Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 393 (2022) 114712
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Variational Bayesian approximation of inverse problems using sparse precision matrices

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*

Received 25 October 2021; received in revised form 30 January 2022; accepted 30 January 2022
Available online 9 March 2022

Abstract

Inverse problems involving partial differential equations (PDEs) are widely used in science and engineering. Although such problems are generally ill-posed, different regularisation approaches have been developed to ameliorate this problem. Among them is the Bayesian formulation, where a prior probability measure is placed on the quantity of interest. The resulting posterior probability measure is usually analytically intractable. The Markov Chain Monte Carlo (MCMC) method has been the go-to method for sampling from those posterior measures. MCMC is computationally infeasible for large-scale problems that arise in engineering practice. Lately, Variational Bayes (VB) has been recognised as a more computationally tractable method for Bayesian inference, approximating a Bayesian posterior distribution with a simpler trial distribution by solving an optimisation problem. In this work, we argue, through an empirical assessment, that VB methods are a flexible and efficient alternative to MCMC for this class of problems. We propose a natural choice of a family of Gaussian trial distributions parametrised by precision matrices, thus taking advantage of the inherent sparsity of the inverse problem encoded in its finite element discretisation. We utilise stochastic optimisation to efficiently estimate the variational objective and assess not only the error in the solution mean 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. A Tensorflow implementation is made publicly available on GitHub.

Keywords: Inverse problems; Bayesian inference; Variational Bayes; Precision matrix; Uncertainty quantification

1. Introduction

The increased availability of measurements from engineering systems allows for the development of new and the improvement of existing computational models, which are usually formulated as partial differential equations. Inferring model parameters from observations of the physical system is termed the *inverse problem* [1–3]. In this work, we consider the inverse problem where the quantities of interest (for example, some material properties) and the observations (e.g., the displacement field) are related through elliptic PDEs. Most inverse problems are non-linear

* Corresponding author at: Department of Mathematics, Imperial College London, London, SW7 2AZ, UK.
** Corresponding author.
E-mail addresses: jan.povala@gmail.com (J. Povala), ik394@cam.ac.uk (I. Kazlauskaite).
[1] Equal contribution.

https://doi.org/10.1016/j.cma.2022.114712
0045-7825/

and ill-posed, meaning that the existence, uniqueness, and/or stability (continuous dependence on the parameters) of the solution are violated [1–3]. These issues are often alleviated through some regularisation, like Tikhonov regularisation [4], that imposes assumptions on the regularity of the solution. Alternatively, the specification of the prior in the Bayesian formulation of inverse problems provides a natural choice for regularisation, and any given regularisation can be interpreted as a specific choice of priors in the Bayesian setting [5]. Furthermore, the Bayesian formulation provides not only a qualitative but also a quantitative estimate of both epistemic and aleatoric uncertainty in 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 and prior assumptions. For these reasons, Bayesian methods have gained popularity in computational mechanics for experimental design and inverse problems with uncertainty quantification; see, e.g., the recent papers by Abdulle and Garegnani [6], Pandita et al. [7], Pyrialakos et al. [8], Ni et al. [9], Sabater et al. [10], Huang et al. [11], Ibrahimbegovic et al. [12], Tarakanov and Elsheikh [13], Michelén Ströfer et al. [14], Carlon et al. [15], Wu et al. [16], Uribe et al. [17], Rizzi et al. [18], Arnst and Soize [19], Beck et al. [20], Betz et al. [21], Chen et al. [22], Asaadi and Heyns [23], Huang et al. [24], Karathanasopoulos et al. [25], Babuška et al. [26] and Girolami et al. [27].

The Bayesian formulation of inverse problems is also 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 [28,29]. One of the leading models for Bayesian inverse problems is Gaussian processes (GPs) which define probability distributions over functions and allow for incorporating observed data to obtain posterior distributions. Given that most posterior distributions in Bayesian inference are analytically intractable, approximation methods need to be resorted to. Two classical approximation schemes are the Markov Chain Monte Carlo (MCMC) and the Laplace approximation. The 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 [30]. Application-specific techniques such as parameter space reduction and state space reduction have been proposed in the literature to help scale up MCMC methods, but these low-rank approximations are not specific to MCMC methods only [31]. 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 centred around the mode of the true posterior, utilising the negative Hessian of the unnormalised posterior log-density [5]. The Hessian is a large dense matrix, where forming each column requires multiple PDE solves; to make such calculations feasible, low-rank approximations are typically used [32,33]. Evidently, the Laplace approximation is not suitable for multi-modal posterior distributions due to the uni-modality of the Gaussian distribution.

1.1. Related work

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 a sampling problem that arises from applying the Bayes rule into an optimisation problem [34–36]. The method finds a solution that minimises the Kullback–Leibler (KL) divergence between the true posterior distribution and a trial distribution from a chosen 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 trial distribution, i.e., its number of free parameters, such that the resulting optimisation problem is computationally tractable, and the approximate posterior adequately captures important aspects of the true posterior.

Further scalability of VB methods is due to advancements in sparse approximations and approximate inference. For instance, sparse GP methods such as Nyström approximation or fully independent training conditional method (FITC) rely on lower-dimensional representations that are defined by a smaller set of so-called inducing points to represent the full GP [37–43]. 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 [44–46]. Sub-sampling the original data and Monte Carlo estimation of the optimisation 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 [47], introducing parametric approximations [48], applying the Lanczos algorithm to efficiently factorise the covariance matrix [49], transforming to an orthogonal basis [50,51], and adapting to compositional models [52].

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 finite elements, 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 [53,54].

1.2. Contributions

In this work, we advocate for the use of GP priors with stochastic variational inference as a principled and efficient way to solve the inverse problems arising in computational mechanics. We show, through an extensive empirical study, that variational Bayes methods provide a flexible and efficient 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 [55] or proof-of-concept on particular 1D inverse problems [56].

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 into account the structure of the problem, as encoded in the FEM discretisation of the PDE. Motivated by previous uses of precision matrices as a way of describing conditional independence [57,58], we leverage the sparse structure of the problems to impose conditional independence in the approximating posterior distribution. This choice of parametrisation results in sparse matrices, which improve the computational and the memory cost of the resulting algorithms. Such parametrisation, 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 optimisation algorithms in the finite element implementations.

The main concern related to VB in statistics stems from the fact that it is constrained by the chosen family of trial distributions, which may not approximate the true posterior distribution well. If the choice of the trial distributions is too restrictive, the estimate of the posterior mean is biased while the uncertainty may be underestimated [59–61]. Furthermore, as noted in previous work, the commonly used mean-field factorisation of the trial distributions does not come with general guarantees on accuracy [62]. However, VB has been demonstrated to work well in practice in a variety of settings [35,63–65]. Recent work on VB has provided some tools for assessing the robustness of the VB estimates [62].

1.3. Overview

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

2. Bayesian formulation of inverse problems

In this section, we review the Bayesian formulation of inverse problems by closely following Stuart [3].

2.1. Forward map and observation model

We are interested in finding $\kappa \in \mathcal{K}$, an input to a model, given $y \in \mathcal{Y}$, a noisy observation of the solution of the model, where $\mathcal{K}, \mathcal{Y}$ are Banach spaces.[2] The mapping is given by

$$y = \mathcal{G}(\kappa) + \eta, \tag{1}$$

where $\mathcal{G} : \mathcal{K} \to \mathcal{Y}$, $\eta \in \mathcal{Y}$ is additive observational noise. We focus on problems where $\mathcal{G}$ maps solutions of elliptic partial differential equations with input $\kappa \in \mathcal{K}$ into the observation space $\mathcal{Y}$. For a suitable Hilbert space $\mathcal{U}$, which we make concrete later, let $\mathcal{A}: \mathcal{K} \to \mathcal{U}$ be a possibly non-linear solution operator of the PDE. For a particular $\kappa \in \mathcal{K}$, the solution $u \in \mathcal{U}$ is

$$u = \mathcal{A}(\kappa). \tag{2}$$

To obtain observations y, we define a projection operator $\mathcal{P}: \mathcal{U} \to \mathcal{Y}$. Consequently, (1) can be written out in full as

$$y = \mathcal{P}(\mathcal{A}(\kappa)) + \eta. \tag{3}$$

2.2. Inference

We solve the inverse problem (1) for κ by finding κ such that the data misfit, $\|y - \mathcal{G}(\kappa)\|_{\mathcal{Y}}$, is minimised. As already mentioned in the introduction, this is typically an ill-posed problem: there may be no solution, it may not be unique, there may exist a dimensionality mismatch between the observations and the quantity being inferred, and it may depend sensitively on y. To proceed, we choose the Bayesian framework for regularising the problem to make it amenable to analysis and practical implementation. We describe our prior knowledge about κ in terms of a prior probability measure μ_0 on the subspace of $\mathcal{K}$ and use Bayes' formula to calculate the posterior probability measure, μ^y, for κ given y. The relationship between the posterior and prior is expressed as

$$\frac{\mathrm{d}\mu^y}{\mathrm{d}\mu_0}(\kappa) = \frac{1}{Z(y)} \exp(-\Phi(\kappa; y)), \tag{4}$$

where $\frac{\mathrm{d}\mu^y}{\mathrm{d}\mu_0}$ is the Radon–Nikodym derivative of μ^y with respect to μ_0, and Φ is the potential function which is determined by the forward problem (1), specifically $\mathcal{G}$ and η. To ensure that μ^y is a valid probability measure, we have $Z(y) = \int_{\mathcal{K}} \exp(-\Phi(\kappa; y))\mathrm{d}\mu_0(\kappa)$.

From here on, we assume that $(\mathcal{Y}, \|\cdot\|_{\mathcal{Y}}) = (\mathbb{R}^{n_y}, \|\cdot\|)$, where $\|\cdot\|$ is the Euclidean norm, and we treat data y and η as vectors, i.e. $\boldsymbol{y}$ and $\boldsymbol{\eta}$. We specify the additive noise vector $\boldsymbol{\eta}$ as the zero-mean Gaussian with covariance matrix $\boldsymbol{\Gamma}$ such that

$$\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma} = \sigma_y^2 \mathbf{I}),$$

where σ_y is the standard deviation of the measurement noise and $\mathbf{I}$ is the identity matrix. We can write Φ conveniently as

$$\Phi(\kappa; \boldsymbol{y}) = \frac{1}{2}\|\mathcal{G}(\kappa) - \boldsymbol{y}\|_{\boldsymbol{\Gamma}}^2, \tag{5}$$

where $\|\cdot\|_{\boldsymbol{\Gamma}}$ is the norm induced by the weighted inner product.[3]

We restrict the space of solutions $\mathcal{K}$ to be a Hilbert space and place a Gaussian prior measure on κ with mean m and covariance operator $\mathcal{C}_\kappa$ such that

$$\mu_0(\kappa) \sim \mathcal{N}(m, \mathcal{C}_\kappa). \tag{6}$$

For detailed assumptions on μ_0, $\mathcal{G}$, and η that are required for deriving the posterior probability measure, we refer the reader to Stuart [3, Sec. 2.4].

[2] Respective norms for Banach spaces $\mathcal{K}$, $\mathcal{Y}$ are $\|\cdot\|_{\mathcal{K}}$ and $\|\cdot\|_{\mathcal{Y}}$.

[3] For any self-adjoint positive operator $\mathcal{T}$, weighted inner product is $\langle\cdot,\cdot\rangle_{\mathcal{T}} = \langle\mathcal{T}^{-1/2}\cdot, \mathcal{T}^{-1/2}\cdot\rangle$, and the induced norm is $\|\cdot\|_{\mathcal{T}} = \|\mathcal{T}^{-1/2}\cdot\|$.

2.2.1. Algorithms

The objective is to find the posterior measure μ^y conditioned on the observations, as dictated by Bayes's rule. The forward map (1) and the respective functions must be discretised. In Bayesian inference there are two possible approaches for discretisation: (1) apply the Bayesian methodology first, discretise afterwards, or (2) discretise first, then apply the Bayesian methodology [3].

The first approach develops the solution of the inference problem in the function space before discretising it. A widely used algorithm of this form is the pre-conditioned Crank–Nicholson (pCN) MCMC scheme, where proposals are based on the prior measure μ_0 and the current state of the Markov chain. The pCN method is a standard choice for high-dimensional sampling problems, as its implementation is well-defined and is invariant to mesh refinement [66,67]. Since we will use this algorithm as one of the baselines, a summary of the algorithm is provided in Appendix C.1. Recently, infinite-dimensional MCMC schemes that leverage the geometry of the posterior to improve the efficiency have been proposed, see Beskos et al. [68]. Other than MCMC schemes, some variational Bayes formulations in function space have been proposed (for example, Minh [69] and Burt et al. [54]), though currently they do not offer a viable computational alternative to the finite-dimensional formulation of variational inference.

The second approach proceeds by first discretising the problem and then deriving the inference method. This approach forms the basis of almost all inference procedures developed in engineering: MCMC algorithms such as Metropolis–Hastings [70,71] or Hamiltonian Monte Carlo (HMC) [72], the Laplace approximation, or variational Bayes [34,73] are used to approximate the posterior. In the discretised formulation, HMC has achieved recognition as the *gold standard* for its good convergence properties, favourable 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 [74]). Therefore, along with the pCN scheme mentioned above, our baseline for inference methods includes the HMC method, and we provide a summary of the HMC scheme in Appendix C.2.

For the rest of the exposition in this paper, we will focus on algorithms in the finite-dimensional case, where we discretise κ to yield a vector $\boldsymbol{\kappa}$. In finite dimensions, probability densities with respect to the Lebesgue measure can be defined, thus leading to a more familiar form of the Bayes's rule:

$$p(\boldsymbol{\kappa} \mid \boldsymbol{y}) = \frac{p(\boldsymbol{y} \mid \boldsymbol{\kappa})\ p(\boldsymbol{\kappa})}{p(\boldsymbol{y})} \propto p(\boldsymbol{y} \mid \boldsymbol{\kappa})\ p(\boldsymbol{\kappa}), \tag{7}$$

where $p(\boldsymbol{\kappa} \mid \boldsymbol{y})$ is the posterior density, $p(\boldsymbol{y} \mid \boldsymbol{\kappa})$ is the likelihood of the observed data $\boldsymbol{y}$ for a given discretised $\boldsymbol{\kappa}$ and is determined by the discretised forward problem (1) and noise $\boldsymbol{\eta}$. The prior density for $\boldsymbol{\kappa}$, which itself may depend on some (hyper-) parameters ψ, is denoted by $p(\boldsymbol{\kappa})$. Next two sections focus on discussing $p(\boldsymbol{y} \mid \boldsymbol{\kappa})$ and $p(\boldsymbol{\kappa})$, respectively.

2.3. Poisson equation and likelihood

Let us consider a specific forward problem where u is the solution to the Poisson problem:

$$-\nabla \cdot (\exp(\kappa(\boldsymbol{x}))\nabla u(\boldsymbol{x})) = f(\boldsymbol{x}), \tag{8}$$

where $\boldsymbol{x} \in \Omega \subset \mathbb{R}^d$, with $d \in \{1, 2, 3\}$, $\kappa(\boldsymbol{x}) \in \mathbb{R}$ is the log-diffusion coefficient, $u(\boldsymbol{x}) \in \mathbb{R}$ is the unknown, and $f(\boldsymbol{x}) \in \mathbb{R}$ is a deterministic forcing term. The boundary conditions have been omitted for brevity. We are given n_y noisy observations $\boldsymbol{y} \in \mathbb{R}^{n_y}$ of the solution u at a finite set of points, $\{\boldsymbol{x}_i\}_{i=1}^{n_y}$. The observation points are collected in the matrix $\mathbf{X} \in \mathbb{R}^{n_y \times d}$. Although this PDE is linear in u for a given κ, the methodology in this paper applies to non-linear cases and can be extended for time-dependent cases such as the inverse problem of inferring initial conditions of a system given observations of the system at a later time.

We discretise the weak form of the Poisson problem (8) with a standard finite element approach. Specifically, the domain of interest Ω is subdivided into a set $\{\omega_e\}_{e=1}^{n_e}$ of non-overlapping elements of size $h = \max_e \operatorname{diam}(\omega_e)$ such that:

$$\Omega = \bigcup_{e=1}^{n_e} \omega_e\,. \tag{9}$$

The unknown field $u(\boldsymbol{x})$ is approximated with Lagrange basis functions $\phi_i(\boldsymbol{x})$ and the respective nodal coefficients $\mathbf{u} = (u_1, \ldots, u_{n_u})^\top$ of the n_u non-Dirichlet boundary mesh nodes by

$$u_h(\boldsymbol{x}) = \sum_{i=1}^{n_u} \phi_i(\boldsymbol{x}) u_i \,. \tag{10}$$

The discretisation of the weak form of the Poisson equation yields the linear system of equations

$$\mathbf{A}(\boldsymbol{\kappa})\mathbf{u} = \mathbf{f}\,, \tag{11}$$

where $\mathbf{A}(\boldsymbol{\kappa}) \in \mathbb{R}^{n_u \times n_u}$ is the stiffness matrix, $\boldsymbol{\kappa} \in \mathbb{R}^{n_\kappa}$ is the vector of log-diffusion coefficients, $\mathbf{f} \in \mathbb{R}^{n_u}$ is the nodal source vector. The stiffness matrix of an element with label e is given by

$$A^e_{ij}(\kappa_e) = \int_{\omega_e} \exp(\kappa_e) \frac{\partial \phi_i(\boldsymbol{x})}{\partial \boldsymbol{x}} \cdot \frac{\partial \phi_j(\boldsymbol{x})}{\partial \boldsymbol{x}} \mathrm{d}\boldsymbol{x}\,, \tag{12}$$

where the log-diffusion coefficient κ_e of the element is assumed to be *constant* within the element. The source vector is discretised as:

$$f_i = \int_\Omega f(\boldsymbol{x}) \phi_i(\boldsymbol{x}) \mathrm{d}x\,. \tag{13}$$

Hence, according to the observation model (5) the likelihood is given by

$$p(\boldsymbol{y} \mid \boldsymbol{\kappa}) = p(\boldsymbol{y} \mid \mathbf{u}(\boldsymbol{\kappa})) = \mathcal{N}(\mathbf{P}\mathbf{A}(\boldsymbol{\kappa})^{-1}\mathbf{f}, \sigma_y^2 \mathbf{I})\,, \tag{14}$$

where the matrix $\mathbf{P}$ represents the discretisation of the observation operator $\mathcal{P}$.

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

$$p(\mathbf{u}) = \int p(\mathbf{u} \mid \boldsymbol{\kappa}) p(\boldsymbol{\kappa}) \mathrm{d}\boldsymbol{\kappa}\,, \tag{15}$$

where $p(\mathbf{u} \mid \boldsymbol{\kappa})$ is deterministic as defined in (11) but $\boldsymbol{\kappa}$ appears in it non-linearly, implying that the inference is not analytically tractable.

Throughout the experiments in the later sections, we either set Dirichlet (essential) boundary conditions everywhere (for example $u(\boldsymbol{x}) = 0$ 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 the Poisson problem (8) for $u(\boldsymbol{x})$ using the finite element method (FEM).

2.4. Prior

As discussed above, we place a Gaussian measure on κ, $\mu_0(\kappa) \sim \mathcal{N}(m, \mathcal{C}_\kappa)$. Properties of samples from the measure depend on mean m and on the spectral properties of the covariance operator $\mathcal{C}_\kappa$. We restrict the space of prior functions to $L^2(\Omega, \mathbb{R})$. Then, operator $\mathcal{C}_\kappa$ can be constructed from the covariance function, $k(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}\big[\big(\kappa(\boldsymbol{x}) - m(\boldsymbol{x})\big)\big(\kappa(\boldsymbol{x}') - m(\boldsymbol{x}')\big)\big]$ as:

$$(\mathcal{C}_\kappa \gamma)(\boldsymbol{x}) = \int_\Omega k(\boldsymbol{x}, \boldsymbol{x}') \gamma(\boldsymbol{x}') \mathrm{d}\boldsymbol{x}', \tag{16}$$

for any $\gamma \in L^2(\Omega, \mathbb{R})$. This formulation is what is commonly referred to as a Gaussian process (GP) with mean function $m(\cdot)$, which we assume to be zero, and covariance function $k(\cdot, \cdot)$ such that

$$\kappa \sim \mathcal{GP}\big(m(\cdot), k(\cdot, \cdot)\big). \tag{17}$$

Even though the process is infinite-dimensional, an instantiation of the process is finite and reduces to a multivariate Gaussian distribution by definition. The covariance function is typically parametrised by a set of hyperparameters ψ. One popular option, which satisfies assumptions about μ_0 as per Stuart [3], is the squared exponential kernel (also called the exponentiated quadratic or the radial basis function (RBF) kernel):

$$k_{\mathrm{SE}}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_\kappa^2 \exp\left(-\frac{r^2}{2\ell_\kappa^2}\right), \tag{18}$$

where $r = \|\boldsymbol{x} - \boldsymbol{x}'\|_2$ is the Euclidean distance between the inputs. It depends on two hyper-parameters $\psi = \{\sigma_\kappa, \ell_\kappa\}$, the scaling parameter σ_κ, and the length-scale ℓ_κ. Note that, $k_{\text{SE}}(\cdot, \cdot)$ is an infinitely smooth function, which implies that so is $\kappa(\cdot)$. The RBF kernel imposes smoothness and stationarity assumptions on the solution; in addition, such choice of kernel offers a way to regularise the resulting optimisation problem. However, depending on the expert knowledge of the true solution, other kernels may be used to impose other assumptions such as periodicity.

Both conditioning and marginalisation of the GP can be done in closed form. In particular, consider the joint model of the values $\boldsymbol{\kappa}$ at training locations $\mathbf{X}$ and the unknown test values $\boldsymbol{\kappa}^*$ at test locations $\mathbf{X}^*$:

$$\begin{bmatrix} \boldsymbol{\kappa} \\ \boldsymbol{\kappa}^* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_\psi(\mathbf{X}, \mathbf{X}) & \mathbf{K}_\psi(\mathbf{X}, \mathbf{X}^*) \\ \mathbf{K}_\psi(\mathbf{X}^*, \mathbf{X}) & \mathbf{K}_\psi(\mathbf{X}^*, \mathbf{X}^*) \end{bmatrix}\right), \tag{19}$$

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 $\boldsymbol{\kappa}^*$ given the values $\boldsymbol{\kappa}$ at $\mathbf{X}$ is:

$$\boldsymbol{\kappa}^* \mid \boldsymbol{\kappa} \sim \mathcal{N}\left(\tilde{\boldsymbol{\kappa}}^*,\ \tilde{\mathbf{K}}\right), \tag{20}$$

where

$$\begin{aligned} \tilde{\boldsymbol{\kappa}}^* &= \mathbf{K}\left(\mathbf{X}^*, \mathbf{X}\right)\left[\mathbf{K}(\mathbf{X}, \mathbf{X})\right]^{-1}\boldsymbol{\kappa} \\ \tilde{\mathbf{K}} &= \mathbf{K}\left(\mathbf{X}^*, \mathbf{X}^*\right) - \mathbf{K}\left(\mathbf{X}^*, \mathbf{X}\right)\left[\mathbf{K}(\mathbf{X}, \mathbf{X})\right]^{-1}\mathbf{K}\left(\mathbf{X}, \mathbf{X}^*\right). \end{aligned} \tag{21}$$

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

In this work, we place a zero-mean Gaussian process prior on $\kappa(\boldsymbol{x})$ 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 $\kappa(\boldsymbol{x})$ is constant on each element of the mesh (we use the same mesh as for discretising $u(\boldsymbol{x})$ and $f(\boldsymbol{x})$). We place the prior on $\boldsymbol{\kappa}$ so that the centroids of the elements are the training points of the GP:

$$p(\boldsymbol{\kappa}) = \mathcal{N}(\mathbf{0}, \mathbf{K}_\psi(\mathbf{X}, \mathbf{X})). \tag{22}$$

3. Variational Bayes approximation

3.1. Variational Bayes

We assume that any hyper-parameters ψ of the prior are fixed, and are only interested in the posterior distribution of $\boldsymbol{\kappa}$. The variational approach proceeds by approximating the true posterior $p(\boldsymbol{\kappa} \mid \boldsymbol{y})$ according to (7) with a trial density $q(\boldsymbol{\kappa})$, which is the minimiser of the discrepancy between a chosen family of trial densities $\mathcal{D}_q$ and the true posterior distribution $p(\boldsymbol{\kappa}|\boldsymbol{y})$ [34,36]. 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:

$$q^*(\boldsymbol{\kappa}) = \underset{q(\boldsymbol{\kappa}) \in \mathcal{D}_q}{\operatorname{argmin}}\ \mathrm{KL}(q(\boldsymbol{\kappa}) \parallel p(\boldsymbol{\kappa} \mid \boldsymbol{y})). \tag{23}$$

Expanding the KL divergence term we obtain

$$\begin{aligned} \mathrm{KL}(q(\boldsymbol{\kappa}) \parallel p(\boldsymbol{\kappa} \mid \boldsymbol{y})) &= \int q(\boldsymbol{\kappa}) \log \frac{q(\boldsymbol{\kappa})}{p(\boldsymbol{\kappa} \mid \boldsymbol{y})} \mathrm{d}(\boldsymbol{\kappa}) \\ &= \mathbb{E}_q\big[\log q(\boldsymbol{\kappa})\big] - \mathbb{E}_q\big[\log p(\boldsymbol{\kappa} \mid \boldsymbol{y})\big] \\ &= \mathbb{E}_q\big[\log q(\boldsymbol{\kappa})\big] - \mathbb{E}_q\big[\log \frac{p(\boldsymbol{y}, \boldsymbol{\kappa})}{p(\boldsymbol{y})}\big] \\ &= \mathbb{E}_q\big[\log q(\boldsymbol{\kappa})\big] - \mathbb{E}_q\big[\log p(\boldsymbol{y}, \boldsymbol{\kappa})\big] + \log p(\boldsymbol{y}). \end{aligned} \tag{24}$$

The last term of the KL divergence, the log-marginal likelihood $\log p(\boldsymbol{y})$, is usually not analytically tractable. However, we use the fact that the KL divergence is non-negative to obtain the bound

$$\log p(\boldsymbol{y}) \geq \mathbb{E}_q\big[\log p(\boldsymbol{y}, \boldsymbol{\kappa})\big] - \mathbb{E}_q\big[\log q(\boldsymbol{\kappa})\big]. \tag{25}$$

This inequality becomes an equality when the trial density $q(\boldsymbol{\kappa})$ and the posterior $p(\boldsymbol{\kappa} \mid \boldsymbol{y})$ are equal. To minimise the KL divergence, it is sufficient to maximise $\mathbb{E}_q\big[\log p(\boldsymbol{y}, \boldsymbol{\kappa})\big] - \mathbb{E}_q\big[\log q(\boldsymbol{\kappa})\big]$, which is commonly referred to as the evidence lower bound (ELBO). The ELBO term can be rewritten as

$$\begin{aligned} \text{ELBO}(q) &= \mathbb{E}_q\big[\log p(\boldsymbol{y} \mid \boldsymbol{\kappa}) + \log p(\boldsymbol{\kappa})\big] - \mathbb{E}_q\big[\log q(\boldsymbol{\kappa})\big] \\ &= \mathbb{E}_q\big[\log p(\boldsymbol{y} \mid \boldsymbol{\kappa})\big] - \text{KL}(q(\boldsymbol{\kappa}) \parallel p(\boldsymbol{\kappa})). \end{aligned} \tag{26}$$

To summarise, the task now becomes:

$$q^*(\boldsymbol{\kappa}) = \underset{q(\boldsymbol{\kappa}) \in \mathcal{D}_q}{\arg\max}\ \mathbb{E}_q\big[\log p(\boldsymbol{y} \mid \boldsymbol{\kappa})\big] - \text{KL}(q(\boldsymbol{\kappa}) \parallel p(\boldsymbol{\kappa})). \tag{27}$$

To maximise the ELBO with a gradient-based optimiser, we need to evaluate it and 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, $\mathbb{E}_q\big[\log p(\boldsymbol{y} \mid \boldsymbol{\kappa})\big]$ involving the likelihood is generally not available. It can be approximated using a Monte Carlo approximation with N_{SVI} samples from the trial density $q(\boldsymbol{\kappa})$ as follows:

$$\mathbb{E}_q\big[\log p(\boldsymbol{y} \mid \boldsymbol{\kappa})\big] \approx \frac{1}{N_{\text{SVI}}} \sum_{i=1}^{N_{\text{SVI}}} \log p(\boldsymbol{y} \mid \boldsymbol{\kappa}^{(i)}), \tag{28}$$

where $\boldsymbol{\kappa}^{(i)}$ is the ith sample from $q(\boldsymbol{\kappa})$. This is done through a reparametrisation trick, as described in Appendix B.1. Our empirical tests show that the value of N_{SVI} in the range of 2–5 provides fast convergence of the optimisation, agreeing with previous literature [63]. 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 [56] but in contrast with the analytic approximation based on the Hessian calculations proposed in Tsilifis et al. [55].

3.2. Specification of trial 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 optimisation of the usually non-convex ELBO becomes harder. A balance must be struck in this trade-off: the family should be rich enough, but the optimisation task should still be computationally tractable.

A practical and widely used variational family is the multivariate Gaussian distribution, parametrised 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 (26) is available in closed form for a GP prior. The choice of the parametrisation 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 trial 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 [75]. A more recent development is embedding parameters of a mean-field approximation in a hierarchical model to induce variational dependencies between latent variables [76,77].

3.2.1. Gaussian trial distribution

Choosing the trial distribution $q(\boldsymbol{\kappa})$ as a multivariate Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ requires optimisation over the mean $\boldsymbol{\mu}$ 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 parametrising the covariance matrix $\boldsymbol{\Sigma}$ using its full Cholesky factor $\boldsymbol{L}$, i.e.,

$$q(\boldsymbol{\kappa}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{L}\boldsymbol{L}^\top). \tag{29}$$

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). Parametrising the components of $\boldsymbol{L}$ automatically

ensures that the covariance matrix $\boldsymbol{\Sigma}$ is positive definite as necessary. The number of parameters to optimise grows as $\mathcal{O}(n_\kappa^2)$ and this leads to a difficult optimisation task that needs to be carefully initialised and parametrised. We refer to this parametrisation 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 optimised, the optimisation task becomes simpler and the number of parameters grows only as $\mathcal{O}(n_\kappa)$. While more computationally efficient and easier to initialise, MFVB ignores much of the dependence structure of the posterior distribution.

3.3. Conditional independence and sparse precision matrices

Instead of parametrising the covariance matrix $\boldsymbol{\Sigma}$, or its Cholesky decomposition $\boldsymbol{L}$, in physical systems it is often advantageous to parametrise the precision matrix, $\boldsymbol{Q}$, where $\boldsymbol{Q} = \boldsymbol{\Sigma}^{-1}$. While a component of the covariance matrix $\boldsymbol{\Sigma}$ expresses *marginal* dependence between the two corresponding random variables, the elements of the precision matrix reflect their *conditional independence* [78]. Or, more specifically, for two components κ_i and κ_j of a Gaussian random vector $\boldsymbol{\kappa}$ we note

$$p(\kappa_i,\, \kappa_j) = p(\kappa_i)p(\kappa_j) \quad \Leftrightarrow \quad \Sigma_{ij} = 0\,, \tag{30}$$

where Σ_{ij} denotes the respective component of $\boldsymbol{\Sigma}$. Furthermore, defining the vector $\boldsymbol{\kappa}_{-\{i,j\}}$ from the random vector $\boldsymbol{\kappa}$ by removing its ith and jth component, we note

$$p(\kappa_i,\, \kappa_j \mid \boldsymbol{\kappa}_{-\{i,j\}}) = p(\kappa_i \mid \boldsymbol{\kappa}_{-\{i,j\}})p(\kappa_j \mid \boldsymbol{\kappa}_{-\{i,j\}}) \quad \Leftrightarrow \quad Q_{ij} = 0\,. \tag{31}$$

That is, $Q_{ij} = 0$ if and only if κ_i is independent from κ_j, *conditional* on all other components of $\boldsymbol{\kappa}$.

A succinct way to represent conditional independence is using an undirected graph whose nodes correspond to the random variables [5]. A graph edge is present between two graph vertices i and j if the corresponding random variables are *not* conditionally independent from each other, given all the other random variables. Or, expressed differently, the edges between the graph vertices correspond to non-zeros in the precision matrix. In our context, each graph vertex represents a finite element and graph edges are introduced according to geometric adjacency of the finite elements as determined by the mesh. To this end, we define the 1-neighbourhood of a finite element as the union of the element itself and of elements sharing a node with the element. The n-neighbourhood is defined recursively as the union of all 1-neighbourhoods of all the elements in the $(n-1)$-neighbourhood. We introduce an edge between two graph vertices when the respective elements are in the same n-neighbourhood.

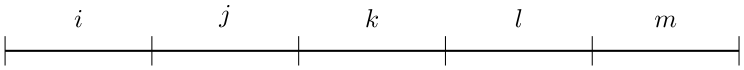
Fig. 1 shows examples of adjacency graphs and the structure of the corresponding precision matrices $\boldsymbol{Q}$ for 5 random variables resulting from a discretisation of a 1D domain with 5 finite elements. In the considered examples the random variables represent the constant log-diffusion coefficient in the elements. As shown in Figs. 1b and 1c choosing a larger n-neighbourhood for graph construction leads to a denser precision matrix. For instance, from the structure of the precision matrix in Fig. 1b, which assumes a 1-neighbourhood structure, we can read for the log-diffusion coefficient of element j the following conditional independence relationship:

$$Q_{ik} = 0 \wedge Q_{il} = 0 \wedge Q_{im} = 0 \Rightarrow p(\kappa_i \mid \kappa_j,\, \kappa_k,\, \kappa_l,\, \kappa_m) = p(\kappa_i \mid \kappa_j)\,. \tag{32}$$

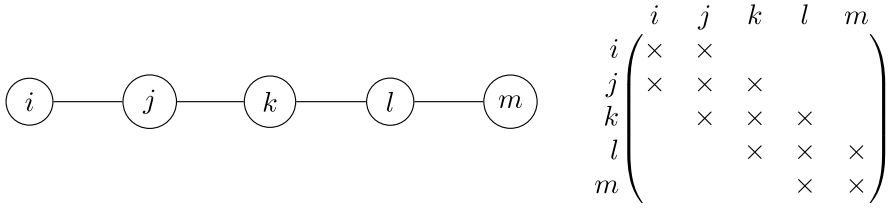
When the coefficient of element j is given, the coefficient of the neighbouring element i is independent from all the remaining coefficients. This is intuitively plausible and in line with physical observations. Clearly, the covariance matrices corresponding to the given sparse precision matrices are dense. Hence, in the considered case the coefficient of element i may still be correlated to the coefficient of element m, i.e. $p(\kappa_i \mid \kappa_m) \neq p(\kappa_i)$. This correlation will most likely be relatively weak given the large distance between the two elements, but knowing the coefficient of element m will certainly restrict the range of possible values for the coefficient of element i.

After obtaining the structure of the precision matrix, which is sparse but, in general, not banded, one can reorder the numbering of the elements in the finite element mesh to reduce its bandwidth. This allows for efficient linear algebra operations. See Cuthill and McKee [79] for an example of a reordering algorithm. Once a minimum bandwidth ordering with $b_{\min}$ 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 $\boldsymbol{Q}$ [80]. Finally, the parameters we optimise are the components of the lower band of size $b_{\min}$ of matrix $\boldsymbol{L}_Q$, so that the approximating distribution reads

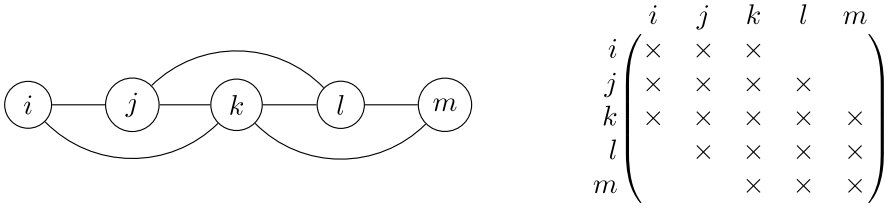
$$q(\boldsymbol{\kappa}) \sim \mathcal{N}\Big(\boldsymbol{\mu}, (\boldsymbol{L}_Q\boldsymbol{L}_Q^\top)^{-1}\Big). \tag{33}$$



(a) Labelling of the five elements.



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



(c) Adjacency graph (left) and the corresponding adjacency matrix (right) based on 2-neighbourhood structure: there is an edge between two vertices if the corresponding elements are in each others 2-neighbourhoods.

Fig. 1. An example of a 1D bar discretised with five elements and two different conditional independence assumptions.

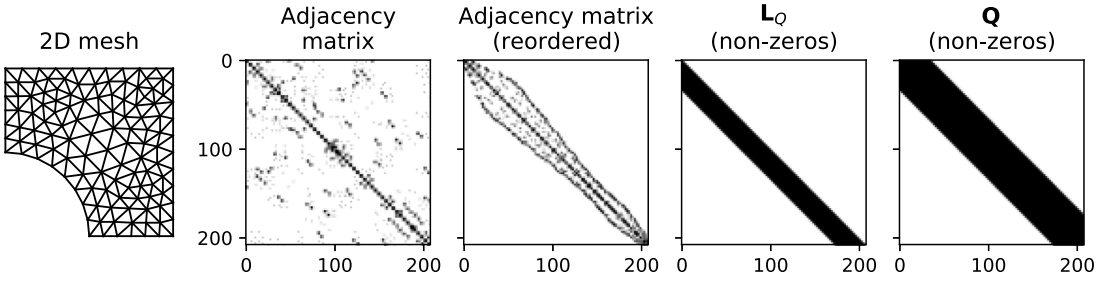


Fig. 2. Sparse precision matrix parametrisation for a 2D problem. A 2-neighbourhood structure is assumed for conditional independence. The structure of the adjacency matrix depends on the specific element numbering. By renumbering the elements, one can obtain a banded adjacency matrix, which is then used to parametrise the Cholesky factor of the precision matrix, as described in Section 3.2.1.

This process of devising a parametrisation for the precision matrix for a more complex mesh in 2D is illustrated in Fig. 2. This approach is computationally efficient – the number of parameters grows as $\mathcal{O}(n_\kappa)$ – and is able to capture dependencies between all the random variables.

3.4. Stochastic optimisation

To maximise the ELBO in (27), we use the ADAM algorithm [81]. ADAM is a member of a larger class of stochastic optimisation methods that have become popular as tools for maximising non-convex cost functions. These methods construct a stochastic estimate of the gradient to perform gradient descent-based optimisation. ADAM, a stochastic gradient descent algorithm with an adaptive step size is one popular algorithm that exhibits a stable

behaviour on many problems and is easy to use without significant tuning. The algorithm uses a per-parameter step size, which is based on the first two moments of the estimate of the gradient for each parameter. Specifically, the step size is proportional to the ratio of the exponential moving average of the 1st moment to the square root of the exponential moving average of the non-centred 2nd moment. At any point, the exponential moving average is computed with decay parameters β_1 and β_2 for the 1st and 2nd moment, respectively. We adopt the parameter values suggested in Kingma and Ba [81]: $\beta_1 = 0.9$ and $\beta_2 = 0.99$. The speed of convergence is further controlled by the learning parameter α which is used to regulate the step size for all parameters in the same way. In our experiments, we set it to 0.01 and let it decay exponentially every 2500 steps (1000 for MFVB), with the decay rate of 0.96. While the ADAM algorithm performs well on a variety of problems, it has been shown that the convergence of this algorithm is poor on some problems [82]. We discuss alternative approaches as potential future work in Section 5.

To monitor convergence, we use 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 optimisation to converge. This rule can be adapted based on the available computational budget.

3.5. The algorithm

The maximisation of the ELBO in (26) involves finding the parameters of the trial distribution $q(\boldsymbol{\kappa})$, i.e. its mean $\boldsymbol{\mu}$ and Cholesky factor $\boldsymbol{L}_{\boldsymbol{Q}}$, that minimise KL divergence between $q(\boldsymbol{\kappa})$ and the posterior $p(\boldsymbol{\kappa}|\mathbf{y})$. Algorithm 1 shows the required steps to compute the ELBO and its gradients with respect to the parameters of the trial distribution. Different from the discussion so far, in Algorithm 1 it is assumed that there are multiple independent observation vectors $\mathbf{y}_i$ with $i \in \{1, 2, \ldots, N_y\}$.

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

Input: Current parameters $\boldsymbol{\mu}$ and $\boldsymbol{L}_{\boldsymbol{Q}}$ of $q(\boldsymbol{\kappa})$
Output: ELBO and its gradients with respect to the parameters of $q(\boldsymbol{\kappa})$
Sample $[\boldsymbol{\kappa}^{(1)}, \boldsymbol{\kappa}^{(2)}, \ldots, \boldsymbol{\kappa}^{(N_{\text{SVI}})}]$ from $q(\boldsymbol{\kappa})$
for *each* $\boldsymbol{\kappa}^{(i)}$ **do**
    Solve for $\mathbf{u}(\boldsymbol{\kappa}^{(i)})$ and obtain gradients with respect to $\boldsymbol{\kappa}$ using the FEM
    $p(y \mid \boldsymbol{\kappa}^{(i)}) \leftarrow \prod_{j=1}^{N_y} p(\mathbf{y}_j \mid \mathbf{u}(\boldsymbol{\kappa}^{(i)}), \sigma_y^2)$ and propagate its gradient with respect to $\boldsymbol{\kappa}^{(i)}$
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 to the parameters of $q(\boldsymbol{\kappa})$ using the reparametrisation trick (see Appendix B.1 and Kingma and Welling [63])
return ELBO, ∇ELBO

4. Examples

We evaluate the efficacy of variational inference first for 1D and 2D Poisson equation examples; a benchmark proposed by Aristoff and Bangerth [83]; and lastly on a multimodal example of the steady-state heat equation. We discretise the examples with a standard finite element method using linear Lagrange basis functions. We compare against two sampling-based inference schemes, Hamiltonian Monte Carlo (HMC) and pre-conditioned Crank–Nicholson Markov Chain Monte Carlo (pCN); both are known to be asymptotically correct as the number of samples increases. The evaluation criteria we use focus on three aspects of an inference scheme: the accuracy with respect to capturing the mean and the variance of the solution; propagation of uncertainty in derived quantities of interest; and the time until convergence of the solution.

To assess the propagation of uncertainty in derived quantities of interest, 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:

$$r(\kappa) = \log \int_{\Gamma_b} e^{\kappa(s)} \nabla u(s) \cdot \boldsymbol{n} \, \mathrm{d}s, \tag{34}$$

where $\boldsymbol{n}$ is a unit vector normal to the boundary Γ_b.

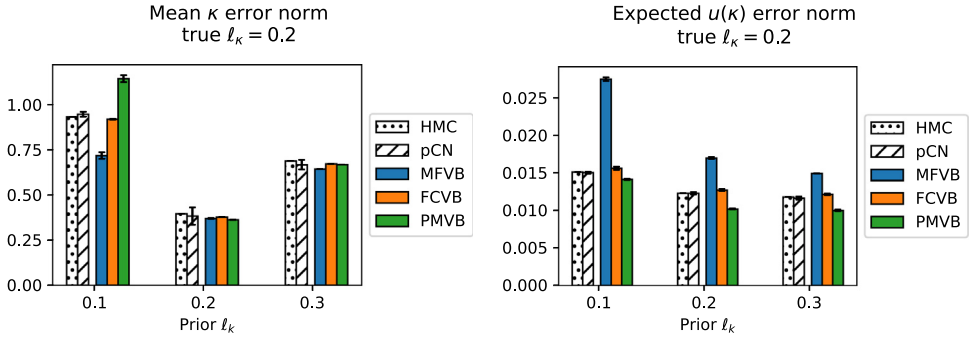


Fig. 3. Mean $\boldsymbol{\kappa}$ error norm for the Poisson 1D problem (left), as defined in (35), and expected solution error norm (right), as defined in (36). Both quantities are estimated using 10,000 samples from the inferred posterior distribution of $\boldsymbol{\kappa}$. Quantitatively, the sampling methods (HMC and pCN) and VB produce comparable results in both metrics, except MFVB parametrisation which captures the mean of $\boldsymbol{\kappa}$ very well, but fails to account for the uncertainty as manifested in high error norm in the solution space. For a qualitative comparison, see Fig. 4 where each row of results corresponds to a different value of the true prior length-scale ℓ_κ.

To quantitatively 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 the mean $\boldsymbol{\kappa}$ error norm. The computation is the Euclidean norm of the error between the true value, $\boldsymbol{\kappa}_{\text{true}}$, and the mean of the obtained samples:

$$\text{Mean } \boldsymbol{\kappa} \text{ error} = \left\| \frac{1}{S} \sum_{s=1}^{S} \boldsymbol{\kappa}^{(s)} - \boldsymbol{\kappa}_{\text{true}} \right\|_2. \tag{35}$$

Further, we compute the expected error in the solution space. This measures how close the solutions corresponding to the samples of $\boldsymbol{\kappa}$ are to the true solution $\mathbf{u}(\boldsymbol{\kappa}_{\text{true}})$. Specifically, we compute

$$\text{Mean } \mathbf{u}(\boldsymbol{\kappa}) \text{ error} = \frac{1}{S} \sum_{s=1}^{S} \left\| \mathbf{u}(\boldsymbol{\kappa}^{(s)}) - \mathbf{u}(\boldsymbol{\kappa}_{\text{true}}) \right\|_2. \tag{36}$$

4.1. Poisson 1D

For this experiment, we assume the unit-line domain, which is discretised into 32 equal-length elements. We impose Dirichlet boundary conditions on both boundaries, specifically we set $u(0) = u(1) = 0$; the forcing is constant everywhere $f(x) = 1$. Unless specified otherwise, all experiments in this section use $N_y = 5$ observations per sensor and the sensor noise $\sigma_y = 0.01$. Sensors are located on each of the discretisation nodes. For the prior on κ, we choose a zero-mean Gaussian process with squared exponential kernel (see Section 2.4 for details). We compare the results for three specifications of the prior length-scale, $\ell_\kappa \in \{0.1, 0.2, 0.3\}$. The length-scale used to generate the data is $\ell_\kappa = 0.2$. For inferences made using data generated by a shorter length-scale, see Appendix A.

4.1.1. VB performs competitively based on error norms

Fig. 3 shows the mean $\boldsymbol{\kappa}$ error norm (35) and the expected solution error norm (36) obtained from 10,000 posterior samples of $\boldsymbol{\kappa}$ from Hamiltonian Monte Carlo (HMC), pre-conditioned Crank–Nicholson MCMC (pCN), as well as VB inference with different parametrisations of the covariance/precision matrix. It is evident that for prior length-scales $\ell_\kappa \in \{0.2, 0.3\}$, the mean $\boldsymbol{\kappa}$ error norms computed by the variational Bayes methods are very close to the estimates from HMC and pCN. For prior $\ell_\kappa = 0.1$, the mean $\boldsymbol{\kappa}$ error norm computed by MFVB is lower than other VB methods and MCMC methods. This is most likely due to MFVB being a much easier optimisation task compared to other VB methods with more optimisation parameters that capture dependencies. For the expected solution error norm, MFVB posterior consistently underestimates the uncertainty in $\boldsymbol{\kappa}$, thus ignoring possible values of $\boldsymbol{\kappa}$ which are consistent with the data. This is further confirmed in the qualitative assessment of uncertainty in the next section.

While MCMC methods are asymptotically correct, in practice, devising efficient samplers for high-dimensional problems within a limited computational budget is still a challenging task and requires substantial hand-tuning. To affirm that all the VB methods provide a good estimate of the mean of $\boldsymbol{\kappa}$, as compared to MCMC methods, is better demonstrated by inspecting Fig. 4 which shows not only the mean but also the posterior uncertainty, which we discuss next.

4.1.2. VB adequately estimates posterior variance

Fig. 4 shows the true values of $\boldsymbol{\kappa}$ (red), the posterior means (black) and plus and minus two times the standard deviation (blue shaded regions) estimated by HMC, pCN, and variational inference with mean-field (MFVB), full covariance (FCVB), and precision matrix (PMVB) parametrisations for different values of prior length-scales.

We observe that the posterior variance estimates computed by HMC, pCN, and full covariance VB are qualitatively very similar, with the estimated uncertainty increasing with increasing distance from the fixed boundary. However, the MFVB solution greatly underestimates posterior variance while computing a reasonable estimate of the posterior mean. The over-confidence of MFVB means that values of $\boldsymbol{\kappa}$ that are consistent with the observed data are ignored; this may lead to poor calibration if the MFVB posterior is used as the true $\boldsymbol{\kappa}$ in downstream tasks or in other contexts. For the PMVB parametrisation, the uncertainty is underestimated to a much lesser extent.

To demonstrate the dependence structure captured by each method, Fig. 5 shows the heatmap of the corresponding precision matrices. Visual inspection suggests that the precision structure inferred using FCVB matches that of MCMC methods while MFVB does not consider covariance relationships by design. The PMVB parametrisation, that takes into account the structure of the problem offers a trade-off between capturing the majority of the correlations in the problem but allowing for more efficient inference due to the sparsity in the resulting matrix. Qualitatively, the PMVB uses only a fraction of the entries in the precision matrix in comparison to the FCVB while consistently achieving a similar ELBO, as demonstrated in Fig. 10.

The observations above are further confirmed by the density plot of our quantity of interest: the log of the total flux on the boundary, shown in Fig. 6. 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 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, pCN, and FCVB schemes. The posterior distribution of FCVB approximately agrees with the MCMC schemes.

For the results obtained using the PMVB scheme, we used the 10-neighbourhood structure to define the adjacency matrix and the non-zero elements of the precision matrix, $\boldsymbol{Q}$ (see Section 3.3). The order of the neighbourhood structure, which corresponds to the precision matrix bandwidth, determines how much dependence within $\boldsymbol{\kappa}$ is captured by the approximating posterior distribution. In Fig. 7, we show how the estimate of the mean and the variance of $\boldsymbol{\kappa}$ changes for different orders of neighbourhood structure. As expected, with the increasing bandwidth, the posterior estimate of $\boldsymbol{\kappa}$ gets closer to the estimate of FCVB, HMC, and pCN (shown in Fig. 4). While there is a significant change in the uncertainty estimate when we increase the bandwidth from 2 to 10, it is less pronounced when we change it from 10 to 20. For this reason, we choose the value of 10 for the PMVB parametrisation in 1D.

4.1.3. VB estimates improve with more observations and decreasing observational noise

The consistency of the posterior refers to the contraction of the posterior distribution to the truth as the data quality increases, *i.e.* either the number of observations increases or observation noise tends to zero. A recent line of work [84–86] showed the posterior consistency for the estimates obtained using popular MCMC schemes such as pCN or unadjusted discretised Langevin algorithm for Bayesian inverse problems based on PDE forward mappings. While similar results are not available for VB methods in infinite-dimensional case, consistency and Bernstein–von Mises type results have been shown for the finite-dimensional case, including Bayesian inverse problems [87,88]. Empirically, our experiments show that for the given family of trial distributions the VB posterior distribution contracts to the true $\boldsymbol{\kappa}$.

Firstly, we show that increasing the number of observations, N_y, results in a more accurate estimate. Given that the observations, $\{\boldsymbol{y}_i\}_{i=1}^{N_y}$, are independent of each other, the likelihood term of the ELBO (see Eq. (26)) is the product of the individual likelihood terms:

$$p(\boldsymbol{y}_1, \dots, \boldsymbol{y}_{N_y} \mid \boldsymbol{\kappa}) = \prod_i^{N_y} p(\boldsymbol{y}_i \mid \boldsymbol{\kappa}). \tag{37}$$

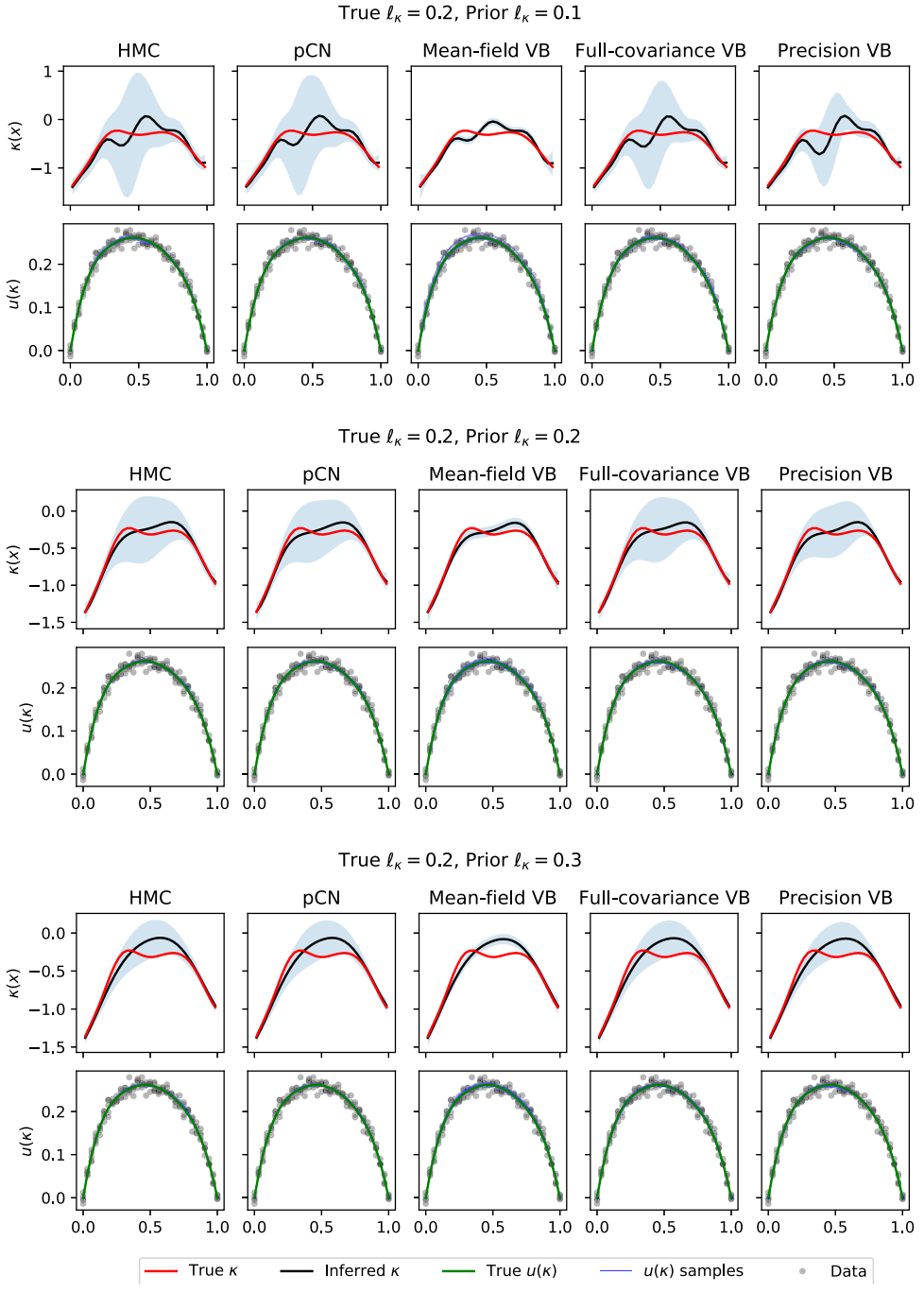


Fig. 4. Top row in each of the three panels show true values of $\boldsymbol{\kappa}$ (red), posterior means (black) and plus and minus two times the standard deviation (blue shaded regions) for HMC, pCN, and VB variants for different values of prior length-scales ℓ_κ. The bottom rows show the data (black), true solution $\mathbf{u}$ (green), solutions for different samples of κ (blue). For the PMVB estimate, the bandwidth is set to 10. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

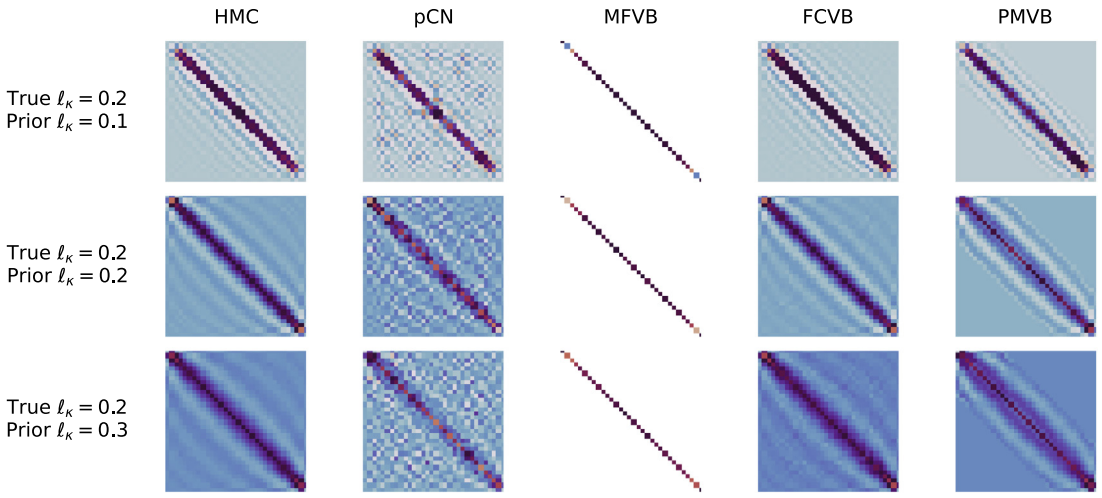


Fig. 5. Precision matrices for each of the considered methods, where true $\ell_\kappa = 0.2$ and each row corresponds to a different value of prior ℓ_κ.

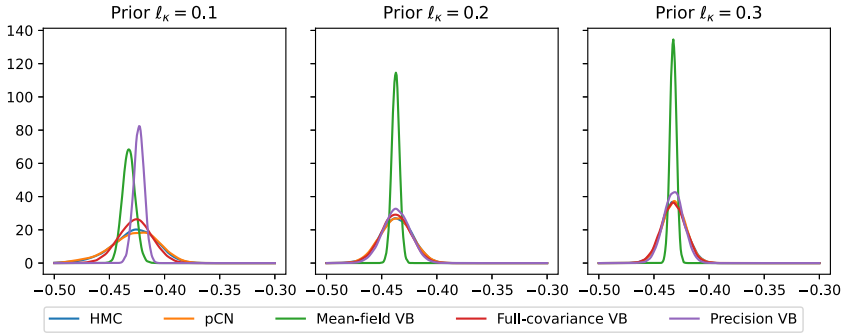


Fig. 6. Log of the boundary flux at the left boundary node ($x = 0$) for the 1D Poisson example. For PMVB, the precision matrix bandwidth of 10 is used.

Secondly, by decreasing the observational noise σ_y we expect the posterior distribution to get closer to the ground truth and with lower uncertainty. Fig. 8 shows the true values of $\boldsymbol{\kappa}$ (red), the posterior mean estimates (black) and plus and minus two times the standard deviation (blue shaded regions) obtained by different variants of variational Bayes for varying 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. 8) or the amount of observational noise (bottom panel in Fig. 8). However, the FCVB and PMVB uncertainty estimates get narrower with increasing number of observations and with decreasing observational noise, which is a desirable behaviour 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 parametrisation. This is not the case for the mean-field VB, providing another indication of uncertainty underestimation for this parametrisation.

4.1.4. VB 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. Table 1 provides the run-times for HMC, MFVB, FCVB, and PMVB. For the HMC column, we also report (shown in brackets) the range of effective sample sizes (ESS) across different components of $\boldsymbol{\kappa}$. For details on ESS, we refer the reader to [30, Ch. 11]. Even with conservative convergence criteria (described in Section 3.4), the computational cost of VB algorithms is up to 25 times lower than that of HMC. To emphasise the computational efficiency of the variational inference, we show the posterior estimates for

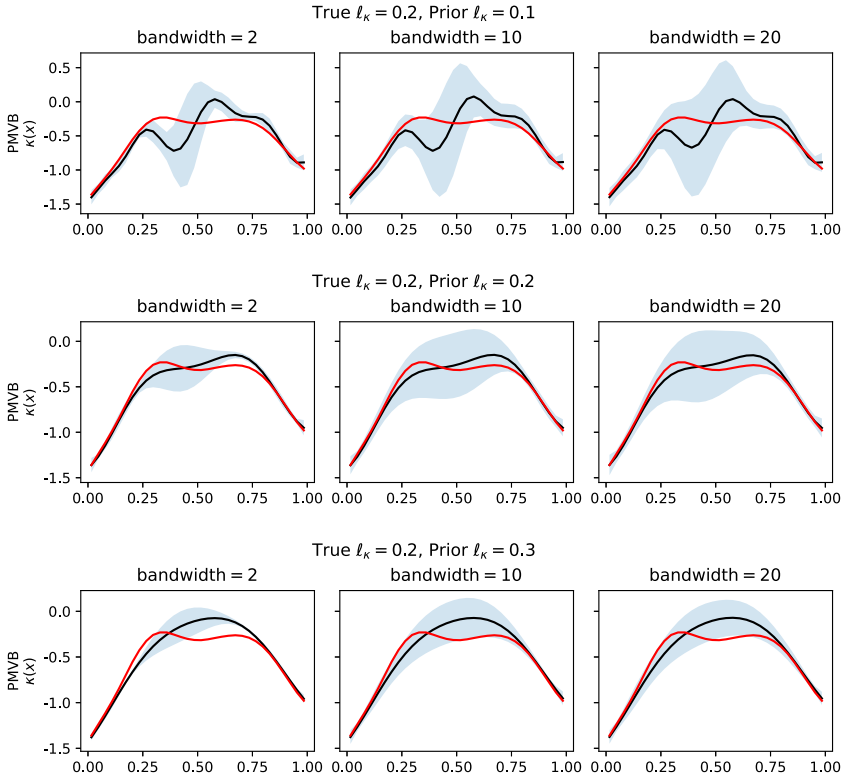


Fig. 7. True values of $\boldsymbol{\kappa}$ (red), posterior means (black) and plus and minus two times the standard deviation (blue shaded region) for different matrix bandwidths of the precision matrix parametrisation of VB. Bandwidth corresponds to the order of neighbourhood structure considered when parametrising $\boldsymbol{Q}$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

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

True ℓ_κ	Prior ℓ_κ	Time (h)				
		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

different number of Monte Carlo samples in the estimation of ELBO. Fig. 9 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 optimisation scheme. Fig. 10 shows that for the FCVB and PMVB parametrisations, where the number of optimised parameters is larger than for MFVB, increasing the number of SVI samples may speed up the convergence of the optimisation. The effect is not as strong for the MFVB parametrisation.

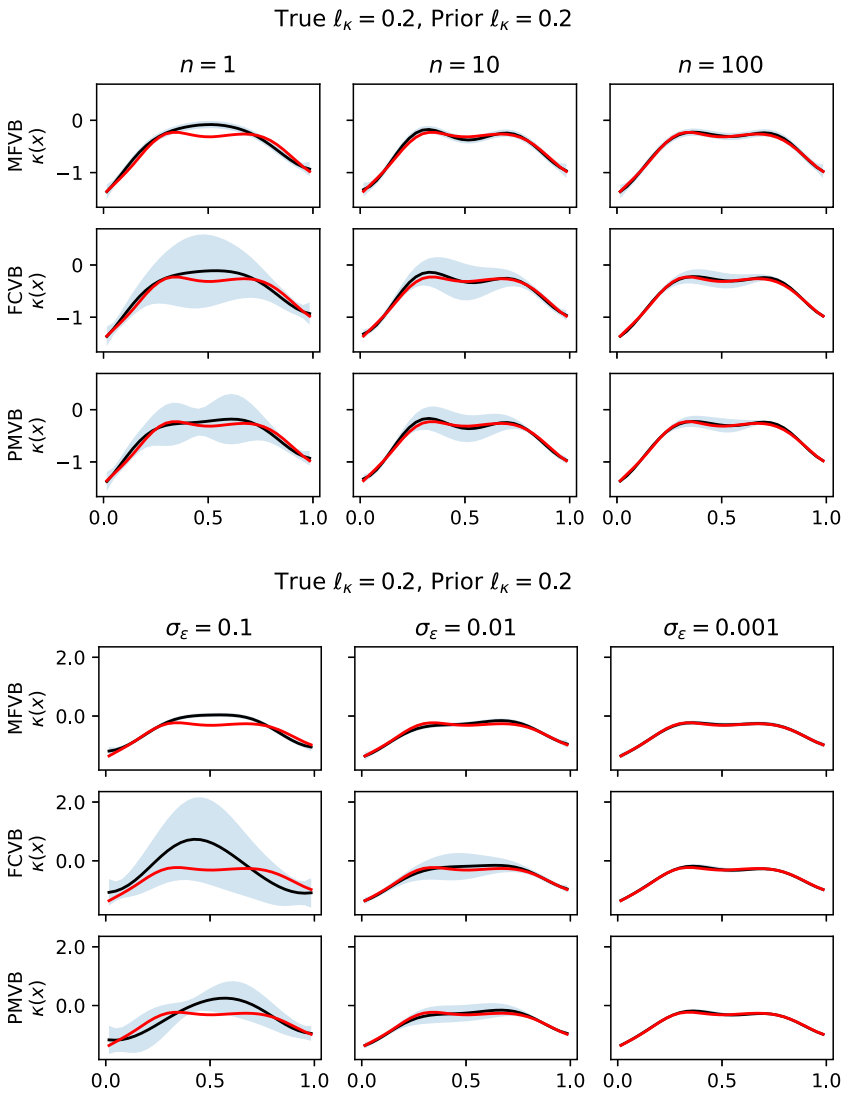


Fig. 8. True values of $\boldsymbol{\kappa}$ (red), posterior means (black) and plus and minus two times the standard deviation (blue shaded regions) for VB with different parametrisations for different number of observations per sensor, $N_y \in \{1, 10, 100\}$ (top panel), and for different values of sensor noise $\sigma_\epsilon \in \{0.1, 0.01, 0.001\}$ (bottom panel). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.2. Poisson 2D

We consider a 2D Poisson problem on the unit-square domain with a circular hole as shown in Fig. 11, with boundary conditions as indicated in the same figure. The problem is discretised with 208 linear triangular elements and 125 nodes. The forcing term is assumed to be constant throughout the domain, $f(\boldsymbol{x}) = 1$. Unless specified otherwise, all experiments in this section use $N_y = 5$ 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. As in the 1D example, we assume a zero-mean GP prior on κ with square exponential kernel with varying length-scale, ℓ_κ, as discussed in Section 2.4.

Firstly, the results in Fig. 12 show that the mean $\boldsymbol{\kappa}$ error of VB methods is very similar to the sampling methods (pCN and HMC). Similarly to the 1D case, the expected solution error norm is highest for MFVB estimate, indicating the lack of capturing the possible values of $\boldsymbol{\kappa}$ for which the solutions, $\mathbf{u}(\boldsymbol{\kappa})$, are consistent with the

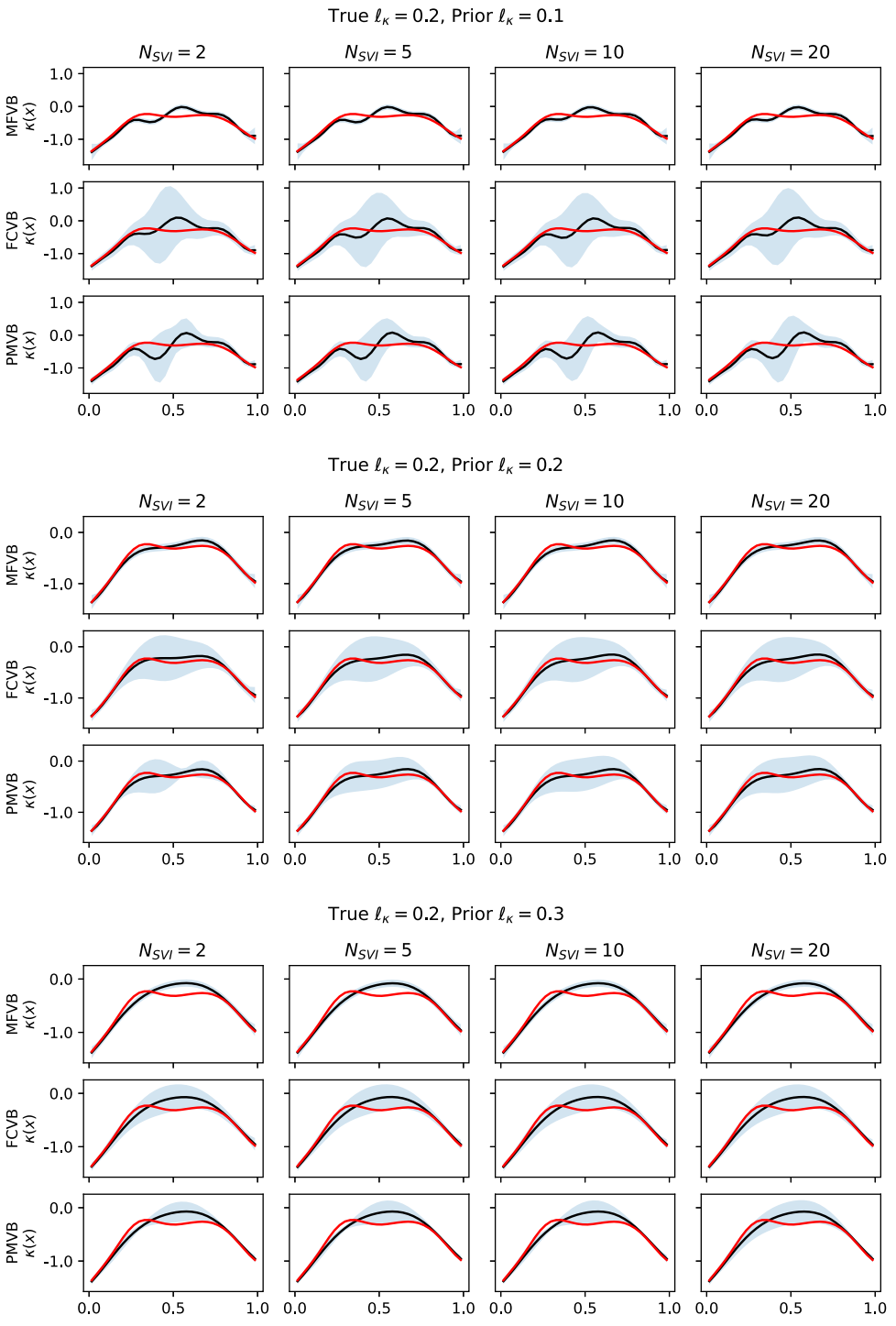


Fig. 9. True values of κ (red), posterior means (black) and plus and minus two times the standard deviation (blue shaded regions) of VB with different parametrisations for varying number of Monte Carlo samples when computing ELBO. Three different length-scales for the prior are shown: 0.1, 0.2, 0.3. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

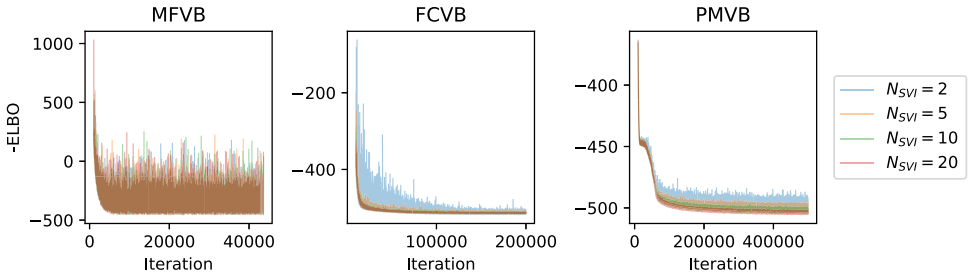


Fig. 10. Negative ELBO trace plot for both MFVB and FCVB for different values of N_{SVI}. For this example, true $\ell_\kappa = 0.2$ and prior $\ell_\kappa = 0.1$.

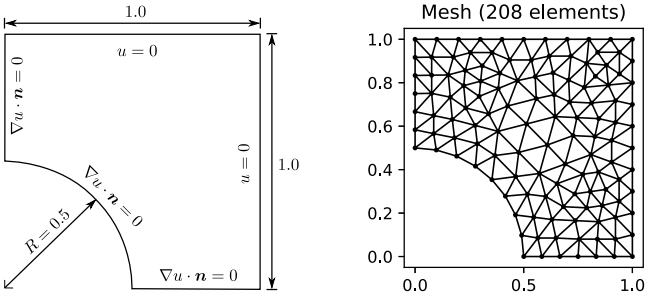


Fig. 11. Left: Specification of the domain for the 2D Poisson problem. Note that we impose Dirichlet boundary conditions $u(x, y) = 0$ when $x = 1$ or $y = 1$. We impose Neumann boundary conditions on the rest of the boundary. Right: a triangular discretisation of the domain.

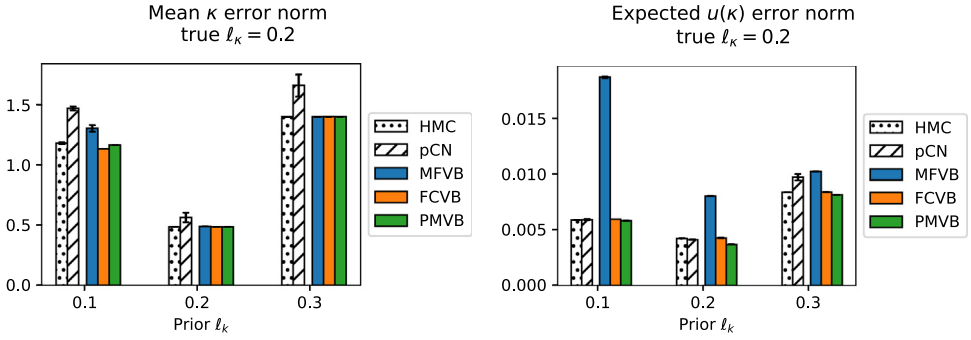


Fig. 12. Mean $\boldsymbol{\kappa}$ error norm for the Poisson 2D problem (left), as defined in (35), and expected solution error norm (right), as defined in (36). Both quantities are estimated using 10,000 samples from the inferred posterior distribution of $\boldsymbol{\kappa}$. Quantitatively, the sampling methods (HMC and pCN) and VB produce comparable results in both metrics, except MFVB parametrisation which captures the mean of $\boldsymbol{\kappa}$ well (as demonstrated in the mean $\boldsymbol{\kappa}$ error norm), but fails to account for the uncertainty as manifested in high error norm in the solution space. For a qualitative comparison, see Figs. 13–15.

observed data. The results also show that both errors are lowest when the prior ℓ_κ matches the length-scale used to generate the data.

Figs. 13–15 show the results for the posterior mean and the standard deviation of $\boldsymbol{\kappa}$, the solution $\mathbf{u}(\boldsymbol{\kappa})$ corresponding to the mean of the posterior. We consider three configurations with prior length-scale $\ell_\kappa \in \{0.1, 0.2, 0.3\}$, where the length-scale used to generate the data is $\ell_\kappa = 0.2$. 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 discussed in Section 4.1, the variance estimates between HMC and FCVB are consistent, especially for longer prior length-scales. There seems to

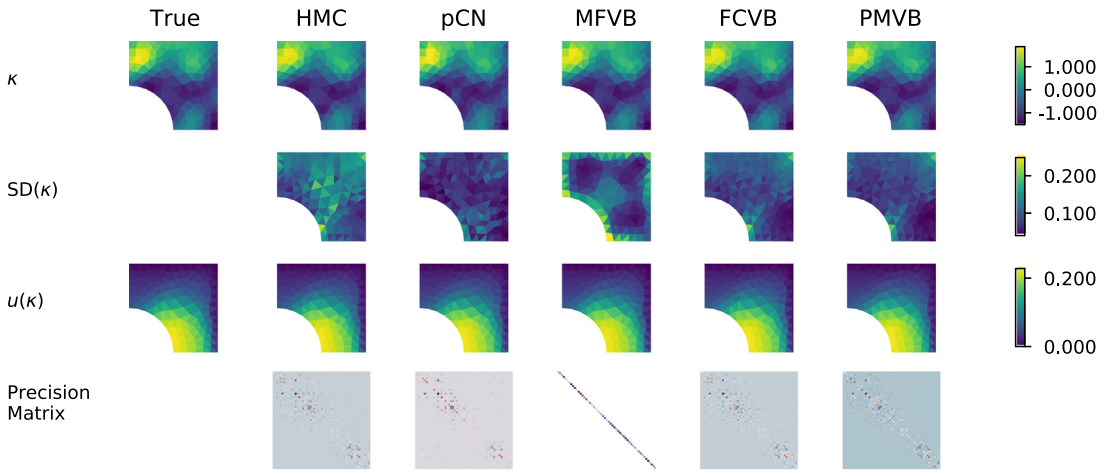


Fig. 13. Posterior mean and standard deviation for $\boldsymbol{\kappa}$ and the corresponding $\mathbf{u}$ for 2D Poisson example with prior length-scale $\ell_\kappa = 0.1$. The bottom row shows the structure of the precision matrix for each inference scheme.

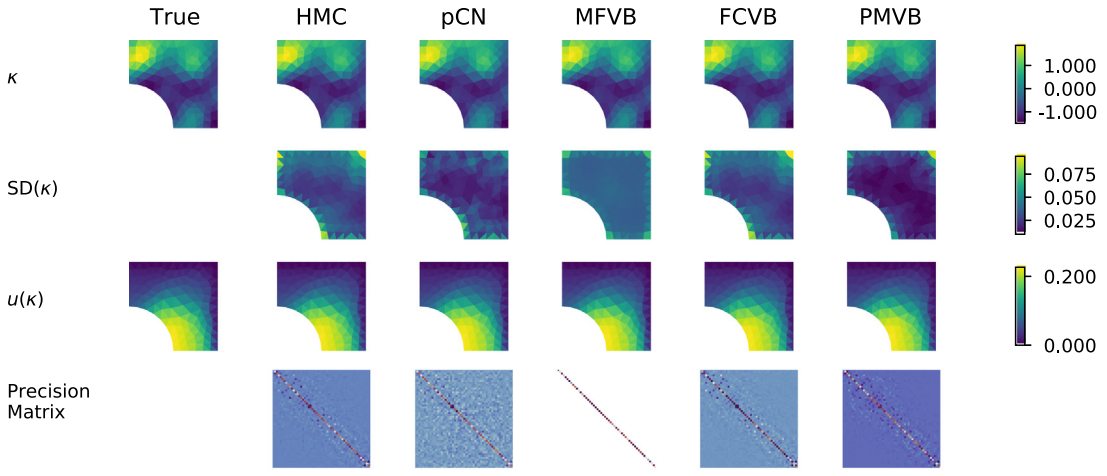


Fig. 14. Posterior mean and standard deviation for $\boldsymbol{\kappa}$ and the corresponding $\mathbf{u}$ for 2D Poisson example with prior length-scale $\ell_\kappa = 0.2$. The bottom row shows the structure of the precision matrix for each inference scheme.

be a discrepancy between the estimates obtained using MFVB and those obtained by other methods. The estimates obtained using precision-matrix parametrisation are qualitatively very close to the FCVB and MCMC estimates.

The bottom rows of Figs. 13–15 show the precision matrices for the inferred posterior distributions. As in the one-dimensional examples, the precision matrices of FCVB and PMVB capture similar dependence structure as the one obtained using HMC, implying that PMVB closes the gap between the over-simplified MFVB and the full covariance VB in terms of the captured dependence relationships while retaining a sparse structure.

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. 16. 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 when prior $\ell_\kappa = 0.1$, where the MFVB estimate is biased as compared to the other methods.

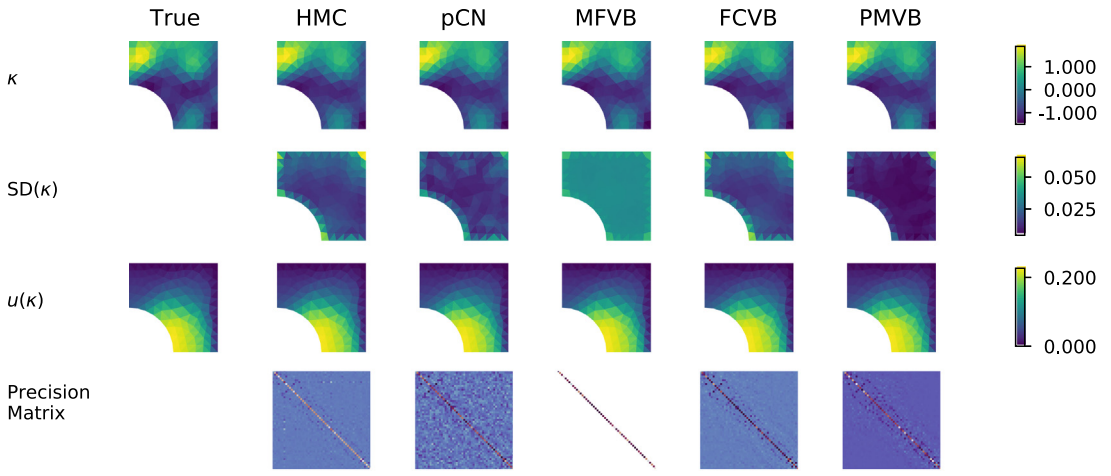


Fig. 15. Posterior mean and standard deviation for $\boldsymbol{\kappa}$ and the corresponding $\mathbf{u}$ for 2D Poisson example with prior length-scale $\ell_\kappa = 0.3$. The bottom row shows the structure of the precision matrix for each inference scheme.

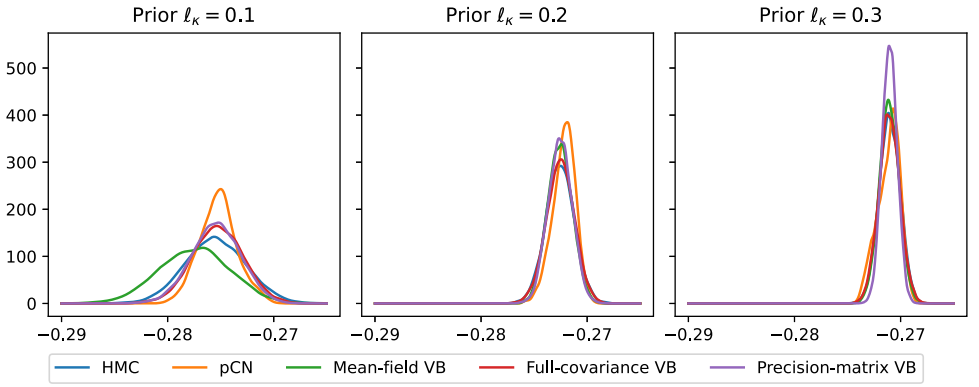


Fig. 16. Log of the total flux computed along the right boundary ($x = 1$). For PMVB, the precision matrix is parametrised using the second-order neighbourhood structure, as shown in Fig. 2.

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 parametrisation.

4.3. Inverse problem benchmark

We evaluate the effectiveness of VB methods on a recently proposed benchmark for Bayesian inverse problems [83]. 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 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 $f(\boldsymbol{x}) = 10$ is constant throughout the domain, and the solution of the PDE is imposed to be zero on all four boundaries.

The benchmark discretises κ using 64 quadrilateral elements, such that κ is constant for each individual element as shown in Fig. 17. The forward solution of the PDE is obtained after discretising u using 32×32 bilinear quadrilateral elements. The locations where the solution is observed are placed on a uniform grid of 169 points

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

True ℓ_κ	Prior ℓ_κ	Time (h)				
		HMC		MFVB	FCVB	PMVB
0.1	0.1	240.6	(930–11 200)	6.4	29.6	28.1
	0.2	295.5	(1537–11 067)	6.6	32.6	28.9
	0.3	242.0	(1057–6068)	7.3	27.3	30.6
0.2	0.1	242.7	(1102–18 235)	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

(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. 17, 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 the 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. 17. 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 are closer to the HMC estimate (our assumed ground truth for the uncertainty) 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 optimisation procedure, which for FCVB involves high-dimensional exploration.

4.4. Multimodal Poisson 1D

One of the advantages of VB is the flexibility of the choice of the trial distribution. To illustrate this, we consider the Poisson equation on the domain $\Omega = (0,\ 1)$ given by

$$\exp(\kappa)\nabla^2 u(x) = 2, \tag{38}$$

where $\exp(\kappa)$ is the conductivity, and the Dirichlet boundary conditions are $u(0) = 0$ and $u(1) = u_R$.

We are interested in inferring the constant conductivity, $\exp(\kappa)$, and the right boundary condition u_R, having obtained multiple measurements of $u(x)$ at $x = 0.5$. We show the solution of this problem in the top part of Fig. 18 where we consider two different combinations of κ and u_R that result in the same solution $u(0.5)$. This implies that there are multiple combinations of the two unknown parameters that result in the same solution at the observation point, making the inference problem ill-posed.

To proceed, we place a prior distribution which is motivated by the domain knowledge: if conductivity is high, so will be the solution u_R at the right boundary. A mixture model provides a convenient way of encoding this prior information in a probability distribution. Specifically, we place the following prior consisting of two bivariate Gaussian distributions on the log of conductivity and the log of boundary condition u_R:

$$\begin{pmatrix} \kappa \\ \log u_R \end{pmatrix} = \frac{1}{2}\mathcal{N}\left(\begin{pmatrix} \log 0.1 \\ \log 1 \times 10^{-8} \end{pmatrix}, 0.5\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right) + \frac{1}{2}\mathcal{N}\left(\begin{pmatrix} \log 2.0 \\ \log 1.0 \end{pmatrix}, 0.5\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right) \tag{39}$$

The contour plot of this prior is shown in the bottom-left part of Fig. 18.

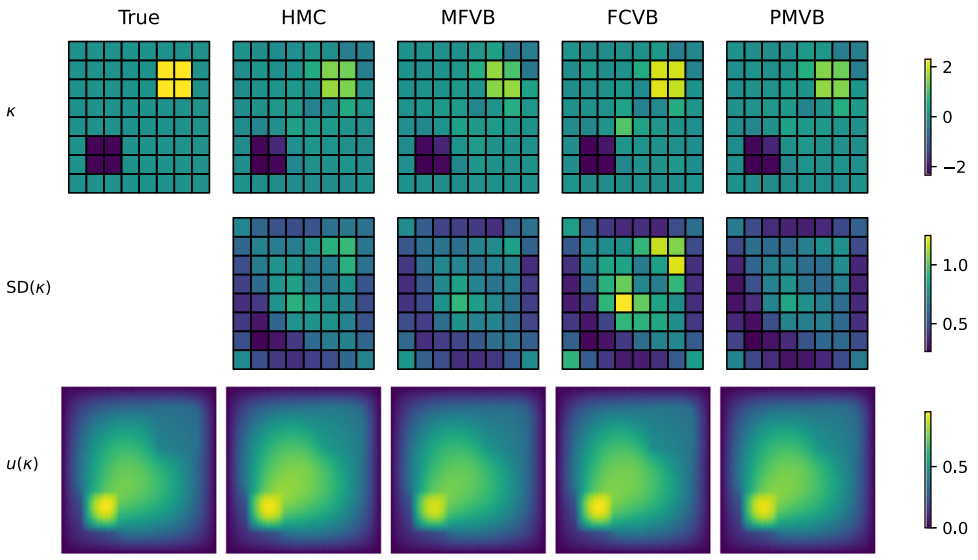


Fig. 17. Posterior mean and standard deviation for $\boldsymbol{\kappa}$ and the corresponding $\mathbf{u}$ for the benchmark example with independent prior for each coefficient of $\boldsymbol{\kappa}$: $\kappa_i \sim \mathcal{N}(0, 1)$.

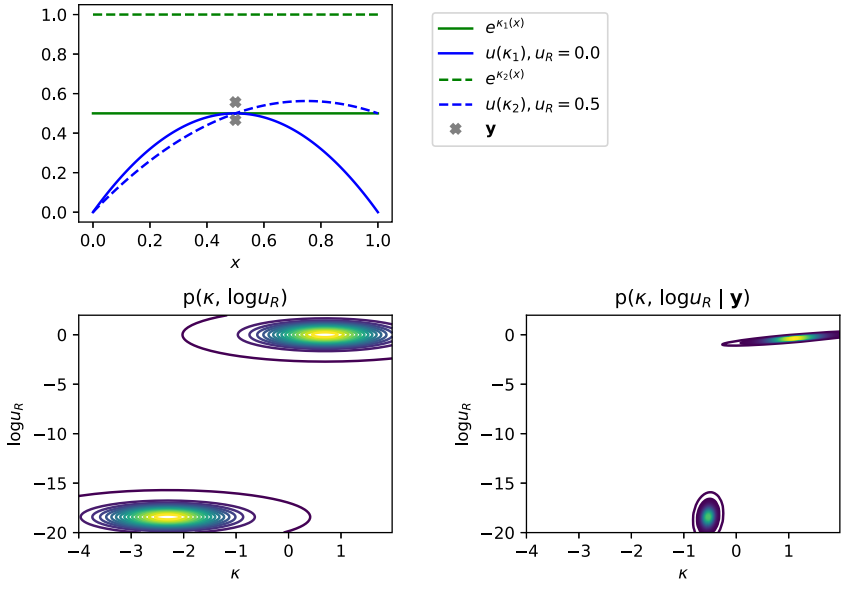


Fig. 18. Multimodal Poisson 1D. *Top*: the solution, u (blue) is shown for two different conductivities and boundary conditions on the right. Two measurements that were taken at the centre of the domain are marked as crosses. *Bottom left*: joint prior distribution for conductivity and the Dirichlet boundary condition on the right at $x = 1$. *Bottom right*: the posterior distribution inferred from the prior distribution and the two measurements (shown in the top panel). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Assuming a Gaussian measurement noise with $\sigma_y = 0.05$, we take two samples of the temperature at the observation point. Following the variational Bayes approach, we restrict the family of trial distributions to be an equally weighted mixture of bivariate Gaussian distributions, each with its own mean and covariance matrix, parametrised by the Cholesky factor. As there is no closed-form expression for the KL divergence between the prior and members of the family of trial distributions, we estimate the KL divergence term in the ELBO using Monte Carlo sampling. As shown in the bottom right panel of Fig. 18, the resulting posterior distribution is bimodal. The distribution is consistent with the physical intuition which we expressed in the prior.

This illustrative example shows that when a proposed model exhibits multi-modality, the flexibility of the variational Bayes methodology allows for specifying a family of trial distributions that can capture that property.

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 optimisation problem. As such, it is naturally integrated with existing FEM solvers, using the gradient calculations from the FEM solvers to optimise the ELBO in VB. Furthermore, the geometry of the problem encoded in the FEM mesh is utilised 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. We note that the inferred variance is important in uncertainty quantification with a probabilistic forward model (for a different load case).

More specifically, our results show that

- the mean of the variational posterior provides an accurate point estimate irrespective of the choice of the parametrisation of the covariance structure,
- the variational approximation with a full-covariance or precision matrix structure adequately estimates posterior uncertainty when compared to HMC and pCN which are known to be asymptotically correct,
- parametrising 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 Bayes 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 estimate various quantities of interest, and
- variational Bayes method is flexible enough to model multimodal posteriors, as illustrated on the steady-state heat equation.

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 existing 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 and taking advantage of low-dimensional projections has potential to further improve computational efficiency [89,90]. Thirdly, the results provided in this paper use standard off-the-shelf optimisation routines; further computational improvements may be achieved using customised algorithms. As a further extension, in some applications it may be informative to consider the uncertainty in the forcing function so that the forward mapping is stochastic, as discussed in [27]. 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.

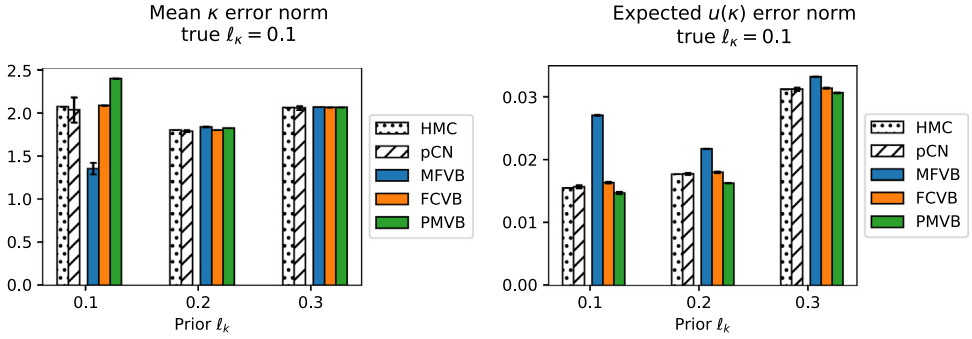


Fig. A.19. Mean $\boldsymbol{\kappa}$ error norm for the Poisson 1D problem (left), as defined in (35), and expected solution error norm (right), as defined in (36). Both quantities are estimated using 10,000 samples from the inferred posterior distribution of $\boldsymbol{\kappa}$. Quantitatively, the sampling methods (HMC and pCN) and VB produce comparable results in both metrics, except MFVB parametrisation which captures the mean of $\boldsymbol{\kappa}$ very well, but fails to account for the uncertainty as manifested in high error norm in the solution space. For a qualitative comparison, see Fig. 4 where each row of results corresponds to a different value of the true prior length-scale ℓ_κ.

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 $\boldsymbol{\kappa}$.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We thank William Baldwin for helpful discussions on the bimodal example and Garoe Dorta for his help with Tensorflow debugging. JP, IK, and MG were supported by the EPSRC, United Kingdom grant EP/P020720/2 for Inference, COmputation and Numerics for Insights into Cities (ICONIC, https://iconicmath.org/). JP was also supported by the EPSRC, United Kingdom grant EP/L015129/1. MG was supported by EPSRC, United Kingdom 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, United Kingdom. FC and EF were supported by the EPSRC, United Kingdom grant EP/T001569/1, particularly the "Digital twins for complex engineering systems"' theme within that grant.

Appendix A. Short length-scale results

Figs. A.19–A.21 show the performance of the proposed method on data generated using a short length-scale. The equivalent plots for a longer length-scale are shown in the main text.

Appendix B. Variational inference

B.1. Reparametrisation trick

Reparametrisation 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, parametrised 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

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

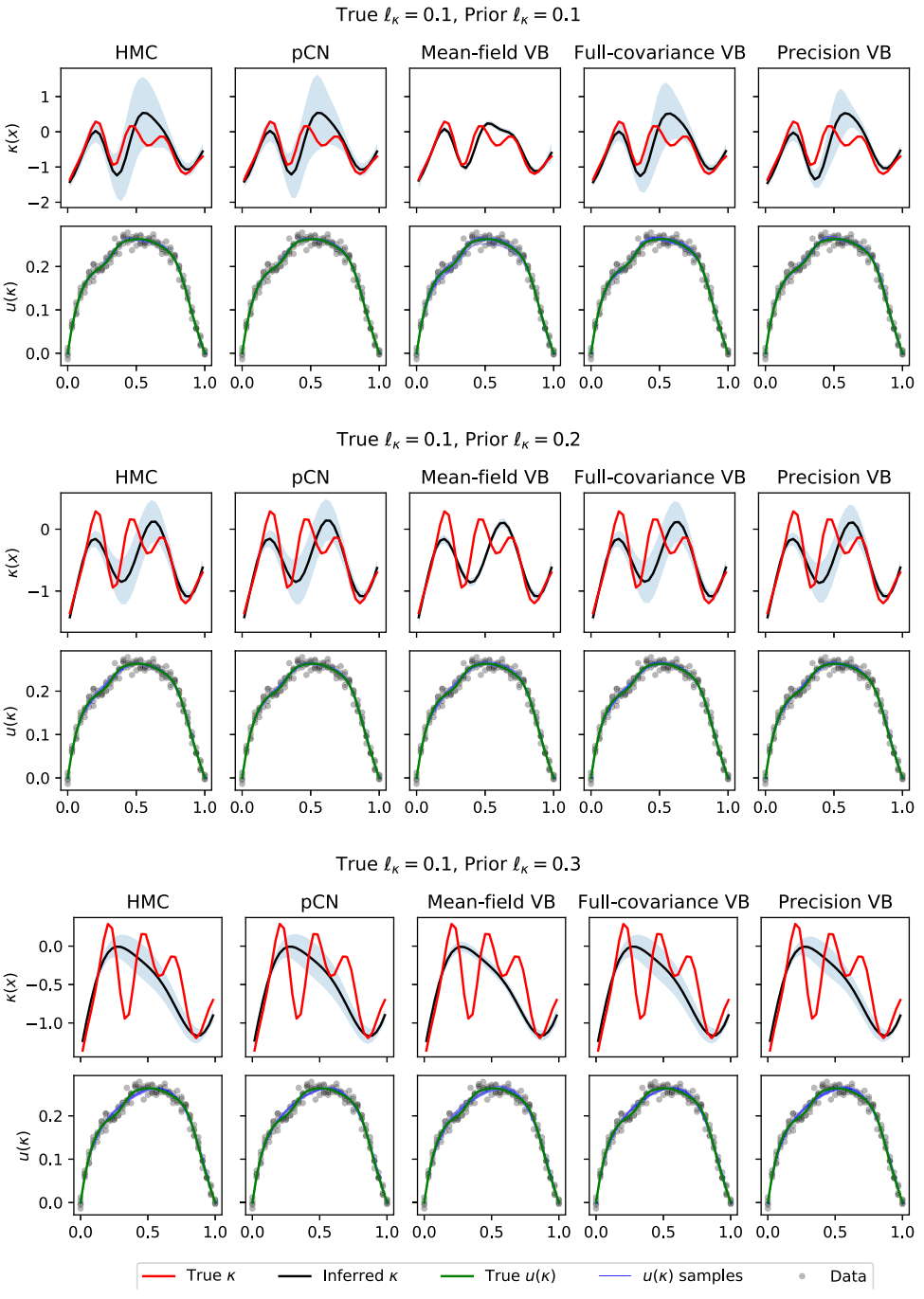


Fig. A.20. Top row in each of the three panels show true values of $\kappa(x)$ (red), posterior means (black) and posterior variances (blue shaded regions) for HMC and VB variants for different values of prior length-scales ℓ_κ. The bottom rows show the data (black), true solution $\mathbf{u}$ (green), solutions for different samples of κ (blue). For the PMVB estimate, the bandwidth is set to 10. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

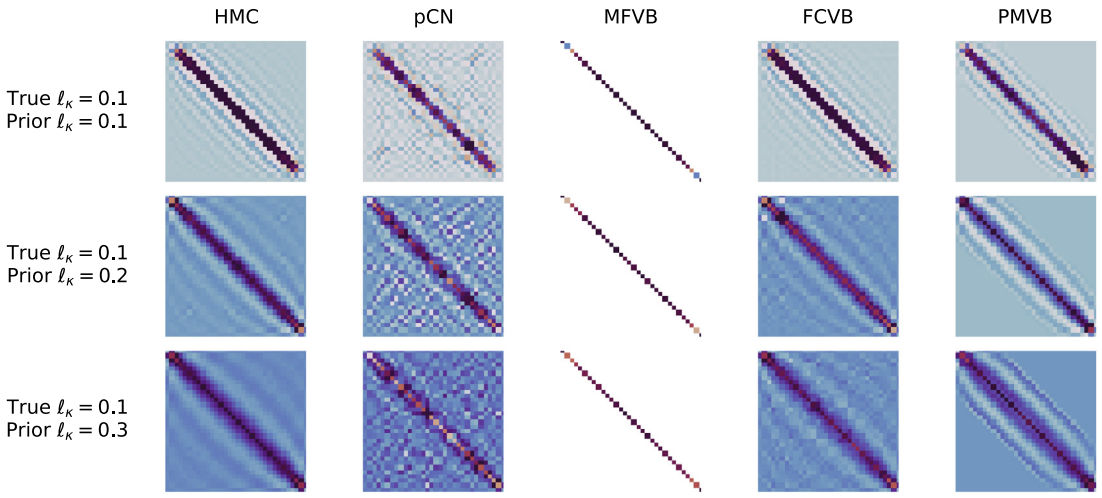


Fig. A.21. Precision matrices for each of the considered methods, where true $\ell_\kappa = 0.1$ and each row corresponds to a different value of prior ℓ_κ.

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,

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

we can use the chain rule of differentiation to obtain

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

Appendix C. Markov Chain Monte Carlo

C.1. Pre-conditioned Crank–Nicholson scheme

We consider the pre-conditioned Crank–Nicholson scheme proposed by Cotter et al. [66]. We summarise the procedure in Algorithm 2.

C.2. Hamiltonian Monte Carlo

Hamiltonian Monte Carlo [72] is a variant of Metropolis–Hastings [70,71] 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 Appendix B.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 or 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 $\mathcal{N}(\mathbf{0}, \mathbf{M})$, where $\mathbf{M}$ is the so called 'mass' matrix. A diagonal matrix

Algorithm 2: PRE-CONDITIONED CRANK–NICHOLSON MCMC [66]

Input: $\Phi(\kappa, \boldsymbol{y}) = -\log p(\boldsymbol{y} \mid \kappa)$: likelihood of the data, $\mu_0(\kappa)$: prior measures, 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.

Output: A list of samples from $\mu^y(\kappa)$.

for $t \leftarrow 1, 2, \ldots$ **do**

  Sample $\xi^{(t)} \sim \mu_0(\kappa)$

  $v^{(t)} \leftarrow \sqrt{\left(1-\beta^2\right)}\kappa^{(t)} + \beta\xi^{(t)}$

  $\kappa^{(t+1)} \leftarrow \begin{cases} v^{(t)} & \text{with probability } \min\left(1, \exp\left(\Phi(\kappa^{(t)}; \boldsymbol{y}) - \Phi(v^{(t)}; \boldsymbol{y})\right)\right) \\ \kappa^{(t)} & \text{otherwise} \end{cases}$

return $[\kappa^{(1)}, \kappa^{(2)}, \ldots]$

Algorithm 3: HAMILTONIAN MONTE CARLO as presented in Gelman et al. [30]

Input: $p(\boldsymbol{\kappa} \mid \boldsymbol{y})$: unnormalised target density, $p(\boldsymbol{\phi})$: momentum density and its mass matrix $\mathbf{M}$, L: leapfrog steps, ϵ: scaling factor

Output: A list of samples from $p(\boldsymbol{\kappa} \mid \boldsymbol{y})$.

for $t \leftarrow 1, 2, \ldots$ **do**

  Sample $\boldsymbol{\phi}$ from $p(\boldsymbol{\phi})$

  for $i \leftarrow 1$ ***to*** L **do**

    $\boldsymbol{\kappa}^* \leftarrow \boldsymbol{\kappa}^{t-1}$

    $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \frac{1}{2}\epsilon \frac{d \log p(\boldsymbol{\kappa}^* | \boldsymbol{y})}{d\boldsymbol{\kappa}}$

    $\boldsymbol{\kappa}^* \leftarrow \boldsymbol{\kappa}^* + \epsilon M^{-1}\boldsymbol{\phi}$

    $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \frac{1}{2}\epsilon \frac{d \log p(\boldsymbol{\kappa}^* | \boldsymbol{y})}{d\boldsymbol{\kappa}}$

    $r \leftarrow \frac{p(\boldsymbol{\kappa}^* | y) p(\boldsymbol{\phi}^*)}{p(\boldsymbol{\kappa}^{t-1} | \boldsymbol{y}) p(\boldsymbol{\phi}^{t-1})}$

    $\boldsymbol{\kappa}^t \leftarrow \begin{cases} \boldsymbol{\kappa}^* & \text{with probability } \min(r, 1) \\ \boldsymbol{\kappa}^{t-1} & \text{otherwise} \end{cases}$

return $[\boldsymbol{\kappa}^1, \boldsymbol{\kappa}^2, \ldots]$

is often chosen to be able to efficiently sample from the momentum distribution. The full steps of the procedure are given in Algorithm 3.

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 scaling factor of the leapfrog step, ϵ, and (iii) the number of leapfrog steps, L. Gelman et al. [30] 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, $(\text{Cov}(\boldsymbol{\kappa} \mid \boldsymbol{y}))^{-1}$. This can be achieved by a pre-run from which the empirical covariance matrix can be computed.

References

[1] A. Tarantola, Inverse Problem Theory and Methods for Model Paramenter Estimation, Society for Industrial and Applied Mathematics, Philadelphia, PA, ISBN: 978-0-89871-792-1, 2005, 978-0-89871-572-9.

[2] J. Kaipio, E. Somersalo, Statistical and Computational Inverse Problems, Springer, New York, ISBN: 978-0-387-22073-4, 2005.

[3] A.M. Stuart, Inverse problems: A Bayesian perspective, Acta Numer. 19 (2010) 451–559, http://dx.doi.org/10.1017/S0962492910000061.

[4] A.N. Tikhonov, V.Y. Arsenin, Solutions of Ill-Posed Problems, in: Scripta Series in Mathematics, Winston, ISBN: 978-0-470-99124-4, 1977.

[5] C.M. Bishop, Pattern Recognition and Machine Learning, in: Information Science and Statistics, Springer, New York, ISBN: 978-0-387-31073-2, 2006.
[6] A. Abdulle, G. Garegnani, A probabilistic finite element method based on random meshes: A posteriori error estimators and Bayesian inverse problems, Comput. Methods Appl. Mech. Engrg. 384 (2021) http://dx.doi.org/10.1016/j.cma.2021.113961.
[7] P. Pandita, P. Tsilifis, N.M. Awalgaonkar, I. Bilionis, J. Panchal, Surrogate-based sequential Bayesian experimental design using non-stationary Gaussian processes, Comput. Methods Appl. Mech. Engrg. 385 (2021) http://dx.doi.org/10.1016/j.cma.2021.114007.
[8] S. Pyrialakos, I. Kalogeris, G. Sotiropoulos, V. Papadopoulos, A neural network-aided Bayesian identification framework for multiscale modeling of nanocomposites, Comput. Methods Appl. Mech. Engrg. 384 (2021) http://dx.doi.org/10.1016/j.cma.2021.113937.
[9] P. Ni, J. Li, H. Hao, Q. Han, X. Du, Probabilistic model updating via variational Bayesian inference and adaptive Gaussian process modeling, Comput. Methods Appl. Mech. Engrg. 383 (2021) http://dx.doi.org/10.1016/j.cma.2021.113915.
[10] C. Sabater, O. Le Maître, P.M. Congedo, S. Görtz, A Bayesian approach for quantile optimization problems with high-dimensional uncertainty sources, Comput. Methods Appl. Mech. Engrg. 376 (2021) http://dx.doi.org/10.1016/j.cma.2020.113632.
[11] Y. Huang, J.L. Beck, H. Li, Y. Ren, Sequential sparse Bayesian learning with applications to system identification for damage assessment and recursive reconstruction of image sequences, Comput. Methods Appl. Mech. Engrg. 373 (2021) http://dx.doi.org/10.1016/j.cma.2020.113545.
[12] A. Ibrahimbegovic, H.G. Matthies, E. Karavelić, Reduced model of macro-scale stochastic plasticity identification by Bayesian inference: Application to quasi-brittle failure of concrete, Comput. Methods Appl. Mech. Engrg. 372 (2020) http://dx.doi.org/10.1016/j.cma.2020.113428.
[13] A. Tarakanov, A.H. Elsheikh, Optimal Bayesian experimental design for subsurface flow problems, Comput. Methods Appl. Mech. Engrg. 370 (2020) http://dx.doi.org/10.1016/j.cma.2020.113208.
[14] C.A. Michelén Ströfer, X.-L. Zhang, H. Xiao, O. Coutier-Delgosha, Enforcing boundary conditions on physical fields in Bayesian inversion, Comput. Methods Appl. Mech. Engrg. 367 (2020) http://dx.doi.org/10.1016/j.cma.2020.113097.
[15] A.G. Carlon, B.M. Dia, L. Espath, R.H. Lopez, R. Tempone, Nesterov-aided stochastic gradient methods using Laplace approximation for Bayesian design optimization, Comput. Methods Appl. Mech. Engrg. 363 (2020) http://dx.doi.org/10.1016/j.cma.2020.112909.
[16] L. Wu, K. Zulueta, Z. Major, A. Arriaga, L. Noels, Bayesian inference of non-linear multiscale model parameters accelerated by a deep neural network, Comput. Methods Appl. Mech. Engrg. 360 (2020) http://dx.doi.org/10.1016/j.cma.2019.112693.
[17] F. Uribe, I. Papaioannou, W. Betz, D. Straub, Bayesian inference of random fields represented with the Karhunen–Loève expansion, Comput. Methods Appl. Mech. Engrg. 358 (2020) http://dx.doi.org/10.1016/j.cma.2019.112632.
[18] F. Rizzi, M. Khalil, R. Jones, J. Templeton, J. Ostien, B. Boyce, Bayesian modeling of inconsistent plastic response due to material variability, Comput. Methods Appl. Mech. Engrg. 353 (2019) 183–200, http://dx.doi.org/10.1016/j.cma.2019.05.012.
[19] M. Arnst, C. Soize, Identification and sampling of Bayesian posteriors of high-dimensional symmetric positive-definite matrices for data-driven updating of computational models, Comput. Methods Appl. Mech. Engrg. 352 (2019) 300–323, http://dx.doi.org/10.1016/j.cma.2019.04.025.
[20] J. Beck, B.M. Dia, L.F. Espath, Q. Long, R. Tempone, Fast Bayesian experimental design: Laplace-based importance sampling for the expected information gain, Comput. Methods Appl. Mech. Engrg. 334 (2018) 523–553, http://dx.doi.org/10.1016/j.cma.2018.01.053.
[21] W. Betz, I. Papaioannou, J.L. Beck, D. Straub, Bayesian inference with subset simulation: Strategies and improvements, Comput. Methods Appl. Mech. Engrg. 331 (2018) 72–93, http://dx.doi.org/10.1016/j.cma.2017.11.021.
[22] P. Chen, U. Villa, O. Ghattas, Hessian-based adaptive sparse quadrature for infinite-dimensional Bayesian inverse problems, Comput. Methods Appl. Mech. Engrg. 327 (2017) 147–172, http://dx.doi.org/10.1016/j.cma.2017.08.016.
[23] E. Asaadi, P.S. Heyns, A computational framework for Bayesian inference in plasticity models characterisation, Comput. Methods Appl. Mech. Engrg. 321 (2017) 455–481, http://dx.doi.org/10.1016/j.cma.2017.04.017.
[24] Y. Huang, J.L. Beck, H. Li, Bayesian system identification based on hierarchical sparse Bayesian learning and Gibbs sampling with application to structural damage assessment, Comput. Methods Appl. Mech. Engrg. 318 (2017) 382–411, http://dx.doi.org/10.1016/j.cma.2017.01.030.
[25] N. Karathanasopoulos, P. Angelikopoulos, C. Papadimitriou, P. Koumoutsakos, Bayesian identification of the tendon fascicle's structural composition using finite element models for helical geometries, Comput. Methods Appl. Mech. Engrg. 313 (2017) 744–758, http://dx.doi.org/10.1016/j.cma.2016.10.024.
[26] I. Babuška, Z. Sawlan, M. Scavino, B. Szabó, R. Tempone, Bayesian inference and model comparison for metallic fatigue data, Comput. Methods Appl. Mech. Engrg. 304 (2016) 171–196, http://dx.doi.org/10.1016/j.cma.2016.02.013.
[27] M. Girolami, E. Febrianto, G. Yin, F. Cirak, The statistical finite element method (statFEM) for coherent synthesis of observation data and model predictions, Comput. Methods Appl. Mech. Engrg. 375 (2021) http://dx.doi.org/10.1016/j.cma.2020.113533.
[28] C. Lu, X. Tang, Surpassing human-level face verification performance on LFW with Gaussian face, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, in: AAAI'15, AAAI Press, ISBN: 0-262-51129-0, 2015, pp. 3811–3819.
[29] A. Solin, S. Cortes, E. Rahtu, J. Kannala, PIVO: Probabilistic Inertial-Visual Odometry for occlusion-robust navigation, in: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, ISBN: 978-1-5386-4886-5, 2018, pp. 616–625, http://dx.doi.org/10.1109/WACV.2018.00073.
[30] A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, D.B. Rubin, Bayesian Data Analysis, Chapman and Hall/CRC, ISBN: 978-1-4398-9820-8, 2013.
[31] T. Cui, Y. Marzouk, K. Willcox, Scalable posterior approximations for large-scale Bayesian inverse problems via likelihood-informed parameter and state reduction, J. Comput. Phys. 315 (2016) 363–387, http://dx.doi.org/10.1016/j.jcp.2016.03.055.
[32] U. Villa, N. Petra, O. Ghattas, hIPPYlib: an extensible software framework for large-scale inverse problems governed by PDEs: Part I: Deterministic inversion and linearized Bayesian inference, ACM Trans. Math. Software 47 (2) (2021) 1–34, http://dx.doi.org/10.1145/3428447.

[33] T. Bui-Thanh, O. Ghattas, J. Martin, G. Stadler, A computational framework for infinite-dimensional Bayesian inverse problems Part I: The linearized case, with application to global seismic inversion, SIAM J. Sci. Comput. 35 (6) (2013) A2494–A2523, http://dx.doi.org/10.1137/12089586X.
[34] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, L.K. Saul, An introduction to variational methods for graphical models, Mach. Learn. 37 (2) (1999) 183–233, http://dx.doi.org/10.1023/A:1007665907178.
[35] D.M. Blei, A. Kucukelbir, J.D. McAuliffe, Variational inference: A review for statisticians, J. Amer. Statist. Assoc. 112 (518) (2017) 859–877, http://dx.doi.org/10.1080/01621459.2017.1285773.
[36] M.I. Jordan, M.J. Wainwright, Graphical models, exponential families, and variational inference, Found. Trends Mach. Learn. 1 (1–2) (2007) 1–305, http://dx.doi.org/10.1561/2200000001.
[37] C. Williams, M. Seeger, Using the Nyström method to speed up kernel machines, in: T. Leen, T. Dietterich, V. Tresp (Eds.), Advances in Neural Information Processing Systems, Vol. 13, MIT Press, 2001.
[38] L. Csató, M. Opper, Sparse on-line Gaussian processes, Neural Comput. 14 (3) (2002) 641–668, http://dx.doi.org/10.1162/089976602317250933.
[39] M.W. Seeger, C.K.I. Williams, N.D. Lawrence, Fast forward selection to speed up sparse Gaussian process regression, in: C.M. Bishop, B.J. Frey (Eds.), Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, in: Proceedings of Machine Learning Research, vol. R4, PMLR, 2003, pp. 254–261.
[40] J. Quiñonero-Candela, C.E. Rasmussen, A unifying view of sparse approximate Gaussian process regression, J. Mach. Learn. Res. 6 (65) (2005) 1939–1959.
[41] E. Snelson, Z. Ghahramani, Sparse Gaussian processes using pseudo-inputs, in: Y. Weiss, B. Schölkopf, J. Platt (Eds.), Advances in Neural Information Processing Systems, Vol. 18, MIT Press, 2006.
[42] M. Titsias, Variational learning of inducing variables in sparse Gaussian processes, in: D. van Dyk, M. Welling (Eds.), Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics, in: Proceedings of Machine Learning Research, vol. 5, PMLR, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 2009, pp. 567–574.
[43] M. Titsias, Variational Model Selection for Sparse Gaussian Process Regression, Tech. Rep., School of Computer Science, University of Manchester, 2008, p. 20.
[44] J. Hensman, M. Rattray, N.D. Lawrence, Fast variational inference in the conjugate exponential family, in: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2, in: NIPS'12, Curran Associates Inc., Red Hook, NY, USA, 2012, pp. 2888–2896.
[45] M.D. Hoffman, D.M. Blei, C. Wang, J. Paisley, Stochastic variational inference, J. Mach. Learn. Res. 14 (2013) 1303–1347.
[46] J. Hensman, N. Fusi, N.D. Lawrence, Gaussian processes for big data, in: Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, in: UAI'13, AUAI Press, Arlington, Virginia, USA, 2013, pp. 282–290.
[47] C.-A. Cheng, B. Boots, Variational inference for Gaussian process models with linear complexity, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 30, Curran Associates, Inc., 2017.
[48] M. Jankowiak, G. Pleiss, J. Gardner, Parametric Gaussian process regressors, in: H.D. III, A. Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 4702–4712.
[49] G. Pleiss, J. Gardner, K. Weinberger, A.G. Wilson, Constant-time predictive distributions for Gaussian processes, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 80, PMLR, 2018, pp. 4114–4123.
[50] H. Salimbeni, C.-A. Cheng, B. Boots, M. Deisenroth, Orthogonally decoupled variational Gaussian processes, in: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 31, Curran Associates, Inc., 2018.
[51] J. Shi, M. Titsias, A. Mnih, Sparse orthogonal variational inference for Gaussian processes, in: S. Chiappa, R. Calandra (Eds.), Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, in: Proceedings of Machine Learning Research, vol. 108, PMLR, 2020, pp. 1932–1942.
[52] H. Salimbeni, M. Deisenroth, Doubly stochastic variational inference for deep Gaussian processes, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 30, Curran Associates, Inc., 2017.
[53] S. Sun, G. Zhang, J. Shi, R.B. Grosse, Functional variational Bayesian neural networks, in: 7th International Conference on Learning Representations, New Orleans, LA, USA, 2019.
[54] D.R. Burt, S.W. Ober, A. Garriga-Alonso, M. van der Wilk, Understanding variational inference in function-space, in: Third Symposium on Advances in Approximate Bayesian Inference, 2021.
[55] P. Tsilifis, I. Bilionis, I. Katsounaros, N. Zabaras, Computationally efficient variational approximations for Bayesian inverse problems, J. Verif. Valid. Uncertain. Quantif. 1 (3) (2016) http://dx.doi.org/10.1115/1.4034102.
[56] D.A. Barajas-Solano, A.M. Tartakovsky, Approximate Bayesian model inversion for PDEs with heterogeneous and state-dependent coefficients, J. Comput. Phys. 395 (2019) 247–262, http://dx.doi.org/10.1016/j.jcp.2019.06.010.
[57] L.S.L. Tan, D.J. Nott, Gaussian variational approximation with sparse precision matrices, Stat. Comput. 28 (2) (2018) 259–275, http://dx.doi.org/10.1007/s11222-017-9729-7.
[58] N. Durrande, V. Adam, L. Bordeaux, S. Eleftheriadis, J. Hensman, Banded matrix operators for Gaussian Markov models in the automatic differentiation era, in: K. Chaudhuri, M. Sugiyama (Eds.), Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics, in: Proceedings of Machine Learning Research, vol. 89, PMLR, 2019, pp. 2780–2789.
[59] D.J.C. MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press, 2003.

[60] B. Wang, D.M. Titterington, Inadequacy of interval estimates corresponding to variational Bayesian approximations, in: R.G. Cowell, Z. Ghahramani (Eds.), Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, in: Proceedings of Machine Learning Research, vol. R5, PMLR, 2005, pp. 373–380.
[61] R.E. Turner, M. Sahani, Two problems with variational expectation maximisation for time series models, in: A.T. Cemgil, D. Barber, S. Chiappa (Eds.), Bayesian Time Series Models, Cambridge University Press, Cambridge, ISBN: 978-0-521-19676-5, 2011, pp. 104–124.
[62] R. Giordano, T. Broderick, M.I. Jordan, Covariances, robustness, and variational Bayes, J. Mach. Learn. Res. 19 (2018) 1981–2029.
[63] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, in: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, 2014.
[64] A.C. Damianou, M.K. Titsias, N.D. Lawrence, Variational inference for latent variables and uncertain inputs in Gaussian processes, J. Mach. Learn. Res. 17 (42) (2016) 1–62.
[65] C. Zhang, J. Butepage, H. Kjellstrom, S. Mandt, Advances in variational inference, IEEE Trans. Pattern Anal. Mach. Intell. 41 (8) (2019) 2008–2026, http://dx.doi.org/10.1109/TPAMI.2018.2889774.
[66] S.L. Cotter, G.O. Roberts, A.M. Stuart, D. White, MCMC methods for functions: Modifying old algorithms to make them faster, Statist. Sci. 28 (3) (2013) 424–446, http://dx.doi.org/10.1214/13-STS421.
[67] F.J. Pinski, G. Simpson, A.M. Stuart, H. Weber, Algorithms for Kullback–Leibler approximation of probability measures in infinite dimensions, SIAM J. Sci. Comput. 37 (6) (2015) A2733–A2757, http://dx.doi.org/10.1137/14098171X.
[68] A. Beskos, M. Girolami, S. Lan, P.E. Farrell, A.M. Stuart, Geometric MCMC for infinite-dimensional inverse problems, J. Comput. Phys. 335 (2017) 327–351, http://dx.doi.org/10.1016/j.jcp.2016.12.041.
[69] H.Q. Minh, Infinite-dimensional log-determinant divergences between positive definite trace class operators, Linear Algebra Appl. 528 (2017) 331–383, http://dx.doi.org/10.1016/j.laa.2016.09.018.
[70] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. 21 (6) (1953) 1087–1092, http://dx.doi.org/10.1063/1.1699114.
[71] W.K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika 57 (1) (1970) 97–109, http://dx.doi.org/10.2307/2334940.
[72] S. Duane, A.D. Kennedy, B.J. Pendleton, D. Roweth, Hybrid Monte Carlo, Phys. Lett. B 195 (2) (1987) 216–222, http://dx.doi.org/10.1016/0370-2693(87)91197-X.
[73] R. Jordan, D. Kinderlehrer, F. Otto, The variational formulation of the Fokker-Planck equation, SIAM J. Math. Anal. 29 (1) (1998) 1–17, http://dx.doi.org/10.1137/S0036141096303359.
[74] B. Carpenter, A. Gelman, M.D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, A. Riddell, Stan : A probabilistic programming language, J. Stat. Softw. 76 (1) (2017) 1–32, http://dx.doi.org/10.18637/jss.v076.i01.
[75] C. Bishop, N. Lawrence, T. Jaakkola, M. Jordan, Approximating posterior distributions in belief networks using mixtures, in: M. Jordan, M. Kearns, S. Solla (Eds.), Advances in Neural Information Processing Systems, Vol. 10, MIT Press, 1998.
[76] D. Tran, D. Blei, E.M. Airoldi, Copula variational inference, in: C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems, 28, Curran Associates, Inc., 2015.
[77] R. Ranganath, D. Tran, D. Blei, Hierarchical variational models, in: M.F. Balcan, K.Q. Weinberger (Eds.), Proceedings of the 33rd International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 48, PMLR, New York, New York, USA, 2016, pp. 324–333.
[78] H. Rue, S. Martino, N. Chopin, Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations, J. R. Stat. Soc. Ser. B Stat. Methodol. 71 (2) (2009) 319–392, http://dx.doi.org/10.1111/j.1467-9868.2008.00700.x.
[79] E. Cuthill, J. McKee, Reducing the bandwidth of sparse symmetric matrices, in: Proceedings of the 1969 24th National Conference, in: ACM '69, Association for Computing Machinery, New York, NY, USA, 1969, pp. 157–172, http://dx.doi.org/10.1145/800195.805928.
[80] H. Rue, L. Held, Gaussian Markov Random Fields: Theory and Applications, in: Monographs on Statistics and Applied Probability, vol. 104, Chapman & Hall/CRC, Boca Raton, ISBN: 978-1-58488-432-3, 2005.
[81] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Y. Bengio, Y. LeCun (Eds.), International Conference on Learning Representations, San Diego, CA, USA, 2015.
[82] S.J. Reddi, S. Kale, S. Kumar, 'On the convergence of Adam and beyond, in: 6th International Conference on Learning Representations (ICLR), Vancouver, BC, 2018.
[83] D. Aristoff, W. Bangerth, A benchmark for the Bayesian inversion of coefficients in partial differential equations, 2021, arXiv:2102.07263.
[84] K. Abraham, R. Nickl, On statistical Calderón problems, Math. Stat. Learn. 2 (2) (2020) 165–216, http://dx.doi.org/10.4171/MSL/14.
[85] F. Monard, R. Nickl, G.P. Paternain, Statistical guarantees for Bayesian uncertainty quantification in non-linear inverse problems with Gaussian process priors, 2020, arXiv:2007.15892.
[86] M. Giordano, R. Nickl, Consistency of Bayesian inference with Gaussian process priors in an elliptic inverse problem, Inverse Problems 36 (8) (2020) http://dx.doi.org/10.1088/1361-6420/ab7d2a.
[87] Y. Wang, D.M. Blei, Frequentist consistency of variational Bayes, J. Amer. Statist. Assoc. 114 (527) (2019) 1147–1161, http://dx.doi.org/10.1080/01621459.2018.1473776.
[88] Y. Lu, A. Stuart, H. Weber, Gaussian approximations for probability measures on R^{d}, SIAM/ASA J. Uncertain. Quantif. 5 (1) (2017) 1136–1165, http://dx.doi.org/10.1137/16M1105384.
[89] J.B. Nagel, B. Sudret, A unified framework for multilevel uncertainty quantification in Bayesian inverse problems, Probab. Eng. Mech. 43 (2016) 68–84, http://dx.doi.org/10.1016/j.probengmech.2015.09.007.
[90] O. Ghattas, K. Willcox, Learning physics-based models from data: Perspectives from inverse problems and model reduction, Acta Numer. 30 (2021) 445–554, http://dx.doi.org/10.1017/S0962492921000064.