. For example, the results of Section 5.1.5 for the Poisson coefficient inverse problem indicate that $O(10^6)$ PDE solves may still be required even with the most efficient MCMC methods. In such cases, hIPPYlib-MUQ can be used as a prototyping environment to study new methods that further exploit problem structure, for example through the use of various reduced models (e.g., Cui et al. [2016b]) or via advanced Hessian approximations that go beyond low rank Alger et al. [2019]; Ambartsumyan et al. [2020].

We also remark on limitations of our software framework. Since we rely on FEniCS for the finite element approximation of PDEs, hIPPYlib-MUQ inherits all the limitations and challenges that come with this software. However, the FEniCS, hIPPYlib, and MUQ developers and user communities offer a rich support base that the users of hIPPYlib-MUQ can build on. Support for alternative finite element implementations is the subject of future work. Another current limitation of hIPPYlib-MUQ is the lack of parallel implementations of MCMC methods. Therefore, the goal for future versions of hIPPYlib-MUQ is to incorporate multilevel parallelism. This will include both the parallel PDE solvers available now and additional parallel MCMC chains, and will allow solutions of even more complex PDE-based Bayesian inverse problems with higher-dimensional parameter spaces.

Software Availability. hIPPYlib-MUQ is distributed under the GNU General Public License version 3 (GPL3). The hIPPYlib-MUQ project is hosted on GitHub (https://github.com/hippylib/hippylib2muq) and use Travis-CI for continuous integration. hIPPYlib-MUQ uses semantic versioning. The results presented in this work were obtained with hIPPYlib-MUQ version 0.3.0, hIPPYlib version 3.1.0, and MUQ version 0.3.5. A Docker image [Merkel 2014] containing the preinstalled software and examples is available at https://hub.docker.com/r/ktkimyu/hippylib2muq. hIPPYlib-MUQ documentation is hosted on ReadTheDocs (https://hippylib2muq.readthedocs.io). Users are encouraged to join the hIPPYlib and MUQ workspaces on Slack to connect with other users, get help, and discuss new features; see https://hippylib.github.io/#slack-channel and https://mituq.bitbucket.io for more information on how to join.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their careful reading of our manuscript and for their valuable comments and suggestions, which helped us improve the quality of the manuscript.

REFERENCES

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.

N. Alger, V. Rao, A. Meyers, T. Bui-Thanh, and O. Ghattas. 2019. Scalable matrix-free adaptive product-convolution approximation for locally translation-invariant operators. *SIAM Journal on Scientific Computing* 41, 4 (2019), A2296–A2328. https://arxiv.org/abs/1805.06018.

Ilona Ambartsumyan, Wajih Boukaram, Tan Bui-Thanh, Omar Ghattas, David Keyes, Georg Stadler, George Turkiyyah, and Stefano Zampini. 2020. Hierarchical matrix approximations of Hessians arising in inverse problems governed by PDEs. *SIAM Journal on Scientific Computing* 42, 5 (2020), A3397–A3426.

Yves F. Atchadé. 2006. An adaptive version for the Metropolis adjusted Langevin algorithm with a truncated drift. *Methodology and Computing in Applied Probability* 8 (2006), 235–254.

Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Alp Dener, Victorand Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Dave A. May, Lois Curfman McInnes, Richard Tran Mills, Todd Munson, Karl Rupp, Patrick Sanan, Barry F. Smith, Stefano Zampini, and Hong Zhang. 2018. PETSc Web page. http://www.mcs.anl.gov/petsc. http://www.mcs.anl.gov/petsc.

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. http://www.mcs.anl.gov/petsc.

Johnathan M. Bardsley, Tiangang Cui, Youssef M. Marzouk, and Zheng Wang. 2020. Scalable optimization-based sampling on function space. *SIAM Journal on Scientific Computing* 42, 2 (2020), A1317–A1347.

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

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

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

Stephen P. Brooks and Andrew Gelman. 1998. General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics* 7, 4 (Dec. 1998), 434–455. https://doi.org/10.1080/10618600.1998.10474787

Jed Brown. 2010. Efficient nonlinear solvers for nodal high-order finite elements in 3D. *Journal of Scientific Computing* 45, 1 (2010), 48–63. https://doi.org/10.1007/s10915-010-9396-8

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 *SC12: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis.* Gordon Bell Prize finalist.

T. Bui-Thanh, O. Ghattas, J. Martin, and G. 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.

Ben Calderhead. 2014. A general construction for parallelizing Metropolis- Hastings algorithms. *Proceedings of the National Academy of Sciences* 111, 49 (2014), 17408–17413.

George Casella and Edward I. George. 1992. Explaining the Gibbs sampler. *The American Statistician* 46, 3 (1992), 167–174.

Patrick R. Conrad and Youssef M. Marzouk. 2013. Adaptive Smolyak pseudospectral approximations. *SIAM Journal on Scientific Computing* 35, 6 (2013), A2643–A2670. https://doi.org/10.1137/120890715

S. L. Cotter, G. O. Roberts, A. M. Stuart, and D. White. 2012. MCMC methods for functions: Modifying old algorithms to make them faster. (2012). submitted.

T. Cui, K. J. H. Law, and Y. M. Marzouk. 2016a. Dimension-independent likelihood-informed MCMC. *J. Comput. Phys.* 304 (2016), 109–137.

Tiangang Cui, Youssef Marzouk, and Karen Willcox. 2016b. Scalable posterior approximations for large-scale Bayesian inverse problems via likelihood-informed parameter and state reduction. *J. Comput. Phys.* 315 (2016), 363–387.

Tiangang Cui and Olivier Zahm. 2021. Data-free likelihood-informed dimension reduction of Bayesian inverse problems. *Inverse Problems* 37, 4 (2021), 045009.

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

Tim J. Dodwell, Christian Ketelsen, Robert Scheichl, and Aretha L. Teckentrup. 2019. Multilevel Markov chain Monte Carlo. *SIAM Rev.* 61, 3 (2019), 509–545.

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

James M. Flegal and Galin L. Jones. 2010. Batch means and spectral variance estimators in Markov chain Monte Carlo. *The Annals of Statistics* 38, 2 (2010), 1034–1070.

Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. 2004. Bayesian data analysis.

Andrew Gelman and Donald B. Rubin. 1992. Inference from iterative simulation using multiple sequences. *Statistical Science* (1992), 457–472.

O. Ghattas and K. Willcox. 2021. Learning physics-based models from data: Perspectives from inverse problems and model reduction. *Acta Numerica* 30 (2021), 445–554. https://doi.org/doi:10.1017/S0962492921000064

Mark Girolami and Ben Calderhead. 2011. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73, 2 (2011), 123–214.

Gene H. Golub and Charles F. Van Loan. 1996. *Matrix Computations* (Third ed.). Johns Hopkins University Press, Baltimore, MD.

Heikki Haario, Eero Saksman, and Johanna Tamminen. 2001. An adaptive metropolis algorithm. *Bernoulli* 7, 2 (Sep. 2001), 223–242. https://doi.org/10.2307/3318737

Martin Hairer, Andrew M. Stuart, and Sebastian J. Vollmer. 2014. Spectral gaps for a Metropolis–Hastings algorithm in infinite dimensions. *The Annals of Applied Probability* 24, 6 (2014), 2455–2490.

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

Jouni Hartikainen and Simo Särkkä. 2010. Kalman filtering and smoothing solutions to temporal Gaussian process regression models. In *2010 IEEE International Workshop on Machine Learning for Signal Processing*. IEEE, 379–384. https://doi.org/10.1109/MLSP.2010.5589113

W. Keith Hastings. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57, 1 (1970), 97–109.

Matthew D. Hoffman and Andrew Gelman. 2014. The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research* 15, 1 (2014), 1593–1623.

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. *J. Comput. Phys.* 296 (September 2015), 348–368. 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. https://doi.org/10.1007/b138659

Finn Lindgren, Håvard 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. https://doi.org/10.1111/j.1467-9868.2011.00777.x

Peter Lindqvist. 2017. *Notes on the p-Laplace Equation*. Number 161. University of Jyväskylä.

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. https://doi.org/10.1007/978-3-642-23099-8

Tristan Marshall and Gareth Roberts. 2012. An adaptive approach to Langevin MCMC. *Statistics and Computing* 22, 5 (Sept. 2012), 10411057. https://doi.org/10.1007/s11222-011-9276-6

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.

Youssef Marzouk, Tarek Moselhy, Matthew Parno, and Alessio Spantini. 2016. *Sampling via Measure Transport: An Introduction*. Springer International Publishing, 1–41. https://doi.org/10.1007/978-3-319-11259-6_23-1

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

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. 1953. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* 21, 6 (1953), 1087–1092. https://doi.org/10.1063/1.1699114

Antonietta Mira et al. 2001. On Metropolis-Hastings algorithms with delayed rejection. *Metron* 59, 3-4 (2001), 231–241.

R. M. Neal. 2010. *Handbook of Markov Chain Monte Carlo*. Chapman & Hall/CRC Press, Chapter MCMC using Hamiltonian dynamics.

Art B. Owen. 2013. Monte Carlo theory, methods and examples. (2013).

Matthew Parno, Andrew Davis, Patrick Conrad, and Y. M. Marzouk. 2014. MIT Uncertainty Quantification (MUQ) Library. https://muq.mit.edu.

Matthew D. Parno and Youssef M. Marzouk. 2018. Transport map accelerated Markov chain Monte Carlo. *SIAM/ASA Journal on Uncertainty Quantification* 6, 2 (2018), 645–682. https://doi.org/10.1137/17M1134640

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.

Frank J. Pinski, Gideon Simpson, Andrew M. Stuart, and Hendrik Weber. 2015. Algorithms for Kullback–Leibler approximation of probability measures in infinite dimensions. *SIAM Journal on Scientific Computing* 37, 6 (2015), A2733–A2757.

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

Carl Edward Rasmussen and Christopher K. I. Williams. 2005. *Gaussian Processes for Machine Learning*. The MIT Press.

Christian P. Robert and George Casella. 2005. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Gareth O. Roberts, Jeffrey S. Rosenthal, et al. 2004. General state space Markov chains and MCMC algorithms. *Probability Surveys* 1 (2004), 20–71.

Gareth O. Roberts and Osnat Stramer. 2003. Langevin diffussions and Metropolis-Hastings algorithms. *Methodology and Computing in Applied Probability* 4 (2003), 337–357.

D. Rudolph and B. Sprungk. 2018. On a generalization of the preconditioned Crank-Nicolson Metropolis algorithm. *Foundations of Computational Mathematics* 18 (2018), 309–343. Issue 2.

S. M. Stigler. 1986. Laplace's 1774 memoir on inverse probability. *Statist. Sci.* 1, 3 (08 1986), 359–363. 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. https://doi.org/10.1017/S0962492910000061

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. *J. Amer. Statist. Assoc.* 81, 393 (1986), 82–86. https://doi.org/10.1080/01621459.1986.10478240

The Trilinos Project Team. 2020 (accessed May 22, 2020). *The Trilinos Project Website*. https://trilinos.github.io.

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

Dootika Vats, James M. Flegal, and Galin L. Jones. 2019. Multivariate output analysis for Markov chain Monte Carlo. *Biometrika* 106, 2 (2019), 321–337.

Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. 2020. Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC (with discussion). *Bayesian Analysis* 16, 2 (2020), 1–26. https://doi.org/10.1214/20-ba1221 arXiv:arXiv:1903.08008v5

Umberto Villa, Noemi Petra, and Omar Ghattas. 2020. hIPPYlib user manual: Version 2. (6 2020). https://doi.org/10.6084/m9.figshare.12510578.v1

Umberto Villa, Noemi Petra, and Omar Ghattas. 2021. hIPPYlib: An extensible software framework for large-scale inverse problems governed by PDEs: Part I: Deterministic inversion and linearized Bayesian inference. *ACM Trans. Math. Softw.* 47, 2, Article 16 (April 2021), 34 pages. https://doi.org/10.1145/3428447

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

Ulli Wolff, Alpha Collaboration, et al. 2004. Monte Carlo errors with less errors. *Computer Physics Communications* 156, 2 (2004), 143–153.

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

Olivier Zahm, Tiangang Cui, Kody Law, Alessio Spantini, and Youssef Marzouk. 2022. Certified dimension reduction in nonlinear Bayesian inverse problems. *Math. Comp.* 91, 336 (2022), 1789–1835.

Received 29 November 2021; revised 30 October 2022; accepted 15 December 2022