Variational Bayesian Inference using Sparse Matrices for PDE Inverse Problems

Jan Povala[a,c,1,*], Ieva Kazlauskaite[b,1], Eky Febrianto[b,c], Fehmi Cirak[b,c], Mark Girolami[b,c]

[a]*Department of Mathematics, Imperial College London, London, SW7 2AZ, UK*
[b]*Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ, UK*
[c]*The Alan Turing Institute, London, NW1 2DB, UK*

Abstract

Inverse problems involving partial differential equations have shown a great practical use in science and engineering. Although such problems are generally ill-posed, different regularization approaches have been developed to ameliorate this problem. Among them is the Bayesian approach, where a prior probability measure is placed on the quantity of interest. The resulting posterior probability measure is in most cases analytically intractable, and approximation techniques must be used. Markov Chain Monte Carlo (MCMC) method has been the go-to method for sampling from those posterior measures. This method, although asymptotically exact, is computationally infeasible for large-size problems that often arise in engineering practice. Variational Bayes (VB) was proposed as a more computationally tractable method for Bayesian inference, approximating a Bayesian posterior distribution with a simpler variational distribution by solving an optimization problem. In this work, we argue, through an extensive empirical assessment, that VB methods are a more flexible, faster, and more scalable alternative to MCMC methods for this class of problems. We propose a natural choice of a family of variational distributions parametrized by precision matrices, thus taking advantage of the sparse structure encoded in the discretization of the problem. We utilize stochastic optimization to efficiently estimate the variational objective and assess not only the expected error in the mean solution but also the ability to quantify the uncertainty of the estimate. We test this on PDEs based on the Poisson equation in 1D and 2D.

Keywords: Bayesian inverse problems, variational inference, precision matrix, uncertainty quantification

1. Introduction

In computational mechanics, one posits a structural model for a system and calibrates it using collected measurements from experiments, such as tensile tests. The increased availability of measurements from engineering systems allows for developing new and improving upon existing computational models. Inferring the system parameters (*e.g.*, some material properties) from observations (*e.g.*, the displacement field) of the physical entities is termed the *inverse problem* (Tarantola, 2005; Kaipio and Somersalo, 2005; Stuart, 2010). Conversely, once the system properties are inferred, the performance of the physical system can be predicted under various conditions, such as different loads applied to an object; this is termed the *forward problem*.

[*] Corresponding author
Email address: `jan.povala11@imperial.ac.uk` (Jan Povala)
[1] Equal contribution.

Most inverse problems, including those that are based on partial differential equation (PDE) forward mappings, are ill-posed, meaning that the existence, uniqueness, and/or stability (continuous dependence on the parameters) of the solution are violated (Stuart, 2010; Kaipio and Somersalo, 2005). These issues are often alleviated through some regularization, like Tikhonov regularization (Tikhonov and Arsenin, 1977), that imposes assumptions on the solutions. Alternatively, the specification of the prior in the Bayesian formulation of inverse problems provides a natural choice for regularization, and any given regularization can be interpreted as a specific choice of priors in the Bayesian setting (Bishop, 2006). Furthermore, the Bayesian formulation provides not only a qualitative but also a quantitative estimate of the solution. In particular, the mean of the posterior probability distribution corresponds to the point estimate of the solution while the credible intervals capture the range of the parameters consistent with the observed measurements. For these reasons, Bayesian methods have gained popularity in computational mechanics for experimental design and inverse problems with uncertainty quantification (Abdulle and Garegnani, 2021; Pandita et al., 2021; Pyrialakos et al., 2021; Ni et al., 2021; Sabater et al., 2021; Huang et al., 2021; Ibrahimbegovic et al., 2020; Tarakanov and Elsheikh, 2020; Michelén Ströfer et al., 2020; Carlon et al., 2020; Wu et al., 2020; Uribe et al., 2020; Rizzi et al., 2019; Arnst and Soize, 2019*a*,*b*; Beck et al., 2018; Betz et al., 2018; Chen et al., 2017; Asaadi and Heyns, 2017; Huang et al., 2017; Karathanasopoulos et al., 2017; Babuška et al., 2016)

The Bayesian formulation of inverse problems is the focal point of probabilistic machine learning, and in recent years significant progress has been made in adapting and scaling machine learning approaches to complex large-scale problems (Lu and Tang, 2015; Solin et al., 2018). One of the leading models for Bayesian inverse problems are Gaussian processes (GPs) which define probability distributions over functions and allow for ways to generalize from observed data. Given that most posterior distributions in Bayesian inference are analytically intractable, approximation methods need to be resorted to. Two classical approximation schemes are Markov Chain Monte Carlo (MCMC) and Laplace approximation (LA). MCMC algorithm proceeds by creating a Markov Chain whose stationary distribution is the desired posterior distribution. Although MCMC provides asymptotic convergence in distribution, devising an efficient, finite-time sampling scheme is challenging, especially in higher dimensions (Gelman et al., 2013). Due to the asymptotic correctness of MCMC, we use it as a benchmark for the experimental studies in this paper. Meanwhile, the Laplace approximation finds a Gaussian density centered around the mode of the true posterior, utilizing the negative Hessian of the unnormalized posterior log-density (Bishop, 2006). The Hessian is a large dense matrix, where forming each column requires multiple PDE solves; to make such calculation feasible, low-rank approximations are typically used (Villa et al., 2021; Bui-Thanh et al., 2013). Additionally, the Laplace approximation is not suitable for multi-modal posterior distributions due to the uni-modality of the Gaussian distribution.

In recent years, advances in variational Bayes (VB) methods have allowed for Bayesian inference to be successfully applied to large data sets. Variational Bayes translates an integration problem that arises from applying the Bayes rule into an optimization problem (Jordan et al., 1999; Blei et al., 2017; Jordan and Wainwright, 2007). The method finds a solution that minimizes the Kullback-Leibler (KL) divergence between the true posterior distribution and a candidate from a proposed family of distributions, for instance, multivariate Gaussian distributions with a specific covariance structure. The strong appeal of VB is that one can explicitly choose the complexity of the variational family such that the resulting optimization problem is computationally tractable, and the approximate posterior adequately captures important aspects of the true posterior. In comparison, the traditional approaches such as MCMC and Laplace approximation do not allow for this explicit flexibility of trading off computational tractability of the problem for a more accurate approximated distribution. Further scalability of VB methods is due to advancements in sparse approximations and approximate inference. For instance, sparse GPs rely on lower-dimensional representations that are defined by a smaller set of so-called inducing points to represent the full GP (Csató and Opper, 2002; Seeger et al., 2003; Quiñonero Candela and Rasmussen, 2005; Snelson and Ghahramani, 2006; Titsias, 2009, 2008). Using this approximation for a data set of size N, algorithmic complexity is reduced from $\mathcal{O}(N^3)$ to $\mathcal{O}(NM^2)$, while storage demands go down from $\mathcal{O}(N^2)$ to $\mathcal{O}(NM)$, where M is a user selected number of inducing variables. To widen the applicability of VB to large datasets and non-conjugate models (combinations of prior distributions and likelihoods that do not result in a closed-form solution), *stochastic variational inference* (SVI) was proposed (Hensman et al., 2012; Hoffman et al., 2013; Hensman et al., 2013). Sub-sampling the original data and Monte Carlo estimation of the optimization objective and its gradients, allows for calibrating complex models using large amounts of data. Multiple further extensions to the sparse SVI framework were proposed, leveraging the Hilbert space formulation of VB (Cheng and Boots, 2017), introducing parametric approximations (Jankowiak et al., 2020), applying Lanczos algorithm to efficiently factorize the covariance matrix (Pleiss et al., 2018), transforming to an orthogonal basis (Salimbeni et al., 2018; Shi et al., 2020), and adapting to compositional models (Salimbeni and Deisenroth, 2017).

The choice of prior is a central task in designing Bayesian models. If the prior is obtained from a domain expert, it is not necessarily less valuable than the data itself; one way of thinking about a prior is by considering how many observations one would be prepared to trade for a prior from an expert – if the expert is very knowledgeable, then one might be prepared to exchange a large part of a dataset to get access to that prior. Translating the expert knowledge into a prior probability distribution is a challenging task, and due to practical considerations, certain choices of priors are preferred for their simplicity and analytic tractability. When inferring values of parameters over a spatial domain, as is typically the case in computational mechanics, GP priors offer a natural way to incorporate the information about the smoothness and other known properties of the solution. We note that while other Bayesian models, such as Bayesian neural networks are gaining interest, it is very difficult to impose functional priors in such models, challenging the effective use of expert knowledge and leading to unrealistic uncertainty estimates (Sun et al., 2019; Burt et al., 2021).

In this work, we advocate for the use of GP priors with stochastic variational inference as a principled and scalable way to solve the inverse problems arising in computational mechanics. We show, through an extensive empirical study, that variational Bayes methods provide a flexible, fast, and scalable alternative to MCMC methods in the context of Bayesian inverse problems based on elliptic PDEs while retaining the ability to quantify uncertainty. While similar directions have been explored in previous work, the focus there is on specific applications, such as parameter estimation problems in models of contamination (Tsilifis et al., 2016) or proof-of-concept on particular 1D inverse problems (Barajas-Solano and Tartakovsky, 2019).

We extend the previous works in multiple aspects, focusing on improving the utility of VB in inverse problems arising from elliptic PDEs and providing a thorough discussion of the empirical results that can be used by practitioners to guide their use of VB in applications. Specifically, we argue that the efficiency of the VB algorithms for PDE based inverse problems can be improved by taking in to account the structure of the problem, as encoded in the FEM discretization of the PDE. Motivated by previous uses of precision matrices as a way of describing conditional independence (Tan and Nott, 2018; Durrande et al., 2019), we leverage the sparse structure of the problems to impose conditional independence in the approximating posterior distribution. This choice of parameterisation results in sparse matrices, which improve the computational and the memory cost of the resulting algorithms. Such parameterisation, combined with stochastic optimisation techniques, allows the method to be scaled up to large problems on 2D domains. Through extensive empirical comparisons, we demonstrate that VB provides high quality point estimates and uncertainty quantification comparable to the estimates attained by MCMC algorithms but with significant computational gains. Finally, we describe how the proposed framework can be seamlessly combined with existing solvers and optimization algorithms in the finite element implementations.

The rest of the paper is structured as follows. In Sec. 2, we define Bayesian inverse problems and detail some inference challenges related to their ill-posedness. In Sec. 3, we give a presentation of the variational inference framework, with strong focus on sparse parametrization resulting from conditional independence. We give details of the experiments and the evaluation criteria, and discuss obtained results for each experiment in Sec. 4. Lastly, Sec. 5 concludes the paper and discusses some promising directions for future work.

2. Bayesian Formulation of Inverse Problems

2.1. Forward map

Let $\Omega \subset \mathbb{R}^d$, $d \in \{1, 2, 3\}$ be the domain and $\kappa : \Omega \mapsto \mathbb{R}$ some function of interest. Let $\mathbf{y}$ be noisy observations of $\mathcal{A}(\kappa)$ at a finite set of N points $\mathbf{x}$ in Ω, where $\mathcal{A}$ is a forward solution of an elliptic partial differential equation. The forward map $\mathcal{A}$ defines the relationship between κ and the solution u of the PDE:

$$u(x) = \mathcal{A}(\kappa)(x). \tag{1}$$

We assume that the forward map is non-linear in κ. Note that the mapping itself may be random; for example, the underlying PDE may contain a random forcing term. Assuming that the vector of observations $\mathbf{y}$ at a set of locations $\boldsymbol{x}$ is corrupted by additive Gaussian noise, we write the model for observations succinctly as

$$\mathbf{y} = \mathbf{u}(\kappa) + \boldsymbol{\epsilon}_{\mathbf{y}}, \qquad \boldsymbol{\epsilon}_{\mathbf{y}} \sim \mathcal{N}(0, \sigma_y^2 \mathbb{I}), \tag{2}$$

where $\mathbf{u}$ is the solution $u(x)$ evaluated at points $\boldsymbol{x}$, σ_y is the standard deviation of the measurement noise and $\mathbb{I}$ is the identity matrix. We wish to infer κ from its indirect noisy observations $\mathbf{y}$.

2.2. Inference

The task of inferring the function κ, which is an infinite-dimensional object, from a finite set of observations $\{(\mathbf{x}, \mathbf{y})\}$ is an ill-posed problem. In general, the ill-posedness of inverse problems manifests itself in different ways: there may be no solution, the solution may not be unique, or the solution may be sensitive to perturbations in the data (Stuart, 2010). These issues can be ameliorated by assuming a certain kind of structure in the function of interest, κ. The two main approaches to solving ill-posed problems are 1) regularization of κ which reduces the space of admissible solutions, and 2) postulating the problem in a Bayesian setting by placing an appropriate prior probability measure μ_0 on κ, and inferring the posterior measure, $\mu^{\mathbf{y}}$, after incorporating the observations $\mathbf{y}$. In this work, we follow the Bayesian approach as it provides a principled way of quantifying uncertainty through probability theory (Stuart, 2010).

To infer κ from the observed data $\{(\mathbf{x}, \mathbf{y})\}$, we incorporate any prior knowledge about κ, such as smoothness or amplitude, into the formulation of the inverse problem through the prior probability measure μ_0. The objective then becomes to find the posterior measure, $\mu^{\mathbf{y}}$, conditioned on the observations, as dictated by the Bayes's rule.

The function of interest, κ, is an infinite-dimensional object, however, any computational algorithm works on a finite-dimensional discretization of the function of interest, thus requiring κ to be discretized at some stage of the inference procedure. For Bayesian inverse problems, there are two ways: 1) apply the Bayesian methodology first, discretize afterwards, or 2) discretize the problem first, then apply the Bayesian methodology (Stuart, 2010).

2.2.1. Inference in function space

The first approach develops the methodology on the function space before discretizing. Given that probability measures on function spaces do not admit probability densities, the Bayes rule is applied directly on the measures as follows:

$$\frac{\mathrm{d}\mu^{\mathbf{y}}}{\mathrm{d}\mu_0}(\kappa) = \frac{1}{Z(\mathbf{y})} \exp(\Phi(\kappa; \mathbf{y})), \tag{3}$$

where $\frac{\mathrm{d}\mu^{\mathbf{y}}}{\mathrm{d}\mu_0}$ is the Radon-Nikodym derivative of $\mu^{\mathbf{y}}$ with respect to μ_0, and Φ is the potential function which is determined by the forward PDE problem and the data likelihood (Stuart, 2010). A widely used algorithm of this form is the preconditioned Crank-Nicholson (pCN) MCMC scheme, where proposals are based on the measure μ_0 and the current state. The pCN method is a standard choice for high-dimensional sampling problems, as its implementation is well-defined (even for infinite dimensional models), and it scales favorably with increasing dimensionality (Cotter et al., 2013; Pinski et al., 2015). Since we will use this algorithm as one of the baselines, we provide a summary of the algorithm in Sec. Appendix B.1. In recent years, some variational Bayes formulations in function space have been proposed (for example, Minh (2017); Burt et al. (2021)), though currently they do not offer a viable computational alternative to the finite-dimensional formulation of variational inference.

2.2.2. Inference in discretized space

The second approach proceeds by first discretizing the problem and then deriving inference methods on the finite-dimensional space. Generally, this simplifies the development of the corresponding computational algorithms. In this framework, MCMC algorithms such as Metropolis-Hastings or Hamiltonian Monte Carlo (HMC) (Gelman et al., 2013), as well as the Laplace approximation, or variational Bayes can be applied. In this finite-dimensional case, the probability densities with respect to the Lebesgue measure can be defined, thus leading to a more familiar formulation of the Bayes's rule:

$$p(\boldsymbol{\kappa} \mid \mathbf{y}) = \frac{p(\mathbf{y} \mid \boldsymbol{\kappa})\; p(\boldsymbol{\kappa})}{p(\mathbf{y})}, \tag{4}$$

where $p(\mathbf{y} \mid \boldsymbol{\kappa})$ is the likelihood of the observed data for a given $\boldsymbol{\kappa}$ and is determined by the forward map $\mathcal{A}(\kappa)$ and the measurement error $\boldsymbol{\epsilon}_{\mathbf{y}}$. The prior on $\boldsymbol{\kappa}$ which itself depends on some parameters ψ is denoted by $p(\boldsymbol{\kappa})$. Note that the boldface of the κ symbol signifies its discretized version.

In the discretized formulation, HMC has achieved recognition as the *gold standard* for its convergence properties, favorable performance on high-dimensional and poorly conditioned problems, and universality of implementation that enables its generic use in many applications through probabilistic programming languages (*e.g.*, Stan (Carpenter et al., 2017)). Therefore, along with the pCN scheme described above, our baseline for inference methods includes the HMC method, and we provide a summary of the HMC scheme in Sec. Appendix B.2.

Regardless of the specific approach taken, MCMC schemes provide *asymptotic* convergence in distribution to the true posterior distribution. This property makes them a robust tool used throughout the scientific community (Gelman et al., 2013). However, devising computationally efficient MCMC schemes in higher dimensions becomes computationally prohibitive for engineering-scale applications, and even schemes such as pCN and HMC that have favorable scaling properties among the MCMC algorithms can only be applied to toy problems. With this in mind, VB was proposed as a more computationally tractable method for Bayesian inference, recasting the integration problem associated with the Bayes's rule to an optimization problem of approximating a Bayesian posterior distribution with a simpler variational distribution. To derive a VB scheme for inferring $p(\boldsymbol{\kappa} \mid \mathbf{y})$ in 4, we require $p(\mathbf{y} \mid \boldsymbol{\kappa})$ and $p(\boldsymbol{\kappa})$, which we derive in the next sections.

2.2.3. Consistency of results

A recent line of work (Abraham and Nickl, 2020; Monard et al., 2020; Giordano and Nickl, 2020) investigates the statistical guarantees of popular MCMC schemes, such as the pCN algorithm and the unadjusted discretized Langevin algorithm, for Bayesian inverse problems based on PDE forward mappings. This work provides algorithmic guarantees in the form of posterior contraction rates (rates at which the posterior mean approaches the ground truth as the dataset size increases) for specific families of inverse problems (including elliptic PDEs (Giordano and Nickl, 2020)). Furthermore, it investigates the computational cost associated with sampling from the posterior measure in situations where the prior measure is Gaussian while the likelihood is implied by a PDE. These results depend on the injectivity properties of the inverse mapping which come from the sensitivity analysis of a particular PDE problem; however, from an algorithmic perspective, only the forward mapping is evaluated (Monard et al., 2020). While similar results for VB are not possible due to the properties outlined above, one may leverage the computational benefits of VB to produce a preconditioned proposal distribution for an MCMC scheme such as pCN as outlined in (Pinski et al., 2015).

2.3. Likelihood, Poisson Equation

The likelihood term, $p(\mathbf{y}|\boldsymbol{\kappa})$, consists of two components. Firstly, the non-linear forward map, $u(x) = \mathcal{A}(\kappa)(x)$, which may itself be random, evaluated at a set of locations, $\boldsymbol{x}$ determines $p(\mathbf{u} \mid \boldsymbol{\kappa})$. Note that when the map is deterministic, $p(\mathbf{u} \mid \boldsymbol{\kappa})$ is a Dirac delta centered at $\mathcal{A}(\kappa)(x)$ evaluated at $\boldsymbol{x}$. Secondly, the likelihood term depends on the assumed model for the measurement errors $\boldsymbol{\epsilon}_{\mathbf{y}}$ so that the observations $\mathbf{y}$ are modeled as $p(\mathbf{y}|\mathbf{u}) \sim \mathcal{N}(\mathbf{u}, \sigma_y^2 \mathbb{I})$. Marginalizing $\mathbf{u}$, we write this concisely as

$$p(\mathbf{y}|\boldsymbol{\kappa}) = \int p(\mathbf{y} \mid \mathbf{u}) p(\mathbf{u} \mid \boldsymbol{\kappa}) \mathrm{d}\mathbf{u}. \tag{5}$$

Let us consider a specific example where u is a solution to the Poisson equation with prescribed boundary conditions:

$$-\nabla \cdot (\exp(\kappa(x)) \nabla u(x)) = f(x), \tag{6}$$

where $\kappa(x) \in \mathbb{R}$ is the log-diffusion coefficient, $u(x) \in \mathbb{R}$ is the unknown, and $f(x) \in \mathbb{R}$ is a deterministic forcing term. Since f is a deterministic forcing term, $u(\kappa)$ is a deterministic mapping. Also, note that while the Poisson equation is linear in u, it is nonlinear in the inverse formulation w.r.t. the parameter κ. Throughout the experiments in the later sections, we either set Dirichlet (essential) boundary conditions everywhere (for example $u(x) = 0$ on $\partial\Omega$), or assume Neumann (natural) boundary conditions on parts of the boundary. The choice will be made explicit in each experiment.

To compute the likelihood, we solve (6) for $u(x)$ using the finite element method (FEM).

2.3.1. FEM Discretization

We discretize the weak form of the governing equations with a standard finite element approach. Specifically, the domain of interest Ω is subdivided into a set $\{\omega_e\}$ of non-overlapping elements of size $h = \max_e \operatorname{diam}(\omega_e)$ such that:

$$\Omega = \bigcup_{e=1}^{n_e} \omega_e. \tag{7}$$

The unknown field $u(x)$ is approximated with Lagrange basis functions $\phi_i(x)$ and the respective nodal coefficients $\mathbf{u} = [\mathbf{u}_1, \ldots, \mathbf{u}_{n_u}]^\top$ of the n_u non-Dirichlet boundary mesh nodes by

$$u_h(x) = \sum_{i=1}^{n_u} \phi_i(x) \mathbf{u}_i. \tag{8}$$

The discretization of the weak form of the Poisson equation yields a linear system of equations:

$$\mathbf{A}(\boldsymbol{\kappa})\mathbf{u} = \mathbf{f} \tag{9}$$

where $\mathbf{A}(\boldsymbol{\kappa}) \in \mathbb{R}^{n_u \times n_u}$ is the stiffness matrix, $\boldsymbol{\kappa} \in \mathbb{R}^{n_{\boldsymbol{\kappa}}}$ is the vector of log-diffusion coefficients, $\mathbf{f} \in \mathbb{R}^{n_u}$ is the nodal source vector. The construction of the system matrix $\mathbf{A}$ and the source vector $\mathbf{f}$ depends on the weak formulation of the governing equations.

The elements in the stiffness matrix are constructed as follows:

$$A^e_{ij}(\boldsymbol{\kappa}_e) = \int_{\omega_e} \exp(\boldsymbol{\kappa}_e) \frac{\partial \phi_i(x)}{\partial x} \cdot \frac{\partial \phi_j(x)}{\partial x} \mathrm{d}x \tag{10}$$

where we assume that the log-diffusion coefficient $\boldsymbol{\kappa}_e$ is *constant* within each element with index e. The source vector is constructed as:

$$\mathbf{f}_i(\kappa_e) = \int_\Omega f(x)\phi_i(x)\mathrm{d}x. \tag{11}$$

Then the mapping from the coefficients $\boldsymbol{\kappa}$ to the solution $\mathbf{u}$ is $\mathbf{u} = \mathbf{A}(\boldsymbol{\kappa})^{-1}\mathbf{f}$. The unconditional distribution of $\mathbf{u}$ is:

$$p(\mathbf{u}) = \int p(\mathbf{u} \mid \boldsymbol{\kappa}) p(\boldsymbol{\kappa}) \mathrm{d}\boldsymbol{\kappa} \tag{12}$$

where $p(\mathbf{u} \mid \boldsymbol{\kappa})$ is deterministic (as defined in Eq. 9) but $\boldsymbol{\kappa}$ appears in it non-linearly, implying that the inference is not analytically tractable.

Deterministic forcing ensures that $p(\mathbf{u}|\boldsymbol{\kappa})$ in (5) is a Dirac delta function, simplifying the likelihood to

$$p(\mathbf{y} \mid \boldsymbol{\kappa}) = p(\mathbf{y} \mid \mathbf{u}(\boldsymbol{\kappa})) = \mathcal{N}(\mathbf{A}(\boldsymbol{\kappa})^{-1}\mathbf{f}, \sigma_y^2 \mathbb{I}), \tag{13}$$

which is the final form for the computation of the likelihood.

2.4. Prior

Gaussian process is a common choice of prior measure for functional spatial data. It is a well-defined measure in a complete Hilbert space, thus offering a convenient way of defining priors over functions. Bayesian inference for inverse problems in the infinite-dimensional setup, as described in Stuart (2010), requires that μ_0 in (3) is a Gaussian measure. The conjugacy properties with Gaussian likelihoods and the ease of specification through the choice of kernel are another reason for its wide adoption (Rasmussen and Williams, 2006).

We denote a GP functional prior, fully specified by a mean function $m(x)$ (typically assumed to be zero) and a covariance function $k(x, x')$, as:

$$g(x) \sim \mathcal{GP}\big(m(x), k(x, x')\big). \tag{14}$$

The index set of the two functions is infinite which allows GPs to be interpreted as non-parametric priors over the space of functions. However, even though the process is infinite-dimensional, an instantiation of the process is finite and reduces to a Gaussian distribution by definition. The covariance function is typically parametrized by a set of hyperparameters ψ. One popular option is the squared exponential kernel (also called the exponentiated quadratic or the radial basis function (RBF) kernel):

$$k_{\mathrm{SE}}(x, x') = \alpha \exp\left(-\frac{r^2}{2\ell^2}\right), \tag{15}$$

where $r = \|x - x'\|$ is the L^2 distance between the inputs. It depends on two parameters $\psi = \{\alpha, \ell\}$, the scaling parameter α, and the length scale ℓ. Note that, $k_{\mathrm{SE}}(\cdot, \cdot)$ is an infinitely smooth function which implies that so is $g(\cdot)$.

Both conditioning and marginalization can be done in closed form. In particular, consider the joint model of the values $\mathbf{g}$ at training locations $\mathbf{x}$ and the test values $\mathbf{g}^*$ at test locations $\mathbf{x}^*$:

$$\begin{bmatrix} \mathbf{g} \\ \mathbf{g}^* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_\psi(\mathbf{x}, \mathbf{x}) & \mathbf{K}_\psi(\mathbf{x}, \mathbf{x}^*) \\ \mathbf{K}_\psi(\mathbf{x}^*, \mathbf{x}) & \mathbf{K}_\psi(\mathbf{x}^*, \mathbf{x}^*) \end{bmatrix}\right), \tag{16}$$

where $\mathbf{K}_\psi(\mathbf{x}, \mathbf{x}^*)$ is the matrix resulting from evaluating $k(\cdot, \cdot)$ at all pairs of training and test points. The conditional distribution of the function values $\mathbf{g}^*$ given the values $\mathbf{g}$ at $\mathbf{x}$ is:

$$\begin{aligned} \mathbf{g}^* | \mathbf{x}, \mathbf{g}, \mathbf{x}^* &\sim \mathcal{N}\left(\tilde{\mathbf{g}}^*, \tilde{\mathbf{K}}\right), \text{ where} \\ \tilde{\mathbf{g}}^* &= \mathbf{K}\left(\mathbf{x}^*, \mathbf{x}\right) \left[\mathbf{K}(\mathbf{x}, \mathbf{x})\right]^{-1} \mathbf{y} \\ \tilde{\mathbf{K}} &= \mathbf{K}\left(\mathbf{x}^*, \mathbf{x}^*\right) - \mathbf{K}\left(\mathbf{x}^*, \mathbf{x}\right) \left[\mathbf{K}(\mathbf{x}, \mathbf{x})\right]^{-1} \mathbf{K}\left(\mathbf{x}, \mathbf{x}^*\right). \end{aligned} \tag{17}$$

The marginal distribution can be recovered by finding the relevant part of the covariance matrix; for example, the marginal of $\mathbf{g}$ given $\mathbf{x}$ is $\mathbf{g} \sim \mathcal{N}\left(0, \mathbf{K}_\psi(\mathbf{x}, \mathbf{x})\right)$.

In this work, we place a Gaussian process prior on $\kappa(x)$ and assume the squared exponential kernel with length scale ℓ_κ and fixed variance $\sigma_\kappa^2 = 1$. As mentioned in the previous section, we assume that $\kappa(x)$ is constant on each element of the mesh (we use the same mesh as for discretizing $u(x)$ and $f(x)$). We place the prior on $\boldsymbol{\kappa}$ so that the centroids of the elements are the training points of the GP:

$$p(\boldsymbol{\kappa}) \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_\psi(\mathbf{x}, \mathbf{x})). \tag{18}$$

3. Variational Bayes Approximation

3.1. Variational Bayes

We define variational Bayes by following the discretize-first approach to inference. We assume that the hyper-parameters ψ are fixed, and we are only interested in the posterior distribution of $\boldsymbol{\kappa}$. The variational approach proceeds by approximating the true posterior with a variational density $q(\boldsymbol{\kappa})$, which is the minimizer of the discrepancy between a chosen family of densities $\mathcal{D}_q$ and the true posterior distribution $p(\boldsymbol{\kappa}|\mathbf{y})$ (Jordan et al., 1999; Jordan and Wainwright, 2007). A typical choice for the measure of discrepancy between distributions is the Kullback-Leibler (KL) divergence (which due to the lack of symmetry is not a metric). To find the approximate posterior distribution we have:

$$q^*(\boldsymbol{\kappa}) = \underset{q(\boldsymbol{\kappa}) \in \mathcal{D}_q}{\arg\min} \ \mathrm{KL}(q(\boldsymbol{\kappa}) \parallel p(\boldsymbol{\kappa} \mid \mathbf{y})). \tag{19}$$

Expanding the KL divergence term we obtain

$$\begin{aligned} \mathrm{KL}(q(\boldsymbol{\kappa}) \parallel p(\boldsymbol{\kappa} \mid \mathbf{y})) &= \int q(\boldsymbol{\kappa}) \log \frac{q(\boldsymbol{\kappa})}{p(\boldsymbol{\kappa} \mid \mathbf{y})} \mathrm{d}(\boldsymbol{\kappa}) \\ &= \mathbb{E}_q\left[\log q(\boldsymbol{\kappa})\right] - \mathbb{E}_q\left[\log p(\boldsymbol{\kappa} \mid \mathbf{y})\right] \\ &= \mathbb{E}_q\left[\log q(\boldsymbol{\kappa})\right] - \mathbb{E}_q\left[\log \frac{p(\mathbf{y}, \boldsymbol{\kappa})}{p(\mathbf{y})}\right] \\ &= \mathbb{E}_q\left[\log q(\boldsymbol{\kappa})\right] - \mathbb{E}_q\left[\log p(\mathbf{y}, \boldsymbol{\kappa})\right] + \log p(\mathbf{y}) \end{aligned} \tag{20}$$

The last term of the KL divergence is usually intractable. However, we use the fact that the KL divergence is non-negative to obtain

$$\log p(\mathbf{y}) \geq \mathbb{E}_q \big[\log p(\mathbf{y}, \boldsymbol{\kappa}) \big] - \mathbb{E}_q \big[\log q(\boldsymbol{\kappa}) \big]. \tag{21}$$

To minimize the KL divergence, it is enough to maximize $\mathbb{E}_q \big[\log p(\mathbf{y}, \boldsymbol{\kappa}) \big] - \mathbb{E}_q \big[\log q(\boldsymbol{\kappa}) \big]$, which is commonly referred to as the evidence lower bound (ELBO). The ELBO term can be further expanded as follows:

$$\begin{aligned} \mathrm{ELBO}(q) &= \mathbb{E}_q \big[\log p(\mathbf{y} \mid \boldsymbol{\kappa}) + \log p(\boldsymbol{\kappa}) \big] - \mathbb{E}_q \big[\log q(\boldsymbol{\kappa}) \big] \\ &= \mathbb{E}_q \big[\log p(\mathbf{y} \mid \boldsymbol{\kappa}) \big] - \mathrm{KL}(q(\boldsymbol{\kappa}) \parallel p(\boldsymbol{\kappa})). \end{aligned} \tag{22}$$

To summarize, the task now becomes:

$$\underset{q \in \mathcal{D}_q}{\arg\max} \; \mathbb{E}_q \big[\log p(\mathbf{y} \mid \boldsymbol{\kappa}) \big] - \mathrm{KL}(q(\boldsymbol{\kappa}) \parallel p(\boldsymbol{\kappa})). \tag{23}$$

To effectively maximize the ELBO, we need to be able to evaluate it and also have access to its gradients with respect to the parameters of $q(\boldsymbol{\kappa})$. Although the KL divergence term of the ELBO is often available in closed form, $\mathbb{E}_q \big[\log p(\mathbf{y} \mid \boldsymbol{\kappa}) \big]$ is generally not available. This term can be approximated using a Monte Carlo approximation with N_{SVI} samples from $q(\boldsymbol{\kappa})$ as follows

$$\mathbb{E}_q \big[\log p(\mathbf{y} \mid \boldsymbol{\kappa}) \big] \approx N_{\mathrm{SVI}}^{-1} \sum_{i=1}^{N_{\mathrm{SVI}}} \log p(\mathbf{y} \mid \boldsymbol{\kappa}^{(i)}), \tag{24}$$

where $\boldsymbol{\kappa}^{(i)}$ is the i-th sample from $q(\boldsymbol{\kappa})$. The value of N_{SVI} can be as low as 1 (Kingma and Welling, 2014), but values in the range of 2–5 will provide faster convergence of the optimisation. This approach is often referred to as stochastic variational inference (SVI). The Monte Carlo approximation is in line with the work in Barajas-Solano and Tartakovsky (2019) but in contrast with the analytic approximation based on the Hessian calculations proposed in Tsilifis et al. (2016).

3.1.1. Guarantees on accuracy

The main concern related to VB stems from the fact that it is constrained by the chosen family of variational distributions, which may not approximate the true posterior distribution well. If the choice of variational family is too restrictive, the estimate of the posterior mean is biased while the uncertainty may be underestimated (MacKay, 2003; Wang and Titterington, 2004; Turner and Sahani, 2011). Furthermore, as noted in previous work, the commonly used mean-field factorization of the variational distributions does not come with general guarantees on accuracy (Giordano et al., 2018). However, VB has been demonstrated to work well in practice in a variety of settings (Kingma and Welling, 2014; Damianou et al., 2016; Blei et al., 2017; Zhang et al., 2019). Recent work on VB has provided some tools for assessing the robustness of the VB estimates. Giordano et al. (2018) present some results on covariance estimates and local robustness measures for mean-field VB which apply to the inverse problem setting given some injectivity assumptions on the inverse mapping, though further work is needed to formalize this. Huggins et al. (2020) further investigate the validity of VB estimates and provide some tools for post-hoc accuracy measures for the posterior point and uncertainty estimates.

3.2. Specification of variational distribution

The specification of the approximating family of distributions determines how much structure of the true posterior distribution is captured by the variational approximation. To model complex relationships between the components of the posterior, a more complex approximating family of distributions is needed. As the richer family of distributions is likely to require more parameters, the optimization task becomes harder. A balance must be struck in this trade-off: the family should be rich enough, but the optimization task should still be computationally tractable.

A practical and widely used variational family is the multivariate Gaussian distribution, parameterized by the mean vector and the covariance matrix. One of the key benefits of this choice is that the KL divergence term of the ELBO in (22) is available in closed form for a GP prior. The choice of the parametrization of the covariance matrix determines how much structure, other than the mean estimate, is captured by the variational family. We discuss this in more detail in the next section.

Numerous approaches have been proposed to extend the variational distribution beyond the Gaussian family. A standard approach in situations when the true posterior distribution is likely to be multimodal is to consider mixtures of variational densities (Bishop et al., 1998). A more recent development is embedding parameters of a mean-field approximation in a hierarchical model to induce variational dependencies between latent variables (Tran et al., 2015; Ranganath et al., 2016). Also recently, an implicit way of approximating the posterior distributions through the propagation of particles has been proposed (Liu and Wang, 2016). Particles are updated using a specific smooth transform that corresponds to the steepest descent direction of the KL divergence. This approach is very flexible and can account for properties such as multi-modality, but it scales poorly with the dimensionality of the problem, and thus various annealing schemes have been proposed to encourage the exploration behavior of the algorithm. However, these methods are still too restrictive in terms of their ability to scale to high-dimensional problems (D'Angelo and Fortuin, 2021).

3.2.1. Gaussian variational distribution

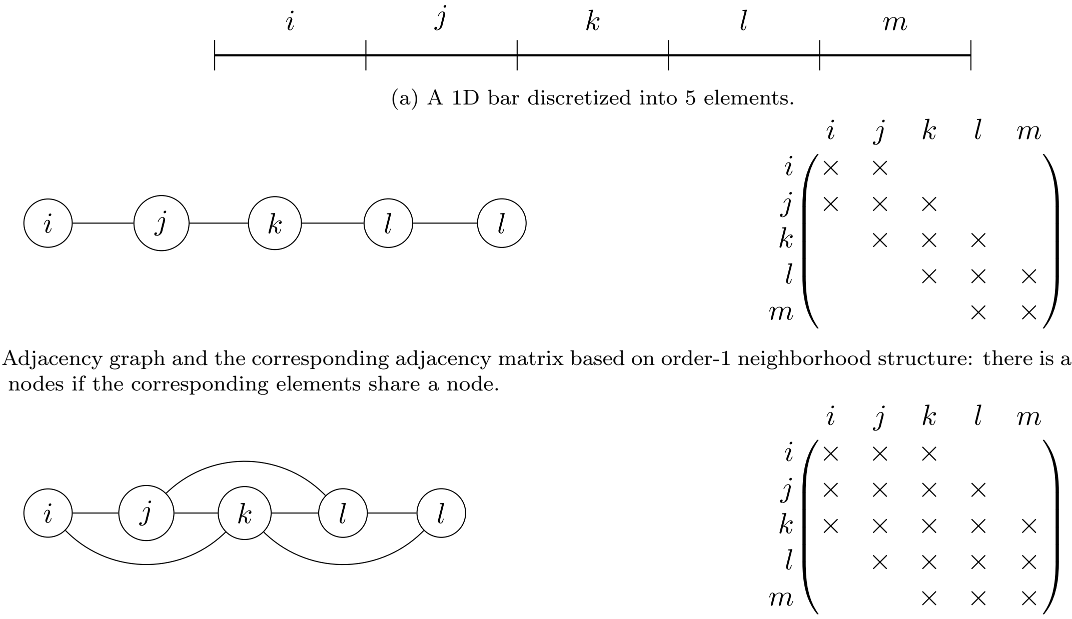
Choosing the variational distribution $q(\boldsymbol{\kappa})$ as a multivariate Gaussian, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, requires optimization over the mean $\boldsymbol{\mu}$ (free-form) and the covariance matrix $\boldsymbol{\Sigma}$. The flexibility in choosing how we specify both of these parameters, especially the covariance matrix, enables us to balance the trade-off between the expressiveness of the approximating distribution and the computational efficiency.

The richest specification corresponds to parametrizing the covariance matrix, $\boldsymbol{\Sigma}$, using its full Cholesky factor $\boldsymbol{L}$:

$$q(\boldsymbol{\kappa}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{L}\boldsymbol{L}^\top). \tag{25}$$

This choice results in a dense covariance matrix that may be able to capture the full covariance structure between the inputs (*i.e.* each input may be correlated with every other input). Parametrizing the components of $\mathbf{L}$ instead of $\boldsymbol{\Sigma}$ automatically ensures that $\boldsymbol{\Sigma}$ is positive definite as necessary. The number of parameters to optimize grows as $\mathcal{O}(n_{\boldsymbol{\kappa}}^2)$ and this leads to a difficult optimization task that needs to be carefully initialized and parameterized. We refer to this parametrization as full-covariance variational Bayes (FCVB).

A much more efficient choice is a diagonal covariance matrix, which is often referred to as mean-field variational Bayes (MFVB). By limiting the number of parameters that need to be optimized, the optimization task becomes simpler and the number of parameters grows only as $\mathcal{O}(n_{\boldsymbol{\kappa}})$. While more computationally efficient and easier to initialize, MFVB ignores much of the dependence structure of the posterior distribution.



(a) A 1D bar discretized into 5 elements.

(b) Adjacency graph and the corresponding adjacency matrix based on order-1 neighborhood structure: there is an edge between two nodes if the corresponding elements share a node.

(c) Adjacency graph and the corresponding adjacency matrix based on order-2 neighborhood structure: there is an edge between two nodes if the corresponding elements share a node or if there is another node which is a neighbor of both nodes.

Figure 1: An example of a 1D bar discretized into 5 elements and two different ways of assuming conditional independence.

3.3. Conditional Independence, Sparse Precision Matrices

Instead of parametrizing the covariance matrix, $\boldsymbol{\Sigma}$, one could parameterize the precision matrix, $\boldsymbol{Q}$, where $\boldsymbol{Q} = \boldsymbol{\Sigma}^{-1}$.

While an element of $\boldsymbol{\Sigma}$ expresses *marginal* dependence between the two corresponding random variables, the elements of the precision matrix reflect the *conditional independence* relationship in the approximating distribution. For a pair of indices i, j, $\boldsymbol{Q}_{i,j} = 0$ if and only if $\boldsymbol{\kappa}_i$ is independent from $\boldsymbol{\kappa}_j$, *conditional* on all other variables, $\{\boldsymbol{\kappa}_k : k \notin \{i, j\}\}$ (Rue et al., 2009). Note that $\boldsymbol{\kappa}_i$ and $\boldsymbol{\kappa}_j$ are still likely to be *marginally* dependent. As a consequence, the presence of conditional independence among random variables will lead to a sparse precision matrix.

A common way to represent conditional independence is using an undirected graph whose nodes correspond to the random variables. An edge is present between two nodes i and j if the corresponding random variables are *not* conditionally independent from each other, given all the other random variables. In our context, we take advantage of the structure of our problem to decide which random variables (elements of $\boldsymbol{\kappa}$) are conditionally independent. We use the adjacency structure imposed by the choice of the discretization of the problem. We add an edge in the adjacency graph if two vertices correspond to immediate or second-order neighbors on the discretization mesh. The edges between the vertices of the graph correspond to non-zeros in the precision matrix. Fig. 1 shows an example of an adjacency graph and the structure of the corresponding precision matrix, $\boldsymbol{Q}$, for 5 random variables resulting from a discretization of a 1D bar. After obtaining the precision matrix, which is sparse, one can reorder the random variables to reduce the bandwidth; this allows for efficient linear algebra operations. See Cuthill and McKee (1969) for an example of a reordering algorithm. Once the minimum bandwidth (b_{min}) ordering has been established, we use the property that the bandwidth of the Cholesky factor $\boldsymbol{L}_{\boldsymbol{Q}}$ of matrix $\boldsymbol{Q}$ is less than or equal to the bandwidth of $\boldsymbol{Q}$ (Rue and Held, 2005, Theorem 2.9). Finally, the parameters we optimize are the components

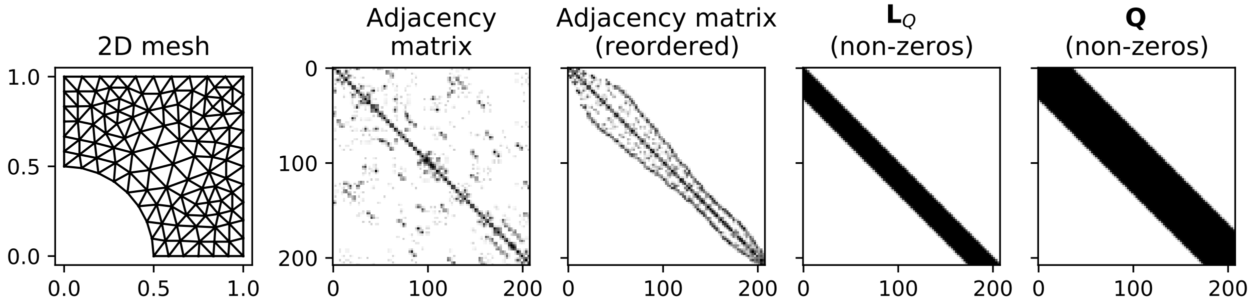


Figure 2: An illustration of how sparse precision matrix parametrization in Eq. (26) is derived. The adjacency matrix encodes the second-order neighborhood structure of the elements in the mesh. By reordering the nodes one can obtain a banded adjacency matrix, which is then used to parameterize the Cholesky factor of the precision matrix, as described in section 3.2.1.

of the lower band of size $b_{\min}$ of matrix $\boldsymbol{L}_Q$, so that the approximating distribution reads

$$q(\boldsymbol{\kappa}) \sim \mathcal{N}\Big(\boldsymbol{\mu}, (\boldsymbol{L}_Q \boldsymbol{L}_Q^\top)^{-1}\Big). \tag{26}$$

This process of devising a parametrization for the precision matrix for a more complex mesh in 2D is shown in Fig. 2. This approach is computationally efficient – the number of parameters grows as $\mathcal{O}(n_{\boldsymbol{\kappa}})$ – but also captures dependencies between the random variables.

3.4. Stochastic optimization

To optimize our variational objective, we use the ADAM algorithm (Kingma and Ba, 2015). ADAM is a member of a larger class of stochastic optimization methods that have become popular as tools for minimizing intractable cost functions. These methods construct a stochastic estimate of the gradient to perform gradient descent-based optimization. In the most basic form, the simplest method is stochastic gradient descent (SGD) which uses an unbiased gradient to perform standard gradient descent (Robbins and Monro, 1951). Due to their usability in the stochastic optimization of neural networks, there has been significant advances in the SGD methodology, utilizing momentum, preconditioning, and various other components to accelerate the convergence of the optimization procedure. ADAM, a stochastic gradient descent algorithm with an adaptive step size is one popular algorithm that exhibits a stable behavior on many problems and is easy to use (e.g. no significant tuning required).

Each sample in the MC estimation in (24) requires drawing a sample for $\boldsymbol{\kappa}$ from the variational distribution and solving the linear system associated with the FEM as derived in (9). For variational approximations, we use adaptive step size through a rule that tracks an exponentially weighted moving average of the decrease in the loss values between successive steps, and stops when that average drops below a threshold. The use of such an adaptive rule gives us a way to track the convergence of the algorithm and provides a conservative estimate for the time it takes for the optimization to converge. This rule can be adapted based on the available computational budget.

3.5. The algorithm

The maximization of the ELBO in (22) involves finding the parameters of $q(\boldsymbol{\kappa})$ that minimize KL between $q(\boldsymbol{\kappa})$ and $p(\boldsymbol{\kappa}|\mathbf{y})$. Algorithm 1 shows the required steps to compute the ELBO and its gradients with respect to the parameters of the variational distribution. This subroutine is called by the optimization algorithm.

Algorithm 1: ELBO estimation and its gradient with respect to the parameters of the variational distribution.

```
Input: Current parameters of $q(\boldsymbol{\kappa})$
Output: ELBO and its gradients with respect to the parameters of $q(\boldsymbol{\kappa})$
Sample $[\boldsymbol{\kappa}^{(1)}, \boldsymbol{\kappa}^{(2)}, \dots, \boldsymbol{\kappa}^{(N_{\text{SVI}})}]$ from $q(\boldsymbol{\kappa})$
for each $\boldsymbol{\kappa}^{(i)}$ do
    Solve $\mathbf{u}(\boldsymbol{\kappa}^{(i)})$ and obtain gradients with respect to $\boldsymbol{\kappa}$ using an FEM solver
    $p(\mathbf{y} \mid \boldsymbol{\kappa}^{(i)}) \leftarrow \prod_{i=1}^{N_{\mathbf{y}}} p(\mathbf{y}_i; \mathbf{u}(\boldsymbol{\kappa}^{(i)}), \sigma_y^2)$ and propagate the gradient with respect to $\boldsymbol{\kappa}^{(i)}$ (using chain rule)
ELBO $\leftarrow N_{\text{SVI}}^{-1} \sum_{i=1}^{N_{\text{SVI}}} \log p(\mathbf{y} \mid \boldsymbol{\kappa}^{(i)}) + \text{KL}(q(\boldsymbol{\kappa}) \parallel p(\boldsymbol{\kappa}))$ and propagate the gradient with respect the parameters of $q(\boldsymbol{\kappa})$ using the reparametrization trick (see section Appendix A.1 and Kingma and Welling (2014))
return ELBO, $\nabla$ELBO
```

4. Examples

We evaluate the efficacy of variational inference on (a) a 1D and 2D Poisson equation examples and (b) a benchmark proposed by Aristoff and Bangerth (2021). We discretize the examples with a standard finite element method using linear Lagrange basis functions as described in the previous section. We compare against two sampling-based inference schemes, Hamiltonian Monte Carlo (HMC) and preconditioned Crank-Nicholson Markov Chain Monte Carlo (pCN) ; these are known to be asymptotically correct as the number of samples increases. The evaluation criteria we use focus on three aspects of the inference scheme, the accuracy with respect to capturing the mean of the solution, the variance of the solution, propagation of uncertainty in derived quantities of interest, and the time until convergence of the solution.

4.1. Quantity of Interest

To assess the propagation of uncertainty, we consider a summary quantity for which a point estimate alone may not be informative enough for downstream tasks. In particular, we compute the log of total boundary flux through the boundary Γ_b:

$$r(\kappa) = \log \int_{\Gamma_b} e^{\kappa(s)} \nabla u(s) \cdot \boldsymbol{n} \, \mathrm{d}s, \tag{27}$$

where $\boldsymbol{n}$ is a unit vector normal to the boundary Γ_b. The quantity $\nabla u(s) \cdot \mathbf{n}$ represents the magnitude of the change in $u(s)$ at the boundary point s. For example, if u represented the temperature of an object, it would represent the loss of energy flowing out of the object.

4.2. Evaluation Criteria

To assess the inference of $\boldsymbol{\kappa}$, we obtain S samples from the posterior distribution of $\boldsymbol{\kappa}$, $\{\boldsymbol{\kappa}^{(s)}\}_{s=1}^{S}$. For synthetic experiments, where we know the true $\boldsymbol{\kappa}$ which generated the observations, we compute two different metrics.

Firstly, the *expected error norm* computes the average Euclidean norm of the error:

$$\text{Expected error} = \frac{1}{S} \sum_{s=1}^{S} \|\boldsymbol{\kappa}^{(s)} - \boldsymbol{\kappa}\|_2. \tag{28}$$

Next, we use an approach that attempts to penalize over-confidence or too uncertain predictions– *scoring rules*. A scoring rule is a function of the probability distribution of the prediction and the realized (true) value. The computed score can subsequently be used to rank different models. For a thorough review of scoring rules, we refer the reader to Gneiting and Raftery (2007). Throughout this work, we use the energy scoring rule, which is defined as follows:

$$\text{Energy score} = \frac{1}{S}\sum_{s=1}^{S} \|\boldsymbol{\kappa}^{(s)} - \boldsymbol{\kappa}\|_{\alpha}^{\beta} - \frac{1}{2S^2}\sum_{i=1}^{S}\sum_{j=1}^{S} \|\boldsymbol{\kappa}^{(i)} - \boldsymbol{\kappa}^{(j)}\|_{\alpha}^{\beta}, \tag{29}$$

where $\|\boldsymbol{\kappa}\|_{\alpha} = \left(\sum_{i=1}^{M} |\kappa_i|^{\alpha}\right)^{1/\alpha}$, $\alpha \in (0, 2]$ $\beta \in (0, \alpha]$. Throughout this work, we use $\alpha = \beta = 1$.

4.3. Poisson 1D

For this experiment, we assume the unit-line domain, which is discretized into 32 equal-length elements. We impose Dirichlet boundary conditions on both boundaries, specifically we set $u(0) = u(1) = 0$; we set the forcing term to be constant everywhere $f(x) = 1$. The quantitative results for the 1D Poisson equation are shown in Fig. 3. Unless specified otherwise, all experiments in this section use $n = 5$ observations per sensor and the sensor noise $\sigma_y = 0.01$. Sensors are located on each of the discretization nodes. We compare the results for three specifications of the prior length scale, $\ell_\kappa = 0.1, 0.2, 0.3$, and two choices of the true length scale, $\ell_\kappa = 0.1, 0.2$.

4.3.1. Mean of variational posterior provides a good point estimate

Fig. 3 shows the score and the expected norm of the posterior mean estimates from Hamiltonian Monte Carlo (HMC), pre-conditioned Crank-Nicholson MCMC (pCN), as well as VB inference with different parametrization of the covariance/precision matrix. It is evident that for all choices of the prior kernel length scale, the posterior means computed by all the variational inference methods are very close to the estimates from HMC and pCN. The small prior lengthscale ($\ell_\kappa = 0.1$) case shows better performance for the MFVB method. One possible explanation is that the short prior length scale induces less spatial dependence in the posterior distribution, thus favoring methods that do not capture those dependencies, such as MFVB.

Among the different parametrization of VB, the posterior mean corresponding to the full covariance typically has a larger expected error suggesting a harder optimization problem for such a parametrization due to more parameters to be estimated and hence a slightly inferior posterior mean estimate in comparison to MFVB. However, a full-covariance parametrization allows us to better capture posterior variance, as detailed in the following sections.

4.3.2. VI adequately estimates posterior variance

Fig. 4 shows the true values of $\boldsymbol{\kappa}$ (red), the posterior means (black) and the corresponding estimate of the posterior variance (blue shaded regions) estimated by HMC and variational inference with mean-field (MFVB), full covariance (FCVB), and precision matrix (PMVB) parametrization for different values of true and prior length scales. We observe that the posterior variance estimates computed by HMC and full covariance VB are qualitatively very similar, with the estimated uncertainty increasing with increasing distance from the fixed boundary. However, the mean-field solution greatly underestimates posterior variance while computing a reasonable estimate of the posterior mean. For the PMVB parametrization, the uncertainty is underestimated to a much lesser extent.

The observations above further confirmed by the density plot of our quantity of interest: the log of the total flux on the boundary, shown in Fig. 5. For this example, we compute the flux on the left boundary at $x = 0$ and show the posterior distribution of this quantity. For longer prior

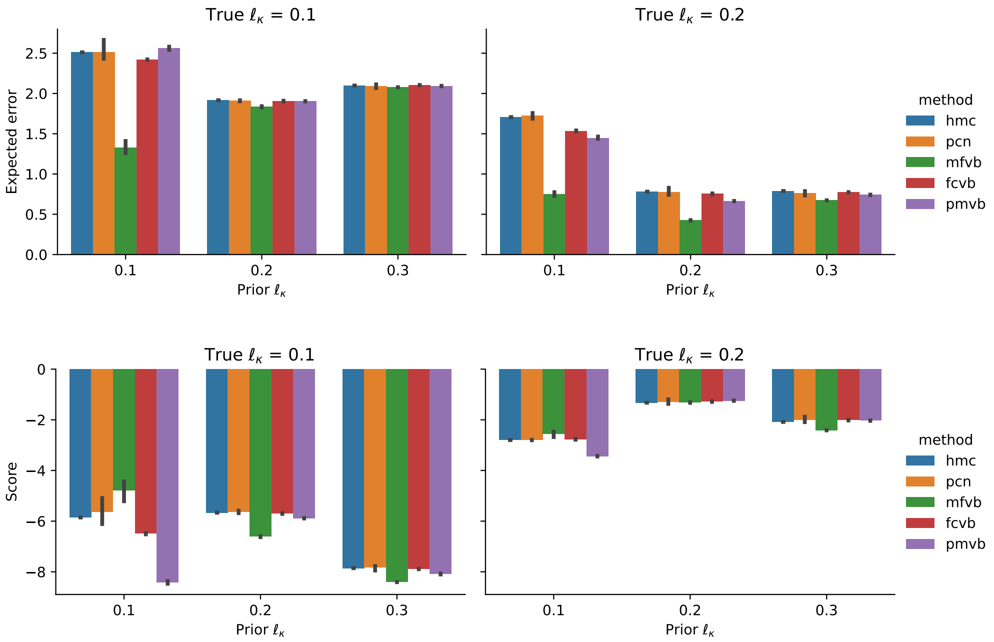


Figure 3: Expected error and energy score for the Poisson 1D problem, estimated using 10,000 samples from the inferred posterior distribution.

length scales, FCVB and PMVB agree with the estimates obtained from pCN and HMC, whereas mean-field VB underestimates the uncertainty. For the short prior length scale ($\ell_\kappa = 0.1$), both PMVB and MFVB underestimate the uncertainty as compared with HMC and pCN schemes. The posterior distribution of FCVB approximately agrees with the MCMC schemes.

For the results obtained using the PMVB scheme, we used the order-10 neighborhood structure to define the adjacency matrix and the non-zero elements of the precision matrix, $\boldsymbol{Q}$ (see Sec. 3.3). The order of the neighborhood structure, which corresponds to the precision matrix bandwidth, determines how much dependence within $\boldsymbol{\kappa}$ is captured by the approximating posterior distribution. In Fig. 6, we show how the estimate of the mean and the variance of $\boldsymbol{\kappa}$ changes for different orders of neighborhood structure. As expected, with the increasing bandwidth, the posterior estimate of $\boldsymbol{\kappa}$ gets closer to the estimate of FCVB and HMC (show in Fig. 4). While there is a significant change in the uncertainty estimate when bandwidth is increase from 2 to 10, it less pronounced when increased from 10 to 20. For this reason, we choose the value of 10 for the PMVB parametrization in 1D.

4.3.3. VI estimates improve with more observations and decreasing observational noise

Fig. 7 shows the true values of $\boldsymbol{\kappa}$ (red), the posterior mean estimates (black), and the corresponding estimates of the posterior variance (blue shaded regions) obtained by variational inference with mean-field or full covariance for different numbers of observations (top panel) and different values of observational noise (bottom panel). We can see that MFVB underestimates the posterior variances and these estimates do not depend on the number of observations (top panel in Fig. 7) or the amount of observational noise (bottom panel in Fig. 7). However, the full covariance uncertainty estimates get narrower with increasing number of observations and with decreasing observational noise, which is a desirable behavior that should be exhibited by any consistent uncertainty estimation method. We can also see that the true solution is contained within the uncertainty bounds for all numbers of observations and noise levels for the full covariance parametrization. This is not the case for the mean-field VB, providing another indication of uncertainty underestimation for this

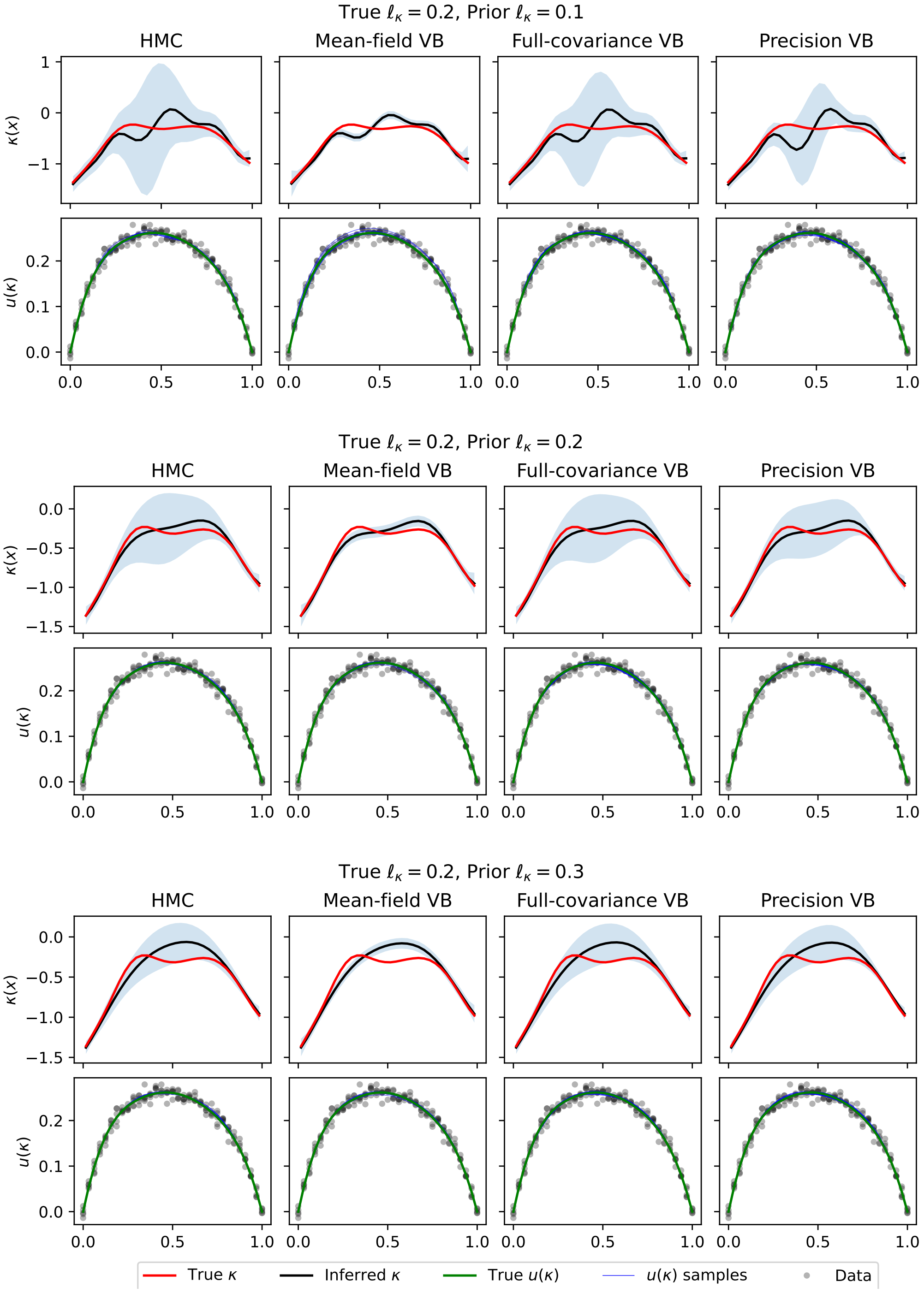


Figure 4: Top row in each of the three panels show true values of $\kappa(x)$ (red), posterior means (black) and posterior variances (blue shaded regions) for HMC and VB variants for different values of prior length scales ℓ_κ. The bottom rows show the data (black), true solution $\mathbf{u}$ (green), solutions for different samples of κ (blue). For the PMVB estimate, the bandwidth is set to 10.

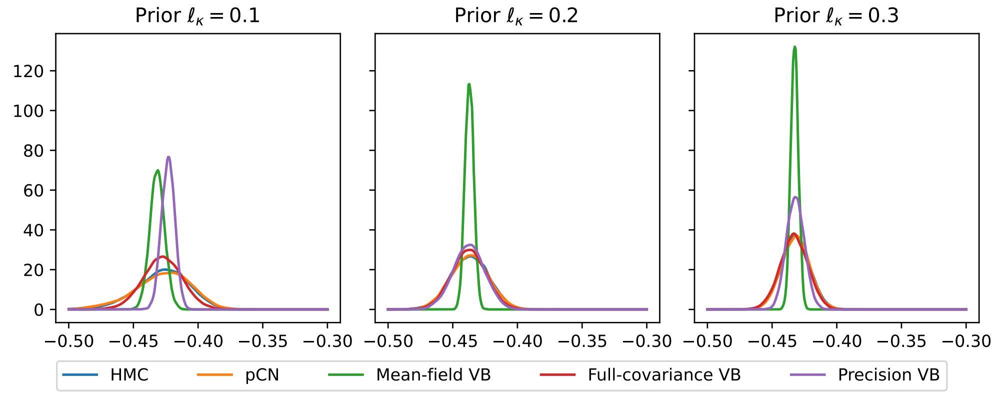


Figure 5: Log of the boundary flux at the left boundary node ($x = 0$) for the 1D Poisson example. For PMVB, the precision matrix bandwidth of 10 is used.

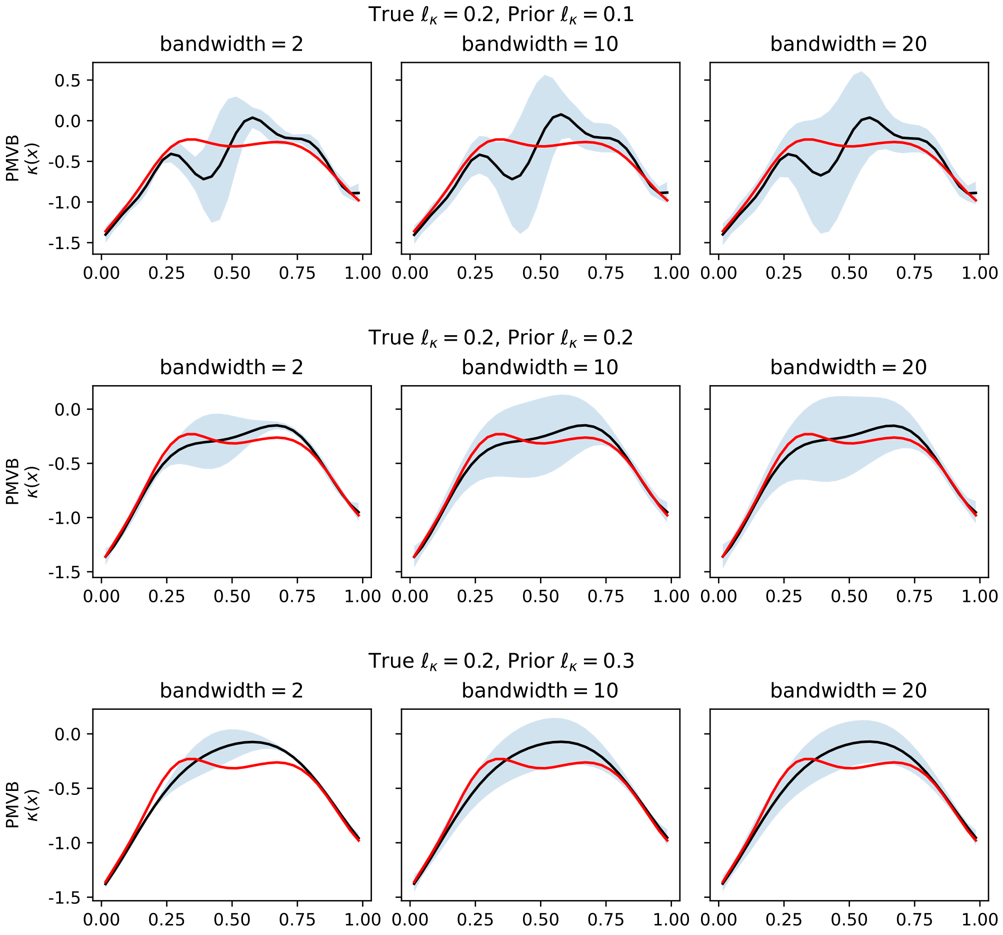


Figure 6: True values of κ (red), posterior means (black), and posterior variances (blue shaded region) for different matrix bandwidths of the precision matrix parametrization. Bandwidth corresponds to the order of neighborhood structure considered when parametrizing $\boldsymbol{Q}$.

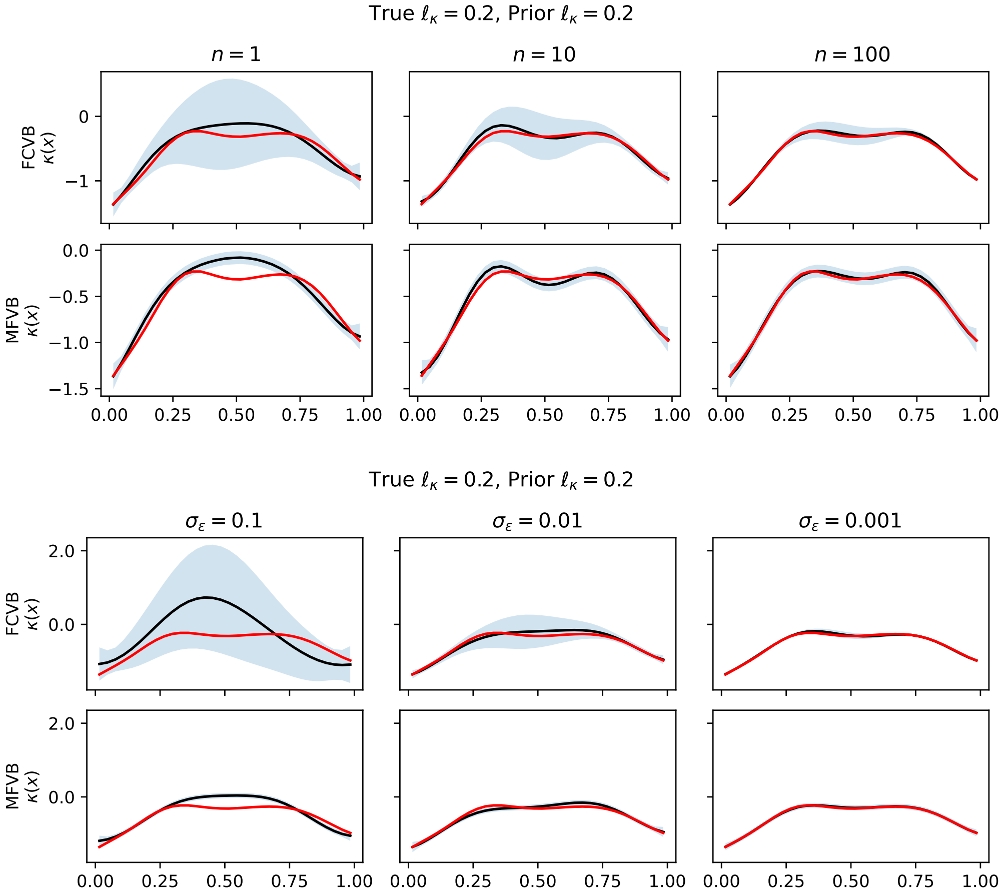


Figure 7: True values of $\kappa(x)$ (red), posterior means (black) and posterior variances (blue shaded regions) for VI with mean-field and full covariance parametrization for different number of observations per sensor, $n = 1, 10, 100$ (top panel), and for different values of sensor noise $\sigma_\epsilon = 0.1, 0.01, 0.001$ (bottom panel).

parametrization.

4.3.4. VI is an order of magnitude faster than HMC

For HMC estimates, we obtain 200,000 samples out of which the first 100,000 are used to calibrate the sampling scheme and are subsequently discarded. The VB algorithm follows as described in Sec. 3. Table 1 provides the run-times for HMC, MFVB, FCVB, and PMVB. For the HMC column we also report the range of effective sample sizes (ESS) across different components of $\boldsymbol{\kappa}$, show in brackets. For details on ESS, we refer the reader to (Gelman et al., 2013, Ch. 11). Even with a conservative convergence criteria (described in Sec. 3.4), the computational cost of VB algorithms is up to 25 times lower than that of HMC. To emphasize the computational efficiency of the variational inference, we show the posterior estimates for different number of Monte Carlo samples in the estimation of ELBO. Fig. 8 shows that on a qualitative level, a low number of samples is sufficient to obtain a good estimate. In particular, even with 2 Monte Carlo samples, the estimates are very similar to the case where $N_{\text{SVI}} = 20$. However, a lower number of samples may result in slower convergence of the optimization scheme. Fig. 9 shows that for the FCVB parametrization where the number of optimized parameters is large, increasing the number of SVI samples may speed up the convergence of the optimization. The effect is not as strong for the MFVB parametrization.

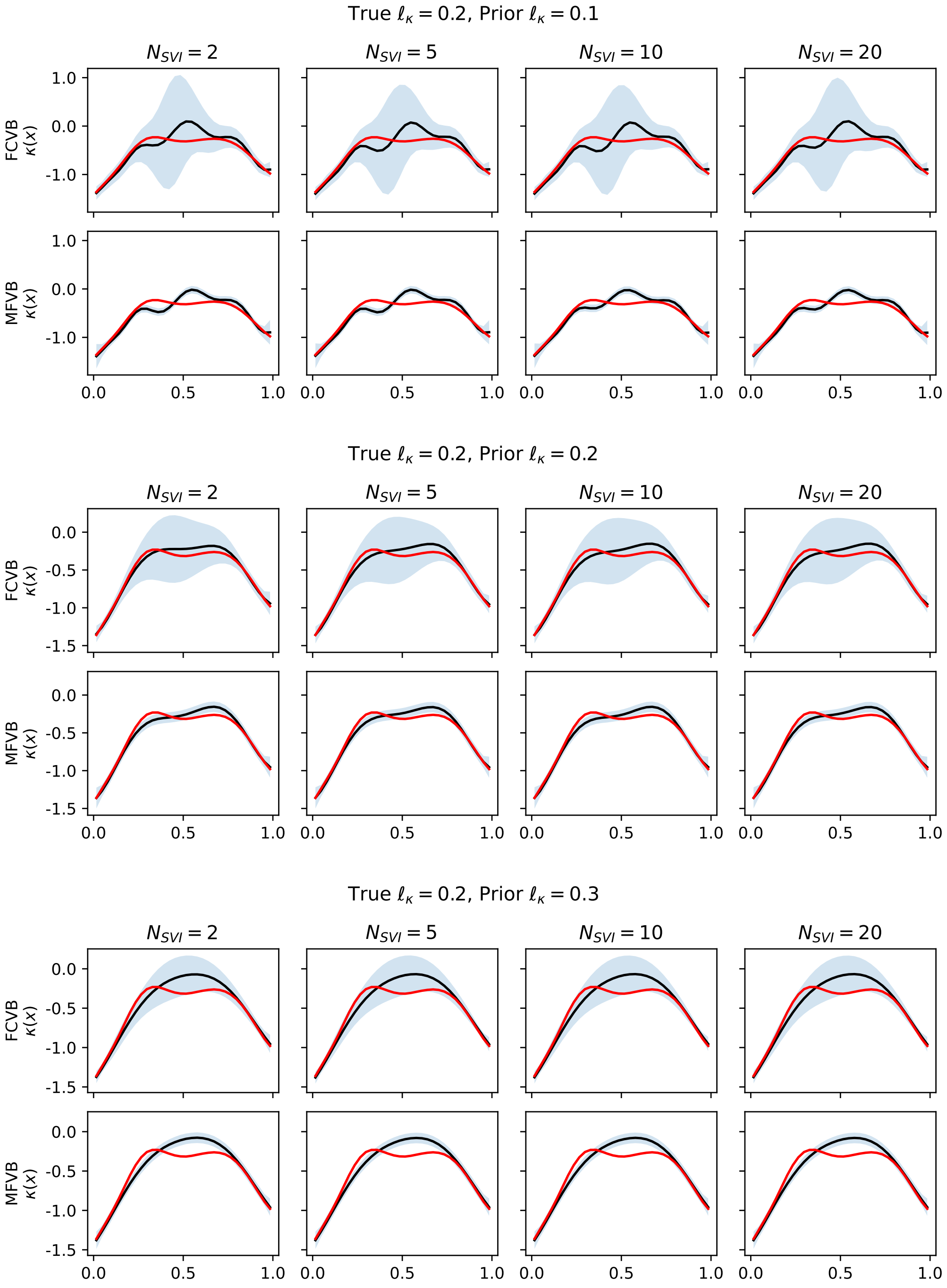


Figure 8: True values of $\kappa(x)$ (red), posterior means (black) and posterior variances (blue shaded regions) for VI with mean-field and full covariance parameterization for different number of Monte Carlo samples when computing ELBO. Three different length scales for the prior are shown: 0.1, 0.2, 0.3.

true ℓ_κ	prior ℓ_κ	Time (hours)				
		HMC		MFVB	FCVB	PMVB
0.1	0.1	15.2	(871–3244)	1.1	3.6	2.1
	0.2	11.1	(1043–4006)	0.7	2.7	2.1
	0.3	7.2	(1130–5408)	0.6	2.3	2.0
0.2	0.1	15.2	(1600–4700)	0.6	2.2	1.8
	0.2	10.4	(1067–3468)	0.6	2.3	2.0
	0.3	7.0	(1487–3969)	0.5	1.7	1.8

Table 1: Run-times for different inference schemes in hours for the Poisson 1D problem. For VB methods, $N_{\mathrm{SVI}} = 3$. The column for HMC includes the range of effective sample sizes (ESS) across different components of $\boldsymbol{\kappa}$.

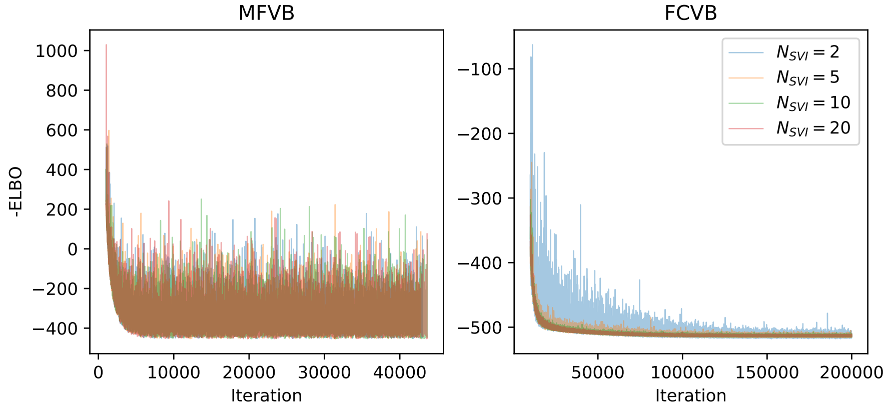


Figure 9: Negative ELBO trace plot for both MFVB and FCVB for different values of N_{SVI}. For this example, true $\ell_\kappa = 0.2$ and prior $\ell_\kappa = 0.1$.

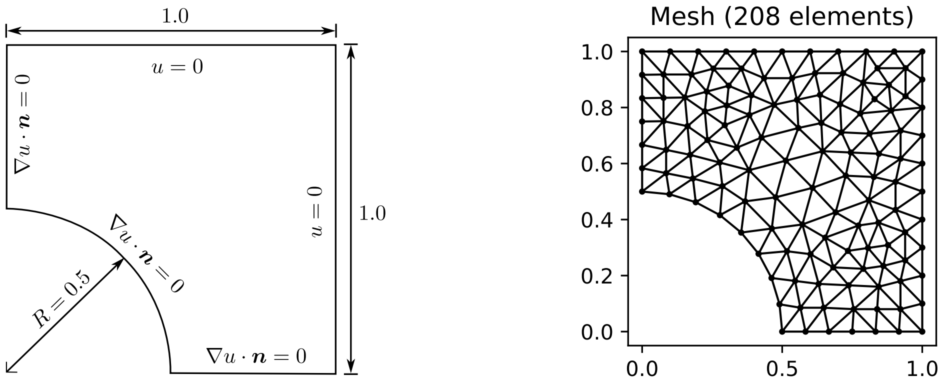

1.0
$u = 0$
$\nabla u \cdot \boldsymbol{n} = 0$
$\nabla u \cdot \boldsymbol{n} = 0$
$u = 0$
1.0
$R = 0.5$
$\nabla u \cdot \boldsymbol{n} = 0$
Mesh (208 elements)
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0

Figure 10: Left: Specification of the domain for the 2D Poisson problem. Note that we impose Dirichlet boundary conditions $u(x, y) = 0$ when $x = 1$ or $y = 1$. We impose Neumann boundary conditions on the rest of the boundary. Right: a triangular discretization of the domain.

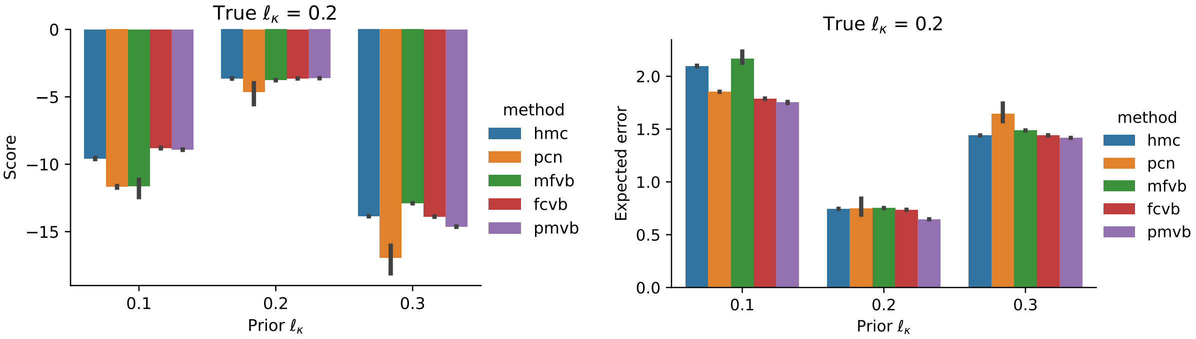


Figure 11: Expected error and the energy score for the Poisson equation 2D example.

4.4. Poisson 2D

We consider a 2D Poisson problem on the unit-square domain with a circular hole as shown in Fig. 10, with boundary conditions as indicated in the same figure. The weak form of the problem is discretized with the finite element mesh consisting of 208 standard linear triangular elements and 125 nodes. The forcing term is assumed to be constant throughout the domain, $f(x) = 1$. Unless specified otherwise, all experiments in this section use $n = 5$ observations per sensor and the sensor noise $\sigma_y = 0.001$ (note that for the 1D example we used $\sigma_y = 0.01$). The sensors are located at each node of the mesh.

Firstly, the results for the expected error norm in the right panel of Fig. 11 show that the mean estimate is very similar for all the considered methods. The results also show that the error is lowest when the prior ℓ_κ matches the length scale used to generate the data. Similarly, for the energy score (left panel), all methods show similar performance except pCN which exhibits higher variance and low values as compared to the other methods. This is due to poor mixing of the corresponding Markov chains. See Fig. B.18 in the appendix for the trace plot of one of the estimated coefficients.

Fig. 12-14 show the results for the posterior mean and the variance of $\boldsymbol{\kappa}$, and the solution $\mathbf{u}$ corresponding to the mean of the posterior. We consider three configurations with prior length scale $\ell_\kappa = 0.1, 0.2, 0.3$ where the true length scale is 0.2. In all cases, the estimates of the posterior mean of $\boldsymbol{\kappa}$ and the corresponding solutions $\mathbf{u}$ are very close to the true values. Similarly to the 1D case

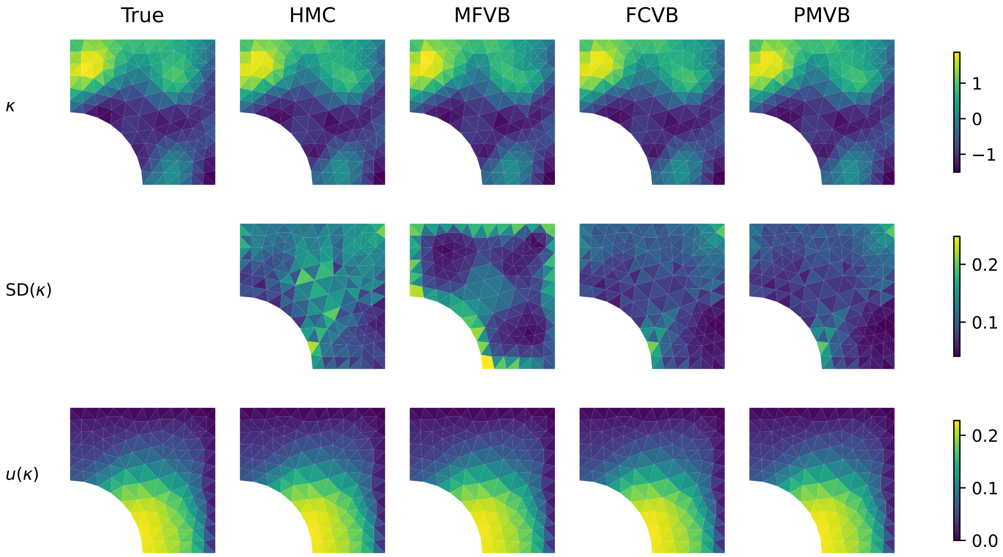


Figure 12: Posterior mean and variance for κ and corresponding $\mathbf{u}$ for 2D Poisson example with prior length scale $\ell_\kappa = 0.1$.

true ℓ_κ	prior ℓ_κ	Time (hours)				
		HMC		MFVB	FCVB	PMVB
0.1	0.1	240.6	(930–11200)	6.4	29.6	28.1
	0.2	295.5	(1537–11067)	6.6	32.6	28.9
	0.3	242.0	(1057–6068)	7.3	27.3	30.6
0.2	0.1	242.7	(1102–18235)	6.2	34.3	27.2
	0.2	264.3	(1304–9848)	7.4	33.7	34.0
	0.3	221.9	(1192–6356)	7.8	31.3	34.0

Table 2: Run-times for different inference schemes in seconds. The number of Monte Carlo samples is $N_{\mathrm{SVI}} = 5$ for all MFVB, FCVB, and PMVB. The column for HMC includes the range of effective sample sizes (ESS) across different components of $\boldsymbol{\kappa}$.

discussed in Sec. 4.3, the variance estimates between HMC and FCVB are consistent, especially for longer prior length scale. There seems to be a discrepancy between the estimates obtained using MFVB and those obtained by FCVB and HMC. The estimates obtained using precision-matrix parametrization are qualitatively very close to the FCVB and HMC estimates.

For the quantity of interest we compute the log of the total flux along the right boundary of the domain ($x = 1$), and the results are shown in Fig. 15. Unlike the 1D case, the posterior estimates of the boundary flux are approximately the same for all the considered methods, except for the mean-field estimate where the prior $\ell_\kappa = 0.1$, where the MFVB estimate is biased as compared to the other methods.

The empirical computational cost for these experiments is given in Table 2. For the HMC experiments, we obtained 250,000 samples out of which the first 125,000 were used to calibrate the sampling scheme and discarded afterwards. The timing results show that HMC takes an order of magnitude longer than variational Bayes, with some variation that depends on the parametrization.

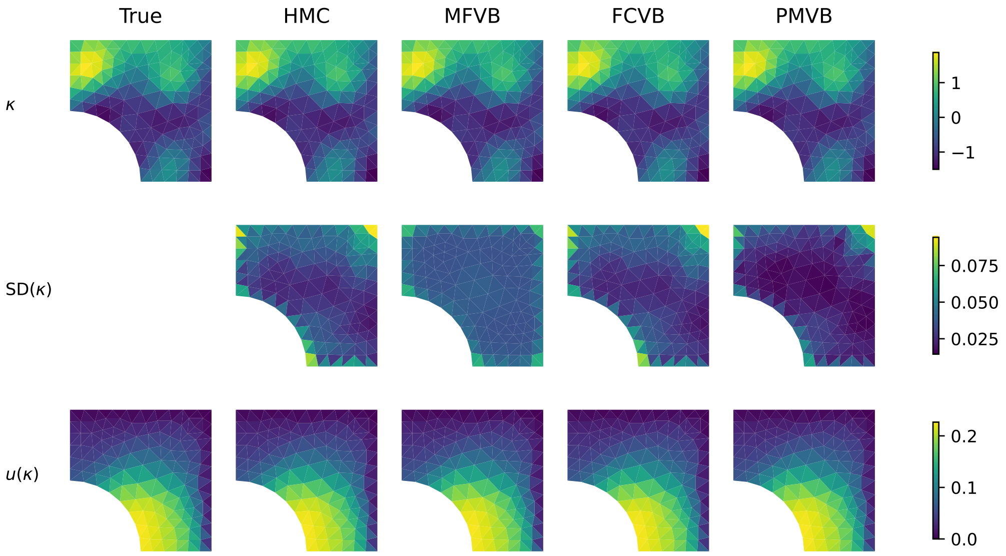


Figure 13: Posterior mean and variance for κ and corresponding $\mathbf{u}$ for 2D Poisson example with prior length scale $\ell_\kappa = 0.2$.

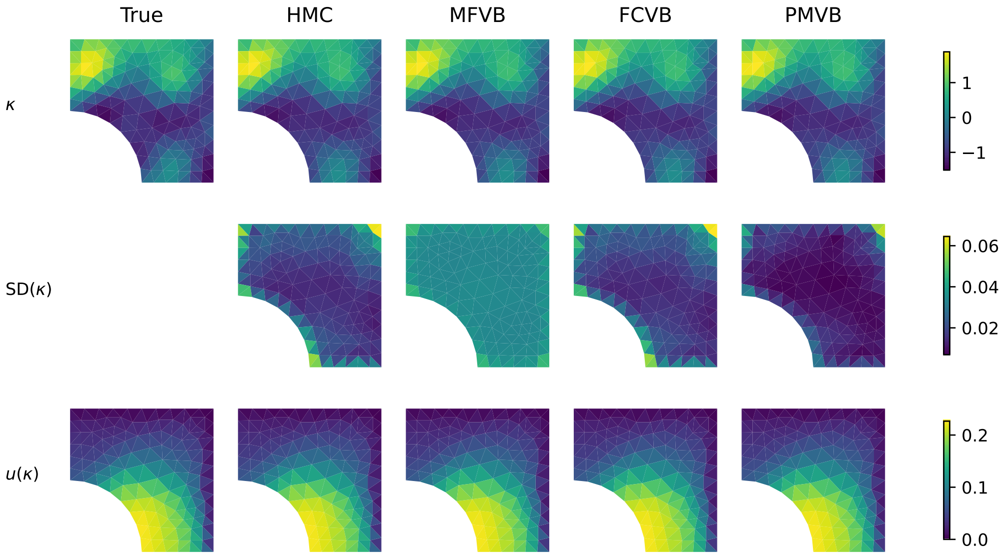


Figure 14: Posterior mean and variance for κ and corresponding $\mathbf{u}$ for 2D Poisson example with prior length scale $\ell_\kappa = 0.3$.

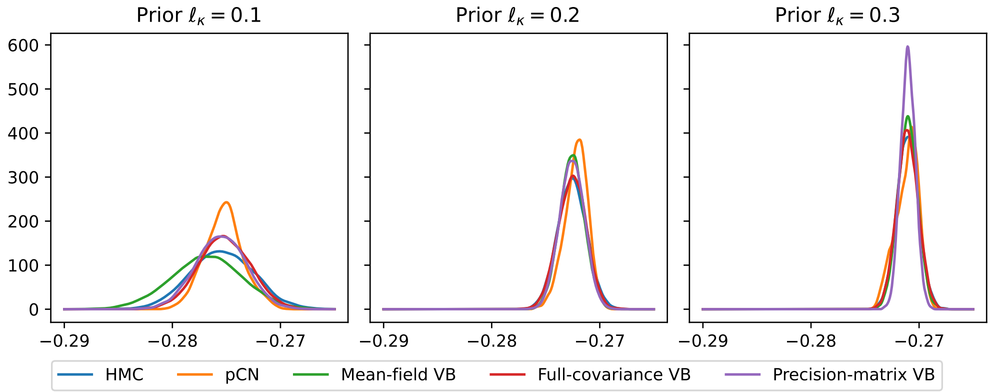


Figure 15: Log of the total flux computed along the right boundary ($x = 1$). For PMVB, the precision matrix is parametrized using the second-order neighborhood structure, as shown in Fig. 2.

4.5. Inverse Problem Benchmark

We evaluate the effectiveness of VB methods on a recently proposed benchmark for Bayesian inverse problems (Aristoff and Bangerth, 2021). The benchmark aims to provide a test case that reflects practical applications, but at the same time is easy to replicate. Like above, the test case is a Poisson equation inverse problem where the task is to recover log-diffusion, κ, from a finite set of noisy observations. The problem domain is a unit square, the forcing function $f(x) = 10$ is constant throughout the domain, and the solution of the PDE is imposed to be zero on all four boundaries.

The benchmark discretizes κ using 64 Q_1 bilinear elements, such that κ is constant for each individual element. The forward solution of the PDE is obtained after discretizing u using a 32×32 $Q1$ elements. The locations where the solution is observed are placed on a uniform grid of 169 points (13×13). The measurements are corrupted by the Gaussian noise with standard deviation $\sigma_y = 0.05$. The authors of the benchmark provide the measurements as well as the true log-diffusion coefficient, κ, which generated the observations. The true log-diffusion coefficient, shown in Fig. 16, is zero throughout the domain, except two regions, where the value is $\log(10)$ and $\log(0.1)$. It is these two jumps that make it a non-trivial test case.

Unlike in the previous examples, we place a prior on κ which does not induce any spatial correlation between any of the $\boldsymbol{\kappa}$ coefficients. The role of the prior is to express our belief about the ranges of the coefficients, rather than any dependencies. Although authors place $\mathcal{N}(\mu = 4, \sigma^2 = 4)$ for each component of $\boldsymbol{\kappa}$ independently, we choose $\mathcal{N}(\mu = 0, \sigma^2 = 1)$ as most of the coefficients of true $\boldsymbol{\kappa}$ are at the baseline level equal to zero, and the fact that the $\boldsymbol{\kappa}$ corresponds to the diffusion parameter on the log-scale, a priori we do not expect such high variance.

We performed the inference using HMC, MFVB, FCVB, and PMVB. The means and standard deviations of inferred log-diffusion coefficients, together with the PDE solutions corresponding to the inferred means, are shown in Fig. 16. The results suggest that the mean estimates of all three methods do capture the jumps and the overall structure of $\boldsymbol{\kappa}$. Specifically, the FCVB estimate of the mean of $\boldsymbol{\kappa}$ is closest to the true value. As for uncertainty quantification, the MFVB and PMVB estimates seem are closer to the HMC estimate (our assumed ground truth) than the FCVB estimate. The FCVB estimate seems to overestimate the uncertainty at a few locations. This is potentially due to being stuck in a local optimum during the optimization procedure, which for FCVB involves high-dimensional exploration.

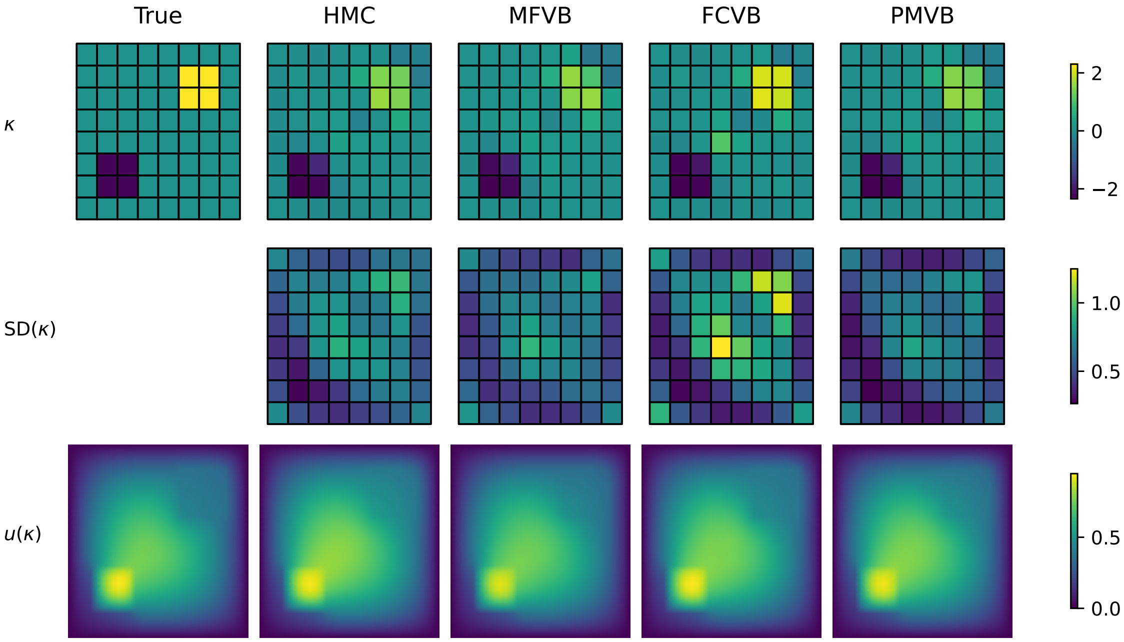


Figure 16: Independent prior for each coefficient of $\boldsymbol{\kappa}$: $\boldsymbol{\kappa}_i \sim \mathcal{N}(0, 1)$

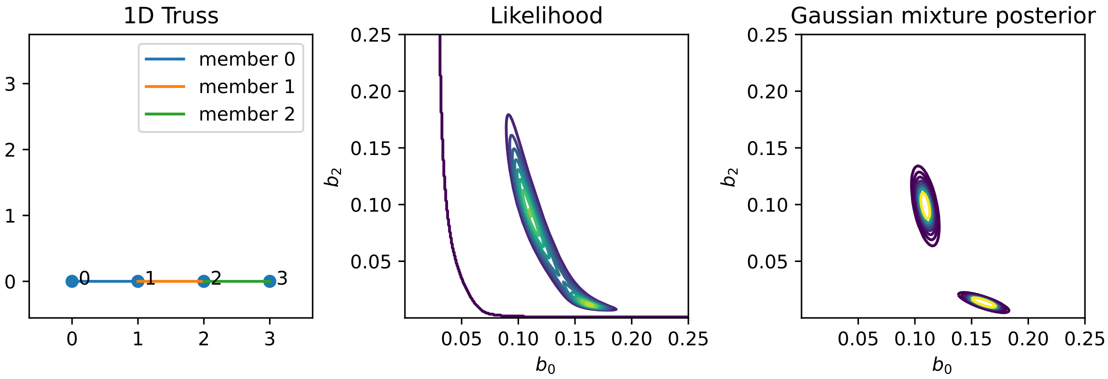


Figure 17: One-dimensional truss discretized using three elements (shown in the left panel). The middle panel shows the likelihood surface for varying stiffness of the first and the third element. The right panel shows the posterior inference of the stiffness after displacements have been observed.

4.6. Bimodal example

One of the advantages of VB over Laplace approximation is the flexibility of the approximating distribution. To illustrate this, we consider an example with a one-dimensional truss which is fixed at one node and contains three degrees of freedom that correspond to the horizontal motion of the three nodes as shown in the left panel of Fig. 17. We assume that the stiffness b_i of each member i is constant within each member and, furthermore, the stiffness of the member 1 is the average of the stiffness of the members at the ends, *i.e.* $b_1 = (b_0 + b_2)/2$. The inverse problem is then defined as: find the unknown stiffness parameters b_0 and b_1, given the displacement vector $\mathbf{d}$ and boundary conditions. To prevent negative or small stiffness, constraints are imposed on the stiffness of each member, $b_i > 0.1$, and Neumann boundary conditions are set to $\mathbf{f} = (0, 1, 0.05)^T$. Due to the constraint on the stiffness of member 1, the image of the forward problem is a manifold with dimension 2 embedded in displacement space $\mathbb{R}^3$. Due to the symmetry in this problem, the likelihood function, shown in center panel of Fig. 17, is bimodal.

We place a multivariate Gaussian as prior on the stiffness parameters, and use a bimodal variational distribution to infer the posterior distribution of the parameters b_0 and b_2 given observed displacement $\mathbf{d} = (0.1, 0.17, 0.23)^T$. Specifically, we consider a mixture of two multivariate Gaussians with equal fixed mixture weights. As there is no closed form expression for the KL divergence between a mixture of Gaussians and a single Gaussian, we estimate the KL divergence term in the ELBO using Monte Carlo sampling. As shown in the right panel of Fig. 17, the resulting posterior distribution is bimodal and recovers the two modes present in the likelihood function. This illustrative example shows that when a proposed model exhibits multi-modality, the flexibility of variational Bayes methodology allows for specifying a family of variational distributions that can capture that property of the model.

5. Conclusions

In this paper, we have presented the variational inference framework for Bayesian inverse problems and investigated its efficacy on problems based on elliptic PDEs. Computationally, Variational Bayes offers a tractable alternative to the intractable MCMC methods, and provides consistent mean and uncertainty estimates on the problems inspired by questions in computational mechanics. VB recasts the integration problem associated with Bayesian inference into an optimization problem. As such, it is naturally integrated with existing FEM solvers, using the gradient calculations from the FEM solvers to optimize the VB objective. Furthermore, the geometry of the problem encoded in the FEM mesh is utilized through the use of a sparse precision matrix that defines the conditional independence structure of the problem. Our results on the 1D and 2D Poisson problems support the claims of accuracy and scalability of VB. In particular, our results show that

- the mean of the variational posterior provides an accurate point estimate irrespective of the choice of the parametrization of the covariance structure,
- the variational approximation with a full-covariance structure and the structured precision structure adequately estimates posterior variance when compared to HMC and pCN which are known to be asymptotically correct,
- parameterizing the multivariate Gaussian distribution using a sparse precision matrix provides a way to balance the trade-off between computational complexity and the ability to capture dependencies in the posterior distribution,
- variational inference provides a good estimate for the mean and the variance of the posterior distribution in a time that is an order of magnitude faster than HMC or pCN,

- the multivariate Gaussian variational family is flexible enough to capture the true posterior distribution with high accuracy,
- the VB estimates may be used effectively in downstream tasks to estimates various quantities of interest,
- variational Bayes method is flexible enough to model multimodal posteriors, as illustrated on the truss example.

Our work may be extended in a number of natural ways that allows for greater adaptivity to the specific problems encountered in applications and integration within exiting frameworks. Firstly, taking advantage of fast implementations of sparse linear algebra routines would further improve the scalability of VB with the structured precision matrix as proposed in our work. Secondly, casting the inverse problem in a multi-level setting has potential to further improve computational efficiency (Nagel and Sudret, 2016). Thirdly, the results provided in this paper use standard off-the-shelf optimization routines; further computational improvements may be achieved using customized algorithms. As a further extension, in some applications it may be informative to consider the uncertainty in the forcing function so the forward mapping is stochastic, as discussed in (Girolami et al., 2021). Finally, one of the aims of our work is to take advantage of the advances in Bayesian inference and adapt the novel algorithms to inverse problems in computational mechanics. As such, any further developments in VB as applied to machine learning and computational statistics problems may be directly applied using the framework proposed in this paper.

6. Implementation

Codes for performing all forms of variational Bayes inference presented in this paper are available on Github at `https://github.com/jp2011/bip-pde-vi`. The user must provide their own PDE solver which accepts $\boldsymbol{\kappa}$ as input parameter and computes $\log p(\boldsymbol{y} \mid \boldsymbol{\kappa})$, together with its gradient with respect to $\boldsymbol{\kappa}$.

Acknowledgements

We would like to thank William Baldwin for providing us the necessary background for and the implementation of the bimodal example; Garoe Dorta for his help with Tensorflow debugging. JP, IK, and MG were supported by the EPSRC grant EP/P020720/2 for Inference, COmputation and Numerics for Insights into Cities (ICONIC, https://iconicmath.org/). JP was also supported by EPSRC (EP/L015129/1). MG was supported by EPSRC grants EP/T000414/1, EP/R018413/2, EP/R034710/1, EP/R004889/1, and a Royal Academy of Engineering Research Chair in Data Centric Engineering.

Appendix A. Variational Inference

Appendix A.1. Reparametrization trick

Reparametrization trick allows computing the gradients of quantities derived from samples from a probability distribution with respect to the parameters $\boldsymbol{\phi}$ of that probability distribution. This holds for probability distributions where samples can be obtained by a deterministic mapping, parametrized by $\boldsymbol{\phi}$, of other random variables.

Let ϵ be a set of random variables. We assume that samples of $\boldsymbol{\kappa} \sim q(\boldsymbol{\kappa}; \boldsymbol{\phi})$ are given by a deterministic mapping

$$\boldsymbol{\kappa} = t(\boldsymbol{\phi}, \epsilon). \tag{A.1}$$

The KL divergence between approximating distribution $q(\boldsymbol{\kappa})$ and the prior $p(\boldsymbol{\kappa})$ is often available in closed form and so are its gradients with respect to $\boldsymbol{\phi}$. To estimate the gradients of the Monte Carlo estimate of the log-likelihood of the data,

$$\mathbb{E}_q\left[\log p(\mathbf{y} \mid \boldsymbol{\kappa})\right] \approx N_{\text{SVI}}^{-1} \sum_{i=1}^{N_{\text{SVI}}} \log p(\mathbf{y} \mid \boldsymbol{\kappa}^{(i)}), \tag{A.2}$$

we can use the chain rule of differentiation to obtain

$$\nabla_{\boldsymbol{\phi}} N_{\text{SVI}}^{-1} \sum_{i=1}^{N_{\text{SVI}}} \log p(\mathbf{y} \mid \boldsymbol{\kappa}^{(i)}) = N_{\text{SVI}}^{-1} \sum_{i=1}^{N_{\text{SVI}}} \nabla_{\boldsymbol{\kappa}} \log p(\mathbf{y} \mid \boldsymbol{\kappa}^{(i)}) \cdot \nabla_{\boldsymbol{\phi}} t(\boldsymbol{\phi}, \epsilon^{(i)}). \tag{A.3}$$

Appendix B. Markov Chain Monte Carlo

Appendix B.1. Pre-Conditioned Crank-Nicholson Scheme

We consider the preconditioned Crank-Nicholson scheme proposed by Cotter et al. (2013). In short, the sampling procedure can be summarized as follows:

1. Initialize $\kappa^{(0)}$ and set $k = 0$.
2. Propose $v^{(k)} = \sqrt{(1-\beta^2)}\kappa^{(k)} + \beta\xi^{(k)}$, $\xi^{(k)} \sim p(\kappa)$.
3. Set $\kappa^{(k+1)} = v^{(k)}$ with probability $\min(1, a(\kappa^{(k)}, v^{(k)}))$, where $a(\kappa_{\text{old}}, \kappa_{\text{new}}) = \exp\big(\log p(y \mid \kappa_{\text{new}}) - \log p(y \mid \kappa_{\text{old}})\big)$
4. Otherwise, set $\kappa^{(k+1)} = \kappa^{(k)}$.
5. Set $k = k + 1$.

The parameter β corresponds to the amount of innovation in the proposal. If the value is small, there is little innovation and the proposed sample will be close to the previous sample.

Appendix B.2. Hamiltonian Monte Carlo

Random walk Metropolis-Hastings algorithm may lead to inefficient explorations of the sample space, especially in the case of oddly-shaped densities and in higher dimensions. Improved proposal densities can alleviate this issue, but as the number of dimensions increases these measures become less effective. Hamiltonian Monte Carlo is a variant of Metropolis-Hastings which takes advantage of the gradients of the target distribution in the proposal, allowing for a more rapid exploration of the sample space, even in a high-dimensional target space. For each component $\boldsymbol{\kappa}_i$ of the target space, the scheme adds a 'momentum' variable $\boldsymbol{\phi}_j$ (note that this is different from $\boldsymbol{\phi}$ used in section Appendix A.1). Subsequently, $\boldsymbol{\kappa}$ and $\boldsymbol{\phi}$ are updated jointly in a series of updates to propose a new sample $(\boldsymbol{\kappa}^*, \boldsymbol{\phi}^*)$ that is then accepted/rejected.

The proposal is largely driven by the momentum variable. The proposal step starts with drawing a new value of $\boldsymbol{\phi}$ from $p(\boldsymbol{\phi})$ which needs to be specified. Then in a series of user-specified steps, L, the momentum variable $\boldsymbol{\phi}$ is updated based on the gradient of the log of the target density, and $\boldsymbol{\kappa}$ is moved based on the momentum. Usually, the distribution of the momentum variable is $\mathcal{N}(0, M)$, where M is the so called 'mass' matrix. A diagonal matrix is often chosen to be able to efficiently sample from the momentum distribution. The full steps of the procedure are given in algorithm 2.

The reason why HMC is suitable for high-dimensional problems is that to efficiently explore the target space, the algorithm exploits gradients of log of the target distribution.

The performance of the algorithm can be tuned in three ways: (i) choice of the momentum distribution $p(\phi)$, which in the version above requires specifying the mass matrix, (ii) adjusting the

Algorithm 2: Hamiltonian Monte Carlo as presented in Gelman et al. (2013)

```
Input: $p(\boldsymbol{\kappa}|\mathbf{y})$: unnormalised target density , $p(\boldsymbol{\phi})$: momentum density and its mass
       matrix $M$, $L$: leapfrog steps, $\epsilon$: scaling factor
Output: A list of samples from $p(\boldsymbol{\kappa}|\mathbf{y})$.
for $t \leftarrow 1, 2, \dots$ do
    Sample $\boldsymbol{\phi}$ from $p(\boldsymbol{\phi})$
    for $i \leftarrow 1$ to $L$ do
        $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \frac{1}{2}\epsilon \frac{d \log p(\boldsymbol{\kappa}|\mathbf{y})}{d\boldsymbol{\kappa}}$
        $\boldsymbol{\kappa} \leftarrow \boldsymbol{\kappa} + \epsilon M^{-1}\boldsymbol{\phi}$
        $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \frac{1}{2}\epsilon \frac{d \log p(\boldsymbol{\kappa}|\mathbf{y})}{d\boldsymbol{\kappa}}$
        $r = \frac{p(\boldsymbol{\kappa}^*|\mathbf{y})p(\boldsymbol{\phi}^*)}{p(\boldsymbol{\kappa}^{t-1}|\mathbf{y})p(\boldsymbol{\phi}^{t-1})}$
        $\boldsymbol{\kappa}^t = \begin{cases} \boldsymbol{\kappa}^* & \text{with probability } \min(r,1) \\ \boldsymbol{\kappa}^{t-1} & \text{otherwise} \end{cases}$
return $[\boldsymbol{\kappa}^1, \boldsymbol{\kappa}^2, \dots]$
```

scaling factor of the leapfrog step, ϵ, and (iii) the number of leapfrog steps, L. Gelman et al. (2013) suggest setting ϵ and L so that $\epsilon L = 1$. They suggest tuning these so that the acceptance rate is about 65%. As for the mass matrix, the authors suggest that it should approximately scale with the inverse covariance matrix of the posterior distribution, $(\mathrm{Cov}(\boldsymbol{\kappa}|\mathbf{y}))^{-1}$. This can be achieved by a pre-run from which the empirical covariance matrix can be computed.

Appendix B.3. Extra plots

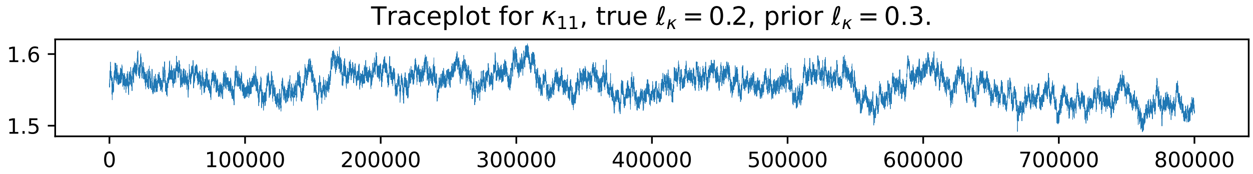


Figure B.18: Trace plot for pCN MCMC method for one of the estimated variables in the 2D Poisson example.

References

Abdulle, A. and Garegnani, G. (2021), 'A probabilistic finite element method based on random meshes: A posteriori error estimators and bayesian inverse problems', *Computer Methods in Applied Mechanics and Engineering* **384**, 113961.

Abraham, K. and Nickl, R. (2020), 'On statistical calderón problems', *Mathematical Statistics and Learning* **2**, 165–216.

Aristoff, D. and Bangerth, W. (2021), A benchmark for the Bayesian inversion of coefficients in partial differential equations. https://arxiv.org/abs/2102.07263.

Arnst, M. and Soize, C. (2019*a*), 'Identification and sampling of bayesian posteriors of high-dimensional symmetric positive-definite matrices for data-driven updating of computational models', *Computer Methods in Applied Mechanics and Engineering* **352**, 300–323.

Arnst, M. and Soize, C. (2019*b*), 'Identification and sampling of Bayesian posteriors of high-dimensional symmetric positive-definite matrices for data-driven updating of computational models', *Computer Methods in Applied Mechanics and Engineering* **352**, 300–323.

Asaadi, E. and Heyns, P. (2017), 'A computational framework for Bayesian inference in plasticity models characterisation', *Computer Methods in Applied Mechanics and Engineering* **321**, 455–481.

Babuška, I., Sawlan, Z., Scavino, M., Szabó, B. and Tempone, R. (2016), 'Bayesian inference and model comparison for metallic fatigue data', *Computer Methods in Applied Mechanics and Engineering* **304**, 171–196.

Barajas-Solano, D. A. and Tartakovsky, A. M. (2019), 'Approximate Bayesian model inversion for PDEs with heterogeneous and state-dependent coefficients', *Journal of Computational Physics* **395**, 247–262.

Beck, J., Dia, B. M., Espath, L. F., Long, Q. and Tempone, R. (2018), 'Fast Bayesian experimental design: Laplace-based importance sampling for the expected information gain', *Computer Methods in Applied Mechanics and Engineering* **334**, 523–553.

Betz, W., Papaioannou, I., Beck, J. L. and Straub, D. (2018), 'Bayesian inference with subset simulation: Strategies and improvements', *Computer Methods in Applied Mechanics and Engineering* **331**, 72–93.

Bishop, C., Lawrence, N., Jaakkola, T. and Jordan, M. (1998), Approximating posterior distributions in belief networks using mixtures, *in* M. Jordan, M. Kearns and S. Solla, eds, 'Advances in Neural Information Processing Systems', Vol. 10, MIT Press.

Bishop, C. M. (2006), *Pattern Recognition and Machine Learning*, Information Science and Statistics, Springer, New York.

Blei, D. M., Kucukelbir, A. and McAuliffe, J. D. (2017), 'Variational Inference: A Review for Statisticians', *Journal of the American Statistical Association* **112**(518), 859–877.

Bui-Thanh, T., Ghattas, O., Martin, J. and Stadler, G. (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), A2494–A2523.

Burt, D. R., Ober, S. W., Garriga-Alonso, A. and van der Wilk, M. (2021), Understanding variational inference in function-space, *in* 'Third Symposium on Advances in Approximate Bayesian Inference'.

Carlon, A. G., Dia, B. M., Espath, L., Lopez, R. H. and Tempone, R. (2020), 'Nesterov-aided stochastic gradient methods using Laplace approximation for Bayesian design optimization', *Computer Methods in Applied Mechanics and Engineering* **363**, 112909.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P. and Riddell, A. (2017), 'Stan : A Probabilistic Programming Language', *Journal of Statistical Software* **76**(1), 1–32.

Chen, P., Villa, U. and Ghattas, O. (2017), 'Hessian-based adaptive sparse quadrature for infinite-dimensional Bayesian inverse problems', *Computer Methods in Applied Mechanics and Engineering* **327**, 147–172. Advances in Computational Mechanics and Scientific Computation—the Cutting Edge.

Cheng, C.-A. and Boots, B. (2017), Variational inference for Gaussian process models with linear complexity, *in* I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, eds, 'Advances in Neural Information Processing Systems', Vol. 30, Curran Associates, Inc.

Cotter, S. L., Roberts, G. O., Stuart, A. M. and White, D. (2013), 'MCMC Methods for Functions: Modifying Old Algorithms to Make Them Faster', *Statistical Science* **28**(3), 424–446.

Csató, L. and Opper, M. (2002), 'Sparse on-line Gaussian processes', *Neural Computation* **14**(3), 641–668.

Cuthill, E. and McKee, J. (1969), Reducing the bandwidth of sparse symmetric matrices, *in* 'Proceedings of the 1969 24th National Conference', ACM '69, Association for Computing Machinery, New York, NY, USA, pp. 157–172.

Damianou, A. C., Titsias, M. K. and Lawrence, N. D. (2016), 'Variational inference for latent variables and uncertain inputs in Gaussian processes', *Journal of Machine Learning Research* **17**(42), 1–62.

D'Angelo, F. and Fortuin, V. (2021), Annealed Stein variational gradient descent, *in* 'Third Symposium on Advances in Approximate Bayesian Inference'.

Durrande, N., Adam, V., Bordeaux, L., Eleftheriadis, S. and Hensman, J. (2019), Banded matrix operators for Gaussian Markov models in the automatic differentiation era, *in* K. Chaudhuri and M. Sugiyama, eds, 'Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics', Vol. 89 of *Proceedings of Machine Learning Research*, PMLR, pp. 2780–2789.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A. and Rubin, D. B. (2013), *Bayesian Data Analysis*, Chapman and Hall/CRC.

Giordano, M. and Nickl, R. (2020), 'Consistency of Bayesian inference with Gaussian process priors in an elliptic inverse problem', *Inverse Problems* **36**(8), 085001.

Giordano, R., Broderick, T. and Jordan, M. I. (2018), 'Covariances, Robustness, and Variational Bayes', *Journal of Machine Learning Research* **19**, 1981–2029.

Girolami, M., Febrianto, E., Yin, G. and Cirak, F. (2021), 'The statistical finite element method (statFEM) for coherent synthesis of observation data and model predictions', *Computer Methods in Applied Mechanics and Engineering* **375**, 113533.

Gneiting, T. and Raftery, A. E. (2007), 'Strictly Proper Scoring Rules, Prediction, and Estimation', *Journal of the American Statistical Association* **102**(477), 359–378.

Hensman, J., Fusi, N. and Lawrence, N. D. (2013), Gaussian processes for big data, *in* 'Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence', UAI'13, AUAI Press, Arlington, Virginia, USA, pp. 282–290.

Hensman, J., Rattray, M. and Lawrence, N. (2012), Fast variational inference in the conjugate exponential family, *in* F. Pereira, C. J. C. Burges, L. Bottou and K. Q. Weinberger, eds, 'Advances in Neural Information Processing Systems', Vol. 25, Curran Associates, Inc.

Hoffman, M. D., Blei, D. M., Wang, C. and Paisley, J. (2013), 'Stochastic Variational Inference', *Journal of Machine Learning Research* **14**, 1303–1347.

Huang, Y., Beck, J. L. and Li, H. (2017), 'Bayesian system identification based on hierarchical sparse Bayesian learning and Gibbs sampling with application to structural damage assessment', *Computer Methods in Applied Mechanics and Engineering* **318**, 382–411.

Huang, Y., Beck, J. L., Li, H. and Ren, Y. (2021), 'Sequential sparse Bayesian learning with applications to system identification for damage assessment and recursive reconstruction of image sequences', *Computer Methods in Applied Mechanics and Engineering* **373**, 113545.

Huggins, J., Kasprzak, M., Campbell, T. and Broderick, T. (2020), Validated variational inference via practical posterior error bounds, *in* S. Chiappa and R. Calandra, eds, 'Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics', Vol. 108 of *Proceedings of Machine Learning Research*, PMLR, pp. 1792–1802.

Ibrahimbegovic, A., Matthies, H. G. and Karavelić, E. (2020), 'Reduced model of macro-scale stochastic plasticity identification by Bayesian inference: Application to quasi-brittle failure of concrete', *Computer Methods in Applied Mechanics and Engineering* **372**, 113428.

Jankowiak, M., Pleiss, G. and Gardner, J. (2020), Parametric Gaussian process regressors, *in* H. D. III and A. Singh, eds, 'Proceedings of the 37th International Conference on Machine Learning', Vol. 119 of *Proceedings of Machine Learning Research*, PMLR, pp. 4702–4712.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S. and Saul, L. K. (1999), 'An Introduction to Variational Methods for Graphical Models', *Machine Learning* **37**(2), 183–233.

Jordan, M. I. and Wainwright, M. J. (2007), 'Graphical Models, Exponential Families, and Variational Inference', *Foundations and Trends® in Machine Learning* **1**, 1–305.

Kaipio, J. and Somersalo, E. (2005), *Statistical and Computational Inverse Problems*, number v. 160 *in* 'Applied Mathematical Sciences', Springer, New York.

Karathanasopoulos, N., Angelikopoulos, P., Papadimitriou, C. and Koumoutsakos, P. (2017), 'Bayesian identification of the tendon fascicle's structural composition using finite element models for helical geometries', *Computer Methods in Applied Mechanics and Engineering* **313**, 744–758.

Kingma, D. P. and Ba, J. (2015), Adam: A method for stochastic optimization, *in* Y. Bengio and Y. LeCun, eds, 'International Conference on Learning Representations', San Diego, CA, USA.

Kingma, D. P. and Welling, M. (2014), Auto-encoding variational bayes, *in* '2nd International Conference on Learning Representations, ICLR'.

Liu, Q. and Wang, D. (2016), Stein variational gradient descent: A general purpose bayesian inference algorithm, *in* 'Advances in Neural Information Processing Systems', Vol. 29.

Lu, C. and Tang, X. (2015), Surpassing human-level face verification performance on lfw with gaussian face, *in* 'Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence', AAAI'15, AAAI Press, p. 3811–3819.

MacKay, D. J. C. (2003), *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, Cambridge, UK ; New York.

Michelén Ströfer, C. A., Zhang, X.-L., Xiao, H. and Coutier-Delgosha, O. (2020), 'Enforcing boundary conditions on physical fields in Bayesian inversion', *Computer Methods in Applied Mechanics and Engineering* **367**, 113097.

Minh, H. Q. (2017), 'Infinite-dimensional log-determinant divergences between positive definite trace class operators', *Linear Algebra and its Applications* **528**, 331–383.

Monard, F., Nickl, R. and Paternain, G. P. (2020), 'Statistical guarantees for bayesian uncertainty quantification in non-linear inverse problems with gaussian process priors'.

Nagel, J. B. and Sudret, B. (2016), 'A unified framework for multilevel uncertainty quantification in bayesian inverse problems', *Probabilistic Engineering Mechanics* **43**, 68–84.

Ni, P., Li, J., Hao, H., Han, Q. and Du, X. (2021), 'Probabilistic model updating via variational bayesian inference and adaptive gaussian process modeling', *Computer Methods in Applied Mechanics and Engineering* **383**, 113915.

Pandita, P., Tsilifis, P., Awalgaonkar, N. M., Bilionis, I. and Panchal, J. (2021), 'Surrogate-based sequential bayesian experimental design using non-stationary gaussian processes', *Computer Methods in Applied Mechanics and Engineering* **385**, 114007.

Pinski, F. J., Simpson, G., Stuart, A. M. and Weber, H. (2015), 'Algorithms for Kullback–Leibler Approximation of Probability Measures in Infinite Dimensions', *SIAM Journal on Scientific Computing* **37**(6), A2733–A2757.

Pleiss, G., Gardner, J., Weinberger, K. and Wilson, A. G. (2018), Constant-time predictive distributions for Gaussian processes, *in* J. Dy and A. Krause, eds, 'Proceedings of the 35th International Conference on Machine Learning', Vol. 80 of *Proceedings of Machine Learning Research*, PMLR, pp. 4114–4123.

Pyrialakos, S., Kalogeris, I., Sotiropoulos, G. and Papadopoulos, V. (2021), 'A neural network-aided bayesian identification framework for multiscale modeling of nanocomposites', *Computer Methods in Applied Mechanics and Engineering* **384**, 113937.

Quiñonero Candela, J. and Rasmussen, C. E. (2005), 'A unifying view of sparse approximate gaussian process regression', *J. Mach. Learn. Res.* **6**, 1939–1959.

Ranganath, R., Tran, D. and Blei, D. (2016), Hierarchical variational models, *in* M. F. Balcan and K. Q. Weinberger, eds, 'Proceedings of The 33rd International Conference on Machine Learning', Vol. 48 of *Proceedings of Machine Learning Research*, PMLR, New York, New York, USA, pp. 324–333.

Rasmussen, C. E. and Williams, C. K. I. (2006), *Gaussian Processes for Machine Learning*, Adaptive Computation and Machine Learning, MIT Press, Cambridge, Mass.

Rizzi, F., Khalil, M., Jones, R., Templeton, J., Ostien, J. and Boyce, B. (2019), 'Bayesian modeling of inconsistent plastic response due to material variability', *Computer Methods in Applied Mechanics and Engineering* **353**, 183–200.

Robbins, H. and Monro, S. (1951), 'A Stochastic Approximation Method', *The Annals of Mathematical Statistics* **22**(3), 400 – 407.

Rue, H. and Held, L. (2005), *Gaussian Markov Random Fields: Theory and Applications*, number 104 *in* 'Monographs on Statistics and Applied Probability', Chapman & Hall/CRC, Boca Raton.

Rue, H., Martino, S. and Chopin, N. (2009), 'Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **71**(2), 319–392.

Sabater, C., Le Maître, O., Congedo, P. M. and Görtz, S. (2021), 'A bayesian approach for quantile optimization problems with high-dimensional uncertainty sources', *Computer Methods in Applied Mechanics and Engineering* **376**, 113632.

Salimbeni, H., Cheng, C.-A., Boots, B. and Deisenroth, M. (2018), Orthogonally decoupled variational gaussian processes, *in* S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi and R. Garnett, eds, 'Advances in Neural Information Processing Systems', Vol. 31, Curran Associates, Inc.

Salimbeni, H. and Deisenroth, M. (2017), Doubly stochastic variational inference for deep gaussian processes, *in* I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, eds, 'Advances in Neural Information Processing Systems 30', Curran Associates, Inc., pp. 4588–4599.

Seeger, M., Williams, C. K. and Lawrence, N. (2003), Fast forward selection to speed up sparse gaussian process regression, *in* 'AISTATS'.

Shi, J., Titsias, M. and Mnih, A. (2020), Sparse orthogonal variational inference for gaussian processes, *in* S. Chiappa and R. Calandra, eds, 'Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics', Vol. 108 of *Proceedings of Machine Learning Research*, PMLR, pp. 1932–1942.

Snelson, E. and Ghahramani, Z. (2006), Sparse gaussian processes using pseudo-inputs, *in* Y. Weiss, B. Schölkopf and J. Platt, eds, 'Advances in Neural Information Processing Systems', Vol. 18, MIT Press.

Solin, A., Cortes, S., Rahtu, E. and Kannala, J. (2018), PIVO: Probabilistic inertial-visual odometry for occlusion-robust navigation, *in* 'Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)'.

Stuart, A. M. (2010), 'Inverse problems: A Bayesian perspective', *Acta Numerica* **19**, 451–559.

Sun, S., Zhang, G., Shi, J. and Grosse, R. B. (2019), Functional variational Bayesian Neural Networks, *in* '7th International Conference on Learning Representations', New Orleans, LA, USA.

Tan, L. S. L. and Nott, D. J. (2018), 'Gaussian variational approximation with sparse precision matrices', *Statistics and Computing* **28**(2), 259–275.

Tarakanov, A. and Elsheikh, A. H. (2020), 'Optimal bayesian experimental design for subsurface flow problems', *Computer Methods in Applied Mechanics and Engineering* **370**, 113208.

Tarantola, A. (2005), *Inverse Problem Theory and Methods for Model Paramenter Estimation*, Society for Industrial and Applied Mathematics, Philadelphia, PA.

Tikhonov, A. N. and Arsenin, V. Y. (1977), *Solutions of ill-posed problems*, Scripta series in mathematics, Winston ; distributed solely by Halsted Press, Washington : New York.

Titsias, M. (2008), Variational Model Selection for Sparse Gaussian Process Regression, Technical report, School of Computer Science, University of Manchester.

Titsias, M. K. (2009), Variational Learning of Inducing Variables in Sparse Gaussian Processes., *in* 'AISTATS', Vol. 5, pp. 567–574.

Tran, D., Blei, D. and Airoldi, E. M. (2015), Copula variational inference, *in* C. Cortes, N. Lawrence, D. Lee, M. Sugiyama and R. Garnett, eds, 'Advances in Neural Information Processing Systems', Vol. 28, Curran Associates, Inc.

Tsilifis, P., Bilionis, I., Katsounaros, I. and Zabaras, N. (2016), 'Computationally Efficient Variational Approximations for Bayesian Inverse Problems', *Journal of Verification, Validation and Uncertainty Quantification* **1**(031004), 1–13.

Turner, R. E. and Sahani, M. (2011), Two problems with variational expectation maximisation for time series models, *in* A. T. Cemgil, D. Barber and S. Chiappa, eds, 'Bayesian Time Series Models', Cambridge University Press, Cambridge, pp. 104–124.

Uribe, F., Papaioannou, I., Betz, W. and Straub, D. (2020), 'Bayesian inference of random fields represented with the Karhunen–Loève expansion', *Computer Methods in Applied Mechanics and Engineering* **358**, 112632.

Villa, U., Petra, N. and Ghattas, O. (2021), 'hIPPYlib: An Extensible Software Framework for Large-Scale Inverse Problems Governed by PDEs: Part I: Deterministic Inversion and Linearized Bayesian Inference', *ACM Transactions on Mathematical Software* **47**(2), 1–34.

Wang, B. and Titterington, D. M. (2004), Inadequacy of interval estimates corresponding to variational bayesian approximations, *in* 'AISTATS', pp. 373–380.

Wu, L., Zulueta, K., Major, Z., Arriaga, A. and Noels, L. (2020), 'Bayesian inference of nonlinear multiscale model parameters accelerated by a Deep Neural Network', *Computer Methods in Applied Mechanics and Engineering* **360**, 112693.

Zhang, C., Bütepage, J., Kjellström, H. and Mandt, S. (2019), 'Advances in variational inference', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **41**(8), 2008–2026.