The statistical finite element method (statFEM) for coherent synthesis of observation data and model predictions

Mark Girolami[a,b], Eky Febrianto[a,b], Ge Yin[a,b], Fehmi Cirak[a,b,*]

[a]*Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ, UK*
[b]*The Alan Turing Institute, London, NW1 2DB, UK*

Abstract

The increased availability of observation data from engineering systems in operation poses the question of how to incorporate this data into finite element models. To this end, we propose a novel statistical construction of the finite element method that provides the means of synthesising measurement data and finite element models. The Bayesian statistical framework is adopted to treat all the uncertainties present in the data, the mathematical model and its finite element discretisation. From the outset, we postulate a statistical generating model which additively decomposes data into a finite element, a model misspecification and a noise component. Each of the components may be uncertain and is considered as a random variable with a respective prior probability density. The prior of the finite element component is given by a conventional stochastic forward problem. The prior probabilities of the model misspecification and measurement noise, without loss of generality, are assumed to have zero-mean and known covariance structure. Our proposed statistical model is hierarchical in the sense that each of the three random components may depend on one or more non-observable random hyperparameters with their own corresponding probability densities. We use Bayes rule to infer the posterior densities of the three random components and the hyperparameters from their known prior densities and a data dependent likelihood function. Because of the hierarchical structure of our statistical model, Bayes rule is applied on three different levels in turn. On level one, we determine the posterior densities of the finite element component and the true system response using the prior finite element density given by the forward problem and the data likelihood. In this step, approximating the prior finite element density with a multivariate Gaussian distribution allows us to obtain a closed-form expression for the posterior. On the next level, we infer the hyperparameter posterior densities from their respective priors and the marginal likelihood of the first inference problem. These posteriors are sampled numerically using the Markov chain Monte Carlo (MCMC) method. Finally, on level three we use Bayes rule to choose the most suitable finite element model in light of the observed data by computing the respective model posteriors. We demonstrate the application and versatility of statFEM with one and two-dimensional examples.

Keywords: finite elements, Bayesian inference, Gaussian processes, stochastic PDEs, physics-informed machine learning, data-centric engineering

---

*Corresponding author
Email address:* `f.cirak@eng.cam.ac.uk` (Fehmi Cirak)

1. Introduction

Most engineering systems, such as structures and machines, are designed using deterministic mathematical models, indeed their finite element discretisations, which depend on material, geometry, loading and other parameters with significant uncertainties. Traditionally, these uncertainties have been taken into account through codified safety factors. Although this approach has been perfected over the centuries, it is known, for instance in structural engineering, that the response of the actual system and the model prediction often bear no resemblance to each other [1, Ch. 9]. More accurate and reliable predictions are essential towards the design of more efficient systems and for making rational decisions about their operation and maintenance. To achieve this, the uncertainties in the model parameters must be taken into account [2, 3]; even so, the predicted uncertainties in the model response can be significant rendering them practically useless. Fortunately, modern engineering systems are more and more equipped with sensor networks that continuously collect data for their in-situ monitoring, see e.g. [4, 5]. The available data includes, for example, strains from fibre-optic Bragg sensor networks or digital image correlation, temperatures from infrared thermography and accelerations. Incorporating this readily available measurement data into finite element models provides a means to infer the uncertain true system behaviour.

To this end, we propose a statistical construction of the finite element method, dubbed as statFEM, which allows one to make predictions about the true system behaviour in light of measurement data. Adopting a Bayesian viewpoint, all uncertainties in the data and model parameters are treated as random variables with suitably chosen prior probability densities, which consolidate any knowledge at hand. Starting from some prior probability densities, Bayes rule provides a coherent formalism to determine their respective posterior densities while making use of the likelihood of the observations. The selected model determines the probability, or the likelihood, that the observed data was produced by the model. See, e.g., the books [6–10] for an introduction to Bayesian statistics and data analysis. Following Kennedy and O'Hagan's seminal work on calibration of computer models [11], we decompose the observed data y into three random components, namely a finite element component u, a model misspecification component d and a measurement noise component e, see Figure 1. We refer to this decomposition as the *statistical generating model*, or in short the *statistical model*, and have additional models corresponding to each of the random components, i.e. u, d and e. The three random variables depend in turn on random parameters with corresponding probability densities. Following standard statistics terminology, we refer to the unknown random parameters as *hyperparameters*. Evidently, the proposed statistical construction has an inherent hierarchical structure, see e.g. [9, Ch. 5]. That is, each of the random variables u, d and e depend in turn on a set of random hyperparameters.

We obtain the prior probability density for the finite element component u by solving a traditional probabilistic forward problem in the form of a stochastic partial differential equation. The source, or forcing, term and the coefficients of the differential operator can all be random. Any unknown variables used for parameterising the respective random fields, e.g., for describing their covariance structure, are treated as hyperparameters. We solve the finite element discretised forward problem with a classical first-order perturbation method [12]. For the misspecification component d, we assume a Gaussian process prior, which for the purposes of illustration is assigned a square exponential kernel, and treat the respective covariance parameters as hyperparameters. The measurement noise e is, as usual, assumed to be independent and identically distributed. It is possible to determine with Bayes rule the joint posterior density of all random variables, i.e. u, d and e and their hyperparameters, and to obtain subsequently the posterior densities of the individual random variables by marginalisation. This leads, however, to a costly very high-dimensional infer-
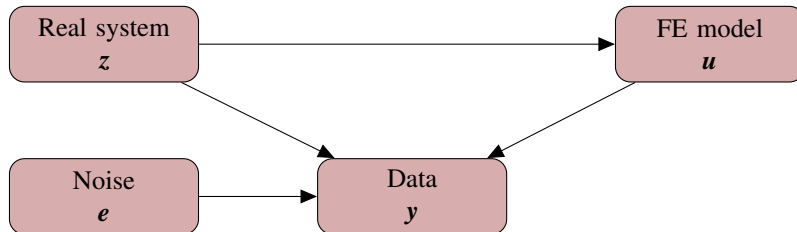
Figure 1: Statistical model underlying statFEM. The observation data y is decomposed as $y = z + e = \rho P u + d + e$, where ρ is a random hyperparameter, P is a suitable projection operator and other variables are all random vectors. Each of the random vectors depend on additional random hyperparameters which have been omitted in this diagram.

ence problem for which advanced sampling schemes are being developed, see e.g. [13–16]. We circumvent the need for costly sampling by using approximate inference, that is, by exploiting the hierarchical structure of the proposed construction and applying Bayes rule on three different levels in turn. The overall approach is akin to the empirical Bayes or evidence approximation techniques prevalent in machine learning, see [17, 18], [19, Ch. 3] and [8, Ch. 5]. Ultimately, we infer, or in other words learn, from the data y the posterior densities of u, d and e and their respective hyperparameters. Moreover, we assess the suitability of different finite element models by computing their Bayes factors [6, 20]. The posteriors are, depending on the level, either analytically approximated or numerically sampled with MCMC. We refer to [21] on mathematical foundations of Bayesian inverse problems for the necessary theory in defining the probability measures required in statFEM.

The seemingly innocuous decomposition of the data y according to Figure 1, as proposed in [11], provides a versatile framework for statistical inference and has been extensively used in the past. The choice of the models for each of the components u, d, and e and the numerical techniques for the treatment of the resulting inference problem leads to a rich set of approaches. In [11] and most subsequent papers, including [22, 23], the component u representing the simulation model is obtained from a given black-box computer code (simulator). The deterministic model response u is approximated with a standard Gaussian process emulator, or surrogate model, [24] using multiple evaluations of the simulator. Obviously, instead of Gaussian process regression any other technique for creating surrogates can be used [25, 26]. The essential advantage of using a surrogate model as a forward model for u is that the inferential framework becomes independent from the complexities of the specific problem considered. Calibration aims to determine the parameters of the deterministic forward model, including its constitutive parameters and forcing and their spatial distributions. However, in many engineering systems the forward model is not deterministic. For instance, the loading of a bridge under operation is inherently random. Similarly, the constitutive parameters of a mechanical part, say a connecting rod in an engine, will indeed be random over its different realisations. The aim of calibration in such cases is, as proposed in this paper, to determine the hyperparameters characterising the random loading or the constitutive parameters. The importance of a random misspecification component d in calibration is meanwhile well-established [27–29].

StatFEM complements the conventional probabilistic finite element method and the Bayesian treatment of inverse problems. As mentioned, we solve the forward problem via a probabilistic finite element method. Over the years, a wide range of techniques have been proposed to solve stochastic partial differential equations using finite elements. They differ in terms of discretisation of the prescribed random inputs, like the forcing or the diffusion coefficient, and the approximation of the solution in the probability domain. For insightful reviews see [30–35]. In this paper, we assume that the random forcing term and the coefficients of the differential operator are both Gaussian processes. When only the forcing is random, the resulting finite element solution is also Gaussian and the solution probability measure can be readily obtained. However, when the coefficients of the differential operator are random, the solution is usually not a Gaussian and it becomes more taxing to solve the forward problem. In addition to the spatial domain the probability domain has to be also discretised or in some way approximated. We choose a perturbation method to approximate the solution in the probability domain [12], but could use any one of the other well-known techniques, like Monte-Carlo [14], Neumann-series [36] or polynomial-chaos expansions with their Galerkin and collocation implementations [37–40]. In contrast to forward problems, inverse problems are considerably more challenging to formulate and to solve, because the same observation can usually be generated by different sets of model parameters. Adopting a Bayesian viewpoint and treating the model parameters as random can resolve the ill-posedness of the inverse problem. See [41–43] for an introduction to Bayesian inversion and [21] for a detailed mathematical analysis. Bayesian inversion is closely related to the calibration framework proposed by Kennedy and O'Hagan [11]. However, in Bayesian inversion literature the model misspecification errors are usually not considered. Different from statFEM, in both Bayesian inversion and [11] the forward model mapping the random parameters to the observations is usually deterministic. Despite the principal differences between statFEM and Bayesian inversion they share a number of algorithmic similarities in terms of implementation. Therefore, many of the numerical techniques developed for efficiently solving large-scale inverse problems, like the treatment of non-local covariance operators [44, 45], representation of random fields [13] or the sampling of posteriors [46], can all be adapted to statFEM.

The outline of this paper is as follows. In Section 2 we review the solution of stochastic partial differential equations with random coefficients and sources with the finite element method. We then introduce in Section 3 the proposed statistical construction of the finite element method. After introducing the underlying statistical generating model we detail the computation of the posterior densities of the finite element solution, the true system response,

the hyperparameters and the finite element model itself. This is followed in Section 4 by the study of one and two-dimensional Poisson problems with the proposed approach. Amongst others, we study the convergence of the computed posterior densities to the true densities used for generating the synthetic observation data with an increasing number of observations. We also illustrate how statFEM reduces to a conventional Bayesian inverse problem when the assumed statistical model is simplified.

2. Probabilistic forward model

The forward problem consists of a stochastic partial differential equation with both a random coefficient and forcing. The coefficient and forcing fields are both assumed to be Gaussian processes. We solve the finite element discretised forward problem with a classical first-order perturbation method so that its solution is also a Gaussian [12]. However, we can use in statFEM any of the established discretisation techniques which can yield the second-order statistics, i.e. mean and covariance, of the solution field; see e.g. the reviews [30–33, 35].

2.1. Governing equations

As a representative stochastic partial differential equation we consider on a domain $\Omega \subset \mathbb{R}^d$ with $d \in \{1, 2, 3\}$ and the boundary $\partial\Omega$ the Poisson equation

$$-\nabla \cdot (\mu(\boldsymbol{x})\nabla u(\boldsymbol{x})) = f(\boldsymbol{x}) \qquad \text{in } \Omega \tag{1a}$$

$$u(\boldsymbol{x}) = 0 \qquad \text{on } \partial\Omega, \tag{1b}$$

where $u(\boldsymbol{x}) \in \mathbb{R}$ is the unknown, $\mu(\boldsymbol{x}) \in \mathbb{R}^+$ is a random diffusion coefficient and $f(\boldsymbol{x}) \in \mathbb{R}$ is a random source term.

To ensure that the diffusion coefficient $\mu(\boldsymbol{x})$ is positive we introduce $\kappa(\boldsymbol{x})$ such that $\mu(\boldsymbol{x}) = \exp(\kappa(\boldsymbol{x}))$. In the following, this new function $\kappa(\boldsymbol{x})$ is referred to as the diffusion coefficient with a slight abuse of terminology. The diffusion coefficient $\kappa(\boldsymbol{x})$ is modelled as a Gaussian process

$$\kappa(\boldsymbol{x}) \sim \mathcal{GP}\left(\overline{\kappa}(\boldsymbol{x}),\, c_\kappa(\boldsymbol{x},\, \boldsymbol{x}')\right) \tag{2}$$

with the mean

$$\mathbb{E}[\kappa(\boldsymbol{x})] = \overline{\kappa}(\boldsymbol{x}) \tag{3}$$

and the covariance

$$\text{cov}\left(\kappa(\boldsymbol{x}),\, \kappa(\boldsymbol{x}')\right) := \mathbb{E}\left[(\kappa(\boldsymbol{x}) - \overline{\kappa}(\boldsymbol{x}))\,(\kappa(\boldsymbol{x}') - \overline{\kappa}(\boldsymbol{x}'))\right] = c_\kappa(\boldsymbol{x},\, \boldsymbol{x}'). \tag{4}$$

Although the specific form of the kernel $c_\kappa(\boldsymbol{x},\, \boldsymbol{x}')$ is inconsequential for the presented approach, we assume for the sake of concreteness a squared exponential kernel of the form

$$c_\kappa(\boldsymbol{x},\, \boldsymbol{x}') = \sigma_\kappa^2 \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\ell_\kappa^2}\right) \tag{5}$$

with the scaling parameter $\sigma_\kappa \in \mathbb{R}^+$ and lengthscale parameter $\ell_\kappa \in \mathbb{R}^+$. However, we stress that this will be a modelling choice in an actual application based on what is understood about the structure and form of the diffusion coefficient.

Similarly, the random source term $f(\boldsymbol{x})$ is modelled as a Gaussian process

$$f(\boldsymbol{x}) \sim \mathcal{GP}\left(\overline{f}(\boldsymbol{x}),\, c_f(\boldsymbol{x},\, \boldsymbol{x}')\right) \tag{6}$$

with the mean

$$\mathbb{E}[f(\boldsymbol{x})] = \overline{f}(\boldsymbol{x}) \tag{7}$$

4

(a) Source $f(x)$
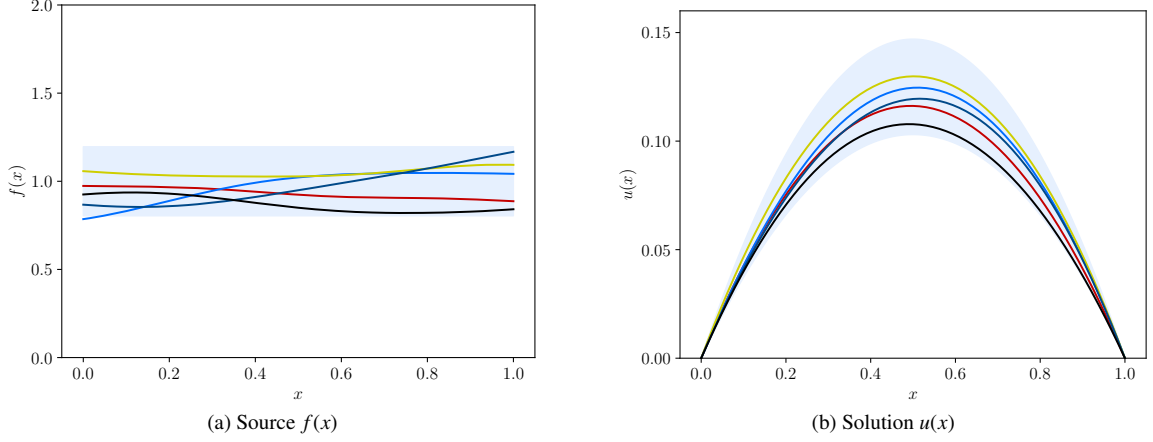
(b) Solution $u(x)$

Figure 2: One-dimensional Poisson problem $\mathrm{d}^2 u / \mathrm{d}x^2 = f(x)$ with a random source with a mean $\overline{f}(x) = 1$ and covariance kernel parameters $\sigma_f = 0.1$ and $\ell_f = 0.4$. The five lines in (a) represent samples drawn from (6) with the corresponding solutions shown in (b). The shaded areas are the 95% confidence regions obtained from (6) and (9), respectively.

and the covariance

$$c_f(\boldsymbol{x}, \boldsymbol{x}') = \mathrm{cov}\left(f(\boldsymbol{x}), f(\boldsymbol{x}')\right) = \mathbb{E}\left[\left(f(\boldsymbol{x}) - \overline{f}(\boldsymbol{x})\right)\left(f(\boldsymbol{x}') - \overline{f}(\boldsymbol{x}')\right)\right] = \sigma_f^2 \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\ell_f^2}\right) \tag{8}$$

with the respective scaling and lengthscale parameters $\sigma_f \in \mathbb{R}^+$ and $\ell_f \in \mathbb{R}^+$.

In Figure 2 an illustrative one-dimensional Poisson example, $\mathrm{d}^2 u / \mathrm{d}x^2 = f(x)$, with a random source and corresponding solution are shown. The diffusion coefficient $\kappa(x) = 1$ is chosen as non-random. The source has the mean $\overline{f}(x) = 1$ and the parameters of the exponential covariance kernel are chosen with $\sigma_f = 0.1$ and $\ell_f = 0.4$. The solution $u(x)$ is given by the push forward measure of the Gaussian process on the forcing, which due to the linearity of the differential operator is also a Gaussian process,

$$u(\boldsymbol{x}) \sim \mathcal{GP}\left(\overline{u}(x), c_u(\boldsymbol{x}, \boldsymbol{x}')\right) = \mathcal{GP}\left(g(\boldsymbol{x}, \boldsymbol{x}') * \overline{f}(\boldsymbol{x}'), g(\boldsymbol{x}, \boldsymbol{x}'') * c_f(\boldsymbol{x}'', \boldsymbol{x}''') * g(\boldsymbol{x}''', \boldsymbol{x}')\right), \tag{9}$$

where $g(\boldsymbol{x}, \boldsymbol{x}')$ is the Greens function of the Poisson problem and $*$ denotes convolution, see e.g. [21, 47]. Due to the smoothing property of the convolution operation the lengthscale of the kernel $c_u(\boldsymbol{x}, \boldsymbol{x}')$ is larger than ℓ_f. Furthermore, the source and the solution are both C^∞ smooth owing to the squared exponential kernel used for the source $f(\boldsymbol{x})$.

2.2. Finite element discretisation

We discretise the weak form of the Poisson equation (1) with a standard finite element approach. Specifically, the domain Ω is subdivided into a set $\{\omega_e\}$ of non-overlapping elements

$$\Omega = \bigcup_{e=1}^{n_e} \omega_e \tag{10}$$

of maximum size

$$h = \max_e \mathrm{diam}(\omega_e). \tag{11}$$

The unknown field $u(\boldsymbol{x})$ is approximated with Lagrange basis functions $\phi_i(\boldsymbol{x})$ and the respective nodal coefficients u_i of the n_u non-Dirichlet boundary mesh nodes by

$$u_h(\boldsymbol{x}) = \sum_{i=1}^{n_u} \phi_i(\boldsymbol{x}) u_i. \tag{12}$$

5

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

$$A(\kappa)u = f, \tag{13}$$

where $A(\kappa) \in \mathbb{R}^{n_u \times n_u}$ is the system matrix, $\kappa \in \mathbb{R}^{n_e}$ is the vector of diffusion coefficients, $u \in \mathbb{R}^{n_u}$ is the vector of nodal coefficients and $f \in \mathbb{R}^{n_u}$ is the nodal source vector.

The diffusion coefficient vector κ is given by the Gaussian process (2). We assume that the diffusion coefficient is constant within each element and collect all element barycentre coordinates in the matrix $X^{(c)} = \left\{ x_e^{(c)} \right\}_{e=1}^{n_e}$. Thus, the diffusion coefficient vector is given by the multivariate Gaussian density

$$\kappa \sim p(\kappa) = \mathcal{N}\left(\overline{\kappa}\left(X^{(c)} \right), C_\kappa\left(X^{(c)}, X^{(c)} \right) \right), \tag{14}$$

where the mean vector $\overline{\kappa}$ and the covariance matrix C_κ are obtained by evaluating the mean (3) and covariance kernel (4) at the respective element barycentres. The system matrix A is assembled from the element system matrices given by

$$A_{ij}^e(\kappa_e) = \int_{\omega_e} \exp(\kappa_e) \frac{\partial \phi_i(x)}{\partial x} \cdot \frac{\partial \phi_j(x)}{\partial x} \, d\omega_e, \tag{15}$$

where κ_e is the diffusion coefficient of the element with the index e. And, the components of the source vector are given by

$$f_i = \int_\Omega f(x)\phi_i(x) \, d\Omega. \tag{16}$$

As introduced in (6), the source is a Gaussian process with the expectation (7) and covariance (8). This implies for the components of the source vector, owing to the linearity of expectation (see e.g. [8]) and Fubini's theorem,

$$\mathbb{E}[f_i] = \mathbb{E}\left[\int_\Omega f(x)\phi_i(x) \, d\Omega \right] = \int_\Omega \mathbb{E}\left[f(x) \right] \phi_i(x) \, d\Omega = \int_\Omega \overline{f}(x)\phi_i(x) \, d\Omega = \overline{f}_i \tag{17a}$$

$$\begin{aligned}
\text{cov}(f_i, f_j) &= \mathbb{E}\left[\int_\Omega \left(f(x) - \overline{f}(x) \right) \phi_i(x) \, d\Omega \int_\Omega \left(f(x') - \overline{f}(x') \right) \phi_j(x') \, d\Omega(x') \right] \\
&= \int_\Omega \int_\Omega \phi_i(x) \mathbb{E}\left[\left(f(x) - \overline{f}(x) \right) \left(f(x') - \overline{f}(x') \right) \right] \phi_j(x') \, d\Omega(x) \, d\Omega(x') \\
&= \int_\Omega \int_\Omega \phi_i(x) c_f(x, x') \phi_j(x') \, d\Omega(x) \, d\Omega(x').
\end{aligned} \tag{17b}$$

The components of the covariance matrix $(C_f)_{ij} = \text{cov}(f_i, f_j)$ are obtained by interpolating the covariance kernel with finite element basis functions, i.e.,

$$(C_f)_{ij} = \sum_k \sum_l \int_\Omega \int_\Omega \phi_i(x)\phi_k(x) c_f(x_k, x_l')\phi_l(x')\phi_j(x') \, d\Omega(x) \, d\Omega(x'). \tag{18}$$

Hence, the source vector is given by the multivariate Gaussian density

$$f \sim p(f) = \mathcal{N}\left(\overline{f}, C_f \right). \tag{19}$$

For a globally supported covariance kernel $c_f(x, x')$, such as the used squared exponential kernel, the covariance matrix C_f is dense. It is non-trivial to efficiently compute and assemble its components. To obtain a more easily computable covariance matrix, notice that in (18) the two integrals over the products of basis functions yield indeed two mass matrices. Replacing the two mass matrices with their lumped versions we obtain the approximation

$$(C_f)_{ij} \approx \left(\int_\Omega \phi_i(x) \, d\Omega \right) c_f(x_i, x_j) \left(\int_\Omega \phi_j(x) \, d\Omega \right). \tag{20}$$

Each of the brackets here corresponds to a source vector with a uniform prescribed source $f(\boldsymbol{x}) = 1$, c.f. (16). The covariance kernel $c_f(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is evaluated at the nodes corresponding to basis functions $\phi_i(\boldsymbol{x})$ and $\phi_j(\boldsymbol{x})$. The introduced approximation makes it possible to assemble and compute the covariance matrix $\boldsymbol{C}_f$ with standard finite element data structures. For a review on similar and other approximation techniques for computing finite element covariance matrices see [30, 31].

Finally, we can write the probability density for the finite element solution vector $\boldsymbol{u}$ for a given diffusion coefficient vector $\boldsymbol{\kappa}$. Solving the discrete system of equations (13) gives

$$\boldsymbol{u} = \boldsymbol{A}(\boldsymbol{\kappa})^{-1}\boldsymbol{f} \,. \tag{21}$$

The right-hand side represents an affine transformation of the source vector $\boldsymbol{f}$ with the multivariate Gaussian density (19) such that

$$\boldsymbol{u} \sim p(\boldsymbol{u}|\boldsymbol{\kappa}) = \mathcal{N}\left(\boldsymbol{A}(\boldsymbol{\kappa})^{-1}\overline{\boldsymbol{f}},\ \boldsymbol{A}(\boldsymbol{\kappa})^{-1}\boldsymbol{C}_f\boldsymbol{A}(\boldsymbol{\kappa})^{-\top}\right) \,. \tag{22}$$

The corresponding unconditional density $p(\boldsymbol{u})$ is obtained by marginalising the joint density $p(\boldsymbol{u}, \boldsymbol{\kappa}) = p(\boldsymbol{u}|\boldsymbol{\kappa})p(\boldsymbol{\kappa})$, which yields

$$p(\boldsymbol{u}) = \int p(\boldsymbol{u}|\boldsymbol{\kappa})p(\boldsymbol{\kappa})\,\mathrm{d}\boldsymbol{\kappa} \,. \tag{23}$$

It is possible to evaluate this integral numerically using, e.g., MC, MCMC or (sparse) quadrature, however it is impractical for large scale problems given that $\boldsymbol{\kappa} \in \mathbb{R}^{n_e}$ is usually a high-dimensional vector. Instead, we use a perturbation method to compute a first order approximation to the density $p(\boldsymbol{u})$ [12]. By explicitly denoting the dependence of the solution $\boldsymbol{u}$ on the random diffusion coefficient $\boldsymbol{\kappa}$ and source $\boldsymbol{f}$, we can write the series expansion

$$\boldsymbol{u}(\boldsymbol{\kappa}, \boldsymbol{f}) = \boldsymbol{u}(\overline{\boldsymbol{\kappa}}, \boldsymbol{f}) + \sum_{e=1}^{n_e} \frac{\partial \boldsymbol{u}(\overline{\boldsymbol{\kappa}}, \boldsymbol{f})}{\partial \kappa_e}(\kappa_e - \overline{\kappa}_e) + \ldots = \boldsymbol{u}^{(0)} + \sum_{e=1}^{n_e} \boldsymbol{u}_e^{(1)} \lambda_e + \ldots, \tag{24}$$

where the coefficients $\boldsymbol{u}^{(0)}, \boldsymbol{u}_e^{(1)}, \ldots$ are obtained by successively differentiating the system equation (13) with respect to the element diffusion coefficients. The two coefficients relevant for a first order approximation are given by

$$\boldsymbol{u}^{(0)} = \boldsymbol{A}(\overline{\boldsymbol{\kappa}})^{-1}\boldsymbol{f} \tag{25a}$$

$$\boldsymbol{u}_e^{(1)} = -\boldsymbol{A}(\overline{\boldsymbol{\kappa}})^{-1}\frac{\partial \boldsymbol{A}(\overline{\boldsymbol{\kappa}})}{\partial \kappa_e}\boldsymbol{u}^{(0)} \,. \tag{25b}$$

Hence, we obtain for the approximate mean and covariance

$$\overline{\boldsymbol{u}} = \mathbb{E}\left[\boldsymbol{u}^{(0)} + \sum_e \boldsymbol{u}_e^{(1)} \lambda_e\right] = \boldsymbol{A}(\overline{\boldsymbol{\kappa}})^{-1}\overline{\boldsymbol{f}} \tag{26a}$$

$$\boldsymbol{C}_u = \mathbb{E}\left[\left(\boldsymbol{u}^{(0)} + \sum_e \boldsymbol{u}_e^{(1)} \lambda_e\right) \otimes \left(\boldsymbol{u}^{(0)} + \sum_e \boldsymbol{u}_e^{(1)} \lambda_e\right)\right] - \overline{\boldsymbol{u}} \otimes \overline{\boldsymbol{u}} \,. \tag{26b}$$

Note that the expectation is over both $\boldsymbol{\kappa}$ and $\boldsymbol{f}$. After lengthy but straightforward algebraic manipulations we obtain

$$\boldsymbol{C}_u = \boldsymbol{A}(\overline{\boldsymbol{\kappa}})^{-1}\boldsymbol{C}_f\boldsymbol{A}(\overline{\boldsymbol{\kappa}})^{-\top} + \sum_e \sum_d (\boldsymbol{C}_\kappa)_{ed}\boldsymbol{A}(\overline{\boldsymbol{\kappa}})^{-1}\frac{\partial \boldsymbol{A}(\overline{\boldsymbol{\kappa}})}{\partial \kappa_e}\boldsymbol{A}(\overline{\boldsymbol{\kappa}})^{-1}\left(\boldsymbol{C}_f + \overline{\boldsymbol{f}} \otimes \overline{\boldsymbol{f}}\right)\boldsymbol{A}(\overline{\boldsymbol{\kappa}})^{-\top}\frac{\partial \boldsymbol{A}(\overline{\boldsymbol{\kappa}})^{\top}}{\partial \kappa_d}\boldsymbol{A}(\overline{\boldsymbol{\kappa}})^{-\top} \,. \tag{27}$$

Finally, we can approximate the density of the finite element solution (23) with the multivariate Gaussian density

$$p(\boldsymbol{u}) = \mathcal{N}\left(\overline{\boldsymbol{u}}, \boldsymbol{C}_u\right) \,. \tag{28}$$

It is clear that the true density $p(\boldsymbol{u})$ according to (23) is usually not a Gaussian. The approximation (28) is only valid when the scaling parameter σ_κ of the variance is relatively small. However, when the diffusion coefficient $\boldsymbol{\kappa}$ is
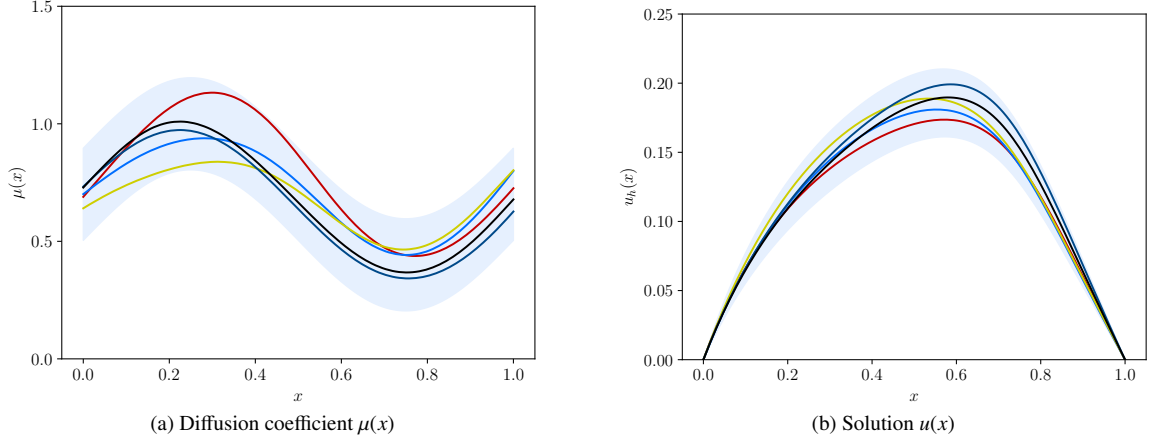
(a) Diffusion coefficient $\mu(x)$ (b) Solution $u(x)$

Figure 3: One-dimensional Poisson problem $-\mathrm{d}(\mu\,\mathrm{d}u/\,\mathrm{d}x)/\,\mathrm{d}x = 1$ with a random diffusion coefficient $\mu(x) = \exp(\kappa(x))$ with a mean $\overline{\kappa}(x) = \ln(0.7 + 0.3\sin(2\pi x))$ and the covariance kernel parameters $\sigma_\kappa = 0.1$ and $\ell_\kappa = 0.25$. The five lines in (a) represent samples drawn from (2) with the corresponding solutions shown in (b). The shaded areas are the 95% confidence regions obtained from (2) and Monte Carlo sampling.

deterministic the density $p(\boldsymbol{u})$ is a Gaussian as can be seen in (22). Furthermore, we can deduce from (27) that there is a fundamental difference in how the source and diffusivity covariance matrices $\boldsymbol{C}_f$ and $\boldsymbol{C}_\kappa$ contribute to $\boldsymbol{C}_u$. The source covariance $\boldsymbol{C}_f$ is always multiplied twice with the inverse of the system matrix, which increases the smoothness of the covariance operator. In contrast, the diffusivity covariance $\boldsymbol{C}_\kappa$ contributes directly with no such smoothing.

In Figure 3 an illustrative one-dimensional Poisson problem, $-\mathrm{d}(\mu(x)\,\mathrm{d}u(x)/\,\mathrm{d}x)/\,\mathrm{d}x = 1$, with a random diffusion coefficient and corresponding finite element solution are shown. The source $f(x) = 1$ is chosen as non-random. The diffusion coefficient $\mu(x) = \exp(\kappa(x))$ has the mean $\overline{\kappa}(x) = \ln(0.7 + 0.3\sin(2\pi x))$ and the parameters of the exponential covariance kernel are chosen with $\sigma_\kappa = 0.1$ and $\ell_\kappa = 0.25$. The one-dimensional problem is discretised with 128 linear finite elements. To assess the accuracy of the approximate mean $\overline{\boldsymbol{u}}$ and covariance $\boldsymbol{C}_u$ according to (26), we compare both with the empirical mean $\overline{\boldsymbol{u}}^{MC}$ and covariance $\boldsymbol{C}_u^{MC}$ obtained by Monte Carlo sampling (23). As depicted in Figure 4 the first-order perturbation and the Monte Carlo results are in good agreement for relatively large σ_κ.
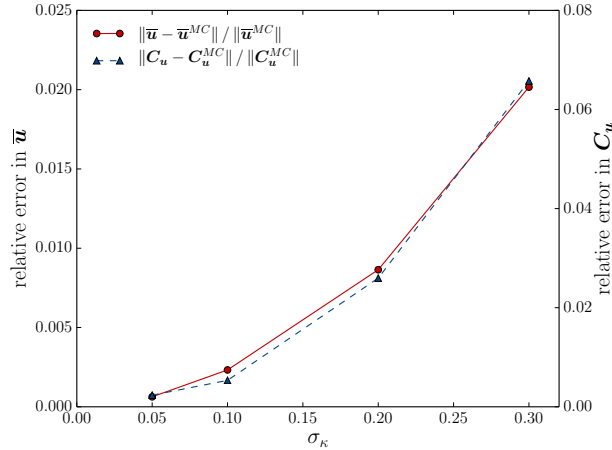

Figure 4: Relative perturbation error of the approximate mean $\overline{\boldsymbol{u}}$ and covariance $\boldsymbol{C}_u$ obtained from (26) for the one-dimensional Poisson problem with a random diffusion coefficient shown in Figure 3. Only the covariance scaling parameter σ_k is varied. The empirical mean $\overline{\boldsymbol{u}}^{MC}$ and covariance $\boldsymbol{C}_u^{MC}$ are obtained by Monte Carlo sampling (23) and can be considered as exact. The errors are measured in the Frobenius norm.

3. Bayesian inference

In this section, we first introduce following [11, 22, 23] the statistical generating model for the true process underlying the observed data. Each of the random variables in this model depends on parameters, like the covariance lengthscale and scaling parameters in the forward model, which may be chosen to be known or unknown with only their prior densities given. The unknown random parameters are referred to as the hyperparameters. We sequentially apply Bayes rule on three different levels to infer, or learn, from the observed data all the random variables and hyperparameters. On level one, in Section 3.2, we derive the posterior finite element and true system response densities with the finite element density derived in Section 2.2 serving as a prior. On level two, in Section 3.3, the posterior densities and point estimates of the hyperparameters are obtained. Finally, on level three, in Section 3.4, we rank different finite element models, e.g., with different mesh sizes or modelling assumptions, based on their ability to explain the observed data.

3.1. Statistical generating model for the observations

In statFEM the observed data vector $\boldsymbol{y} \in \mathbb{R}^{n_y}$ is, as graphically illustrated in Figure 5, additively composed in three components

$$\boldsymbol{y} = \boldsymbol{z} + \boldsymbol{e} = \rho \boldsymbol{P} \boldsymbol{u} + \boldsymbol{d} + \boldsymbol{e} \,. \tag{29}$$

That is, the observed data vector is equal to the unknown true system response $\boldsymbol{z} \in \mathbb{R}^{n_y}$ and a random observation error, i.e. noise, $\boldsymbol{e} \in \mathbb{R}^{n_y}$. In turn, the true system response is characterised with the parameter $\rho \in \mathbb{R}^+$ scaled projected finite element solution $\boldsymbol{P}\boldsymbol{u}$ and the mismatch error, or model inadequacy, $\boldsymbol{d} \in \mathbb{R}^{n_y}$. The matrix $\boldsymbol{P} \in \mathbb{R}^{n_y \times n_u}$ projects the finite element solution to the observed data space and consists of the finite element basis functions $\phi_i(\boldsymbol{x})$ evaluated at the n_y observation points. Of course, the observations in $\boldsymbol{y}$ can correspond to almost any physical quantity of interest, like the flux $\nabla u_h(\boldsymbol{x})$, which can be obtained from the solution $u_h(\boldsymbol{x})$ by applying a linear operator. In such cases the projection matrix $\boldsymbol{P}$ is the discretisation of the linear operator in question.

Although the mismatch error $\boldsymbol{d}$ is not known, we approximate its distribution using a Gaussian Process

$$\boldsymbol{d} \sim p(\boldsymbol{d}|\sigma_d, \ell_d) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}_d) \,, \tag{30}$$

and choose as a kernel the squared exponential kernel

$$c_d(\boldsymbol{x}, \boldsymbol{x}') = \sigma_d^2 \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\ell_d^2}\right) \tag{31}$$

with the parameters $\sigma_d \in \mathbb{R}^+$ and $\ell_d \in \mathbb{R}^+$. The covariance matrix $\boldsymbol{C}_d \in \mathbb{R}^{n_y} \times \mathbb{R}^{n_y}$ is obtained by evaluating the kernel at the n_y observation locations. It is straightforward to consider other covariance kernels or a linear combination of covariance kernels, see e.g. [24, Ch. 4].

Furthermore, as usual, we assume that the observation error $\boldsymbol{e}$ has the multivariate Gaussian density

$$\boldsymbol{e} \sim p(\boldsymbol{e}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}_e) \tag{32}$$

with the diagonal covariance matrix $\boldsymbol{C}_e = \sigma_e^2 \boldsymbol{I}$.

All the variables in the decomposition (29) are Gaussians so that the observed data vector $\boldsymbol{y}$ has the conditional density

$$p(\boldsymbol{y}|\boldsymbol{u}) = \mathcal{N}(\rho \boldsymbol{P} \boldsymbol{u}, \boldsymbol{C}_d + \boldsymbol{C}_e) \,. \tag{33}$$

This density is the likelihood of observing the data $\boldsymbol{y}$ for a given finite element solution $\boldsymbol{u}$. The likelihood depends in addition to the scaling parameter ρ on a number of parameters, including the introduced kernel scaling and lengthscale parameters, not all of which are known from the outset, see Figure 5. The linear summation $\boldsymbol{C}_d + \boldsymbol{C}_e$ in the likelihood indicates an identifiability issue, as discussed in detail in [11, 22]. We enforce weak identifiability by employing prior distributions on the hyperparameters defining each $\boldsymbol{C}_d$ and $\boldsymbol{C}_e$, see also Section 3.3.
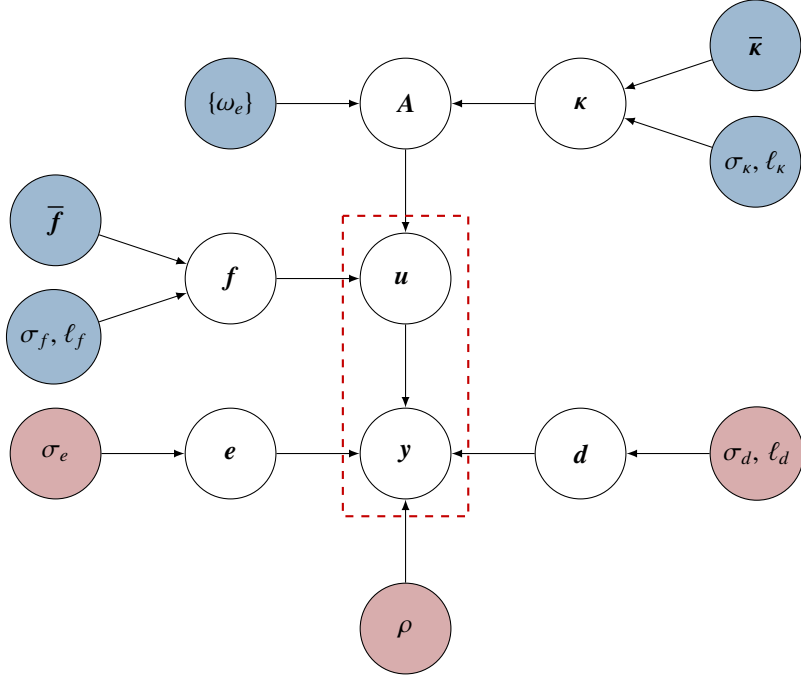
Figure 5: Graphical model of statFEM. The coloured circles represent the (possibly unknown) parameters and the empty circles denote the random variables that are either observed or derived. The parameters of the forward problem introduced in Section 2 are shown in blue and the parameters of the statistical generating model (29) are shown in red. Usually, some or all of the parameters in the shaded circles are known and need not be inferred. The remaining unknown random hyperparameters are inferred from the observations $\boldsymbol{y}$ and the finite element solution $\boldsymbol{u}$ using the statistical generating model.

In passing, we note that in a non-Bayesian context the unknown hyperparameters are determined by maximising the likelihood (33) and the obtained values are referred to as MLE estimates. Clearly, the likelihood

$$p(\boldsymbol{y}|\boldsymbol{u}) \propto \exp\left(-\frac{1}{2}\left(\rho\boldsymbol{Pu} - \boldsymbol{y}\right)\left(\boldsymbol{C}_d + \boldsymbol{C}_e\right)^{-1}\left(\rho\boldsymbol{Pu} - \boldsymbol{y}\right)\right) \tag{34}$$

has its maximum at $\boldsymbol{y} = \rho\boldsymbol{Pu}$. Hence, MLE prefers models which match the observation vector $\boldsymbol{y}$ as closely as possible irrespective of the true system response and measurement errors. As widely discussed in the literature, this gives rise to overly complex models prone to overfitting, see [6, 8, 19, 43, 46].

3.2. Posterior finite element and true system response densities

We use Bayes rule to update the finite element density of the forward problem (28) with the available observed data in line with the postulated statistical model (29). All the hyperparameters are assumed to be known or, in other words, all the mentioned densities are conditioned on the hyperparameters.

3.2.1. Single observation vector

To begin with, we consider only one single observation vector $\boldsymbol{y}$. The posterior finite element density $p(\boldsymbol{u}|\boldsymbol{y})$ conditioned on observed data $\boldsymbol{y}$ is given by

$$p(\boldsymbol{u}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{u})p(\boldsymbol{u})}{p(\boldsymbol{y})} \tag{35}$$

with the likelihood $p(\boldsymbol{y}|\boldsymbol{u})$ in (33), the prior $p(\boldsymbol{u})$ in (28) and the marginal likelihood, or the evidence,

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y}|\boldsymbol{u})p(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u}\,, \tag{36}$$

which ensures that the posterior $p(\boldsymbol{u}|\boldsymbol{y})$ is a probability distribution integrating to one. The likelihood $p(\boldsymbol{y}|\boldsymbol{u})$ is a function of $\boldsymbol{u}$ and measures the fit of the statistical generating model (29) to the given observation $\boldsymbol{y}$. On the other hand, the prior $p(\boldsymbol{u})$ reflects our knowledge of the system before any observations are made. The marginal likelihood $p(\boldsymbol{y})$ is the probability of observing the known (fixed) observation $\boldsymbol{y}$ averaged over all possible finite element solutions $\boldsymbol{u}$. Marginal likelihood plays a key role in Bayesian statistics as will be detailed in the following sections. As shown in the Appendix A.2, the posterior density is a multivariate Gaussian and is given by

$$p(\boldsymbol{u}|\boldsymbol{y}) = \mathcal{N}(\overline{\boldsymbol{u}}_{|y},\, \boldsymbol{C}_{u|y}),\tag{37a}$$

where

$$\overline{\boldsymbol{u}}_{|y} = \boldsymbol{C}_{u|y}\left(\rho \boldsymbol{P}^\mathsf{T}\,(\boldsymbol{C}_d + \boldsymbol{C}_e)^{-1}\,\boldsymbol{y} + \boldsymbol{C}_u^{-1}\overline{\boldsymbol{u}}\right) \quad \text{and} \quad \boldsymbol{C}_{u|y} = \left(\rho^2 \boldsymbol{P}^\mathsf{T}(\boldsymbol{C}_d + \boldsymbol{C}_e)^{-1}\boldsymbol{P} + \boldsymbol{C}_u^{-1}\right)^{-1}.\tag{37b}$$

Likewise, the marginal likelihood is a Gaussian and can be obtained by analytically evaluating (36), or more easily by revisiting the decomposition (29). All the variables in (29) are multivariate Gaussians so that the marginal likelihood simply reads

$$p(\boldsymbol{y}) = \mathcal{N}\left(\rho \boldsymbol{P}\overline{\boldsymbol{u}},\, \boldsymbol{C}_d + \boldsymbol{C}_e + \rho^2 \boldsymbol{P}\boldsymbol{C}_u\boldsymbol{P}^\mathsf{T}\right).\tag{38}$$

In (37), we can see when $\boldsymbol{C}_d + \boldsymbol{C}_e$ is small in comparison to $\boldsymbol{C}_u$ (in some norm) the mean $\overline{\boldsymbol{u}}_{|y}$ tends to $\boldsymbol{y}/\rho$ and the covariance $\boldsymbol{C}_{u|y}$ to $(\boldsymbol{C}_d + \boldsymbol{C}_e)/\rho^2$. However, when $\boldsymbol{C}_d + \boldsymbol{C}_e$ is relatively large the mean $\overline{\boldsymbol{u}}_{|y}$ tends to $\overline{\boldsymbol{u}}$ and the covariance $\boldsymbol{C}_{u|y}$ to $\boldsymbol{C}_u$. These bounds are reasonable reminding ourselves that the density of the unobserved true system response $\boldsymbol{z} = \rho \boldsymbol{P}\boldsymbol{u} + \boldsymbol{d}$ is given by

$$p(\boldsymbol{z}|\boldsymbol{y}) = \mathcal{N}\left(\rho \boldsymbol{P}\overline{\boldsymbol{u}}_{|y},\, \rho^2 \boldsymbol{P}\boldsymbol{C}_{u|y}\boldsymbol{P}^\mathsf{T} + \boldsymbol{C}_d\right).\tag{39}$$

With the mentioned bounds, for relatively small $\boldsymbol{C}_d + \boldsymbol{C}_e$ the mean of $\boldsymbol{z}$ tends to $\boldsymbol{y}$ and its covariance to $2\boldsymbol{C}_d + \boldsymbol{C}_e$. In contrast, for large $\boldsymbol{C}_d + \boldsymbol{C}_e$ the mean tends to $\rho\overline{\boldsymbol{u}}$ and the covariance to $\boldsymbol{C}_u + \boldsymbol{C}_d$.

In terms of implementation, usually the number of observations points is significantly smaller than the unknowns in the finite element method, i.e. $n_y \ll n_u$. Hence, the required inversion of large dense matrices of size $n_u \times n_u$ in (37) can be avoided by using the Woodbury matrix identity, see Appendix A.2.

3.2.2. Multiple observation vectors

In engineering applications usually the same sensors are used to repeatedly sample a set of observation vectors $\{\boldsymbol{y}_i\}_{i=1}^{n_o}$. For notational convenience we collect the set of observation vectors in a matrix $\boldsymbol{Y} \in \mathbb{R}^{n_y \times n_o}$. The posterior finite element density $p(\boldsymbol{u}|\boldsymbol{Y})$ conditioned on all the observations $\boldsymbol{Y}$ is once again given by

$$p(\boldsymbol{u}|\boldsymbol{Y}) = \frac{p(\boldsymbol{Y}|\boldsymbol{u})p(\boldsymbol{u})}{p(\boldsymbol{Y})}.\tag{40}$$

The physically sensible assumption of statistical independence between the n_o observations yields the likelihood

$$p(\boldsymbol{Y}|\boldsymbol{u}) = p(\boldsymbol{y}_1|\boldsymbol{u})p(\boldsymbol{y}_2|\boldsymbol{u})\cdots p(\boldsymbol{y}_{n_o}|\boldsymbol{u}) = \prod_{i=1}^{n_o} p(\boldsymbol{y}_i|\boldsymbol{u})\tag{41}$$

and the marginal likelihood, or the evidence,

$$p(\boldsymbol{Y}) = \int \prod_{i=1}^{n_o} p(\boldsymbol{y}_i|\boldsymbol{u})p(\boldsymbol{u})\,\mathrm{d}\boldsymbol{u}.\tag{42}$$

Following the same steps as in the preceding Section 3.2.1 we obtain the posterior density

$$p(\boldsymbol{u}|\boldsymbol{Y}) = \mathcal{N}(\overline{\boldsymbol{u}}_{|Y},\, \boldsymbol{C}_{u|Y}),\tag{43a}$$

where

$$\overline{\boldsymbol{u}}_{|Y} = \boldsymbol{C}_{u|Y}\left(\rho\boldsymbol{P}^\mathsf{T}\left(\boldsymbol{C}_d + \boldsymbol{C}_e\right)^{-1}\sum_{i=1}^{n_o}\boldsymbol{y}_i + \boldsymbol{C}_u^{-1}\overline{\boldsymbol{u}}\right) \quad \text{and} \quad \boldsymbol{C}_{u|Y} = \left(\rho^2 n_o \boldsymbol{P}^\mathsf{T}(\boldsymbol{C}_d + \boldsymbol{C}_e)^{-1}\boldsymbol{P} + \boldsymbol{C}_u^{-1}\right)^{-1}. \tag{43b}$$

Ostensibly, with an increase in the number of observation vectors n_o the covariance $\boldsymbol{C}_{u|Y}$ tends to zero and, in turn, the mean $\overline{\boldsymbol{u}}_{|Y}$ tends to the empirical mean of the observations $\sum \boldsymbol{y}_i/n_o$. For later reference, we note that the marginal likelihood is because of the statistical independence assumption between the observations given by

$$p(\boldsymbol{Y}) = \prod_{i=1}^{n_o} p(\boldsymbol{y}_i), \tag{44}$$

where $p(\boldsymbol{y}_i)$ is the marginal likelihood (36) of each of the readings.

3.3. Hyperparameter learning

The marginal likelihood $p(\boldsymbol{Y})$ given in (44), or $p(\boldsymbol{y})$ in (36), is critical for determining the hyperparameters of the statistical model (29). To begin with, we collect the parameters introduced so far in the vector

$$\boldsymbol{w} := \begin{pmatrix} \rho & \sigma_\kappa & \ell_\kappa & \sigma_r & \ell_r & \kappa_1 & \kappa_2 & \ldots & \kappa_{n_e} & \sigma_d & \ell_d & \sigma_e \end{pmatrix}^\mathsf{T} \in \mathbb{R}^{n_w}. \tag{45}$$

Some of these parameters may be known or unknown with only their prior densities given. To sidestep the issue of non-identifiability of the parameters it is important that the priors are informative. In the following, $\boldsymbol{w}$ includes only the *unknown* hyperparameters so that its dimension varies depending on the considered problem. The hyperparameters are estimated from the observed data by applying the Bayes formula one more time. To this end, note that the marginal likelihood $p(\boldsymbol{Y})$ in (44) is indeed conditioned on the hyperparameter vector $\boldsymbol{w}$ so that we write more succinctly $p(\boldsymbol{Y}|\boldsymbol{w})$. Consequently, the Bayes formula for obtaining the posterior density of the parameter vector reads

$$p(\boldsymbol{w}|\boldsymbol{Y}) = \frac{p(\boldsymbol{Y}|\boldsymbol{w})p(\boldsymbol{w})}{\int p(\boldsymbol{Y}|\boldsymbol{w})p(\boldsymbol{w})\,\mathrm{d}\boldsymbol{w}}. \tag{46}$$

As usual, the prior $p(\boldsymbol{w})$ encodes any information that we might have prior to making the observation $\boldsymbol{Y}$. In choosing $p(\boldsymbol{w})$ it is justified to assume that all the hyperparameters are statistically independent such that

$$p(\boldsymbol{w}) = \prod_{i=1}^{n_w} p(w_i). \tag{47}$$

Moreover, the normalisation constant in the denominator of (46) can be omitted when only a point-estimate is needed or when the posterior is sampled with MCMC. In that case it is sufficient to consider just

$$p(\boldsymbol{w}|\boldsymbol{Y}) \propto p(\boldsymbol{Y}|\boldsymbol{w})p(\boldsymbol{w}). \tag{48}$$

It bears emphasis that this posterior is in contrast to the posterior (35) analytically intractable. The optimal hyperparameter vector $\boldsymbol{w}^*$ referred to as the maximum posteriori (MAP) estimate is given by

$$\boldsymbol{w}^* = \arg\max_{\boldsymbol{w}} p(\boldsymbol{Y}|\boldsymbol{w})p(\boldsymbol{w}). \tag{49}$$

This, often non-convex, optimisation problem can be numerically solved with conventional algorithms, see e.g. [48]. Alternatively, as in this paper, we can use the expectation of the parameter vector

$$\overline{\boldsymbol{w}} = \mathbb{E}[\boldsymbol{w}] = \int \boldsymbol{w}\,p(\boldsymbol{w}|\boldsymbol{Y})\,\mathrm{d}\boldsymbol{w}, \tag{50}$$

as a point estimate. The expectation is obtained by sampling $p(\boldsymbol{w}|\boldsymbol{Y})$ using MCMC, see Appendix A.3, and then computing the empirical mean

$$\overline{\boldsymbol{w}} \approx \frac{1}{N} \sum_i \boldsymbol{w}^{(i)} \quad \text{with } \boldsymbol{w}^{(i)} \sim p(\boldsymbol{w}|\boldsymbol{Y}) \, . \tag{51}$$

The empirical variance of the samples $w^{(i)}$ represents the uncertainty in the obtained point estimate. In contrast to sampling, the optimisation problem (49) does not yield such a variance estimate. Evidently, the two estimates (49) and (51) will have different values and either one or both of them might be sufficient to characterise $p(\boldsymbol{w}|\boldsymbol{Y})$. When $p(\boldsymbol{w}|\boldsymbol{Y})$ is multimodal, usually, both are inadequate. In such cases it is better to take the entire distribution $p(\boldsymbol{w}|\boldsymbol{Y})$ and to marginalise out $\boldsymbol{w}$ whenever a distribution depends, i.e., is conditioned, on $\boldsymbol{w}$.

3.4. Model comparison and hypothesis testing

The marginal likelihood $p(\boldsymbol{Y})$ in (44), or $p(\boldsymbol{y})$ in (36), plays also a key role in Bayesian model comparison, see e.g. [6, 20]. Without loss of generality, we consider in the following finite element models which differ only in terms of mesh resolution, specifically, maximum element size h. However, the same approach can be applied to finite element models that have, for instance, differing domain geometries or boundary conditions or are based on fundamentally different mathematical models.

We aim to compare the fidelity of two finite element models, i.e. the two meshes $\mathcal{M}_1$ and $\mathcal{M}_2$, in explaining the observed data $\boldsymbol{Y}$. As previously mentioned, the marginal likelihood $p(\boldsymbol{Y})$ is obtained by averaging the likelihood $p(\boldsymbol{Y}|\boldsymbol{u})$ over all possible finite element solutions $\boldsymbol{u}$. The marginal likelihood is conditioned on the specific mesh used for computing $\boldsymbol{u}$ so that we denote it either with $p(\boldsymbol{Y}|\mathcal{M}_1)$ or $p(\boldsymbol{Y}|\mathcal{M}_2)$. Because the true system response $\boldsymbol{z}$ is unknown it is from the outset unclear which of the two models is sufficient to explain the observed data $\boldsymbol{Y}$. Obviously, the model with the finer mesh yields the more accurate finite element solution. However, in light of inherent observation errors and model inadequacy the coarser mesh may indeed be sufficient to explain the observed data. To quantify the fidelity of the two models, we compute their posterior densities using Bayes formula, i.e.,

$$p(\mathcal{M}_1|\boldsymbol{Y}) = \frac{p(\boldsymbol{Y}|\mathcal{M}_1)p(\mathcal{M}_1)}{p(\boldsymbol{y}|\mathcal{M}_1)p(\mathcal{M}_1) + p(\boldsymbol{Y}|\mathcal{M}_2)p(\mathcal{M}_2)} \, , \tag{52a}$$

$$p(\mathcal{M}_2|\boldsymbol{Y}) = \frac{p(\boldsymbol{Y}|\mathcal{M}_2)p(\mathcal{M}_2)}{p(\boldsymbol{y}|\mathcal{M}_1)p(\mathcal{M}_1) + p(\boldsymbol{Y}|\mathcal{M}_2)p(\mathcal{M}_2)} \, , \tag{52b}$$

where the two priors $p(\mathcal{M}_1)$ and $p(\mathcal{M}_2)$ encode our subjective preference for either one of the models or any prior information available. If we do not have any prior information, we can choose them with $p(\mathcal{M}_1) \propto 1$ and $p(\mathcal{M}_2) \propto 1$. Furthermore, for model comparison rather than the absolute values of the two posteriors their ratio, referred to as the Bayes factor,

$$\frac{p(\mathcal{M}_1|\boldsymbol{Y})}{p(\mathcal{M}_2|\boldsymbol{Y})} = \frac{p(\boldsymbol{Y}|\mathcal{M}_1)p(\mathcal{M}_1)}{p(\boldsymbol{Y}|\mathcal{M}_2)p(\mathcal{M}_2)} \, , \tag{53}$$

is more meaningful. A ratio larger than one indicates a preference for $\mathcal{M}_1$ and a value smaller a preference for $\mathcal{M}_2$ [20].

3.5. Predictive observation density

We can use the posterior finite element density derived in Section 3.2 and the likelihood according to our statistical model to compute the predictive observation density at locations where there are no observations. The unknown predictive distribution at the $n_{\tilde{y}}$ non-observed locations of interest are collected in a vector $\tilde{\boldsymbol{y}} \in \mathbb{R}^{n_{\tilde{y}}}$. First, we consider the conditional joint distribution

$$p(\tilde{\boldsymbol{y}}, \boldsymbol{u}|\boldsymbol{Y}) = p(\tilde{\boldsymbol{y}}|\boldsymbol{u}, \boldsymbol{Y})p(\boldsymbol{u}|\boldsymbol{Y}) = p(\tilde{\boldsymbol{y}}|\boldsymbol{u})p(\boldsymbol{u}|\boldsymbol{Y}) \, , \tag{54}$$

where we used the statistical independence of the random vector $\tilde{\boldsymbol{y}}$ and the observation matrix $\boldsymbol{Y}$. Then, marginalising out the finite element solution we obtain for the predictive observation density

$$p(\tilde{\boldsymbol{y}}|\boldsymbol{Y}) = \int p(\tilde{\boldsymbol{y}}|\boldsymbol{u})p(\boldsymbol{u}|\boldsymbol{Y}) \, \mathrm{d}\boldsymbol{u} \, . \tag{55}$$

The likelihood is given in (33) and the posterior finite element density in (43), or (37) in case of a single reading. Denoting the matrices corresponding to the $n_{\tilde{y}}$ non-observed locations with $\tilde{P} \in \mathbb{R}^{n_{\tilde{y}} \times n_u}$, $\tilde{C}_d \in \mathbb{R}^{n_{\tilde{y}} \times n_{\tilde{y}}}$ and $\tilde{C}_e \in \mathbb{R}^{n_{\tilde{y}} \times n_{\tilde{y}}}$, the likelihood reads

$$p(\tilde{y}|u) = \mathcal{N}\left(\rho\tilde{P}u, \; \tilde{C}_d + \tilde{C}_e\right) . \tag{56}$$

In (55), both terms in the integrand are Gaussians so that the integral can be analytically evaluated, c.f. Appendix A.2 and the references therein, yielding

$$p(\tilde{y}|Y) = \mathcal{N}\left(\rho\tilde{P}\overline{u}_{|Y}, \; \tilde{C}_d + \tilde{C}_e + \rho^2 \tilde{P} C_{u|Y} \tilde{P}^{\mathsf{T}}\right) . \tag{57}$$

The mean is the with ρ scaled mean of the posterior finite element density and the covariance is the sum of all contributions to the overall uncertainty in prediction.

4. Examples

In this section, we apply statFEM to one- and two-dimensional Poisson problems. All examples are discretised with a standard finite element approach using linear Lagrange basis functions. After establishing the convergence of the probabilistic forward problem we study the convergence of the posterior densities with an increasing number of observation points n_y and readings n_o. In addition, we illustrate how statFEM reduces to conventional Bayesian inversion when the statistical model (29) is simplified to $y = u + e$ and the mapping of the model parameters to the finite element solution becomes deterministic. Usually, the finite element covariance matrix C_u given by (26) becomes ill-conditioned when the covariance lengthscale for the source ℓ_f or for the diffusion coefficient ℓ_κ is larger than the characteristic element size h. Therefore, in all the examples, we use instead of C_u the stabilised covariance matrix $C_u + 0.001(\sigma_f^2 + \sigma_\kappa^2)I$.

4.1. One-dimensional problem

We seek the solution of the one-dimensional Poisson-Dirichlet problem

$$-\frac{\mathrm{d}}{\mathrm{d}x}\left(\mu(x)\frac{\mathrm{d}u}{\mathrm{d}x}\right) = f(x) \qquad \text{in } \Omega = (0, \, 1) \tag{58a}$$

$$u(x) = 0 \qquad \text{on } x = 0 \text{ and } x = 1 \, , \tag{58b}$$

where either the diffusion coefficient $\mu(x)$ or the source $f(x)$ is random. As mentioned, to ensure that the diffusion coefficient $\mu(x)$ is positive we consider the auxiliary variable $\kappa(x) = \ln(\mu(x))$. That is, in MCMC sampling $\kappa(x) \in \mathbb{R}$ is the unknown variable and the positive diffusion coefficient is $\mu(x) = \exp(\kappa(x))$, see Appendix A.3.

4.1.1. Convergence of the forward finite element density for random source

To establish the convergence of the discretised probabilistic forward problem, we consider the Poisson-Dirichlet problem (58) with a deterministic diffusion coefficient $\mu(x) = 1$ and a random source with a mean $\overline{f}(x) = 1$, covariance scaling parameter $\sigma_f = 0.2$ and a lengthscale parameter $\ell_f \in \{0.25, 0.5, 1.0\}$. The exact solution $u(x)$ is a Gaussian process (9) with a mean $\overline{u}(x)$ and covariance $c_u(x, \, x')$. The required Greens function $g(x, \, x')$ can be easily analytically obtained.

According to (22) the density of the finite element solution is a multivariate Gaussian $p(u) = \mathcal{N}(\overline{u}, \, C_u)$ with a mean $\overline{u} = A^{-1}\overline{f}$ and covariance $C_u = A^{-1}C_f A^{-\mathsf{T}}$. The mean is identical to the solution of a Poisson-Dirichlet problem with a deterministic source and as such its convergence characteristics are well studied, see e.g. [49]. Therefore, we focus here on the convergence of the covariance. The covariance of the finite element approximation $u_h(x) = \sum_i \phi_i(x)u_i = \phi(x)^{\mathsf{T}}u$ is given by

$$
\begin{aligned}
c_{u_h}(x, \, x') = \mathrm{cov}(u_h(x), \, u_h(x')) &= \mathbb{E}\left[\left(\sum_i \phi_i(x)\left(u_i - \overline{u}_i\right)\right)\left(\sum_j \phi_j(x')(u_j - \overline{u}_j)\right)\right] \\
&= \sum_i \sum_j \phi_i(x)\phi_j(x')\mathbb{E}\left[(u_i - \overline{u}_i)(u_j - \overline{u}_j)\right] = \phi(x)^{\mathsf{T}}C_u\phi(x') .
\end{aligned}
\tag{59}
$$

14

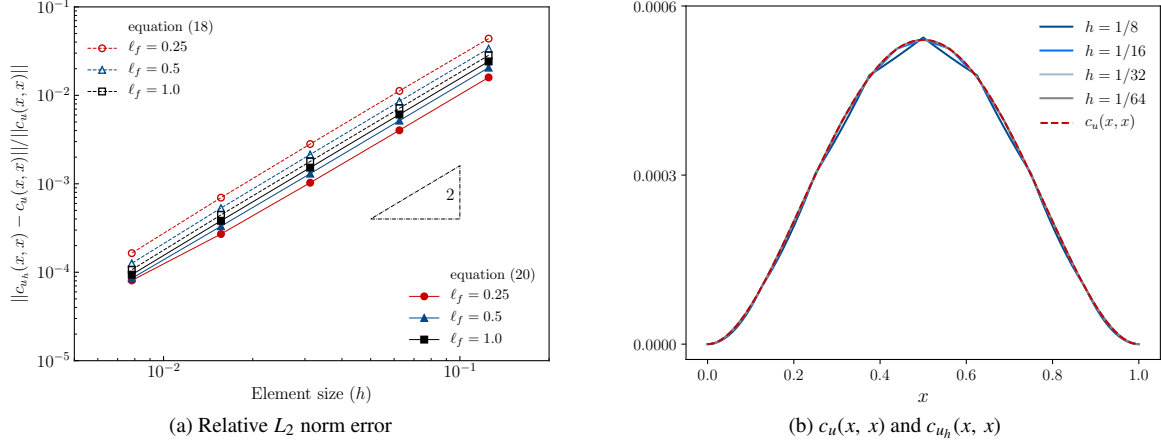(a) Relative L_2 norm error

(b) $c_u(x, x)$ and $c_{u_h}(x, x)$

Figure 6: One-dimensional problem. Convergence of the finite element variance $c_{u_h}(x, x)$ of the forward problem. Source covariance matrix $\boldsymbol{C}_f$ either exact (18) or approximate (20).

In Figure 6a the L_2 norm of the approximation error of the variance $\|c_{u_h}(x, x) - c_u(x, x)\|$ as a function of the element size h is shown. The covariance $\boldsymbol{C}_f$ of the source is computed either according to (18) or its approximation (20) and using 10 quadrature points per element. In both cases, the error converges with a rate of around two irrespective of the lengthscale parameter ℓ_f. However, the approximation (20) leads to a smaller error than the exact expression (18), especially for smaller ℓ_f. Figure 6b depicts the variance $c_u(x, x)$ and its finite element approximation $c_{u_h}(x, x)$ for $\ell_f = 0.5$ on successively finer meshes. As expected, the finite element approximation converges to the exact solution when the mesh is refined.

4.1.2. Posterior finite element density and system response for random source

We assume that the true system response is given by the Gaussian process

$$z(x) \sim \mathcal{GP}\left(\overline{z}(x), \, g(x, x'') * c_z(x'', x''') * g(x''', x')\right) \tag{60}$$

with a mean

$$\overline{z}(x) = \frac{1}{5}\sin(\pi x) + \frac{1}{50}\sin(7\pi x) \tag{61}$$

and a squared exponential covariance kernel

$$c_z(x, x') = 0.0225 \exp\left(-2(x - x')^2\right). \tag{62}$$

Of course, this true system response $z(x)$ is in practice not known. Irrespective of $z(x)$ we choose a finite element model with a deterministic diffusion coefficient $\mu(x) = 1$ and a random source with a mean $\overline{f}(x) = \pi^2/5$ and squared-exponential covariance parameters $\sigma_f = 0.3$ and $\ell_f = 0.25$. The uniform finite element discretisation consists of 32 elements. The system response $z(x)$ and the finite element solution $u_h(x)$, i.e. their mean and 95% confidence regions, are depicted in Figure 7. The finite element solution is able to roughly capture the overall true system response but not its details.

In practice, the true response is not known and we can only observe $\boldsymbol{y} = \boldsymbol{z} + \boldsymbol{e}$ at the n_y observation points. We consider the $n_y \in \{4, 11, 33\}$ observation points shown in Figure 8. We sample at each of the n_y observation points $n_o \in \{1, 10, 100, 1000\}$ repeated readings. Hence, we sample a synthetic observation matrix $\boldsymbol{Y} \in \mathbb{R}^{n_y \times n_o}$ from the Gaussian process

$$y(x) \sim \mathcal{GP}\left(\overline{z}(x), \, g(x, x'') * c_z(x'', x''') * g(x''', x') + 2.5 \cdot 10^{-5} \delta_{xx'}\right), \tag{63}$$
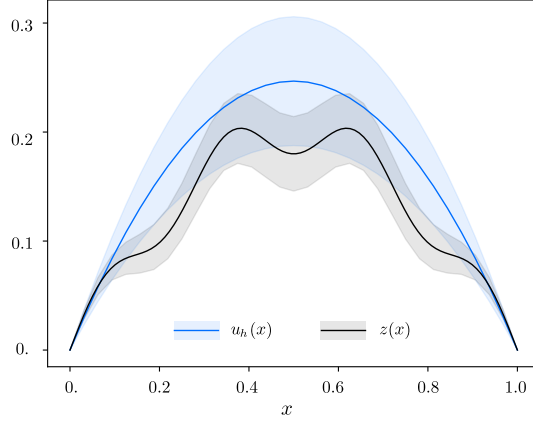
15

Figure 7: One-dimensional problem. Finite element solution $u_h(x)$ and the true system response $z(x)$. The solid lines represent the respective means and the shaded areas the 95% confidence regions.
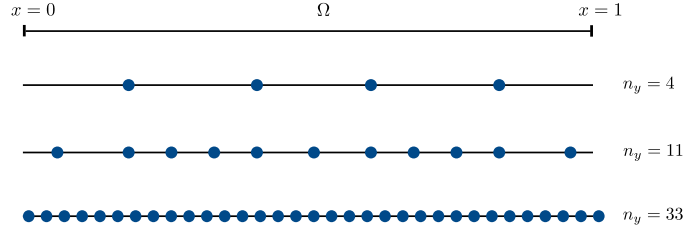

Figure 8: One-dimensional problem. Location of the $n_y \in \{4, 11, 33\}$ observations points of the data vector $\boldsymbol{y}$.

where $\delta_{xx'}$ is the Kronecker delta and $\sigma_e^2 = 2.5 \cdot 10^{-5}$ is the observation noise. See Appendix A.1 for sampling from a Gaussian process.

Before computing the posterior finite element density $p(\boldsymbol{u}|\boldsymbol{Y})$, we first determine the (unknown) hyperparameters of the statistical generating model (29). In this example, amongst the parameters summarised in the graphical model in Figure 5, only the scaling parameter ρ and the mismatch covariance parameters σ_d and ℓ_d are assumed to be unknown. We collect these hyperparameters in the vector $\boldsymbol{w} = (\rho, \sigma_d, \ell_d)^\top$. The posterior density of the hyperparameters is given by $p(\boldsymbol{w}|\boldsymbol{Y}) \propto p(\boldsymbol{Y}|\boldsymbol{w})p(\boldsymbol{w})$, see (48). To make the inference problem more challenging we assume a non-informative prior $p(\boldsymbol{w}) \propto 1$ so that the posterior is proportional to the likelihood, i.e. $p(\boldsymbol{w}|\boldsymbol{Y}) \propto p(\boldsymbol{Y}|\boldsymbol{w})$. We sample $p(\boldsymbol{w}|\boldsymbol{Y})$ using a standard MCMC algorithm while enforcing the positivity of the hyperparameters by sampling on the log scale, see Appendix A.3. For each combination of n_y and n_o, we run 20000 iterations with an average acceptance ratio of 0.287. In Figure 9 the obtained normalised histograms for $p(\rho|\boldsymbol{Y})$, $p(\sigma_d|\boldsymbol{Y})$ and $p(\ell_d|\boldsymbol{Y})$ for $n_y = 11$ are depicted. We can observe that the standard deviations become significantly smaller with increasing n_o. In Tables 1, 2 and 3 the empirical mean and standard deviations of these three plots and other n_y and n_o combinations are given. When either n_y, n_o or both are increased, the standard deviation becomes smaller. Depending on the considered application it may be easier to increase either n_y or n_o.

With the density of the hyperparameters $p(\boldsymbol{w}|\boldsymbol{Y})$ at hand, it is possible to evaluate the posterior density of the finite element solution $p(\boldsymbol{u}|\boldsymbol{Y})$ given by (43). As discussed in Section 3.3, we use the empirical mean $\overline{\boldsymbol{w}} = \mathbb{E}[\boldsymbol{w}]$ of the hyperparameters $\boldsymbol{w} \sim p(\boldsymbol{w}|\boldsymbol{Y})$ as an point estimate in evaluating the posterior density $p(\boldsymbol{u}|\boldsymbol{Y})$. The obtained posterior densities for different combinations of n_y and n_o are depicted in Figure 10. Each observation sampled from the Gaussian process (63) is depicted with a dot. It is remarkable that already a single set of readings can achieve a significant improvement of the finite element mean. In all cases the 95% confidence regions become smaller with increasing number of readings n_o. When both n_y and n_o are increased the posterior mean $\overline{\boldsymbol{u}}_{|Y}$ converges to the true process mean $\overline{z}(x)$ and the covariance $\boldsymbol{C}_{u|Y}$ converges to zero. In the corresponding Figure 11 the obtained posterior densities $p(\boldsymbol{z}|\boldsymbol{Y})$ for the inferred true system response according to (39) are shown. As expected with increasing n_y and n_o the inferred density $p(\boldsymbol{z}|\boldsymbol{Y})$ converges to the, in this example known, true density given by the Gaussian pro-
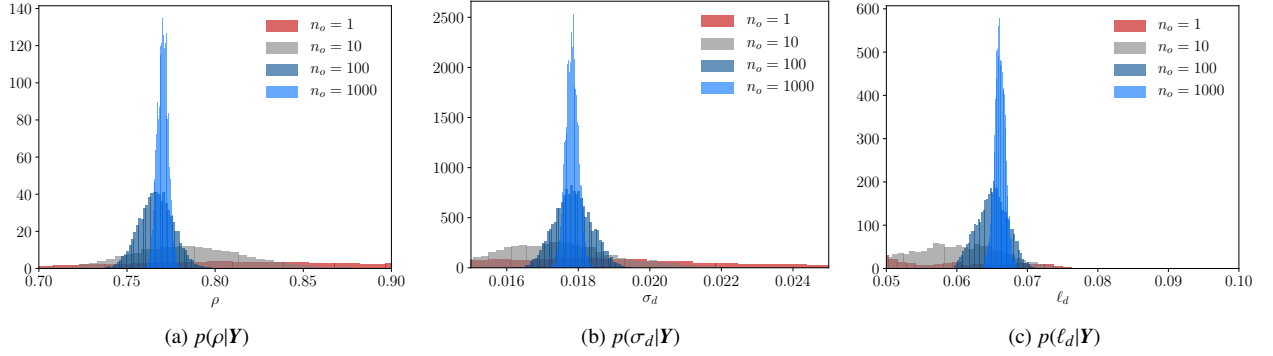
Figure 9: One-dimensional problem with random source. Posteriors of the parameters ρ, σ_d and ℓ_d for $n_y = 11$ and $n_o \in \{1, 10, 100, 1000\}$ sampled with MCMC.

	$n_o = 1$	$n_o = 10$	$n_o = 100$	$n_o = 1000$
$n_y = 4$	0.80118 ± 0.16823	0.78730 ± 0.03518	0.79015 ± 0.01208	0.77836 ± 0.00380
$n_y = 11$	0.85740 ± 0.11607	0.78549 ± 0.03349	0.76662 ± 0.00961	0.77024 ± 0.00312
$n_y = 33$	0.76473 ± 0.11476	0.79153 ± 0.03072	0.77200 ± 0.00980	0.77135 ± 0.00311

Table 1: One-dimensional problem with random source. Empirical mean and standard deviation of $p(\rho|Y)$.

	$n_o = 1$	$n_o = 10$	$n_o = 100$	$n_o = 1000$
$n_y = 4$	0.04610 ± 0.02150	0.02292 ± 0.00328	0.02310 ± 0.00128	0.02340 ± 0.00046
$n_y = 11$	0.01823 ± 0.00471	0.01734 ± 0.00164	0.01784 ± 0.00051	0.01782 ± 0.00017
$n_y = 33$	0.02976 ± 0.01463	0.01969 ± 0.00186	0.01860 ± 0.00053	0.01874 ± 0.00018

Table 2: One-dimensional problem with random source. Empirical mean and standard deviation of $p(\sigma_d|Y)$.

	$n_o = 1$	$n_o = 10$	$n_o = 100$	$n_o = 1000$
$n_y = 4$	0.23291 ± 0.06648	0.08071 ± 0.03899	0.17373 ± 0.01346	0.17703 ± 0.00527
$n_y = 11$	0.04447 ± 0.01359	0.05925 ± 0.00696	0.06505 ± 0.00215	0.06610 ± 0.00075
$n_y = 33$	0.08967 ± 0.01828	0.07577 ± 0.00496	0.07309 ± 0.00164	0.07436 ± 0.00052

Table 3: One-dimensional problem with random source. Empirical mean and standard deviation of $p(\ell_d|Y)$.

cess (60). Only very few n_y and n_o yield a very good approximation to the true process mean $\bar{z}(x)$. While the overall shape of the inferred and the true process confidence regions are in good agreement there are some differences close to the boundaries. These are related to the assumed covariance structure for the model mismatch vector $\boldsymbol{d}$. The chosen squared exponential kernel is unable to provide a better approximation to the covariance of the true process.

4.1.3. Posterior finite element density and system response for random diffusivity

We consider the case when the diffusion coefficient is random and the source is deterministic. We aim to compute the posterior finite element and true system densities $p(\boldsymbol{u}|Y)$ and $p(\boldsymbol{z}|Y)$ for observation matrices Y sampled from the Gaussian process (63). The source is chosen to be $f(x) = \pi^2/5$ and the diffusion coefficient is given by the Gaussian process

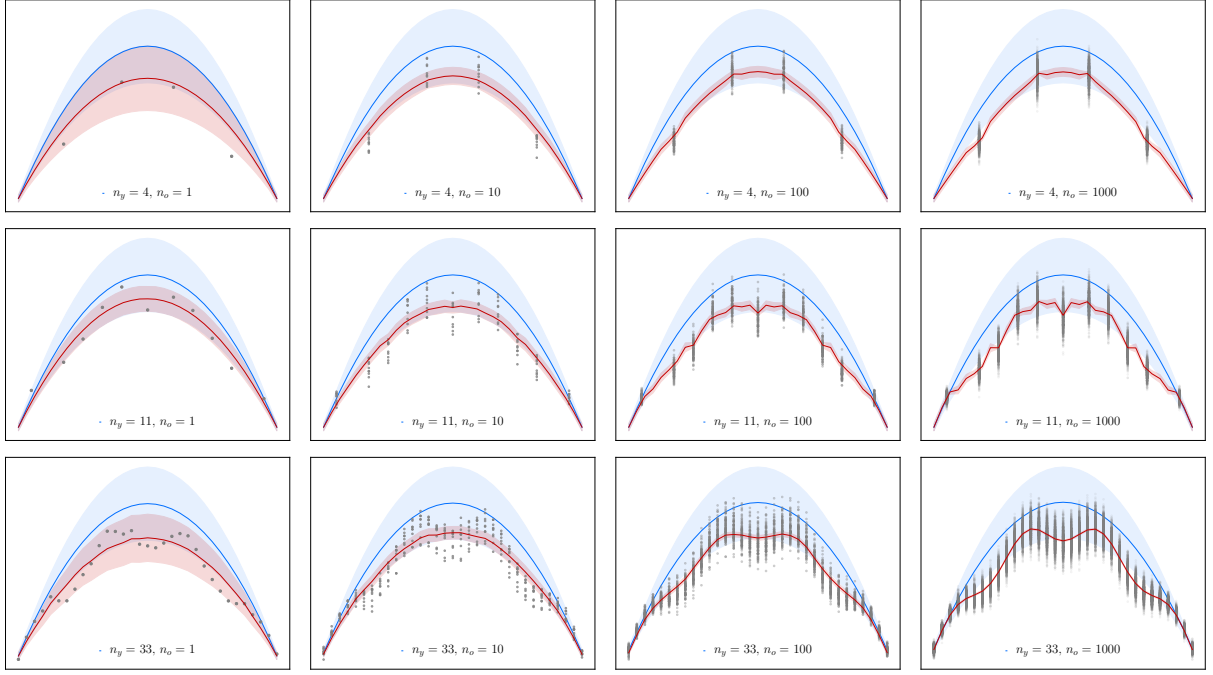$$\kappa(x) \sim \mathcal{GP}\left(1.0,\ 0.0225 \exp\left(-8(x - x')^2\right)\right). \tag{64}$$

17

Figure 10: One-dimensional problem with random source. Finite element density $p(\boldsymbol{u}|\boldsymbol{Y})$ conditioned on observation data (black dots). The blue lines represent the mean $\overline{\boldsymbol{u}}$ and the red lines the conditioned mean $\overline{\boldsymbol{u}}_{|Y}$. The shaded areas denote the corresponding 95% confidence regions. In each row the number of sensors n_y and each column the number of readings n_o is constant.
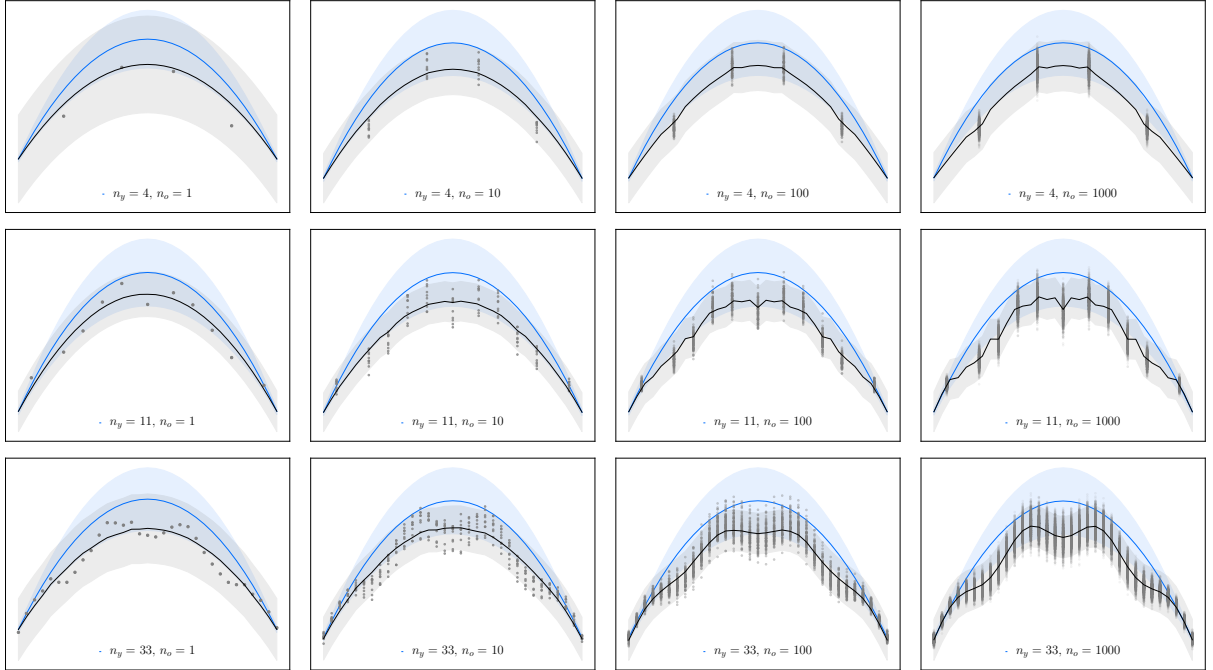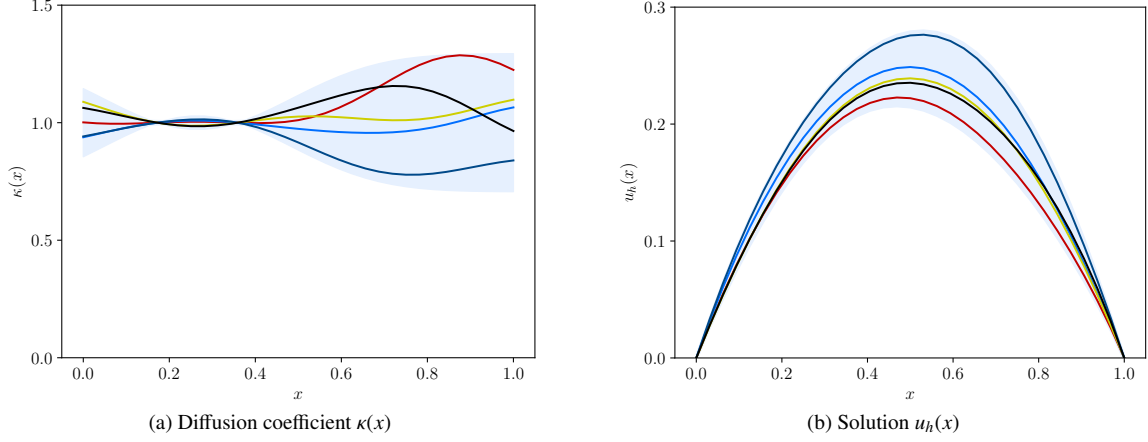

Figure 11: One-dimensional problem with random source. Inferred true system density $p(z|\boldsymbol{Y})$ conditioned on observation data (black dots). The blue lines represent the mean $\overline{\boldsymbol{u}}$ and the black lines the conditioned mean $\overline{z}_{|Y}$. The shaded areas denote the corresponding 95% confidence regions. In each row the number of sensors n_y and each column the number of readings n_o is constant.

(a) Diffusion coefficient $\kappa(x)$	(b) Solution $u_h(x)$

Figure 12: One-dimensional problem with random diffusivity. The diffusion coefficient is a Gaussian process with a mean $\bar{\kappa}(x) = 1$ and covariance kernel parameters $\sigma_\kappa = 0.15$ and $\ell_\kappa = 0.25$. The five lines in (a) represent samples drawn from the Gaussian process (64) conditioned on $\kappa(11/64) = \kappa(23/64) = 1.0$. The corresponding lines in (b) show the solution. The shaded areas denote the corresponding 95% confidence regions.

The diffusion coefficient within each element is assumed to be constant. Hence, this Gaussian process is discretised by sampling the diffusion coefficient $\kappa \in \mathbb{R}^{n_e}$ at the element centres $X^{(c)}$ to yield the multivariate Gaussian density $p(\kappa) = \mathcal{N}(\bar{\kappa}, C_\kappa)$. When the diffusion coefficient in some of the finite elements is known it can be taken into account by conditioning the density $p(\kappa)$ on those known values, see Appendix A.1. In this example, we assume that the diffusion coefficient $\kappa(x = 11/64) = 1.0$ and $\kappa(x = 23/64) = 1.0$ are known. In Figure 12 five samples of the so conditioned diffusion coefficient and the 95% confidence region are shown. The corresponding finite element solutions are obtained by solving the forward problem with the given diffusion coefficient distribution. Evidently, the mapping between the diffusion coefficient κ and the finite element solution u is nonlinear. As discussed in Section 2.2, we approximate this mapping with a first order perturbation method yielding the approximate density $p(u) = \mathcal{N}(\bar{u}, C_u)$, see (28).

As in Section 4.1.2, we first determine the unknown hyperparameters of the statistical generating model before computing the posterior densities $p(u|Y)$ and $p(u|Z)$. The unknowns in this example are again the scaling parameter ρ and the mismatch covariance parameters σ_d and ℓ_d, which are collected in the vector $w = (\rho, \sigma_d, \ell_d)^\mathsf{T}$. We sample the posterior $p(w|Y) \propto p(Y, w)p(w)$ using standard MCMC and a non-informative prior $p(w) \propto 1$. We consider $n_o \in \{1, 10, 100, 1000\}$ repeated readings sampled from (63) at the $n_y \in \{4, 33\}$ locations shown in Figure 8.

We evaluate next the posterior finite element density $p(u|Y)$. In Figure 13 the posterior and the prior finite element densities are compared for different n_y and n_o combinations. The prior mean and the 95% confidence region are slightly asymmetric due to the asymmetry of the diffusion coefficient, see Figure 12a. When n_y and n_o are low the posterior mean $\bar{u}|_Y$ and the 95% confidence regions are asymmetric as well. However, with increasing n_y and n_o the posterior mean converges to the symmetric true process mean $\bar{z}(x)$ and the posterior covariance converges to zero. The inferred true system density $p(z|Y)$ is shown in Figure 14. It can be observed that for $n_y = 33$ observation locations and increasing number of readings n_o that both the mean $\bar{u}|_z$ and the covariance $C_{z|Y}$ of the inferred posterior show good agreement with the known true system density. These results assert the success of statFEM in inferring the true system density given a reasonable amount of observation data.

4.1.4. Posterior diffusion coefficient density

We contrast now statFEM to conventional Bayesian treatment of inverse problems. In statFEM our primary aim is to infer the posterior finite element and true system densities $p(u|Y)$ and $p(z|Y)$. While in inverse problems the aim is to infer the posterior densities of certain model parameters, like the posterior diffusion coefficient density $p(\kappa^{(a)}|Y)$. The vector $\kappa^{(a)}$ represents the coefficients, or the parameters, used for discretising the diffusion coefficient. Commonly
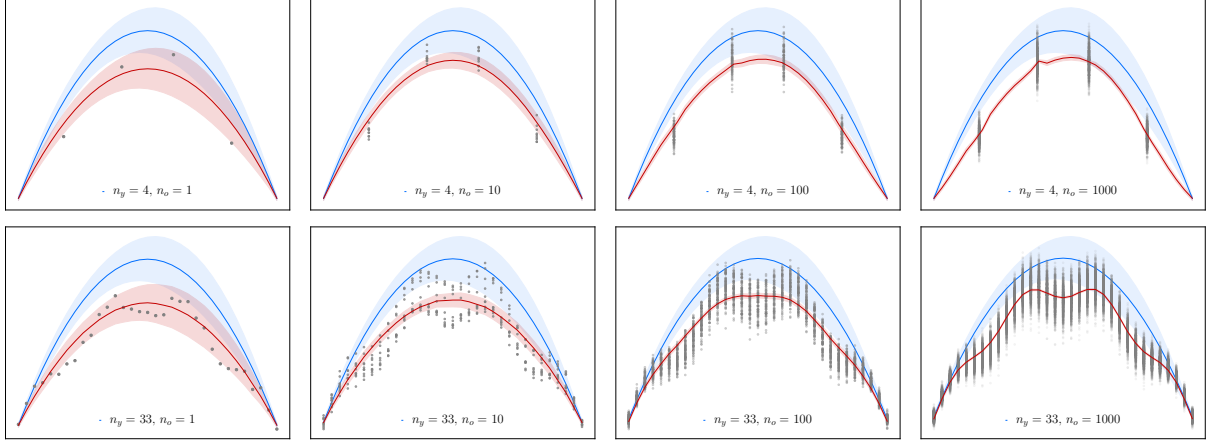
Figure 13: One-dimensional problem with random diffusivity. Finite element density $p(\boldsymbol{u}|\boldsymbol{Y})$ conditioned on observation data (black dots). The blue lines represent the mean $\overline{\boldsymbol{u}}$ and the red lines the conditioned mean $\overline{\boldsymbol{u}}_{|Y}$. The shaded areas denote the corresponding 95% confidence regions. In each row the number of sensors n_y and each column the number of readings n_o is constant.
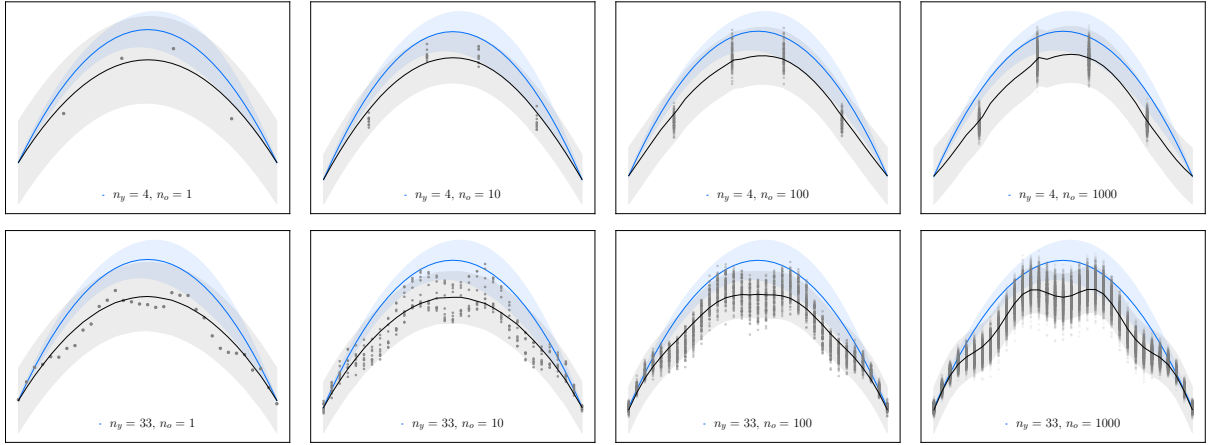

Figure 14: One-dimensional problem with random diffusivity. Inferred true system density $p(\boldsymbol{z}|\boldsymbol{Y})$ conditioned on observation data (black dots). The blue lines represent the mean $\overline{\boldsymbol{u}}$ and the black lines the conditioned mean $\overline{\boldsymbol{z}}_{|Y}$. The shaded areas denote the corresponding 95% confidence regions. In each row the number of sensors n_y and each column the number of readings n_o is constant.

used approaches for discretising the diffusion coefficient can be expressed in the form

$$\kappa(x) = \sum_i \psi_i(x)\kappa_i^{(a)} = \boldsymbol{\psi}(x)^\mathsf{T}\boldsymbol{\kappa}^{(a)}\,, \tag{65}$$

where $\boldsymbol{\psi}(x)$ is a set of basis functions and $\boldsymbol{\kappa}^{(a)}$ are their coefficients. Usually, the $\boldsymbol{\psi}(x)$ are chosen as either the Lagrange, B-spline, Karhunen-Loeve or Gaussian process basis functions, see [13, 44, 46].

In this example, we discretise the diffusion coefficient using Gaussian process basis functions. Specifically, $\boldsymbol{\kappa}^{(a)}$ are the diffusion coefficient values at the *anchor* points with the coordinates $\boldsymbol{x}^{(a)} = (0, 0.25, 0.5, 0.75, 1)^\mathsf{T}$. We obtain the basis functions $\boldsymbol{\psi}^{(a)}$ by conditioning a Gaussian process on the anchor point coefficients $\boldsymbol{\kappa}^{(a)}$, see (A.5) in Appendix A.1. The chosen Gaussian process has a zero mean and a squared exponential kernel with $\sigma_{\kappa^{(a)}} = 1$ and $\ell_{\kappa^{(a)}} = 0.32$. The obtained basis functions are infinitely smooth owing to the chosen squared exponential kernel and their shape is controlled by $\ell_{\kappa^{(a)}}$.

The vector of the element centre diffusion coefficients $\boldsymbol{\kappa}(\boldsymbol{X}^{(c)})$ required for finite element analysis is obtained by evaluating (65). When the source $\boldsymbol{f}$ is deterministic, as is usually the case in inverse problems, the finite element

density (22) conditioned on the anchor point coefficients $\boldsymbol{\kappa}^{(a)}$ is given by

$$p(\boldsymbol{u}|\boldsymbol{\kappa}^{(a)}) = \boldsymbol{\delta}\left(\boldsymbol{u} - \boldsymbol{A}\left(\boldsymbol{\kappa}^{(a)}\right)^{-1}\boldsymbol{f}\right), \tag{66}$$

where $\boldsymbol{\delta}(\cdot)$ denotes a Dirac measure. That is, the forward problem in conventional inverse problems is deterministic. On the contrary, in statFEM the forward problem has always a well defined distribution as given by (28). The marginal likelihood corresponding to (66) is according to (36) given by

$$p(\boldsymbol{y}|\boldsymbol{\kappa}^{(a)}, \dots) = \int p(\boldsymbol{y}|\boldsymbol{u}, \dots)p(\boldsymbol{u}|\boldsymbol{\kappa}^{(a)})\,\mathrm{d}\boldsymbol{u} = \mathcal{N}\left(\rho \boldsymbol{P}\boldsymbol{A}\left(\boldsymbol{\kappa}^{(a)}\right)^{-1}\boldsymbol{f}, \boldsymbol{C}_d + \boldsymbol{C}_e\right), \tag{67}$$

where we made in contrast to (38) the conditioning of the marginal likelihood on the anchor point coefficients and other hyperparameters, represented by dots, explicit.

In this example we choose a deterministic source $f(x) = 1$ and consider $n_y = 33$ observation points located at each of the nodes of the uniform finite element mesh with $n_e = 32$ elements and $n_o \in \{1, 5, 25, 50\}$ repeated readings. We sample the synthetic observation matrix $\boldsymbol{Y} \in \mathbb{R}^{n_y \times n_o}$ from the Gaussian

$$p(\boldsymbol{y}|\boldsymbol{\kappa}^{(a)}, \dots) = \mathcal{N}\left(\boldsymbol{P}\boldsymbol{A}\left(\boldsymbol{\kappa}_z^{(a)}\right)^{-1}\boldsymbol{f}, 0.01^2\boldsymbol{I}\right), \tag{68}$$

where $\boldsymbol{\kappa}_z^{(a)} = \{\ln 0.7, \ln 1, \ln 0.7, \ln 0.4, \ln 0.7\}$ are the true coefficients at the respective anchor points $\boldsymbol{x}^{(a)}$. In the underlying statistical generating model (29) the hyperparameters have been chosen with $\rho = 1$, $\sigma_d = 0$ and $\sigma_e = 0.01$. In addition, ρ and σ_d are chosen to be deterministic, i.e. their priors and posteriors are fixed to the given values, and σ_e is a random variable. Under these conditions the statistical generating model reduces to $\boldsymbol{y} = \boldsymbol{u} + \boldsymbol{e}$ as is usually assumed in inverse problems.

In hyperparameter learning we consider $\boldsymbol{w} = (\boldsymbol{\kappa}^{(a)\top}, \sigma_e)^\top$ as the unknown parameters to be inferred from the observation matrix $\boldsymbol{Y}$ sampled from (68). According to (48) the hyperparameter posterior is given by $p(\boldsymbol{w}|\boldsymbol{Y}) \propto p(\boldsymbol{Y}|\boldsymbol{w})p(\boldsymbol{w})$. We choose as priors the Gaussian densities

$$p(\boldsymbol{\kappa}^{(a)}) = \mathcal{N}\left((\ln 0.8, \ln 1.1, \ln 0.8, \ln 0.5, \ln 0.8)^\top, 0.0025\,\boldsymbol{I}\right), \quad p(\sigma_e) = \mathcal{N}(0.0075, 4 \cdot 10^{-6}). \tag{69}$$

Notice that the mean of the priors are different from the true generating process parameters used in (68) for sampling the synthetic observations. We sample the posterior $p(\boldsymbol{w}|\boldsymbol{Y})$ with MCMC using 50000 iterations with an acceptance ratio of around 0.3. Each n_o requires one MCMC run so that in total four runs are performed. The inferred densities $p(\boldsymbol{w}|\boldsymbol{Y})$ are shown in Figure 15. The red vertical lines indicate the mean of the priors and the black lines the mean of the true generating process. Even with only one set of readings, $n_o = 1$, the mean of the posteriors are visually very different from the mean of the priors. When n_o becomes larger the mean of the posteriors converge indeed towards the true generating process parameter values and the standard deviation of the posteriors become smaller. In Figure 16 the mean of the true diffusion coefficient $\mu_z(x)$ and the inferred diffusion coefficient $\mu(x)$ over the domain are shown. The anchor points and their coefficients are denoted by black dots. As visually apparent with increasing number of readings the inferred diffusion coefficient converges towards the true diffusion coefficient.

4.1.5. Finite element mesh selection

We consider Bayesian model comparison for selecting a finite element mesh which can best explain the observed data. That is, for a given observation matrix $\boldsymbol{Y}^{(i)}$ we are looking for the finite element mesh $\mathcal{M}_j$ with the highest posterior probability $p(\mathcal{M}_j|\boldsymbol{Y}^{(i)})$. In this example, we examine four different uniform finite element meshes $\{\mathcal{M}_j\}_{j=1}^4$ with the respective finite element sizes $h_1 = 1/4$, $h_2 = 1/8$, $h_3 = 1/16$ and $h_4 = 1/32$. Furthermore, we choose a deterministic diffusion coefficient with $\mu(x) = 1$ and a random source with the mean

$$\overline{f}(x) = \frac{\pi^2}{5}\sin(\pi x) + \frac{49\pi^2}{50}\sin(7\pi x) \tag{70}$$

and the covariance parameters $\sigma_f = 0.2$ and $\ell_f = 0.25$.

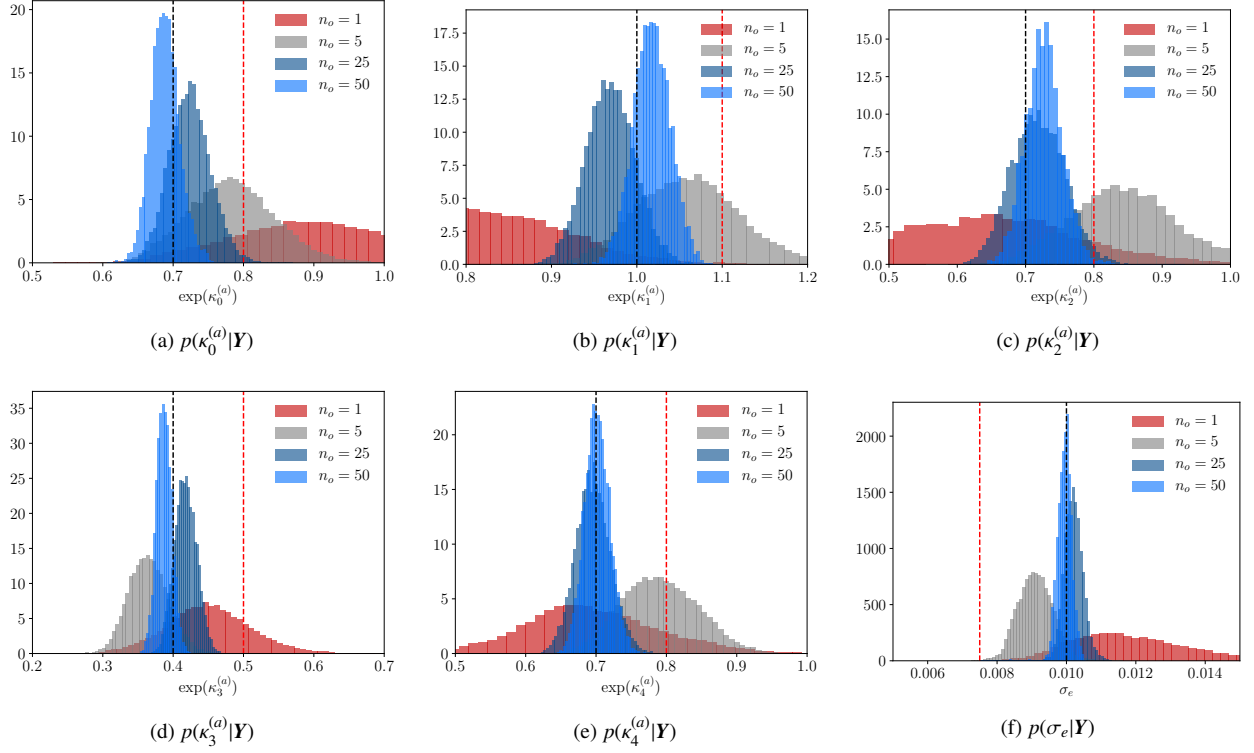(a) $p(\kappa_0^{(a)}|\boldsymbol{Y})$ (b) $p(\kappa_1^{(a)}|\boldsymbol{Y})$ (c) $p(\kappa_2^{(a)}|\boldsymbol{Y})$

(d) $p(\kappa_3^{(a)}|\boldsymbol{Y})$ (e) $p(\kappa_4^{(a)}|\boldsymbol{Y})$ (f) $p(\sigma_e|\boldsymbol{Y})$

Figure 15: One-dimensional conventional Bayesian inverse problem. Posteriors of the parameters $\boldsymbol{w} = (\boldsymbol{\kappa}^{(a)\top}, \sigma_e)^\top$ for $n_o \in \{1, 5, 25, 50\}$ obtained with MCMC. The red vertical lines indicate the mean of the prior and the black lines the true generating process parameters.
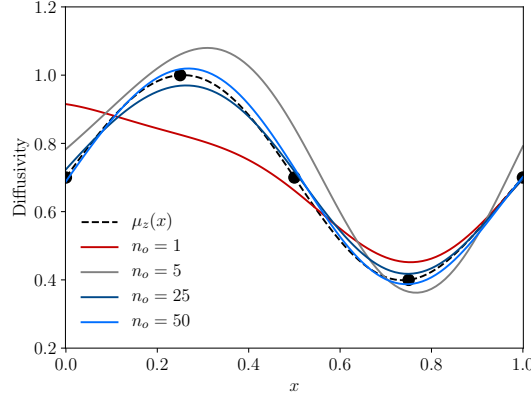
Figure 16: One-dimensional conventional Bayesian inverse problem. Comparison of the inferred diffusion coefficient $\mu(x)$ with the true diffusion coefficient $\mu_z(x)$ (dashed black line). The five black dots denote the anchor points for $\mu_z(x)$.

Four synthetic observation matrices $\left\{\boldsymbol{Y}^{(i)}\right\}_{i=1}^4$ are sampled from the marginal likelihood (38) with the scaling parameter $\rho = 0.8$, the mismatch covariance parameters $\sigma_d = 0.005$ and $\ell_d = 0.3$, and the sensor noise $\sigma_e = 0.005$. Each mesh $\mathcal{M}_i$ has the corresponding observation matrix $\boldsymbol{Y}^{(i)} \in \mathbb{R}^{n_y \times n_o}$ with $n_y \in \{11, 33\}$ observation locations and $n_o = 100$ readings. See Figure 8 for the location of the observation points. Thus, each column of $\boldsymbol{Y}^{(i)}$ is sampled from

22

(a) $\log p(\mathcal{M}_j|\mathbf{Y}^{(1)})$

(b) $\log p(\mathcal{M}_j|\mathbf{Y}^{(2)})$

(c) $\log p(\mathcal{M}_j|\mathbf{Y}^{(3)})$

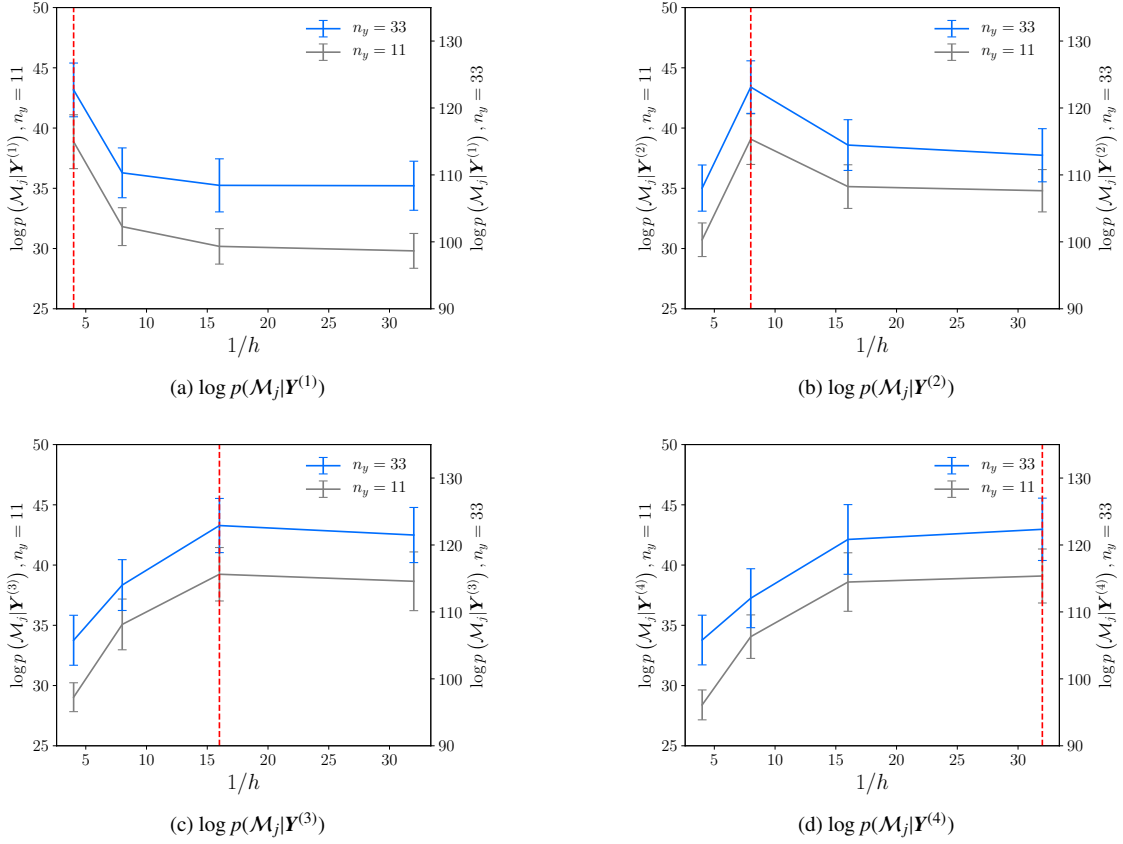(d) $\log p(\mathcal{M}_j|\mathbf{Y}^{(4)})$

Figure 17: Comparison of one-dimensional models. Mesh log-posterior probability $p(\mathcal{M}_j|\mathbf{Y}^{(i)})$ over the inverse element size $1/h$ for the four different meshes $\{\mathcal{M}_j\}_{j=1}^4$ and data sets $\{\mathbf{Y}^{(i)}\}_{i=1}^4$. In each plot the legend on the left refers to $n_y = 11$ and on the right to $n_y = 33$ observation locations.

the marginal likelihood

$$p(\mathbf{y}|\mathcal{M}_i) = \mathcal{N}\left(0.8\mathbf{P}\overline{\mathbf{u}}(\mathcal{M}_i),\ 0.64\mathbf{P}\mathbf{C}_u(\mathcal{M}_i)\mathbf{P}^\top + \mathbf{C}_d + \mathbf{C}_e\right), \tag{71}$$

where the finite element mean $\overline{\mathbf{u}}(\mathcal{M}_i)$ and covariance $\mathbf{C}_u(\mathcal{M}_i)$ are obtained with the mesh $\mathcal{M}_i$.

As in preceding sections, prior to computing the posterior probabilities $p(\mathcal{M}_j|\mathbf{Y}^{(i)})$ we first infer the hyperparameters $\mathbf{w} = (\rho, \sigma_d, \ell_d)^\top$ of the statistical generating model. In doing so we choose a non-informative prior $p(\mathbf{w}) \propto 1$. The determined hyperparameters have very similar values like the ones in the generating density (71), confirming the consistency of the proposed approach. Once the hyperparameters are known we compute the posterior probabilities $p(\mathcal{M}_j|\mathbf{Y}^{(i)}) \propto p(\mathbf{Y}^{(i)}|\mathcal{M}_j)p(\mathcal{M}_j)$ with the marginal likelihood $p(\mathbf{Y}^{(i)}|\mathcal{M}_j)$ given in (44). Assuming a non-informative prior $p(\mathcal{M}_j) \propto 1$ we have $p(\mathcal{M}_j|\mathbf{Y}^{(i)}) \propto p(\mathbf{Y}^{(i)}|\mathcal{M}_j)$.

Figure 17 shows the log-posterior probability $\log p\left(\mathcal{M}_j|\mathbf{Y}^{(i)}\right)$ of the four different meshes and the four different observation matrices. Considering that there are two different observation arrangements $n_y \in \{11, 33\}$, there are in total $2 \times 16\ \mathcal{M}_j$ and $\mathbf{Y}^{(i)}$ combinations. The bars in Figure 17 indicate the standard deviations of $\log p\left(\mathcal{M}_j|\mathbf{Y}^{(i)}\right)$ obtained by sampling each 50 times. Clearly, the maximum of $\log p(\mathcal{M}_j|\mathbf{Y}^{(i)})$ is always where j is equal to i. That is, for a given data set the most probable mesh is the one with which the data has been generated. We can use this information to choose the most suitable mesh for a given data set. When two meshes have a similar log-posterior probability we can choose for computational efficiency the coarser one. To explain Figure 17, note that an observation matrix generated with a coarse mesh will lack the higher frequencies of the solution field. For such an observation

23

(a) Geometry and boundary conditions

(b) Isocontours of source $f(x)$ drawn from a Gaussian process with $\overline{f}(x) = 1.0$, $\sigma_f = 0.3$ and $\ell_f = 0.15$ and its corresponding solution $u_h(x)$
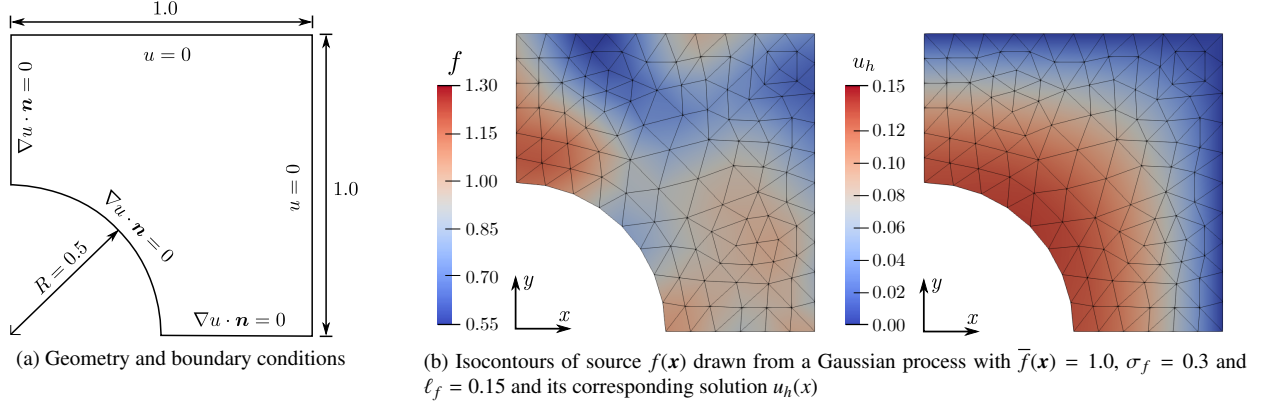
Figure 18: Plate with a hole.

matrix there is no need to use a complex computational model with a finer mesh. Consequently, Bayesian inference allows us to identify the simplest possible model as stipulated by the well-known Occam's razor principle [6, Ch. 28].

4.2. Plate with a hole

As a two-dimensional example, we study a Poisson problem on a unit-square with a circular hole shown in Figure 18a. The boundary conditions on the five edges are chosen as indicated in the figure. In this example only the source $f(x)$ is chosen as random. We discretise the weak form with the finite element mesh shown in Figure 18b consisting of 208 standard linear triangular elements and 125 nodes.

4.2.1. Posterior finite element density and system response for random source

The deterministic diffusion coefficient is assumed to be $\mu(x) = 1.0$ and the random source $f(x)$ is a Gaussian process with a mean $\overline{f}(x) = 1.0$ and a squared exponential covariance kernel with the parameters $\sigma_f = 0.3$ and $\ell_f = 0.15$. As introduced in Section 2.2, the density of the source vector f is given by the multivariate Gaussian $p(f) = \mathcal{N}(\overline{f}, C_f)$ and the density of the finite element solution by $p(u) = \mathcal{N}(A^{-1}\overline{f}, A^{-1}C_f A^{-T})$. In Figure 18b a representative source distribution and its respective finite element solution are depicted. As visually apparent and discussed in Section 2.2 the solution field $u_h(x)$ is significantly smoother than the source field $f(x)$ owing to the smoothing property of the inverse differential, i.e. Laplace, operator.

We consider as the true system response $z(x)$ the solution of a second much finer finite element model. The fine mesh is obtained by repeated quadrisection of the coarse mesh shown in Figure 18b and has 53248 elements. The random source of the fine finite element model is a Gaussian process

$$g(x) \sim \mathcal{GP}\left(\overline{g}(x), c_g(x, x')\right) \tag{72}$$

with the mean

$$\overline{g}(x) = \frac{1}{2} + \frac{1}{2}\sin(\pi\|x\|) + 3\sin(7\pi\|x\|) \tag{73}$$

and the squared exponential covariance kernel with the parameters $\sigma_g = 0.1$ and $\ell_g = 0.2$. The density of the corresponding finite element solution on the fine mesh is given by

$$z = \mathcal{N}(\overline{z}, C_z) = \mathcal{N}\left(A_g^{-1}\overline{g}, A_g^{-1}C_{u_g}A_g^{-T}\right), \tag{74}$$

The true system response $z(x) = \sum_i \phi_i(x)z_i$ and the finite element solution $u_h(x) = \sum_i \phi_i(x)u_i$ are compared in Figure 19. Both fields are plotted along the diagonal of the domain, i.e. the line with $x^{(2)} = x^{(1)}$. As described above the source terms and meshes chosen for $z(x)$ and $u_h(x)$ are very different. Their difference represents the model mismatch.
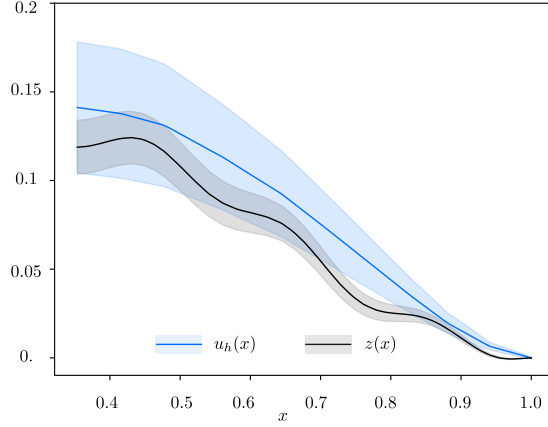
24

Figure 19: Plate with a hole. Finite element solution $u_h(x)$ and the true system response $z(x)$ along the diagonal of the domain with $x^{(2)} = x^{(1)}$. The solid lines represent the respective means and the shaded areas the 95% confidence regions.
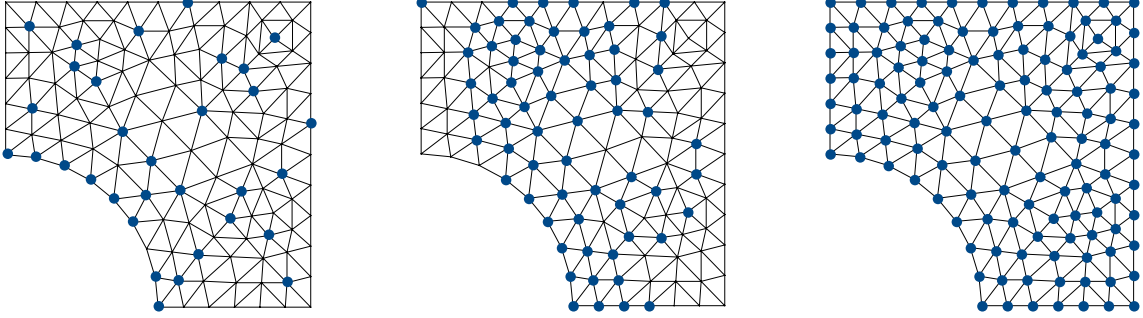

Figure 20: Plate with a hole. Location of the $n_y \in \{32, 64, 125\}$ observations points for the data vector $\boldsymbol{y}$ chosen according to a Sobol sequence. The observation points are located at the finite element nodes marked with dots.

We sample the synthetic observation matrix $\boldsymbol{Y} \in \mathbb{R}^{n_y \times n_o}$ from the multivariate Gaussian

$$\boldsymbol{y} = \mathcal{N}\left(A_g^{-1}\overline{\boldsymbol{g}},\ A_g^{-1} C_g A_g^{-\top} + 2.5 \cdot 10^{-5} \boldsymbol{I}\right) \tag{75}$$

with the observation noise $\sigma_e^2 = 2.5 \cdot 10^{-5}$. The selected $n_y \in \{32, 64, 125\}$ observation points are all located at the finite element nodes, see Figure 18b. They are distributed according to a Sobol sequence so that the set observation points are nested [50]. We sample at each of the n_y sample points $n_o = \{1, 10, 100, 1000\}$ repeated readings.

As for the one-dimensional example in Section 4.1.2, only the parameters $\boldsymbol{w} = (\rho, \sigma_d, \ell_d)^\top$ of the statistical generating model are assumed to be unknown. Choosing an uninformed prior $p(\boldsymbol{w}) \propto 1$ we sample the posterior density of the hyperparameters $p(\boldsymbol{w}|\boldsymbol{Y})$ using a standard MCMC algorithm, see Appendix A.3. For each combination of n_y and n_o we run 50000 iterations with an average acceptance ratio of 0.254. In Figure 21 the obtained histograms for $p(\rho|\boldsymbol{Y})$, $p(\sigma_d|\boldsymbol{Y})$ and $p(\ell_d|\boldsymbol{Y})$ for $n_y = 64$ are depicted. The observed overall trends are very similar to the one-dimensional example. The standard deviations become smaller with increasing n_o. In Tables 4, 5 and 6 the mean and standard deviations of these three plots and other n_y and n_o combinations are given. As to be expected when either n_y, n_o or both are increased the standard deviation becomes smaller.

The inferred posterior finite element densities $p(\boldsymbol{u}|\boldsymbol{Y})$ computed according to (43) for different number of observation points n_y and readings n_o are shown in Figure 22. We used again the empirical averages $\overline{\boldsymbol{w}} = \mathbb{E}[\boldsymbol{w}]$ of the hyperparameters as point estimates. As for the one-dimensional example, the posterior mean $\overline{\boldsymbol{u}}_{|Y}$ converges with increasing n_y and n_o to the true system response mean $\overline{z}$. At the same time, the posterior covariance $C_{u|Y}$ converges to zero. Obviously, the finite element solution space cannot represent the true system response so that some differences between $\overline{\boldsymbol{u}}_{|Y}$ and $\overline{z}$ remain, see, e.g., the plot for $n_y = 125$ and $n_o = 1000$. In Figure 23 the inferred true system

(a) $p(\rho|Y)$ (b) $p(\sigma_d|Y)$ (c) $p(\ell_d|Y)$
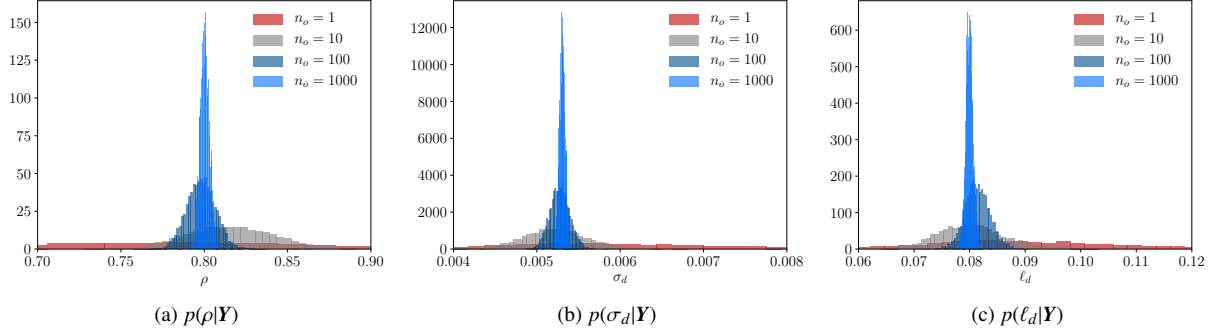
Figure 21: Plate with a hole. Posteriors of the parameters ρ, σ_d and ℓ_d for $n_y = 64$ and $n_o \in \{1, 10, 100, 1000\}$ obtained with MCMC.

	$n_o = 1$	$n_o = 10$	$n_o = 100$	$n_o = 1000$
$n_y = 32$	0.81665 ± 0.09841	0.81056 ± 0.02826	0.80928 ± 0.00943	0.81085 ± 0.00289
$n_y = 64$	0.81051 ± 0.10648	0.81710 ± 0.02713	0.79831 ± 0.00853	0.80026 ± 0.00278
$n_y = 125$	0.78432 ± 0.08124	0.82240 ± 0.02761	0.79888 ± 0.00861	0.79806 ± 0.00271

Table 4: Plate with a hole. Empirical mean and standard deviation of $p(\rho|Y)$.

	$n_o = 1$	$n_o = 10$	$n_o = 100$	$n_o = 1000$
$n_y = 32$	0.00678 ± 0.00229	0.00498 ± 0.00043	0.00546 ± 0.00016	0.00545 ± 0.00005
$n_y = 64$	0.00595 ± 0.00145	0.00521 ± 0.00038	0.00527 ± 0.00011	0.00530 ± 0.00003
$n_y = 125$	0.00444 ± 0.00058	0.00503 ± 0.00024	0.00463 ± 0.00008	0.00469 ± 0.00002

Table 5: Plate with a hole. Empirical mean and standard deviation of $p(\sigma_d|Y)$.

	$n_o = 1$	$n_o = 10$	$n_o = 100$	$n_o = 1000$
$n_y = 32$	0.28759 ± 0.17127	0.06962 ± 0.01399	0.08344 ± 0.00447	0.08644 ± 0.001188
$n_y = 64$	0.08726 ± 0.02194	0.08049 ± 0.00637	0.08141 ± 0.00228	0.080011 ± 0.00066
$n_y = 125$	0.06002 ± 0.00752	0.06870 ± 0.00429	0.06949 ± 0.00133	0.06949 ± 0.00041

Table 6: Plate with a hole. Empirical mean and standard deviation of $p(\ell_d|Y)$.

response densities $p(z|Y)$ according to (39) are shown. The two means $\overline{z}_{|Y}$ and $\overline{u}_{|Y}$ are identical. In comparison to the true system covariance C_z, the obtained posterior covariance $C_{z|Y}$ is slightly smaller towards the centre and larger towards the boundary of the plate. To achieve a better match between C_z and $C_{z|Y}$ it is necessary to model the mismatch covariance differently. The used squared exponential kernel (31) depends on a scalar scaling parameter σ_d. A possible remedy would involve the modelling of σ_d as a spatially varying field.

5. Conclusions

We introduced the statistically constructed finite element method, statFEM, that provides a means for coherent synthesis of observation data and finite element discretised mathematical models. Thus, statFEM can be interpreted as a physics-informed machine learning or Bayesian learning technique. The mathematical models used in engineering practice are highly misspecified due to inherent uncertainties in loading, material properties, geometry, and the many inevitable modelling assumptions. StatFEM enables the fusion of observation data from in-situ monitoring of engineering systems with the possibly severely misspecified finite element model. This is conceptually different from
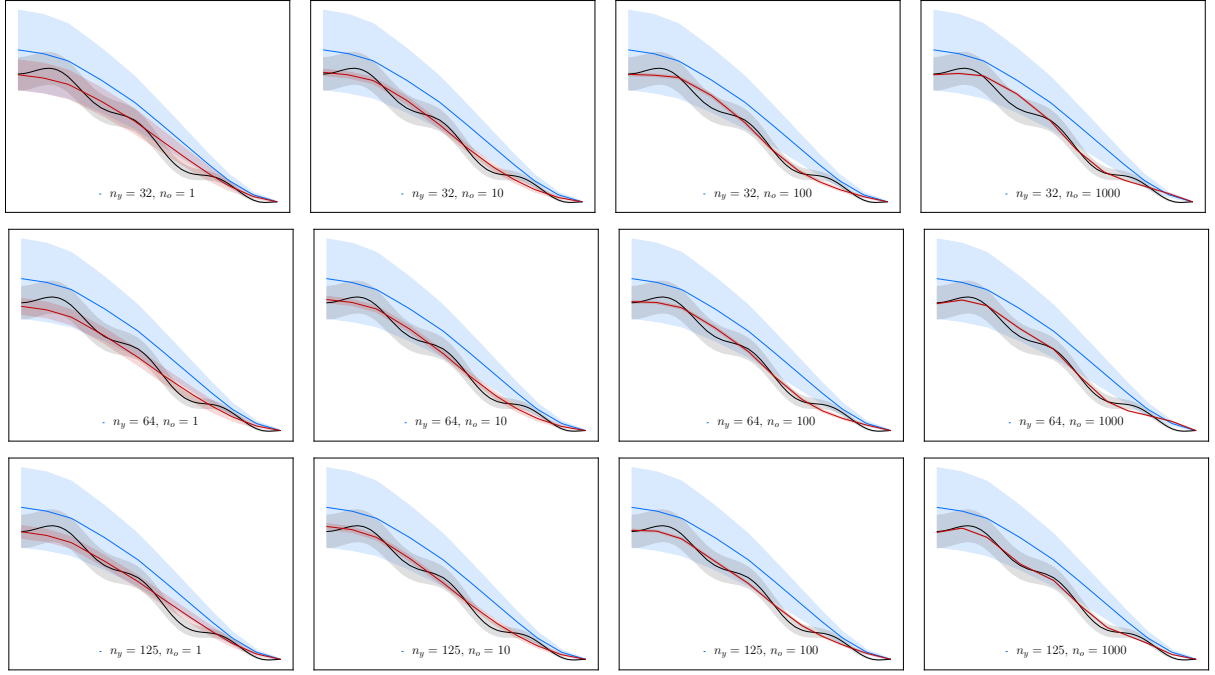
Figure 22: Plate with a hole. Finite element density $p(\boldsymbol{u}|\boldsymbol{Y})$ conditioned on observation data sampled from the multivariate Gaussian density (75) (black line). The blue lines represent the mean $\overline{\boldsymbol{u}}$ and the red lines the conditioned mean $\overline{\boldsymbol{u}}_{|Y}$. The shaded areas denote the corresponding 95% confidence regions. In each row the number of sensors n_y and each column the number of readings n_o is constant. All plots along the diagonal of the domain with $x^{(2)} = x^{(1)}$
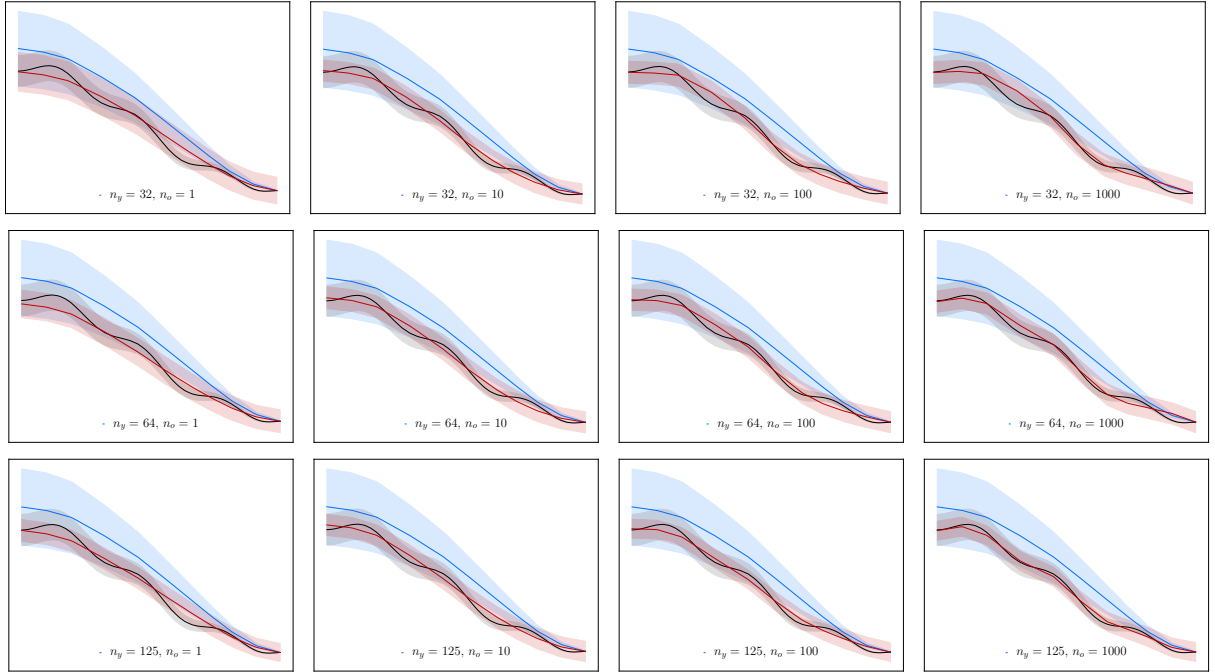

Figure 23: Plate with a hole. Inferred true system density $p(z|\boldsymbol{Y})$ conditioned on observation data sampled from the multivariate Gaussian density (75) (black line). The blue lines represent the mean $\overline{\boldsymbol{u}}$ and the red lines the conditioned mean $\overline{z}_{|Y}$. The shaded areas denote the corresponding 95% confidence regions. In each row the number of sensors n_y and each column the number of readings n_o is constant. All plots along the diagonal of the domain with $x^{(2)} = x^{(1)}$

traditional Bayesian inversion or calibration, which aim to learn the parameters, like the material and geometry, of the model. In statFEM the probabilistic finite element model provides the prior density which is used in determining the posterior densities of the random variables and hyperparameters of the postulated statistical generating model. As numerically demonstrated, with increasing number of observations the obtained posterior densities converge towards the true system density. Informally, choosing a more informative prior, or in some sense better finite element model, enables us to approximate the true system density with less data. In engineering practice observational data is usually scarce so that informative priors are important. When data is abundant instead of statFEM one could argue that a purely data-driven approach, like Gaussian process regression, may be sufficient if the system under study is relatively simple, see e.g. [51]. Building upon the Bayesian statistics framework, statFEM provides a wide range of techniques to compare and interrogate models and data of different fidelity and resolution, which we only partially exploited in this paper.

In closing, a number of possible extensions of statFEM are noteworthy. We used in this paper, purely for illustrative purposes, the squared exponential kernel as a covariance kernel. Especially, for the mismatch variable the use of other kernels or linear combination of kernels from Gaussian process regression literature appears promising, see e.g. [24, Ch. 4]. Moreover, we obtained the prior density of the finite element solution by approximating the forward problem in the random domain by a first-order perturbation method. Although this method has certain advantages, like ease of extensibility to nonlinear and nonstationary problems [52], the random perturbations must be relatively small. Advanced approaches for solving stochastic partial differential equations do not have this limitation and may provide more informative priors. Furthermore, we apply in statFEM Bayes rule on several levels in turn which requires that hyperparameter densities are replaced with point estimates. Alternatively, it is possible to obtain the density of random variables and hyperparameters by marginalisation from a high-dimensional joint density. The joint density can be sampled, for instance, with the Metropolis-within-Gibbs algorithm; see [13] for an application of this approach in the Bayesian inversion context. A possible further extension of statFEM concerns the refinement of the postulated statistical model to incorporate computational models of different fidelity by incorporating ideas from recursive co-kriging [53–55]. This would allow, for instance, the synthesis of observation data with a detailed expensive to evaluate 3D elasticity model and an empirical engineering formula. Finally, extension of statFEM to time-evolving linear and nonlinear Korteweg-de Vries equation has been considered in [56].

Acknowledgement

Appendix A. Appendix

Appendix A.1. Gaussian process regression

The diffusion coefficient $\kappa(x) \sim \mathcal{GP}(\overline{\kappa}(x), c_\kappa(x, x'))$ is assumed to be a Gaussian process and is approximated as constant in each finite element giving the multivariate Gaussian density (14), repeated here for convenience,

$$\kappa \sim p(\kappa) = \mathcal{N}\left(\overline{\kappa}\left(X^{(c)}\right), C_\kappa\left(X^{(c)}, X^{(c)}\right)\right).$$

A common operation is to condition this density on a set of prescribed coefficients $\kappa^{(a)} = \{\kappa_i^{(a)}\}_{i=1}^{n_a}$ at the respective anchor points $X^{(a)} = \{x_i^{(a)}\}_{i=1}^{n_a}$. The joint probability of the anchor point diffusion vector $\kappa^{(a)} \in \mathbb{R}^{n_a}$ and the element diffusion vector $\kappa \in \mathbb{R}^{n_e}$ is given by

$$p(\kappa^{(a)}, \kappa) = \mathcal{N}\left(\begin{bmatrix} \overline{\kappa}\left(X^{(a)}\right) \\ \overline{\kappa}\left(X^{(c)}\right) \end{bmatrix}, \begin{bmatrix} C_\kappa\left(X^{(a)}, X^{(a)}\right) & C_\kappa\left(X^{(a)}, X^{(c)}\right) \\ C_\kappa\left(X^{(c)}, X^{(a)}\right) & C_\kappa\left(X^{(c)}, X^{(c)}\right) \end{bmatrix}\right). \tag{A.1}$$
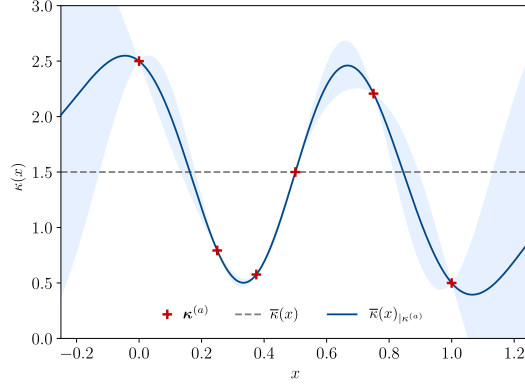
Figure A.24: Illustrative Gaussian process regression example. The coefficients $\boldsymbol{\kappa}^{(a)}$ at the six anchor points (in red) are prescribed. The solid line represents the conditioned mean $\overline{\boldsymbol{\kappa}}_{|\kappa^{(a)}}$ and the shaded area the 95% confidence region obtained from the covariance $\boldsymbol{C}_{\kappa|\kappa^{(a)}}$.

The mean and the covariance matrix are obtained by evaluating the prescribed mean $\overline{\kappa}(\boldsymbol{x})$ and covariance kernel $c_\kappa(\boldsymbol{x}, \boldsymbol{x}')$ at the respective points. According to standard results, see e.g. [24, Chapter 2], the density of $\boldsymbol{\kappa}$ conditioned on $\boldsymbol{\kappa}^{(a)}$ is given by

$$\boldsymbol{\kappa}_{|\kappa^{(a)}} \sim p\left(\boldsymbol{\kappa}|\boldsymbol{\kappa}^{(a)}\right) = \mathcal{N}\left(\overline{\boldsymbol{\kappa}}_{|\kappa^{(a)}}, \boldsymbol{C}_{\kappa|\kappa^{(a)}}\right), \tag{A.2}$$

where

$$\overline{\boldsymbol{\kappa}}_{|\kappa^{(a)}} = \overline{\boldsymbol{\kappa}}\left(\boldsymbol{X}^{(c)}\right) + \boldsymbol{C}_\kappa\left(\boldsymbol{X}^{(c)}, \boldsymbol{X}^{(a)}\right)\boldsymbol{C}_\kappa\left(\boldsymbol{X}^{(a)}, \boldsymbol{X}^{(a)}\right)^{-1}\left(\boldsymbol{\kappa}^{(a)} - \overline{\boldsymbol{\kappa}}\left(\boldsymbol{X}^{(a)}\right)\right), \tag{A.3a}$$

$$\boldsymbol{C}_{\kappa|\kappa^{(a)}} = \boldsymbol{C}_\kappa\left(\boldsymbol{X}^{(c)}, \boldsymbol{X}^{(c)}\right) - \boldsymbol{C}_\kappa\left(\boldsymbol{X}^{(c)}, \boldsymbol{X}^{(a)}\right)\boldsymbol{C}_\kappa\left(\boldsymbol{X}^{(a)}, \boldsymbol{X}^{(a)}\right)^{-1}\boldsymbol{C}_\kappa\left(\boldsymbol{X}^{(a)}, \boldsymbol{X}^{(c)}\right). \tag{A.3b}$$

If needed, we sample from this density by first computing the Cholesky decomposition $\boldsymbol{C}_{\kappa|\kappa^{(a)}} = \boldsymbol{L}\boldsymbol{L}^\mathsf{T}$ and then sampling the Gaussian white noise $\boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ to obtain

$$\boldsymbol{\kappa}_{|\kappa^{(a)}} = \overline{\boldsymbol{\kappa}}_{|\kappa^{(a)}} + \boldsymbol{L}\boldsymbol{e}. \tag{A.4}$$

As an example for Gaussian process regression, in Figure A.24 the approximation of $\kappa(\boldsymbol{x})$ over a one-dimensional domain $\Omega = [0, 1]$ is illustrated. The six prescribed anchor point coefficients $\boldsymbol{\kappa}^{(a)}$ lie on the curve $1.5 + \cos(3\pi x)$. The prescribed mean is $\overline{\kappa}(x) = 1.5$ and the parameters of the squared exponential kernel are $\sigma_\kappa = 1$ and $\ell_\kappa = 0.2$. The depicted conditioned mean and the 95% confidence region are obtained from (A.3a) and (A.3b).

Finally, the conditioned mean A.3a can be used to define the interpolating basis functions $\boldsymbol{\psi}(\boldsymbol{x})$. Specifically, choosing the mean of the Gaussian process as $\overline{\kappa}(\boldsymbol{x}) = 0$ we can define

$$\overline{\kappa}(\boldsymbol{x})_{|\kappa^{(a)}} = \boldsymbol{C}_\kappa\left(\boldsymbol{x}, \boldsymbol{X}^{(a)}\right)\boldsymbol{C}_\kappa\left(\boldsymbol{X}^{(a)}, \boldsymbol{X}^{(a)}\right)^{-1}\boldsymbol{\kappa}^{(a)} = \boldsymbol{\psi}(\boldsymbol{x})^\mathsf{T}\boldsymbol{\kappa}^{(a)}. \tag{A.5}$$

Appendix A.2. Computation of the posterior finite element density

In deriving the posterior finite element density $p(\boldsymbol{u}|\boldsymbol{y})$ and several other places we make use of the fact that the product of two Gaussian densities is again a Gaussian, see e.g. [8, 19]. To see this, it is sufficient to focus on the argument of the respective exponential function and to bring it into a quadratic form. The normalisation constant for the so obtained Gaussian can be determined by inspection. The steps in obtaining the quadratic form are as follows

$$\begin{aligned} p(\boldsymbol{u}|\boldsymbol{y}) &= \frac{p(\boldsymbol{y}|\boldsymbol{u})p(\boldsymbol{u})}{p(\boldsymbol{y})} \propto p(\boldsymbol{y}|\boldsymbol{u})p(\boldsymbol{u}) \\ &\propto \exp\left((\rho\boldsymbol{P}\boldsymbol{u} - \boldsymbol{y})^\mathsf{T}(\boldsymbol{C}_d + \boldsymbol{C}_e)^{-1}(\rho\boldsymbol{P}\boldsymbol{u} - \boldsymbol{y})\right)\exp\left((\overline{\boldsymbol{u}} - \boldsymbol{u})^\mathsf{T}\boldsymbol{C}_u^{-1}(\overline{\boldsymbol{u}} - \boldsymbol{u})\right) \\ &= \exp\left(\boldsymbol{u}^\mathsf{T}\boldsymbol{B}\boldsymbol{u} - 2\boldsymbol{a}^\mathsf{T}\boldsymbol{u} + \dots\right), \end{aligned} \tag{A.6}$$

where

$$B = \rho^2 P^\mathsf{T} (C_d + C_e)^{-1} P + C_u^{-1}$$
$$a = \rho P^\mathsf{T} (C_d + C_e)^{-1} y + C_u^{-1} \overline{u} \,.$$

By completion of the square, the last expression in (A.6) is evidently proportional to

$$p(u|y) \propto \exp\left(\left(B^{-1} a - u \right)^\mathsf{T} B \left(B^{-1} a - u \right) \right) . \tag{A.7}$$

After determining the normalisation constant by inspection the posterior is given by

$$p(u|y) = \frac{1}{\sqrt{(2\pi)^{n_u} |B|}} \exp\left(\left(B^{-1} a - u \right)^\mathsf{T} B \left(B^{-1} a - u \right) \right) . \tag{A.8}$$

By comparison with (37) we can conclude

$$C_{u|y} = B^{-1}, \quad \overline{u}_{|y} = C_{u|y} a \,. \tag{A.9}$$

To avoid the inversion of large dense matrices, i.e. C_u, in evaluating $p(u|y)$, we use the Sherman-Morrison-Woodbury identity to obtain

$$C_{u|y} = C_u - C_u P^\mathsf{T} \left(\frac{1}{\rho^2} (C_d + C_e) + P C_u P^\mathsf{T} \right)^{-1} P C_u \tag{A.10a}$$

$$\overline{u}_{|y} = C_{u|y} \left(\rho P^\mathsf{T} (C_d + C_e)^{-1} y + C_u^{-1} \overline{u} \right) . \tag{A.10b}$$

Note that $C_u \in \mathbb{R}^{n_u \times n_u}$, $P \in \mathbb{R}^{n_y \times n_u}$ and $C_d, C_e \in \mathbb{R}^{n_y \times n_y}$ so that the bracket expression to be inverted is a dense matrix of dimension $n_y \times n_y$, where $n_y \ll n_u$ in most applications. When a direct solver is used, it is possible to speed up the computation of $P C_u P^\mathsf{T}$, which involves according to (27), the evaluation of terms like

$$P A^{-1} C_f A^{-1} P^\mathsf{T} = R^\mathsf{T} C_f R \,. \tag{A.11}$$

Here, it is sufficient to factorise the sparse system matrix A only once and to obtain R by column-wise back and forward substitution of P.

For completeness, we note that the posterior density $p(u|y)$ can be alternatively obtained from the joint density of $p(u, y)$, see [11, 43]. The statistical model introduced in Section 3.1 is given by

$$y = \rho P u + d + e \,, \tag{A.12}$$

and the random vectors have the densities $u \sim N(\overline{u}, C_u)$, $d \sim N(0, C_d)$ and $e \sim N(0, C_e)$. The joint density is defined by

$$p(u, y) = \mathcal{N}\left(\begin{pmatrix} \overline{u} \\ \overline{y} \end{pmatrix}, \begin{pmatrix} \mathbb{E}\left((u - \overline{u}) \otimes (u - \overline{u}) \right) & \mathbb{E}\left((u - \overline{u}) \otimes (y - \overline{y}) \right) \\ \mathbb{E}\left((y - \overline{y}) \otimes (u - \overline{u}) \right) & \mathbb{E}\left((y - \overline{y}) \otimes (y - \overline{y}) \right) \end{pmatrix} \right), \tag{A.13}$$

where the expectations must be taken over the random vectors u, d and e. After introducing the statistical model and the densities of the random vectors we obtain the joint density

$$p(u, y) = \mathcal{N}\left(\begin{pmatrix} \overline{u} \\ \rho P \overline{u} \end{pmatrix}, \begin{pmatrix} C_u & \rho C_u P^\mathsf{T} \\ \rho P C_u & \rho^2 P C_u P^\mathsf{T} + C_d + C_e \end{pmatrix} \right) . \tag{A.14}$$

The respective conditional density $p(u|y)$ according to Section Appendix A.1 yields for the covariance $C_{u|y}$ the same expression as (A.10a) and for the mean $\overline{u}_{|y}$ a slightly different expression than (A.10b).

Appendix A.3. Markov chain Monte Carlo method (MCMC)

We obtain the posterior of the statistical generating model parameters, i.e. $p(\boldsymbol{w}|\boldsymbol{Y}) \propto p(\boldsymbol{Y}|\boldsymbol{w})p(\boldsymbol{w})$, by sampling with the MCMC Metropolis algorithm. To avoid numerical stability issues with products of small densities in the likelihood $p(\boldsymbol{Y}|\boldsymbol{w})$ the logarithm of the posterior is considered. As the proposal density we choose a normal distribution $q(\boldsymbol{w}|\boldsymbol{v}) = \mathcal{N}(\boldsymbol{v}, \sigma_q^2 \boldsymbol{I})$ with the algorithmic parameter σ_q.

For a given sample $\boldsymbol{w}^{(i)}$ one MCMC iteration consists of the following steps:

S1. Sample $\boldsymbol{w} \sim q(\boldsymbol{w}|\boldsymbol{w}^{(i)})$.

S2. Compute acceptance probability

$$\alpha = \min\left(0, \ln\left(p(\boldsymbol{w}|\boldsymbol{Y})\right) - \ln(p\left(\boldsymbol{w}^{(i)}|\boldsymbol{Y}\right))\right). \tag{A.15}$$

S3. Generate a uniform random number $u \sim \mathcal{U}[0, 1]$.

S4. Set new sample to

$$\boldsymbol{w}^{(i+1)} = \begin{cases} \boldsymbol{w} & \text{if } \ln(u) < \alpha \\ \boldsymbol{w}^{(i)} & \text{if } \ln(u) \geq \alpha \end{cases}. \tag{A.16}$$

The non-negativity of the parameters collected in $\boldsymbol{w}$ can be enforced by choosing a prior density with non-negative support or applying a parameter transformation. When the prior has a non-negative support any negative proposal $\boldsymbol{w}$ will yield a zero posterior $p(\boldsymbol{w}|\boldsymbol{Y})$ and will be rejected in Step S4. As a result, all the collected samples $\{\boldsymbol{w}^{(i)}\}$ will be non-negative. The many redundant samples make this approach, however, inefficient. To avoid this, we consider the parameter transformation $\widetilde{\boldsymbol{w}} = \ln(\boldsymbol{w})$, where the logarithm is applied component-wise, and sample in the transformed domain. Rather than rewriting the above algorithm, the transformation can be taken into account by simply replacing the acceptance probability in Step S2 with

$$\alpha = \min\left(0, \ln\left(p\left(\exp(\boldsymbol{w})|\boldsymbol{Y}\right)\right) + \sum_j w_j - \ln\left(p\left(\exp(\boldsymbol{w}^{(i)})|\boldsymbol{Y}\right)\right) - \sum_j w_j^{(i)}\right), \tag{A.17}$$

where the two additional terms represent the Jacobian of the transformation of the posterior density. The samples $\{\boldsymbol{w}^{(i)}\}$ are now in the transformed domain and have to be transformed back. The so obtained samples $\{\exp(\boldsymbol{w}^{(i)})\}$ are all positive and have the desired distribution.

Moreover, in our computations we choose the algorithmic parameter σ_q for the proposal density $q(\boldsymbol{w}|\boldsymbol{v})$ so that the acceptance ratio in Step S4 is around 0.25. There are a number of efficient algorithms available to automate the selection of σ_q [57]. We discard the the first $25\% - 30\%$ of the obtained samples to account for the burn-in phase. For further details on MCMC algorithms see [14].

References

[1] J. Heyman, Structural Analysis: A Historical Approach, Cambridge University Press, 1998.

[2] J. T. Oden, R. Moser, O. Ghattas, Computer predictions with quantified uncertainty, part I, SIAM News 43 (2010) 1–3.

[3] C. J. Roy, W. L. Oberkampf, A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing, Computer Methods in Applied Mechanics and Engineering 200 (2011) 2131–2144.

[4] W. Lin, L. J. Butler, M. Z. E. B. Elshafie, C. R. Middleton, Performance assessment of a newly constructed skewed half-through railway bridge using integrated sensing, Journal of Bridge Engineering 24 (2018) 04018107–1–04018107–14.

[5] S. K. Everton, M. Hirsch, P. Stravroulakis, R. K. Leach, A. T. C, Review of in-situ process monitoring and in-situ metrology for metal additive manufacturing, Materials & Design 95 (2016) 431–445.

[6] D. J. C. MacKay, Information Theory, Inference and Learning Algorithms, Cambridge University Press, 2003.

[7] D. S. Sivia, J. Skilling, Data Analysis: A Bayesian Tutorial, Oxford University Press, 2006.

[8] K. P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.

[9] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, D. B. Rubin, Bayesian Data Analysis, CRC Press, third edn., 2014.

[10] S. Rogers, M. Girolami, A First Course in Machine Learning, Chapman and Hall/CRC, 2016.

[11] M. C. Kennedy, A. O'Hagan, Bayesian calibration of computer models, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63 (2001) 425–464.

[12] W. K. Liu, T. Belytschko, A. Mani, Random field finite elements, International Journal for Numerical Methods in Engineering 23 (1986) 1831–1845.

[13] Y. M. Marzouk, H. N. Najm, Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems, Journal of Computational Physics 228 (2009) 1862–1902.

[14] C. Robert, G. Casella, Monte Carlo Statistical Methods, Springer, second edn., 2004.

[15] N. Petra, J. Martin, G. Stadler, O. Ghattas, A computational framework for infinite-dimensional Bayesian inverse problems, Part II: Stochastic Newton MCMC with application to ice sheet flow inverse problems, SIAM Journal on Scientific Computing 36 (2014) A1525–A1555.

[16] A. Beskos, M. Girolami, S. Lan, P. E. Farrell, A. M. Stuart, Geometric MCMC for infinite-dimensional inverse problems, Journal of Computational Physics 335 (2017) 327–351.

[17] D. J. C. MacKay, Bayesian interpolation, Neural Computation 4 (1992) 415–447.

[18] D. J. C. MacKay, Comparison of approximate methods for handling hyperparameters, Neural Computation 11 (1999) 1035–1068.

[19] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[20] R. E. Kass, A. E. Raftery, Bayes factors, Journal of the American Statistical Association 90 (1995) 773–795.

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

[22] M. J. Bayarri, J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C.-H. Lin, J. Tu, A framework for validation of computer models, Technometrics 49 (2007) 138–154.

[23] D. Higdon, M. Kennedy, J. C. Cavendish, J. A. Cafeo, R. D. Ryne, Combining field data and computer simulations for calibration and prediction, SIAM Journal on Scientific Computing 26 (2004) 448–466.

[24] C. K. I. Williams, C. E. Rasmussen, Gaussian Processes for Machine Learning, MIT Press, 2006.

[25] B. Peherstorfer, K. Willcox, M. Gunzburger, Survey of multifidelity methods in uncertainty propagation, inference, and optimization, SIAM Review 60 (2018) 550–591.

[26] A. Forrester, A. Sobester, A. Keane, Engineering Design via Surrogate Modelling: A Practical Guide, John Wiley & Sons, 2008.

[27] J. Brynjarsdóttir, A. O'Hagan, Learning about physical parameters: the importance of model discrepancy, Inverse Problems 30 (2014) 1–24.

[28] Y. Ling, J. Mullins, S. Mahadevan, Selection of model discrepancy priors in Bayesian calibration, Journal of Computational Physics 276 (2014) 665–680.

[29] C. Jiang, Z. Hu, Y. Liu, Z. P. Mourelatos, D. Gorsich, P. Jayakumar, A sequential calibration and validation framework for model uncertainty quantification and reduction, Computer Methods in Applied Mechanics and Engineering 368 (2020) 113172:1 – 113172:30.

[30] H. G. Matthies, C. E. Brenner, C. G. Bucher, C. G. Soares, Uncertainties in probabilistic numerical analysis of structures and solids — stochastic finite elements, Structural Safety 19 (1997) 283–336.

[31] B. Sudret, A. Der Kiureghian, Stochastic finite element methods and reliability: a state-of-the-art report, Tech. Rep. UCB/SEMM-2000/08, Department of Civil & Environmental Engineering, University of California, Berkeley, 2000.

[32] G. Stefanou, The stochastic finite element method: past, present and future, Computer Methods in Applied Mechanics and Engineering 198 (2009) 1031–1051.

[33] D. Xiu, Numerical Methods for Stochastic Computations: A Spectral Method Approach, Princeton University Press, 2010.

[34] G. J. Lord, C. E. Powell, T. Shardlow, An Introduction to Computational Stochastic PDEs, Cambridge University Press, 2014.

[35] M. Aldosary, J. Wang, C. Li, Structural reliability and stochastic finite element methods, Engineering Computations 35 (2018) 2165–2214.

[36] F. Yamazaki, M. Shinozuka, G. Dasgupta, Neumann expansion for stochastic finite element analysis, Journal of Engineering Mechanics 114 (1988) 1335–1354.

[37] R. G. Ghanem, P. D. Spanos, Stochastic Finite Elements: A Spectral Approach, Springer, 1991.

[38] D. Xiu, G. E. Karniadakis, Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos, Computer Methods in Applied Mechanics and Engineering 191 (2002) 4927–4948.

[39] I. Babuska, R. Tempone, G. E. Zouraris, Galerkin finite element approximations of stochastic elliptic partial differential equations, SIAM Journal on Numerical Analysis 42 (2004) 800–825.

[40] D. Xiu, J. S. Hesthaven, High-order collocation methods for differential equations with random inputs, SIAM Journal on Scientific Computing 27 (2005) 1118–1139.

[41] J. T. Oden, R. Moser, O. Ghattas, Computer predictions with quantified uncertainty, part II, SIAM News 43 (2010) 1–4.

[42] A. Tarantola, Inverse Problem Theory and Methods for Model Parameter Estimation, SIAM, 2005.

[43] J. Kaipio, E. Somersalo, Statistical and Computational Inverse Problems, Springer, 2006.

[44] 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 Journal on Scientific Computing 35 (2013) A2494–A2523.

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

[46] A. Vigliotti, G. Csányi, V. S. Deshpande, Bayesian inference of the spatial distributions of material properties, Journal of the Mechanics and Physics of Solids 118 (2018) 74–97.

[47] H. Owhadi, Bayesian numerical homogenization, Multiscale Modeling & Simulation 13 (2015) 812–828.

[48] C. Zhu, R. H. Byrd, P. Lu, J. Nocedal, Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization, ACM Transactions on Mathematical Software (TOMS) 23 (1997) 550–560.

[49] A. Ern, J.-L. Guermond, Theory and Practice of Finite Elements, Springer, 2003.

[50] I. M. Sobol, Uniformly distributed sequences with an additional uniform property, USSR Computational Mathematics and Mathematical Physics 16 (1976) 236–242.

[51] M. A. Bessa, P. Glowacki, M. Houlder, Bayesian machine learning in metamaterial design: fragile becomes supercompressible, Advanced Materials 31 (1904845) (2019) 1–6.

[52] W. K. Liu, G. Besterfield, T. Belytschko, Transient probabilistic systems, Computer Methods in Applied Mechanics and Engineering 67 (1988) 27–54.

[53] M. C. Kennedy, A. O'Hagan, Predicting the output from a complex computer code when fast approximations are available, Biometrika 87 (2000) 1–13.

[54] P. Perdikaris, D. Venturi, J. O. Royset, G. E. Karniadakis, Multi-fidelity modelling via recursive co-kriging and Gaussian–Markov random

fields, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 471 (2015) 20150018–1–20150018–23.

[55] P. Perdikaris, M. Raissi, A. Damianou, N. D. Lawrence, G. E. Karniadakis, Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 473 (2017) 20160751–1–20160751–16.

[56] C. Duffin, E. Crips, T. Stemier, M. Girolami, Statistical finite elements for misspecified models, Preprint .

[57] C. Andrieu, J. Thoms, A tutorial on adaptive MCMC, Statistics and Computing 18 (2008) 343–373.