


A Surrogate Modelling Approach Based on Nonlinear Dimension Reduction for Uncertainty Quantification in Groundwater Flow Models

C. Gadd¹ · W. Xing¹ · M. Mousavi Nezhad¹ ·
A. A. Shah¹ 

Received: 13 September 2017 / Accepted: 13 April 2018 / Published online: 25 May 2018
© The Author(s) 2018

Abstract In this paper, we develop a surrogate modelling approach for capturing the output field (e.g. the pressure head) from groundwater flow models involving a stochastic input field (e.g. the hydraulic conductivity). We use a Karhunen–Loève expansion for a log-normally distributed input field and apply manifold learning (local tangent space alignment) to perform Gaussian process Bayesian inference using Hamiltonian Monte Carlo in an abstract feature space, yielding outputs for arbitrary unseen inputs. We also develop a framework for forward uncertainty quantification in such problems, including analytical approximations of the mean of the marginalized distribution (with respect to the inputs). To sample from the distribution, we present Monte Carlo approach. Two examples are presented to demonstrate the accuracy of our approach: a Darcy flow model with contaminant transport in 2-d and a Richards equation model in 3-d.

Keywords Groundwater flow models · Uncertainty quantification · Surrogate model · Karhunen–Loève expansion · Manifold learning

1 Introduction

Groundwater contamination, caused by landfills, wastewater seepage, hazardous chemical spillage, dumping of toxic substances or discharge from industrial processes (Karatzas 2017), is a major concern for both public and environmental health. Understanding the mechanisms and predicting the transport of contaminants through soils is therefore an important topic in groundwater flow modelling.

The control of groundwater quality relies on knowledge of the transport of chemicals to the groundwater through soil. The efficacy of remedial treatment and management of contaminated land depends on the accuracy of models used for the simulation of flow and

✉ A. A. Shah
Akeel.Shah@warwick.ac.uk

¹ School of Engineering, University of Warwick, Coventry CV47AL, UK

solute transport. Modelling and simulation of hydraulic phenomena in soil are, however, hampered by the complex and heterogeneous nature of soils, as well as the broad range of influential factors involved. A number of simplified models have been developed to describe the small-scale physical, chemical (Boi et al. 2009; Foo and Hameed 2009; Vomvoris and Gelhar 1990), and biological mechanisms (Schfer et al. 1998; Barry et al. 2002) that affect unsaturated flow and contaminant transport.

A current challenge in modelling solute transport in soils lies in characterizing and quantifying the uncertainties engendered by the natural heterogeneity of the soil. Such uncertainty can be vital for decision-making. Despite strong evidence from field-scale observations and experimental studies in relation to the effects of soil heterogeneity on the transport of contaminants (Al-Tabbaa et al. 2000; Kristensen et al. 2010), relatively few numerical models incorporate the effects of this uncertainty (Feyen et al. 1998; Aly and Peralta 1999; Sreekanth and Datta 2011a, 2014; Herckenrath et al. 2011).

Monte Carlo (MC) sampling is the default method for investigating uncertainties in a system (e.g. propagating uncertainty in the inputs), including in the context of groundwater flow modelling (Fu and Gomez-Hernandez 2009; Paleologos et al. 2006; Kourakos and Harter 2014; Maxwell et al. 2007; Herckenrath et al. 2011). MC estimates are extracted from multiple runs of the model using different realizations of the inputs, sampled from some distribution. While convergence is guaranteed as the number of runs increases, the slow rate of convergence demands (typically) a few thousand runs in order to extract reliable estimates of the statistics. If the model is computationally expensive, such a brute-force approach can be extremely time-consuming or perhaps even infeasible (Maxwell et al. 2007). Analytical stochastic methods have also been employed (Gelhar and Axness 1983; Gelhar 1986). Such methods can be useful for conceptual understanding of the transport process but are not applicable to practical scenarios.

Such limitations and shortcomings could be resolved in theory by using surrogate models (also known as metamodels, emulators or simply surrogates) in place of the complex numerical codes; that is, computationally efficient approximations of the codes based on data-driven or reduced-order model (ROM) approaches. Surrogate models have been used in a limited number of groundwater flow modelling problems (Aly and Peralta 1999; Bhattacharjya and Datta 2005; Kourakos and Mantoglou 2009; Sreekanth and Datta 2011b; Ataie-Ashtiani et al. 2014) (we refer to Razavi et al. 2012; Ketabchi and Ataie-Ashtiani 2015 for reviews of the topic) and are typically based on artificial neural networks (ANNs) for approximating a small number of outputs within an optimization task. For example, Bhattacharjya and Datta used an ANN to approximate the salt concentration in pumped water at 8 pumping wells for 3 different times, in order to maximize the total withdrawal of water from a coastal aquifer while limiting the salt concentration (Bhattacharjya and Datta 2005). Similarly, Kourakos and Mantoglou used an ANN model to optimize 34 well pumping rates in a coastal aquifer (Kourakos and Mantoglou 2009).

Another popular surrogate modelling approach is the stochastic collocation method (Babuška et al. 2007) in which the approximate response is constrained to a subspace, typically spanned by a generalized Polynomial Chaos basis (Xiu and Karniadakis 2002). The coefficients in this basis are approximated via a collocation scheme. While these schemes yield good convergence rates, they scale poorly with the number of collocation points (Rajabi et al. 2015). Although sparse grid methods based on the Smolyak algorithm (Smolyak 1963) help to alleviate the increased computational burden, the resulting schemes are still severely limited by the input space dimensionality and tend to perform poorly with limited observations (Xiu and Hesthaven 2005; Xiu 2007; Nobile et al. 2008; Ma and Zabaras 2009).

When data are scarce, we may turn to statistical Bayesian approaches such as Gaussian process (GP) regression. GPs are stochastic processes used for inferring nonlinear and latent functions. They are defined as a families of normally distributed random variables, indexed in this case by the input variable(s). GPs were first used for surrogate models in the seminal papers of Currin et al. (1988) and Sacks et al. (1989). The first applications of GP surrogate models to uncertainty quantification can be found in O'Hagan and Kingman (1978). Kernel methods such as GP models are well-established tools for analysing the relationships between input data and corresponding outputs of complex functions. Kernels encapsulate the properties of functions in a computationally efficient manner and provide flexibility in terms of model complexity (the functions used to approximate the target function) though variation of the functional form and parameters of the kernel.

GPs excel when data are scarce since they make a priori assumptions with regard to the relationship between data points. Comparatively, ANNs make fewer a priori assumptions and as a result require much larger data sets; they are, therefore, infrequently used for uncertainty quantification tasks. In the context of groundwater flow, very few applications of GPs can be found (Bau and Mayer 2006; Hemker et al. 2008; Borgonovo et al. 2012), the most likely explanations for which are the difficulty in implementing multioutput GP models and the lack of available information on, and software for GP modelling in comparison with ANNs. Existing applications again deal with low-dimensional outputs; e.g. in Bau and Mayer (2006), the authors use a GP model to learn 4 well extraction rates for a pump-and-treat optimization problem.

Our aim in this paper is to develop a surrogate model for the values of a field variable in a groundwater flow model, e.g. the pressure, pressure head or flow velocity, at a high number of points in the spatial domain, in order to propagate uncertainty in a stochastic field input, e.g. the hydraulic conductivity. In such cases, simplified covariance structures (Conti and O'Hagan 2010) for the output space (response surface) or dimensionality reduction for the input and/or output space can be used. In Higdon et al. (2008) Higdon et al. use principal component analysis (PCA) to perform linear, non-probabilistic dimensionality reduction on the response in order to render a GP model tractable (independent learning of a small number of PCA coefficients). Such linear approaches (PCA, multidimensional scaling, factor analysis) are applicable only when data lie in or near a linear subspace of the output space.

For more complex response surfaces, manifold learning (nonlinear dimensionality reduction) can be employed, using, for example, kernel principal component analysis (kPCA), diffusion maps (Xing et al. 2016 or isomaps Xing et al. 2015). In contrast, kPCA was used to perform nonlinear, non-probabilistic dimensionality reduction of the input space in Ma and Zabaras (2011). This can be useful when the input space is generated from observations (experimental data), but when the form is specified we can use linear dimension reduction methods such as the Karhunen–Loève (KL) expansion (Wong 1971).

In this paper we use manifold learning in the form of local tangent space alignment (LTSA) (Zhang and Zha 2004) to perform Bayesian inference (GP regression/emulation with Markov Chain Monte Carlo) in an abstract feature space and use an inverse (pre-image) map to obtain the output field at a finite number of points for an arbitrary input. In contrast to diffusion maps, isomaps and kPCA, LTSA is a local method in that it approximates points on the manifold on localized regions (patches), rather than directly seeking a global basis for the feature space. This can potentially provide more accurate results, although this is of course dependent upon the sampling methodology for the points and the quality of the reconstruction mapping.

The aforementioned approach is combined with a Karhunen–Loève expansion for a log-normally distributed input field and a framework for UQ is developed. We derive analytical forms for the output distribution by pushing the feature space Gaussian distribution through a

locally linear reconstruction map. Additionally, we derive analytical estimates of the moments of the predictive distribution via approximate marginalization of the stochastic input. To sample from the hyperparameter and signal precision posteriors, we employ a Hamiltonian Monte Carlo scheme and use MC sampling to approximately marginalize the stochastic input distribution. The accuracy of the approach is demonstrated via two examples: a linear, steady-state Darcy's Law with a contaminant mass balance in a 2-d domain (aquifer) and a time-dependent Richards equation evaluated at a fixed time in a 3-d domain. In both cases we consider a stochastic hydraulic conductivity input.

The rest of the paper is organized as follows. In Sect. 2 we provide a detailed problem statement and outline the proposed solution. In Sect. 3 we outline LTSA, and in Sect. 4 we outline GP regression. In Sect. 5 we provide full details of the coupling of the methods and we demonstrate how the approach can be used to perform UQ tasks. In Sect. 6 we present the examples and discuss the results.

2 Problem Statement

Consider a well-defined steady-state partial differential equation (PDE) with a scalar, isotropic random field input (e.g. a permeability or hydraulic conductivity), and a response (output) consisting of a scalar field, e.g. pressure head, concentration or flow velocity. We may generalize our approach to multiple or vector fields but in order to simplify the presentation we focus on a single scalar field. We can also apply the method we develop to dynamic problems by focusing on the spatial field at a given fixed time (the second example we present). For an arbitrary input field realization, solutions to the PDE are found using a numerical code (simulator, or solver) on a spatial mesh with k_y fixed degrees of freedom, e.g. grid points in a finite difference grid, control volume centres in a finite volume mesh or spatial nodes in a finite element mesh combined with a nodal basis.

We denote the input field by $K(\mathbf{x})$, where $\mathbf{x} \in \mathcal{R} \subset \mathbb{R}^d$, $d \in \{1, 2, 3\}$ denotes a spatial location and the notation makes explicit the spatial dependence. The model output (a scalar field) is denoted by $u(\mathbf{x}; K)$, i.e. it is a function of \mathbf{x} that is parameterized by $K(\mathbf{x})$. The random input $K(\mathbf{x})$ is defined on the whole of \mathcal{R} and therefore requires a discrete (finite-dimensional) approximation in order to obtain a numerical solution. Let $\mathbf{x}_k \in \mathcal{R}^d$, $k = 1, \dots, k_y$ be a set of nodes or grid points and suppose that the simulator yields discrete approximations $\{u_k = u(\mathbf{x}_k; K)\}_{k=1}^{k_y}$ of the output field $u(\mathbf{x}; K)$ in each run. Our goal is to approximate these simulator outputs for an arbitrary K .

2.1 Input Model: Karhunen–Loève Expansion

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, with sample space Ω , event space \mathcal{F} and probability measure \mathbb{P} . We can explicitly signify the randomness of the input by writing $K(\mathbf{x}, \omega)$, where $\omega \in \Omega$. For simplicity, and where it will not cause confusion, we suppress the dependence on ω (the same applies to other random processes). We assume that $K(\mathbf{x})$ is log-normal (to avoid unphysical, i.e. negative, realizations), so is of the form $K(\mathbf{x}, \omega) = \exp(Z(\mathbf{x}, \omega))$, where $Z(\mathbf{x}, \omega)$ is a normally distributed field (a GP¹ indexed by \mathbf{x}). For each $\mathbf{x} \in \mathcal{R}$, $Z(\mathbf{x}, \cdot) : \Omega \rightarrow \mathbb{R}$ is a random variable defined on the (common) probability space $(\Omega, \mathcal{F}, \mathbb{P})$. For a fixed $\omega \in \Omega$,

¹ Technically the process is a random field if the index (here \mathbf{x}) lies in \mathbb{R}^L where $L > 1$ but the convention in the great majority of the literature is to use the term Gaussian *process* even in such cases.

$Z(\cdot, \omega) : \mathcal{R} \rightarrow \mathbb{R}$ is a deterministic function of \mathbf{x} called a realization or sample path of the process. The mean and covariance functions of $Z(\mathbf{x}, \omega)$ are defined as:

$$\begin{aligned}
 m_Z(\mathbf{x}) &= \mathbb{E}[Z(\mathbf{x}, \omega)] = \int_{\Omega} Z(\mathbf{x}, \omega) d\mathbb{P}(\omega), \\
 c_Z(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(Z(\mathbf{x}, \omega) - m_Z(\mathbf{x}))(Z(\mathbf{x}', \omega) - m_Z(\mathbf{x}'))],
 \end{aligned}
 \tag{1}$$

respectively, in which $\mathbb{E}[\cdot]$ is the expectation operator. Given the covariance and mean functions for $Z(\mathbf{x}, \omega)$, the most widely used finite-dimensional approximation is based on a Karhunen–Loève (KL) expansion (Wong 1971). Assume that $Z(\mathbf{x}, \omega)$ is mean-square continuous ($\lim_{\delta\mathbf{x} \rightarrow \mathbf{0}} \mathbb{E}[(Z(\mathbf{x} + \delta\mathbf{x}, \omega) - Z(\mathbf{x}, \omega))^2] = 0$) and that $Z(\mathbf{x}, \omega) \in L^2(\mathcal{R} \times \Omega)$ ($\int_{\mathcal{R}} \mathbb{E}[|Z(\mathbf{x}, \omega)|^2] < \infty$), and is thus a second-order process. The KL theorem states that we may express $Z(\mathbf{x}, \omega)$ as a linear combination of deterministic $L^2(\mathcal{R})$ -orthonormal functions $w_j(\mathbf{x})$, with random $L^2(\Omega)$ -orthonormal coefficients $\xi_j(\omega)$:

$$Z(\mathbf{x}, \omega) = m_Z(\mathbf{x}) + \sum_{j=1}^{\infty} \sqrt{\lambda_j} \xi_j(\omega) w_j(\mathbf{x}),
 \tag{2}$$

where $\lambda_1 \geq \lambda_2 \geq \dots > 0$ and $\{w_j(\mathbf{x})\}_{j=1}^{\infty}$ are, respectively, the eigenvalues and eigenfunctions of an integral operator with kernel $c_Z(\mathbf{x}, \mathbf{x}')$:

$$\int_{\mathcal{R}} c_Z(\mathbf{x}, \mathbf{x}') w_j(\mathbf{x}') d\mathbf{x}' = \lambda_j w_j(\mathbf{x}).
 \tag{3}$$

The random coefficients are given by:

$$\xi_j(\omega) = \frac{1}{\sqrt{\lambda_j}} \int_{\mathcal{R}} (Z(\mathbf{x}', \omega) - m_Z(\mathbf{x}')) w_j(\mathbf{x}') d\mathbf{x}',
 \tag{4}$$

and are independent, standard normal ($\xi_j \sim \mathcal{N}(0, 1)$), with $\text{Var}(\sqrt{\lambda_j} \xi_j(\omega)) = \lambda_j$, where $\text{Var}(\cdot)$ denotes the variance operator. Here and throughout, $\mathcal{N}(\cdot, \cdot)$ denotes a normal distribution, in which the first argument is the mean (mean vector) and the second is the variance (covariance matrix).

The sum (2) can be truncated by virtue of the decay in the eigenvalues for increasing j . Discretizing the eigenvalue problem (3) using finite differencing at the nodes $\mathbf{x}_k \in \mathcal{R}$, $k = 1, \dots, k_y$, assuming that they are uniformly distributed, leads to an eigenvalue problem for the covariance matrix $\mathbf{C} = [c_Z(\mathbf{x}_k, \mathbf{x}_j)]_{k,j=1}^{k_y}$:

$$\mathbf{C} \mathbf{w}_j = \lambda_j \mathbf{w}_j,
 \tag{5}$$

where the k th component $w_{j,k}$ of $\mathbf{w}_j \in \mathbb{R}^{k_y}$, $j = 1, \dots, k_y$, is equivalent to the evaluation of eigenfunction w_j at the node \mathbf{x}_k , $k = 1, \dots, k_y$. Defining the random vector $\mathbf{Z} := (Z(\mathbf{x}_1), \dots, Z(\mathbf{x}_{k_y}))^T : \Omega \rightarrow \mathbb{R}^{k_y}$, we can write:

$$\mathbf{Z} = \mathbf{m}_Z + \sum_{j=1}^{k_y} \sqrt{\lambda_j} \xi_j(\omega) \mathbf{w}_j,
 \tag{6}$$

where $\mathbf{m}_Z = (m_Z(\mathbf{x}_1), \dots, m_Z(\mathbf{x}_{k_y}))^T$ and $\xi_j \sim \mathcal{N}(0, 1)$ are independent random variables (note that we have kept the notation ξ_j and λ_j used in the continuous case in order to avoid notational clutter). This provides discrete realizations of $Z(\mathbf{x}, \omega)$, and the expansion in (6) can be truncated by virtue of the decay in λ_j for some $k_{\xi} < k_y$, chosen so that the generalized

variance satisfies $\sum_{j=1}^{k_\xi} \sqrt{\lambda_j} / \sum_{j=1}^{k_y} \sqrt{\lambda_j} > \vartheta$ for some specified tolerance $0 < \vartheta < 1$. We can then obtain discrete realizations $\mathbf{K} = (K_1, \dots, K_{k_y})^T$ of $K(\mathbf{x}, \omega)$ via:

$$K_k = K(\mathbf{x}_k, \omega) = \exp \left(m_Z(\mathbf{x}_k) + \sum_{j=1}^{k_\xi} \sqrt{\lambda_j} \xi_j(\omega) w_{j,k} \right). \tag{7}$$

The discrete input \mathbf{K} can then be replaced by the random vector defined by $\boldsymbol{\xi} = (\xi_1, \dots, \xi_{k_\xi})^T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the coefficients of which are independent standard normal. We may then write $u(\mathbf{x}_k; \boldsymbol{\xi})$ for the KL approximation to $u(\mathbf{x}_k; K)$, at the nodes $\{\mathbf{x}_k\}_{k=1}^{k_y}$.

We note that different methods, including different quadrature rules or the use of projection schemes and Nystrom methods (Wan and Karniadakis 2006) can be used to solve the eigenvalue problem (3), all of which lead to a generalized eigenvalue problem in place of (5) (Betz et al. 2014). For example, if the finite element method is used, we may express the eigenfunctions as $w_j(\mathbf{x}) = \sum_k l_{j,k} \psi_k$ in terms of the finite element basis $\{\psi_k\}_{k=1}^{k_y}$ and perform a Galerkin projection of (3) onto $\text{span}(\psi_1, \dots, \psi_{k_y})$ to yield a generalized eigenvalue problem for $\{\lambda_j\}_{j=1}^{k_y}$ and the undetermined coefficients $\{l_{j,k}\}_{j,k=1}^{k_y}$ (Ghanem and Spanos 2003).

2.2 Statement of the Surrogate Model Problem

The simulator can now be considered as a mapping $\eta : \mathcal{X} \rightarrow \mathcal{Y}$ (assumed to be continuous and injective), where $\boldsymbol{\xi} \in \mathcal{X} \subset \mathbb{R}^{k_\xi}$ is the permissible *input space* and $\mathbf{y} \in \mathcal{Y} \subset \mathbb{R}^{k_y}$ is the permissible *output space* or *response surface* consisting of the discrete field:

$$\mathbf{y} = \eta(\boldsymbol{\xi}) := (u(\mathbf{x}_1; \boldsymbol{\xi}), \dots, u(\mathbf{x}_{k_y}; \boldsymbol{\xi}))^T. \tag{8}$$

Our aim is to develop a surrogate to make fast, online predictions of $\eta(\boldsymbol{\xi})$, using *training data* from a limited number of solver runs at the *design points* $\boldsymbol{\xi}_n, n = 1, \dots, N$. The training data can be expressed compactly as a matrix $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N]^T \in \mathbb{R}^{N \times k_y}$ and we can define $\boldsymbol{\Xi} = [\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_N]^T \in \mathbb{R}^{N \times k_\xi}$. The data set is thus $\mathcal{D}' = \{\boldsymbol{\Xi}, \mathbf{Y}\}$.

The high dimensionalities of the (original) input and output spaces pose great challenges for surrogate model development. The input space dimensionality can be reduced as described above. The intrinsic dimensionality of the output space is significantly lower than k_y by virtue of correlations between outputs for different inputs, as well as physical constraints imposed by the simulator. This suggests that we treat \mathcal{Y} as a manifold and use manifold learning/dimensionality reduction to perform Bayesian inference on a low-dimensional (feature) space \mathcal{F} that is locally homeomorphic to \mathcal{Y} . Below we introduce the manifold learning method employed, before recasting the emulation problem as one of inference in the feature space, together with a pre-image (inverse) mapping to obtain solutions in \mathcal{Y} for arbitrary inputs $\boldsymbol{\xi}$.

3 Dimensionality Reduction and Manifold Learning: Feature Space Representations

Roughly speaking, a k_z -dimensional manifold \mathcal{Y} is a set for which all points can be parameterized by k_z independent variables. A parameterization is called a coordinate system (or a chart) and it is not necessarily the case that a single coordinate system can describe the entire

manifold. To characterize the manifold in such cases, we can introduce overlapping patches, each with its own system of (non-unique) coordinates.

Formally speaking, a smooth k_z -manifold is defined as a topological space \mathcal{Y} that is equipped with a maximal open cover $\{U_\alpha\}_{\alpha \in \Gamma}$ consisting of coordinate neighbourhoods (or patches) U_α , together with a collection of homeomorphisms (coordinate charts) $\phi_\alpha : U_\alpha \rightarrow \phi_\alpha(U_\alpha) \subset \mathbb{R}^{k_z}$ onto open subsets $\phi_\alpha(U_\alpha) \subset \mathbb{R}^{k_z}$ such that $\phi_\alpha(U_\alpha \cap U_\beta)$ and $\phi_\beta(U_\alpha \cap U_\beta)$ are open in \mathbb{R}^{k_z} ; we say that ϕ_α and ϕ_β are compatible. Moreover, the transition maps defining a change of coordinates $\phi_\beta \circ \phi_\alpha^{-1}$ are diffeomorphisms for all $\alpha, \beta \in \Gamma$.

Let $\mathcal{A} = \{(U_\alpha, \phi_\alpha)\}_{\alpha \in \Gamma}$ be an atlas on \mathcal{Y} ($\{U_\alpha\}_{\alpha \in \Gamma}$ is a cover and the $\{\phi_\alpha\}_{\alpha \in \Gamma}$ are pairwise compatible). Two smooth curves $\gamma_0, \gamma_1 : \mathbb{R} \rightarrow \mathcal{Y}$ are called **y**-equivalent at a point $\mathbf{y} \in \mathcal{Y}$ if for every $\alpha \in \Gamma$ with $\mathbf{y} \in U_\alpha$, we have $\gamma_0(0) = \gamma_1(0) = \mathbf{y}$ and furthermore $(d/dt)|_{t=0} \phi_\alpha(\gamma_0(t)) = (d/dt)|_{t=0} \phi_\alpha(\gamma_1(t))$. With this equivalence relation, the equivalence class of a smooth curve γ with $\gamma(0) = \mathbf{y}$ is denoted $[\gamma]_{\mathbf{y}}$ and the *tangent space* $T_{\mathbf{y}}\mathcal{Y}$ of \mathcal{Y} at \mathbf{y} is the set of equivalence classes $\{[\gamma]_{\mathbf{y}} : \gamma(0) = \mathbf{y}\}$. The tangent space is a k_z -dimensional vector space, which is seen more clearly by identifying $T_{\mathbf{y}}\mathcal{Y}$ with the set of all derivations at \mathbf{y} [linear maps from $C^\infty(\mathcal{Y})$ to \mathbb{R} satisfying the derivation (Liebnitz) property].

We assume that the output space $\mathcal{Y} \supset \mathbf{Y}$ is a manifold of dimension $k_z \ll k_y$ embedded in \mathbb{R}^{k_y} . Representations of points in \mathcal{Y} and corresponding representations in the feature or latent space $\mathcal{F} \subset \mathbb{R}^{k_z}$ can be related by some smooth and *unknown* function $\mathbf{f} : \mathcal{F} \rightarrow \mathcal{Y}$. *Manifold learning* is concerned with the reconstruction of \mathbf{f} and its inverse given data points on the manifold, whereas *dimensionality reduction* is concerned with the representation of given points in \mathcal{Y} by corresponding points in the feature space \mathcal{F} . Here we are interested primarily in dimensionality reduction and use *Local Tangent Space Alignment* (LTSA) (Zhang and Zha 2004). The tangent space at a point \mathbf{y} provides a low-dimensional linear approximation of points in a neighbourhood of \mathbf{y} . We can approximate each point \mathbf{y} in a data set using a basis for $T_{\mathbf{y}}\mathcal{Y}$ and use these approximations to find low-dimensional representations in a global coordinate system, by aligning the tangent spaces using local affine transformations (Zhang and Zha 2004). We note that this assumes the existence of a single chart (homeomorphism) \mathbf{f}^{-1} .

Consider a noise-free model in which the data \mathbf{Y} are generated by the smooth function \mathbf{f} defined above:

$$\mathbf{y} = \mathbf{f}(\mathbf{z}) = (f_1(\mathbf{z}), \dots, f_{k_y}(\mathbf{z}))^T, \tag{9}$$

where $\mathbf{z} = (z_1, \dots, z_{k_z})^T \in \mathcal{F}$ is a latent/feature vector (i.e. the low-dimensional representation of the point \mathbf{y}). Under the assumption that \mathbf{f} is smooth, it can be approximated using a first-order Taylor expansion in a neighbourhood $\Omega(\mathbf{z})$ of a point \mathbf{z} : $\mathbf{f}(\widehat{\mathbf{z}}) = \mathbf{f}(\mathbf{z}) + \mathbf{J}_{\mathbf{f}}(\mathbf{z}) \cdot (\widehat{\mathbf{z}} - \mathbf{z}) + \mathcal{O}(\|\widehat{\mathbf{z}} - \mathbf{z}\|^2)$, $\forall \widehat{\mathbf{z}} \in \Omega(\mathbf{z})$, where $\mathbf{J}_{\mathbf{f}}(\mathbf{z}) \in \mathbb{R}^{k_y \times k_z}$ is the Jacobi matrix of \mathbf{f} at \mathbf{z} , the i, j th entry of which is $\partial f_i / \partial z_j$. Here and throughout, $\|\cdot\|$ denotes a standard Euclidean norm.

A basis for the tangent space $T_{\mathbf{y}}\mathcal{Y}$ of \mathcal{Y} (a k_z -dimensional linear subspace of \mathbb{R}^{k_y}) at $\mathbf{y} = \mathbf{f}(\mathbf{z})$ is given by the span of the column vectors of $\mathbf{J}_{\mathbf{f}}$. The vector $\widehat{\mathbf{z}} - \mathbf{z}$ then gives the coordinate of $\mathbf{f}(\widehat{\mathbf{z}})$ in the affine subspace $\mathbf{f}(\mathbf{z}) + T_{\mathbf{y}}\mathcal{Y}$. $\mathbf{J}_{\mathbf{f}}$ cannot be computed explicitly without knowledge of \mathbf{f} . Suppose we can express $T_{\mathbf{y}}\mathcal{Y}$ in terms of a matrix $\mathbf{Q}_{\mathbf{z}}$, the columns of which form an orthonormal basis for $T_{\mathbf{y}}\mathcal{Y}$:

$$\mathbf{J}_{\mathbf{f}}(\mathbf{z}) \cdot (\widehat{\mathbf{z}} - \mathbf{z}) = \mathbf{Q}_{\mathbf{z}} \boldsymbol{\pi}_{\mathbf{z}}^*, \tag{10}$$

where $\boldsymbol{\pi}_z^* = \mathbf{Q}_z^T \mathbf{J}_f(\mathbf{z}) \cdot (\widehat{\mathbf{z}} - \mathbf{z}) \equiv \mathbf{P}_z (\widehat{\mathbf{z}} - \mathbf{z})$ is still unknown. Combining Eq. (10) with the Taylor expansion, we can, however, find an approximation of $\boldsymbol{\pi}_z^*$ consisting of an orthogonal projection of $\mathbf{f}(\widehat{\mathbf{z}}) - \mathbf{f}(\mathbf{z})$ onto $T_y\mathcal{Y}$:

$$\boldsymbol{\pi}_z \equiv \mathbf{Q}_z^T (\mathbf{f}(\widehat{\mathbf{z}}) - \mathbf{f}(\mathbf{z})) = \boldsymbol{\pi}_z^* + \mathcal{O}(\|\widehat{\mathbf{z}} - \mathbf{z}\|^2), \tag{11}$$

provided that the basis \mathbf{Q}_z is known for each \mathbf{z} . Truncating this expansion, the global coordinate \mathbf{z} then satisfies:

$$\int \int_{\Omega(\mathbf{z})} \|\mathbf{P}_z (\widehat{\mathbf{z}} - \mathbf{z}) - \boldsymbol{\pi}_z\| d\widehat{\mathbf{z}} \approx 0. \tag{12}$$

If the Jacobian is of full column rank, we can find a local affine transformation:

$$\widehat{\mathbf{z}} - \mathbf{z} \approx \mathbf{P}_z^{-1} \boldsymbol{\pi}_z \equiv \mathbf{L}_z \boldsymbol{\pi}_z. \tag{13}$$

The transformation \mathbf{L}_z aligns the local coordinate with the global coordinate $\widehat{\mathbf{z}} - \mathbf{z}$ for $f(\widehat{\mathbf{z}})$. We then find the global coordinate $\widehat{\mathbf{z}}$ and affine transformation \mathbf{L}_z by minimizing $\int \int_{\Omega(\mathbf{z})} \|\widehat{\mathbf{z}} - \mathbf{z} - \mathbf{L}_z \boldsymbol{\pi}_z\| d\widehat{\mathbf{z}}$.

We note that the orthogonal basis \mathbf{Q}_z for each tangent space is still unknown. Consider a data set $\mathbf{y}_n, n = 1, \dots, N$, sampled with noise $\epsilon_n, n = 1, \dots, N$, from the underlying nonlinear manifold:

$$\mathbf{y}_n = \mathbf{f}(\mathbf{z}_n) + \epsilon_n. \tag{14}$$

For any \mathbf{y}_n , let $\mathbf{Y}_n = [\mathbf{y}_{n_1} \dots \mathbf{y}_{n_P}]$ be the matrix containing the P nearest neighbours, including \mathbf{y}_n , where distances are measured using the standard Euclidean metric. The best k_z -dimensional local affine subspace approximation for the points in \mathbf{Y}_n is given by:

$$\arg \min_{\mathbf{y}, \mathbf{\Pi}, \mathbf{Q}} \sum_{k=1}^P \|\mathbf{y}_{n_k} - (\mathbf{y} + \mathbf{Q}\boldsymbol{\pi}_k)\|_2^2 = \arg \min_{\mathbf{y}, \mathbf{\Pi}, \mathbf{Q}} \|\mathbf{Y}_n - (\mathbf{y}\mathbf{e}^T + \mathbf{Q}\mathbf{\Pi})\|_2^2, \tag{15}$$

where the orthonormal matrix \mathbf{Q} has k_z columns, $\mathbf{\Pi} = [\boldsymbol{\pi}_1 \dots \boldsymbol{\pi}_P]$ and \mathbf{e} is a vector of all ones. The optimal \mathbf{y} is given by the mean of $\{\mathbf{y}_{n_k}\}_k$, denoted $\bar{\mathbf{y}}_n$, and the optimal \mathbf{Q} is given by \mathbf{Q}_n , the columns of which are the k_z left singular vectors of $\mathbf{Y}_n (\mathbf{I} - \mathbf{e}\mathbf{e}^T/P)$ corresponding to the k_z largest singular values. Lastly, $\mathbf{\Pi}$ is given by $\mathbf{\Pi}_n$:

$$\mathbf{\Pi}_n = \mathbf{Q}_n^T \mathbf{Y}_n \left(\mathbf{I} - \frac{1}{P} \mathbf{e}\mathbf{e}^T \right) = [\boldsymbol{\pi}_1^{(i)}, \dots, \boldsymbol{\pi}_K^{(i)}], \tag{16}$$

where $\boldsymbol{\pi}_k^{(i)} = \mathbf{Q}_n^T (\mathbf{y}_{n_k} - \bar{\mathbf{y}}_n)$. Consequently:

$$\mathbf{y}_{n_k} = \bar{\mathbf{y}}_n + \mathbf{Q}_n \boldsymbol{\pi}_k^{(l)} + \varphi_k^{(l)}, \tag{17}$$

where $\varphi_k^{(l)} = (\mathbf{I} - \mathbf{Q}_n \mathbf{Q}_n^T) (\mathbf{y}_{n_k} - \bar{\mathbf{y}}_n)$ is the reconstruction error. Having minimized the local reconstruction error, we would like to find the global coordinates $\mathbf{Z} = [\mathbf{z}_1 \dots \mathbf{z}_N] \in \mathbb{R}^{k_z \times N}$, corresponding to data points \mathbf{Y} , given the local coordinates $\boldsymbol{\pi}_k^{(l)}$. The global coordinates \mathbf{z}_{n_k} of the corresponding points \mathbf{y}_{n_k} are chosen to respect the local geometry as determined by the $\boldsymbol{\pi}_k^{(l)}$:

$$\begin{aligned} \mathbf{z}_{n_k} &= \bar{\mathbf{z}}_n + \mathbf{L}_n \boldsymbol{\pi}_k^{(l)} + \epsilon_k^{(l)}, \quad k = 1, \dots, P, \quad l = 1, \dots, N, \\ \mathbf{Z}_n &= \frac{1}{P} \mathbf{Z}_n \mathbf{e}\mathbf{e}^T + \mathbf{L}_n \mathbf{\Pi}_n + \mathbf{E}_n, \end{aligned} \tag{18}$$

where $\bar{\mathbf{z}}_n$ is the mean of $\{\mathbf{z}_{nk}\}_k$, $\mathbf{Z}_n = [\mathbf{z}_{n1} \dots \mathbf{z}_{np}]$ and $\mathbf{E}_n = [\epsilon_1^{(l)} \dots \epsilon_p^{(l)}]$, given by $\mathbf{E}_n = \mathbf{Z}_n(\mathbf{I} - \mathbf{e}\mathbf{e}^T/P) - \mathbf{L}_n\mathbf{\Pi}_n$. We find the latent points and local affine transformations \mathbf{L}_n that minimize the local reconstruction error $\|\mathbf{E}_n\|_F$, in which $\|\cdot\|_F$ denotes a Frobenius norm. The optimal \mathbf{L}_n are given by $\mathbf{L}_n = \mathbf{Z}_n(\mathbf{I} - \mathbf{e}\mathbf{e}^T/P)\mathbf{\Pi}_n^+$, and consequently the errors are given by $\mathbf{E}_n = \mathbf{Z}_n(\mathbf{I} - \mathbf{e}\mathbf{e}^T/P)(\mathbf{I} - \mathbf{\Pi}_n^+\mathbf{\Pi}_n)$, where $\mathbf{\Pi}_n^+$ is the Moor–Penrose pseudo-inverse of $\mathbf{\Pi}_n$. We define a 0-1 selection matrix $\mathbf{S}_n \in \mathbb{R}^{N \times P}$ such that $\mathbf{Z}\mathbf{S}_n = \mathbf{Z}_n$. The global coordinates can then be selected according to a minimization of the overall reconstruction error:

$$\arg \min_{\mathbf{Z}:\mathbf{Z}^T\mathbf{Z}=\mathbf{I}} \sum_n \|\mathbf{E}_n\|_F^2 = \arg \min_{\mathbf{Z}:\mathbf{Z}^T\mathbf{Z}=\mathbf{I}} \|\mathbf{Z}\mathbf{S}\mathbf{W}\|_F^2, \tag{19}$$

where $\mathbf{S} = [\mathbf{S}_1 \dots \mathbf{S}_N]$, and $\mathbf{W} = \text{diag}(\mathbf{W}_1, \dots, \mathbf{W}_N)$, in which $\mathbf{W}_n = (\mathbf{I} - \mathbf{e}\mathbf{e}^T/P)(\mathbf{I} - \mathbf{\Pi}_n^+\mathbf{\Pi}_n)$. The constraint $\mathbf{Z}^T\mathbf{Z} = \mathbf{I}$ ensures that the solutions are unique. The vector \mathbf{e} is an eigenvector of $\mathbf{B} \equiv \mathbf{S}\mathbf{W}\mathbf{W}^T\mathbf{S}^T \in \mathbb{R}^{N \times N}$ corresponding to a zero eigenvalue. Arranging the eigenvalues in increasing order, the optimal \mathbf{Z} is given by $\mathbf{Z}' = [\zeta_2 \dots \zeta_{k_z+1}]^T$, where $\zeta_2, \dots, \zeta_{k_z+1} \in \mathbb{R}^N$ are the eigenvectors of \mathbf{B} corresponding to the $(k_z + 1)^{\text{st}}$ smallest eigenvalues excluding the first (zero) eigenvalue. This defines a map $\mathbf{f}^- : \mathbf{y} \mapsto \mathbf{z}$, $\mathbf{z} = \mathbf{f}^-(\mathbf{y})$ that approximates $\mathbf{f}^{-1} : \mathcal{Y} \rightarrow \mathcal{F}$ for the given data points:

$$\mathbf{z}_n = \mathbf{f}^{-1}(\mathbf{y}_n) \approx \mathbf{f}^-(\mathbf{y}_n) = \mathbf{z}'_{n,\dots} \tag{20}$$

where $\mathbf{z}'_{n,\dots}$ is the n th column of \mathbf{Z}' .

Fixing the number of neighbours assumes that the manifold has a certain smoothness, while using the same number of neighbours for every tangent space assumes a global smoothness. These assumptions may result in inaccurate predictions, in which case we can use adaptive algorithms (Zou and Zhu 2011; Zhang et al. 2012; Wei et al. 2008). Similar adaptations can be made for other issues, such as robustness in the presence of noise (Zhan and Yin 2011).

We remark that LTSA is a nonparametric technique, in that an explicit form of \mathbf{f} is not available. This means that the *out-of-sample* problem does not have a parametric (explicit) solution. In other words, application of LTSA (the map \mathbf{f}^-) to a point that was not in the data set can only be achieved by rerunning the entire algorithm with an updated data set that appends the new point. Nonparametric solutions to the out-of-sample problem have been developed, and one that is applicable to LTSA can be found in Li et al. (2005).

If we map points $\mathbf{y} \in \mathcal{Y}$ to \mathcal{F} using \mathbf{f}^- and perform inference in \mathcal{F} , an approximation of \mathbf{f} is required in order to make predictions in the physical space \mathcal{Y} . This is referred to as the *pre-image* problem in manifold learning methods: given a point in the low-dimensional space, find a mapping to the original space (manifold). We outline an approximation of the pre-image map in the next section.

3.1 Pre-image Problem: Reconstruction of Points in the Manifold \mathcal{Y}

Given a point $\mathbf{z} \in \mathcal{F}$ in latent space, we require the corresponding point in the original physical space $\mathbf{y} \in \mathcal{Y}$. Let \mathbf{z}_k be the neighbour nearest to \mathbf{z} . According to Eq. 18:

$$\boldsymbol{\pi}_*^{(k)} = \mathbf{L}_k^{-1}(\mathbf{z} - \bar{\mathbf{z}}_k) - \mathbf{L}_k^{-1}\boldsymbol{\epsilon}_*^{(k)}. \tag{21}$$

By Eq. 17 we can also define:

$$\mathbf{y} = \bar{\mathbf{y}}_k + \mathbf{Q}_k\boldsymbol{\pi}_*^{(k)} + \varphi_*^{(k)}. \tag{22}$$

Consequently, we have the following approximate pre-image mapping $\hat{\mathbf{f}} : \mathcal{F} \rightarrow \mathcal{Y}$ (approximation of \mathbf{f}):

$$\begin{aligned} \mathbf{y} = \mathbf{f}(\mathbf{z}) &\approx \hat{\mathbf{f}}(\mathbf{z}) = \bar{\mathbf{y}}_k + \mathbf{Q}_k \left(\mathbf{L}_k^{-1} (\mathbf{z} - \bar{\mathbf{z}}_k) - \mathbf{L}_k^{-1} \epsilon_*^{(k)} \right) + \varphi_*^{(k)} \\ &= \bar{\mathbf{y}}_k + \mathbf{Q}_k \mathbf{L}_k^{-1} (\mathbf{z} - \bar{\mathbf{z}}_k) + \mathcal{E}, \end{aligned} \tag{23}$$

where $k = \arg \min_n \|\mathbf{z} - \mathbf{z}_n\|$ and $\mathcal{E} = -\mathbf{Q}_k \mathbf{L}_k^{-1} \epsilon_*^{(k)} + \varphi_*^{(k)}$ incorporates the error terms.

4 Gaussian Process Emulation in Feature Space

In Sect. 2.2, the surrogate model problem was defined as one of approximating the simulator mapping $\eta : \mathcal{X} \rightarrow \mathcal{Y}$ given the data set $\mathcal{D}' = \{\mathbf{\Xi}, \mathbf{Y}\}$ derived from runs of the simulator at selected design points $\{\xi_n\}_{n=1}^N$. We can instead consider the simulator as a mapping $\eta_{\mathcal{F}} \equiv \mathbf{f}^{-1} \circ \eta : \mathcal{X} \rightarrow \mathcal{F}$ from the input space to the feature space, i.e. $\eta_{\mathcal{F}}(\cdot) = \mathbf{f}^{-1}(\eta(\cdot))$. Application of LTSA to points on the manifold approximates this mapping with $\mathbf{f}^- \approx \mathbf{f}^{-1}$. The original data set $\mathcal{D}' = \{\mathbf{\Xi}, \mathbf{Y}\}$ is replaced by the equivalent data set $\mathcal{D} = \{\mathbf{\Xi}, \mathbf{Z}\}$ or $\mathcal{D} = \{(\xi_n, \mathbf{z}_n)\}_{n=1}^N$, where $\mathbf{z}_n = \mathbf{f}^-(\mathbf{y}_n) \approx \mathbf{f}^{-1}(\mathbf{y}_n) = \mathbf{f}^{-1}(\eta(\xi_n)) = \eta_{\mathcal{F}}(\xi_n)$, and our aim is now to approximate the mapping $\eta_{\mathcal{F}}(\cdot)$. Returning a general point $\mathbf{z} = \eta_{\mathcal{F}}(\xi)$ to the corresponding point \mathbf{y} in the space \mathcal{Y} is discussed in the next section.

In this work, a GP model is used to infer the mapping $\eta_{\mathcal{F}} : \xi \mapsto \mathbf{z}$ by treating it as a realization of a (Gaussian) stochastic process indexed by the inputs ξ . Specifically, we learn each component of \mathbf{z} separately (assuming independence) using a *scalar* GP model. Here and throughout, $\mathcal{GP}(\cdot, \cdot)$ denotes a GP, in which the first argument is the mean function and the second is the covariance (kernel) function.

Let $z_{n,i}, i = 1, \dots, k_z$, denote the i th component of $\mathbf{z}_n, n = 1, \dots, N$. The probabilistic model is $z_{n,i} = h_i(\xi_n) + \eta_{n,i}$, in which the signal noise distribution is $p(\eta_{n,i}) = \mathcal{N}(0, \beta_i^{-1}) \forall n$, where β_i is the precision. We assume independent GP priors $h_i(\xi) \sim \mathcal{GP}(0, c_h(\xi, \xi'; \theta_i))$, where $c_h(\xi, \xi'; \theta_i)$ is the kernel function (of the same form across i) in which θ_i is a vector of hyperparameters pertaining to component i . The latent functions $h_i(\xi), i = 1, \dots, k_z$, can be thought of as independent draws from the GP. Using the notation $h_{n,i} = h_i(\xi_n)$ we can define a matrix $\mathbf{H} \in \mathbb{R}^{N \times k_z}$ with columns $\mathbf{h}_{:,i} = (h_{1,i}, \dots, h_{N,i})^T$ ($\mathbf{z}_{:,i}$ similarly defines the vector of the i th features across all samples). By the independence assumption:

$$p(\mathbf{H} | \mathbf{\Xi}, \Theta) = \prod_{i=1}^{k_z} p(\mathbf{h}_{:,i} | \mathbf{\Xi}, \theta_i), \tag{24}$$

where $\Theta = [\theta_1 \dots \theta_{k_z}]$. By the properties of GPs, we have $p(\mathbf{h}_{:,i} | \mathbf{\Xi}, \theta_i) = \mathcal{N}(\mathbf{0}, \mathbf{C}_i)$, where $\mathbf{C}_i \in \mathbb{R}^{N \times N}$ is a kernel matrix, the n, m th entry of which is $c_h(\xi_n, \xi_m; \theta_i)$. Thus:

$$\begin{aligned} p(\mathbf{Z} | \mathbf{\Xi}, \Theta, \beta) &= \int \prod_{i=1}^{k_z} \prod_{n=1}^N p(z_{n,i} | h_{n,i}, \beta_i) p(\mathbf{h}_{:,i} | \mathbf{\Xi}, \theta_i) d\mathbf{H} \\ &= \prod_{i=1}^{k_z} \mathcal{N}(\mathbf{0}, \mathbf{C}_i + \beta_i^{-1} \mathbf{I}), \end{aligned} \tag{25}$$

where $p(\mathbf{z}_{:,i} | \mathbf{h}_{:,i}, \beta_i) = \mathcal{N}(\mathbf{h}_{:,i}, \beta_i^{-1} \mathbf{I})$ by virtue of the noise model, and $\beta = (\beta_1, \dots, \beta_{k_z})^T$.

We place gamma priors on all hyperparameters θ_i and signal noise precisions β_i . The parameterization of these priors is determined through an initial maximum likelihood run. We choose these parameters such that the mean is equal to the maximum likelihood value, and so that we obtain an appropriate variance. Let $\mathbf{z} \in \mathcal{F}$ be the feature vector corresponding to the test (new) input ξ . The predictive distribution for the i th component z_i of \mathbf{z} ($i = 1, \dots, k_z$) is given by:

$$\begin{aligned}
 p(z_i | \xi, \mathcal{D}, \theta_i, \beta_i) &= \mathcal{N}(\mu_i(\xi), \sigma_i^2(\xi)), \\
 \mu_i(\xi) &= \mathbf{c}_h(\xi, \Xi; \theta_i)^T \left(\mathbf{C}_i + \beta_i^{-1} \mathbf{I} \right)^{-1} \mathbf{z}_{:,i}, \\
 \sigma_i^2(\xi) &= c_h(\xi, \xi; \theta_i) - \mathbf{c}_h(\xi, \Xi; \theta_i)^T \left(\mathbf{C}_i + \beta_i^{-1} \mathbf{I} \right)^{-1} \mathbf{c}_h(\xi, \Xi; \theta_i),
 \end{aligned}
 \tag{26}$$

where $\mathbf{c}_h(\xi, \Xi; \theta_i) = (c_h(\xi_1, \xi; \theta_i), \dots, c_h(\xi_N, \xi; \theta_i))^T \in \mathbb{R}^N$ is the cross-covariance between \mathbf{z} and \mathbf{z}_n , $n = 1, \dots, N$. Thus, the latent variable GP prediction is distributed as:

$$\begin{aligned}
 p(\mathbf{z} | \xi, \mathcal{D}, \Theta, \beta) &= \mathcal{N}(\boldsymbol{\mu}_z(\xi), \Sigma_z(\xi)), \\
 \boldsymbol{\mu}_z(\xi) &= (\mu_1(\xi), \dots, \mu_{k_z}(\xi))^T, \\
 \Sigma_z(\xi) &= \text{diag}(\sigma_1^2(\xi), \dots, \sigma_{k_z}^2(\xi)),
 \end{aligned}
 \tag{27}$$

where the components of $\boldsymbol{\mu}_z(\xi) \in \mathcal{F}$ are given by the second of Eqs. (26) and $\Sigma_z(\xi) \in \mathbb{R}^{k_z \times k_z}$ is a diagonal covariance matrix, in which the i th diagonal element corresponds to the predictive variance given by the third of Eqs. (26), while the off-diagonal elements are zero due to the assumption of independent GPs across i .

4.1 Sampling Hyperparameter Posterior with Hybrid Monte Carlo

We explore the hyperparameter posterior distributions using a hybrid Monte Carlo (HMC) scheme. HMC is a Metropolis method that uses gradient information. It exploits Hamiltonian dynamics to explore state spaces based on the likelihood probability, and consequently limits the random walk behaviour. The Hamiltonian is defined as an energy function in terms of a position vector $\mathbf{q}(t)$ and a momentum vector $\mathbf{p}(t)$ at time t (unrelated to the time component in the solver): $H(\mathbf{q}(t), \mathbf{p}(t)) = E_U(\mathbf{q}(t)) + E_K(\mathbf{p}(t))$, where $E_U(\mathbf{q})$ is the potential energy and $E_K(\mathbf{p})$ is the kinetic energy, the sum of which is constant. The evolution of this system is then defined by the partial derivatives of the Hamiltonian:

$$\frac{d\mathbf{p}}{dt} = -\frac{\partial H}{\partial \mathbf{q}}, \quad \frac{d\mathbf{q}}{dt} = +\frac{\partial H}{\partial \mathbf{p}}.
 \tag{28}$$

We define the potential energy as the negative log likelihood with an additive constant C , chosen for convenience: $E_U(\mathbf{q}(t)) = -\log(\text{likelihood}(\mathbf{q}(t))) - \log(\text{prior}(\mathbf{q}(t)))$. Furthermore, following convention we define the kinetic energy as:

$$E_K(\mathbf{p}(t)) = \frac{1}{2} \mathbf{p}(t) \mathbf{M}_K^{-1} \mathbf{p}(t),
 \tag{29}$$

where \mathbf{M}_K is a symmetric, positive definite mass matrix, chosen to be a scalar multiple of the identity matrix. With this choice, the potential energy is the negative log probability density of a multivariate Gaussian distribution with covariance \mathbf{M}_K and matches the classical definition of potential energy.

5 Predictions

The physical models we consider have an unknown, stochastic input (e.g. the hydraulic conductivity). This represents a lack of knowledge of the input, which induces a random variable response (e.g. the pressure head). Quantifying the distribution over the response is referred to as a pushforward or *forward problem*. The *pushforward measure* is the distribution over the response, or quantity of interest derived from the response.² Based on the methods of the preceding sections, we now present an emulation framework for interrogating the pushforward measure (the response distribution). We begin by describing in the next section how a single realization of the random variable response may be obtained given a single realization of the stochastic input. In Sect. 5.2, we then discuss how to quantify the pushforward measure (extract relevant statistics of the response).

5.1 Outputs Conditioned on Inputs

Due to the nature of the emulator, the prediction of a point $\mathbf{z} \in \mathcal{F}$ is normally distributed. This distribution captures uncertainty in the predictions as a consequence of limited and noise corrupted data. A common challenge when using reduced dimensional representations is analytically propagating this distribution through a nonlinear, pre-image map [in this case $\hat{\mathbf{f}} : \mathcal{F} \ni \mathbf{z} \mapsto \mathbf{y} \in \mathcal{Y}$ defined by Eq. (23)] for a test input ξ .

Analytically propagating a distribution through a nonlinear mapping is often not feasible. Instead we could repeatedly sample from the feature space response distribution (over $\mathbf{z} \in \mathcal{F}$) and apply the pre-image map to find the distribution over the corresponding $\mathbf{y} \in \mathcal{Y}$. Examples that use this latter approach include kernel principal component analysis and Gaussian process latent variable models (in the latter case, approximations can be obtained using the projected process approximation). Since the manifold consists of aligned (tangent) hyperplanes, however, we are able to derive locally linear pre-image maps that can be used for mapping distributions defined on local tangent spaces. The latent variable GP prediction \mathbf{z} is distributed according to Eq. (27). Restricting to a single tangent space, it is a straightforward task to push this distribution through Eq. (23) to obtain a normal distribution for the corresponding $\mathbf{y} \in \mathcal{Y}$:

$$\begin{aligned}
 p(\mathbf{y}|\xi, \mathcal{D}, \Theta, \beta) &= \mathcal{N}(\boldsymbol{\mu}_{\mathbf{y}}(\xi), \Sigma_{\mathbf{y}}(\xi)), \\
 \boldsymbol{\mu}_{\mathbf{y}}(\xi) &= \bar{\mathbf{y}}_k + \mathbf{Q}_k \mathbf{L}_k^{-1} (\boldsymbol{\mu}_{\mathbf{z}}(\xi) - \bar{\mathbf{z}}_k), \\
 \Sigma_{\mathbf{y}}(\xi) &= \mathbf{Q}_k \mathbf{L}_k^{-1} \Sigma_{\mathbf{z}}(\xi) (\mathbf{Q}_k \mathbf{L}_k^{-1})^T,
 \end{aligned}
 \tag{30}$$

where $k = \arg \min_n \|\boldsymbol{\mu}_{\mathbf{z}}(\xi) - \mathbf{z}_n\|$, $\boldsymbol{\mu}_{\mathbf{y}}(\xi) \in \mathbb{R}^{k_y}$, and $\Sigma_{\mathbf{y}}(\xi) \in \mathbb{R}^{k_y \times k_y}$. This result is particularly useful for scenarios in which knowledge of the correlations between response features is required. Without this result we would require a large number of samples to estimate the covariance (tens of thousands). If, however, we are only interested in samples of the distribution (27), i.e. making predictions at specified inputs, then it is more memory efficient to sample from the predictive distribution (27) and use the pre-image map (23). When the support of this distribution is large, the accuracy of the local approximation breaks down and we must first sample the latent features before applying the pre-image map.

² Let $\mathbb{P}_{\mathcal{X}}$ be a measure on $(\mathcal{X}, \mathcal{F}_{\mathcal{X}})$. The pushforward measure of $\mathbb{P}_{\mathcal{X}}$ under $\eta : (\mathcal{X}, \mathcal{F}_{\mathcal{X}}, \mathbb{P}_{\mathcal{X}}) \rightarrow (\mathcal{Y}, \mathcal{F}_{\mathcal{Y}}, \mathbb{P}_{\mathcal{Y}})$ is defined as $\mathbb{P}_{\mathcal{Y}}(F) = \mathbb{P}_{\mathcal{X}} \circ \eta^{-1}(F)$ for $F \in \mathcal{F}_{\mathcal{Y}}$. We characterize the measures by their probability density functions (pdfs) with respect to Lebesgue measure. In this work a Gaussian distribution is placed on the inputs.

5.2 Marginalizing the Stochastic Input

Having obtained a distribution over the response for a stochastic input realization, we now consider the problem of obtaining a distribution over the response marginalized over the stochastic input. We assume that the input is normally distributed:

$$p(\boldsymbol{\xi}) = \mathcal{N}(\boldsymbol{\mu}_\xi, \Sigma_\xi), \tag{31}$$

for some mean vector $\boldsymbol{\mu}_\xi$ (equal to $\mathbf{0}$ in this case) and covariance matrix Σ_ξ (equal to \mathbf{I} in this case). We wish to evaluate:

$$p(y|\mathcal{D}, \boldsymbol{\Theta}, \boldsymbol{\beta}) = \hat{\mathbf{f}}(p(\mathbf{z}|\mathcal{D}, \boldsymbol{\Theta}, \boldsymbol{\beta})) = \hat{\mathbf{f}}\left(\int p(\mathbf{z}|\boldsymbol{\xi}', \mathcal{D}, \boldsymbol{\Theta}, \boldsymbol{\beta})p(\boldsymbol{\xi}')d\boldsymbol{\xi}'\right), \tag{32}$$

where $\hat{\mathbf{f}}$ is the (measurable) pre-image map and $p(\mathbf{z}|\boldsymbol{\xi}, \mathcal{D}, \boldsymbol{\Theta}, \boldsymbol{\beta})$ is defined in Eq. (27). Since the input $\boldsymbol{\xi}$ appears nonlinearly in the inverse of the \mathbf{z} predictive distribution covariance $\sigma^2(\boldsymbol{\xi})$, we are unable to find a closed form solution to the integral in (32), i.e. the marginal distribution over \mathbf{z} . The moments of this marginal distribution can, on the other hand, be found analytically, although we will not know the family of distributions to which these moments belong.

Let us focus on the i th feature of \mathbf{z} . We wish to find the first two moments, i.e. the mean and variance, of the marginal distribution $p(z_i|\mathcal{D}, \boldsymbol{\theta}_i, \beta_i)$. We can then push these moments through the pre-image map to obtain analytical solutions. This can be repeated for each i by virtue of the independence assumption. To begin, $p(z_i|\mathcal{D}, \boldsymbol{\theta}_i, \beta_i)$ is approximated as a Gaussian with mean m and variance v (Girard and Murray-Smith 2003), which, from ‘‘Appendix A’’, are given by:

$$m = \mathbb{E}_\xi [c_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)]^T (\mathbf{C}_i + \beta_i^{-1}\mathbf{I})^{-1} \mathbf{z}_{:,i} \tag{33}$$

and:

$$v = \mathbb{E}_\xi [c_h(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i)] - m^2 - \left[(\mathbf{C}_i + \beta_i^{-1}\mathbf{I})^{-1} - \left((\mathbf{C}_i + \beta_i^{-1}\mathbf{I})\mathbf{z}_{:,i} \right)^2 \right] \mathbb{E}_\xi [c_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)^T c_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)]. \tag{34}$$

where $\mathbb{E}_\xi[\cdot]$ and $\text{Var}_\xi(\cdot)$ are the expectation and variance with respect to $\boldsymbol{\xi}$, respectively. Calculation of these moments involves expectations of the kernel with respect to the stochastic input distribution on the unknown and unobserved test inputs:

$$\begin{aligned} \mathbb{E}_\xi [c_h(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i)] &= \int c_h(\boldsymbol{\xi}', \boldsymbol{\xi}'; \boldsymbol{\theta}_i)p(\boldsymbol{\xi}')d\boldsymbol{\xi}', \\ \mathbb{E}_\xi [c_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)] &= \int c_h(\boldsymbol{\xi}', \boldsymbol{\Xi}; \boldsymbol{\theta}_i)p(\boldsymbol{\xi}')d\boldsymbol{\xi}', \\ \mathbb{E}_\xi [c_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)^T c_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)] &= \int c_h(\boldsymbol{\xi}', \boldsymbol{\Xi}; \boldsymbol{\theta}_i)^T c_h(\boldsymbol{\xi}', \boldsymbol{\Xi}; \boldsymbol{\theta}_i)p(\boldsymbol{\xi}')d\boldsymbol{\xi}'. \end{aligned} \tag{35}$$

The analytic tractability of these integrals is dependent upon the choice of kernel and stochastic input distribution. One example of a kernel is the commonly used squared exponential, for which the integrals are derived in ‘‘Appendix B’’. Once calculated, the mean can be pushed through the local pre-image mapping (23). Since we expect that the variance, on the other hand, will span more than one tangent space, predictions of the variance using this method may be inaccurate.

Since we cannot sample from the approximate marginal of the analytical approach, further analysis requires MC to fully characterize the distribution (32). Again it is sufficient to demonstrate the procedure for a single latent (feature space) dimension i . Using MC we obtain a marginalized predictive distribution expressed as the sum of normally distributed random variables, which itself is non-Gaussian:

$$\begin{aligned} p(z_{\cdot,i}|\mathcal{D}, \theta_i, \beta_i) &= \int p(z_{\cdot,i}|\xi', \mathcal{D}, \theta_i, \beta_i) p(\xi') d\xi' \\ &\simeq \frac{1}{Q} \sum_{q=1}^Q p(z_{\cdot,i}|\xi^{(q)}, \mathcal{D}, \theta_i, \beta_i) \\ &= \frac{1}{Q} \sum_{q=1}^Q \mathcal{N}(\mu(\xi^{(q)}), \sigma^2(\xi^{(q)})), \end{aligned} \quad (36)$$

where $\xi^{(q)} \sim \mathcal{N}(\mu_\xi, \Sigma_\xi)$, θ_i and β_i are samples from the hyperparameter and signal noise posteriors (for the i th feature), and the approximation converges as $Q \rightarrow \infty$. Each sampled latent variable can then be pushed through the pre-image map. Latent variables found in this way are draws from the marginalized distribution $p(z_{\cdot,i}|\mathcal{D}, \theta_i, \beta_i)$ and we obtain multiple marginalized distributions [one for each (θ_i, β_i)] from which we can estimate the statistics of the response. Algorithm 1 describes the procedure. Note that we use a * superscript in order to avoid confusion between MC samples and training points. Each \mathbf{Y}_i^* in Algorithm 1 can be interrogated to find any property of the pushforward measure (mean, standard deviation and higher-order moments). We can use kernel density estimation (KDE, also known as Parzen–Rosenblatt window) (Simonoff 1996) to approximate the pdf given a finite number of samples, or find the moments of the density. We use Gaussian kernel function with a suitably small bandwidth.

Algorithm 1 Sample from the push forward measure

- 1: $S \leftarrow$ Number of hyperparameter posterior samples
 - 2: $Q \leftarrow$ Number of draws from the input distribution $p(\xi)$
 - 3: $\{\xi_q^*\}_{q=1}^Q \leftarrow$ Dense set of Q draws from $p(\xi)$
 - 4: **for** $s \leftarrow 1$ to S **do**
 - 5: $\Theta_s, \beta_s \leftarrow$ Sample from hyperparameter and signal precision posteriors
 - 6: **for** $q \leftarrow 1$ to Q **do**
 - 7: $\mathbf{z}_{s,q}^* \leftarrow$ Sample from Eq. (27) using $\Theta_s, \beta_s, \xi_q^*$
 - 8: $\mathbf{y}_{s,q}^* \leftarrow$ Application of pre-image map (23) to $\mathbf{z}_{s,q}^*$
 - 9: **end for**
 - 10: $\mathbf{Y}_s^* \leftarrow [\mathbf{y}_{s,1}^* \dots \mathbf{y}_{s,Q}^*]^T$.
 - 11: **end for**
 - 12: Interrogate $\{\mathbf{Y}_s^*\}_{s=1}^S$ to extract statistics or distributions
-

6 Results and Discussion

We now assess the performance of the proposed method on two example partial differential equation problems: a Darcy flow problem with a contaminant mass balance, modelling steady-state groundwater flow in a 2-d porous medium; and Richards equation, modelling single-

phase flow through a 3-d porous medium. As explained in Sect. 5, the analysis includes: (i) predictions that are conditioned on an input; and (ii) predictions that are marginalized over the stochastic input.

When making conditioned predictions, we use the conditional predictive distribution (30) for \mathbf{y} , or the distribution (27) for \mathbf{z} in conjunction with the pre-image map (23). As explained in 4.1, we place a prior over the hyperparameters Θ and signal variances β and use a HMC scheme to sample from the posterior distributions over these parameters. Each sample can be used to obtain a different normal predictive distribution, conditioned on an input. We are therefore able to see how the predictive mean and variance change with respect to the uncertainty in the GP parameters. In the results, we plot the expectation and standard deviation of first two predictive distribution moments.

For the forward UQ problem we marginalize the conditional predictive distributions over a stochastic input (Eq. 32) to obtain the pushforward measure (non analytically). We are able to analytically find the mean using (A2) and (A3) together with the pre-image map, or, using Algorithm 1, sample from the marginalized distribution via MC (Eq. 36).

The accuracies of both the point predictions and the predictions of the pushforward measure are assessed by comparison with the true values obtained with the simulator (on the test inputs $\{\xi_q^*\}_{q=1}^Q$). We run the solver for each test input to generate the true response, denoted $\tilde{\mathbf{y}}_q^*$. For the UQ comparison we again approximate the pdf using KDE (or simply extract the moments) based on $\{\tilde{\mathbf{y}}_q^*\}_{q=1}^Q$. The latter approximation is guaranteed to converge to the truth as the number of test inputs increases.

6.1 Darcy Flow: Non-point Source Pollution

The first example is a linear model of steady-state groundwater flow in 2-d. The approach was developed by Kourakos et al. (2012) and implemented in the mSim package.³ The model comprises Darcy’s law and a contaminant mass balance in a 2-d polygonal domain Ω of total area 18.652 km² containing wells and a stream, and subdivided into polygonal regions of different land use (Fig. 5). Full details of the model and the numerical method can be found in Kourakos et al. (2012). Below we provide a brief description. The model equations are given by:

$$\begin{aligned} \nabla \cdot (K \nabla h) &= Q \\ R \frac{\partial C}{\partial t} &= \nabla \cdot (\mathbf{D} \nabla C) - \nabla \cdot (\mathbf{v} C) = G \end{aligned} \tag{37}$$

where $K(\mathbf{x})$ is the hydraulic conductivity, $h(\mathbf{x})$ is the pressure head, $C(\mathbf{x}, t)$ is the contaminant concentration, R is the retardation factor, \mathbf{D} is the dispersion tensor, \mathbf{v} is the fluid velocity, and Q and G represent sources/sinks. The contaminant transport equation is replaced by a 1-d approximation and is solved through an ensemble of one-dimensional streamline-based solutions (Kourakos et al. 2012).

The contaminant balance and flow (Darcy) equations are decoupled. The latter is solved using the finite element method based on triangular elements and first-order (linear) shape functions. The boundary conditions are given by: (i) a constant head equal to 30 m on the left boundary; (ii) a general head boundary equal to 40 m with conductance equal 160 m³ day⁻¹ on the right boundary; and (iii) no flow on the top and bottom boundaries. Each land

³ See http://subsurface.gr/joomla/msim_doc/twoD_examples_help.html for full details of the implementation, including the domain, mesh generation and boundary conditions. Last accessed 29 August 2017.

use polygon is assigned its own recharge rate. Stream rates are assigned directly to nodes. (Any node closer than 10 m to the stream is considered to be part of the stream.)

We assume that $K(\mathbf{x})$ is log-normally distributed and treat it as an input. The output field upon which we focus is the pressure head, that is, $u(\mathbf{x}; K) = h(\mathbf{x})$ in the notation of Sect. 2. We use the input model described in Sect. 2, defining a discretized random field corresponding to realizations of $K(\mathbf{x}) = \exp(Z(\mathbf{x}))$ at the nodes $\{\mathbf{x}_k\}_{k=1}^{k_y} \subset \mathcal{R}$ on the finite element mesh. The covariance function for the random field $Z(\mathbf{x})$ is given by:

$$c_Z(\mathbf{x}, \mathbf{x}') = \sigma_Z^2 \exp \left\{ -\frac{(x_1 - x'_1)^2}{l_1^2} - \frac{(x_2 - x'_2)^2}{l_2^2} \right\}, \quad \mathbf{x} = (x_1, x_2)^T \in \mathcal{R}, \quad (38)$$

where l_1 and l_2 are correlation lengths. This separable form was suggested in Zhang and Lu (2004) and is used extensively in the literature to model hydraulic permeability fields (often by setting the correlation lengths equal). The generalized variance (value of k_ξ) was chosen to satisfy $\sum_{j=1}^{k_\xi} \sqrt{\lambda_j} / \sum_{j=1}^{k_y} \sqrt{\lambda_j} > 0.98$.

Both the training and test input samples were drawn independently: $\xi_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $n = 1, \dots, N$ to yield $\{\mathbf{y}_n\}_{n=1}^N$ for training; and $\xi_q \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $q = 1, \dots, Q$ to yield $\{\tilde{\mathbf{y}}_q\}_{q=1}^Q$ for testing and the forward problem (UQ). We set $Q = 5000$ and $N \in \{25, 50, 75, 100\}$. Running the solver with an input generated using the KL truncation necessarily leads to a response surface with intrinsic dimension at most k_ξ , which was therefore the value chosen for the approximating manifold dimension k_z . In all of the results presented below, $k_y = 1933$ nodes (elements) were used in the simulation. The number of neighbours P in the LTSa algorithm was chosen according to the error between the solver response and the predictive mean at the test points. We define a scaled measure of error on each test point as follows:

$$e_q = \|\tilde{\mathbf{y}}_q^* - \bar{\mathbf{y}}_q^*\| / \|\tilde{\mathbf{y}}_q^*\|, \quad q = 1, \dots, Q, \quad (39)$$

where $\tilde{\mathbf{y}}_q^*$ is the response predicted by the solver, and $\bar{\mathbf{y}}_q^*$ is the point recovered by application of the pre-image map (23) on the GP predictive mean (26). The scaling ensures that the errors are comparable and can be interpreted as percentage errors.

We present results for three stochastic input models:

- M1** We set $m_Z = \ln(40)$ and $\sigma_Z^2 = 0.2$, yielding⁴ a mean for $k(\mathbf{x})$ of 44.2 m day⁻¹, which is close to the default value in the mSim package, and a standard deviation of 13.63 m day⁻¹. The correlation lengths were chosen as $l_1 = 2000$ m and $l_2 = 1000$ m, which correspond to dimensionless values of 1/3 and 2/7, respectively. These choices require $k_\xi = 5$ input dimensions to capture 98% of the generalized variance.
- M2** We set $m_Z = \ln(36.18)$ and $\sigma_Z^2 = 0.4$, again yielding a mean 44.2 m day⁻¹ and a standard deviation of 18.80 m day⁻¹. We set $l_1 = 2000$ m and $l_2 = 1000$ m. $k_\xi = 5$ captures 98% of the generalized variance.
- M3** We set $m_Z = \ln(40)$, and $\sigma_Z^2 = 0.4$ and reduce the correlation lengths to $l_1 = 1000$ m and $l_2 = 500$ m (dimensionless values of 1/6 and 1/7, respectively). We now require $k_\xi = 15$ to capture 98% of the generalized variance.

For model **M1**, the distributions of $\{e_q\}_{q=1}^Q$ for different training set sizes N are shown as boxplots for increasing values of P in Fig. 1. The performance of the emulator is good even for $N = 25$ training points (maximum e_q of approximately e^{-3}), although there is a clear decrease in the error when N is increased to 100. The relationship between the errors

⁴ If $Z(\mathbf{x})$ has a mean and variance of μ and v , then the mean and variance of the log-normal process $\exp(Z(\mathbf{x}))$ are $\mu' = \exp(\mu + v/2)$ and $v' = \exp(2\mu + v)(\exp(v) - 1)$, respectively.

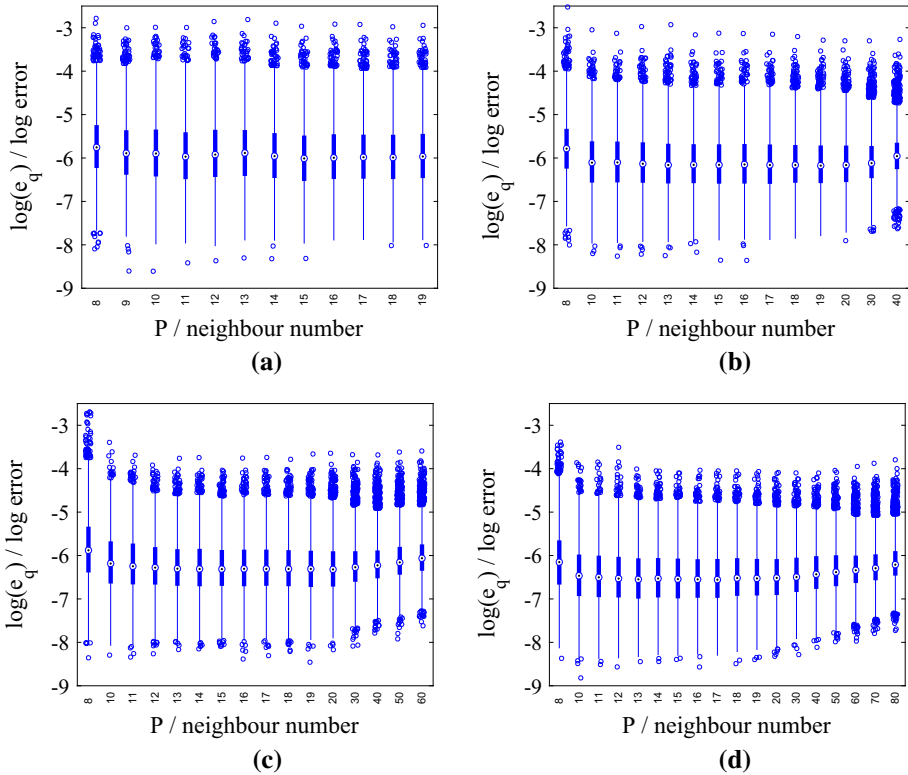


Fig. 1 Log normalized error $\ln(e_q)$ in the normalized pressure head prediction for an emulator trained on $N = 25, 50, 75$ and 100 points \mathbf{y}_n and tested with $Q = 5000$ test points $\tilde{\mathbf{y}}_q^*$ for different nearest neighbour numbers P (model **MI**). Predictions were obtained by averaging over hyperparameter and precision posterior samples. **a** 25 training points. **b** 50 training points. **c** 75 training points. **d** 100 training points

and P is more complicated. The errors are high for $P < 8$ (not shown in the boxplots) at all values of N and decrease as P increases. This is due to the linear approximation of points in local tangent spaces via PCA in the LTSa algorithm. As more points are added, the approximation improves. As P is increased beyond a certain value, however, the errors increase (this is most clearly visible for $N = 100$). The reason for this behaviour is that for large enough neighbourhood sizes the linear approximation breaks down. Thus, there is an optimal choice of P for each value of N and the higher the value of N the more sensitive are the errors to the value of P . In the subsequent results we use $P = 15$ unless otherwise specified.

In Fig. 2 we plot the *normalized* pressure head prediction (for each coordinate of the predicted pressure head we subtract the mean and divide by the standard deviation) corresponding to the highest e_q for both $N = 25$ and $N = 50$ (using $P = 15$). The normalization highlights the differences between the true values and the predictions (the errors) more clearly. The predicted means of the means (middle row) are the mean predictions averaged over all hyperparameter and precision samples. Also shown (bottom row) are the standard deviations of the predictions averaged over all hyperparameter and precision samples. We observe that the prediction at $N = 75$ is highly accurate, while the prediction at $N = 25$ is still reasonably accurate even in this worst case (an outlier in Fig. 1). For both values of N , the true values lie

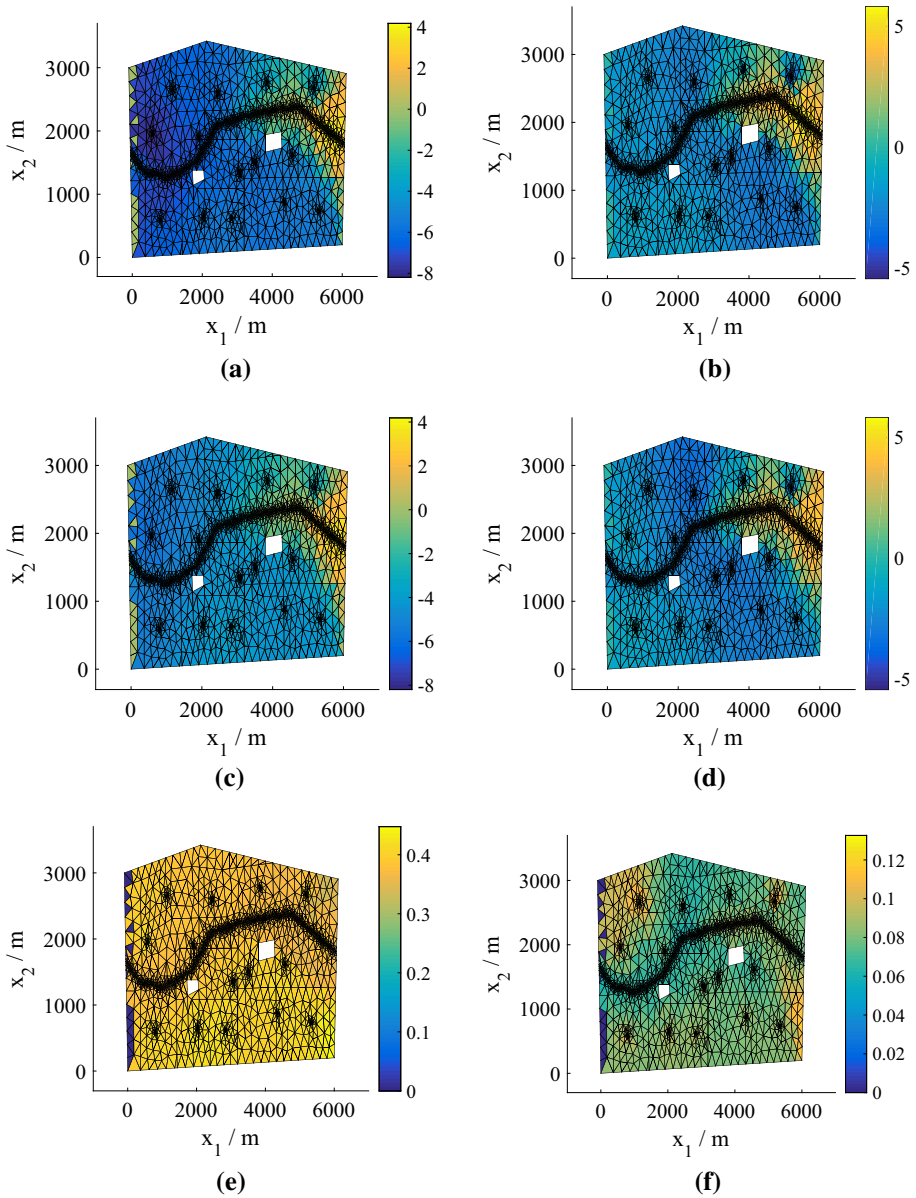


Fig. 2 The test predictive mean and standard deviation of the normalized pressure head for the point with highest error from an emulator using $P = 15$, corresponding to the relevant boxplot in Fig. 1, for both 25 and 75 training points (model **M1**). **a** True value, $N = 25$. **b** True value, $N = 75$. **c** Mean of the mean, $N = 25$. **d** Mean of the mean, $N = 75$. **e** Mean of the standard deviation, $N = 25$. **f** Mean of the standard deviation, $N = 75$

within the credible regions. In Fig. 3 we show the corresponding predictions for cases where the errors are close to the medians. Both predictions are highly accurate and again the true values lie inside the credible regions.

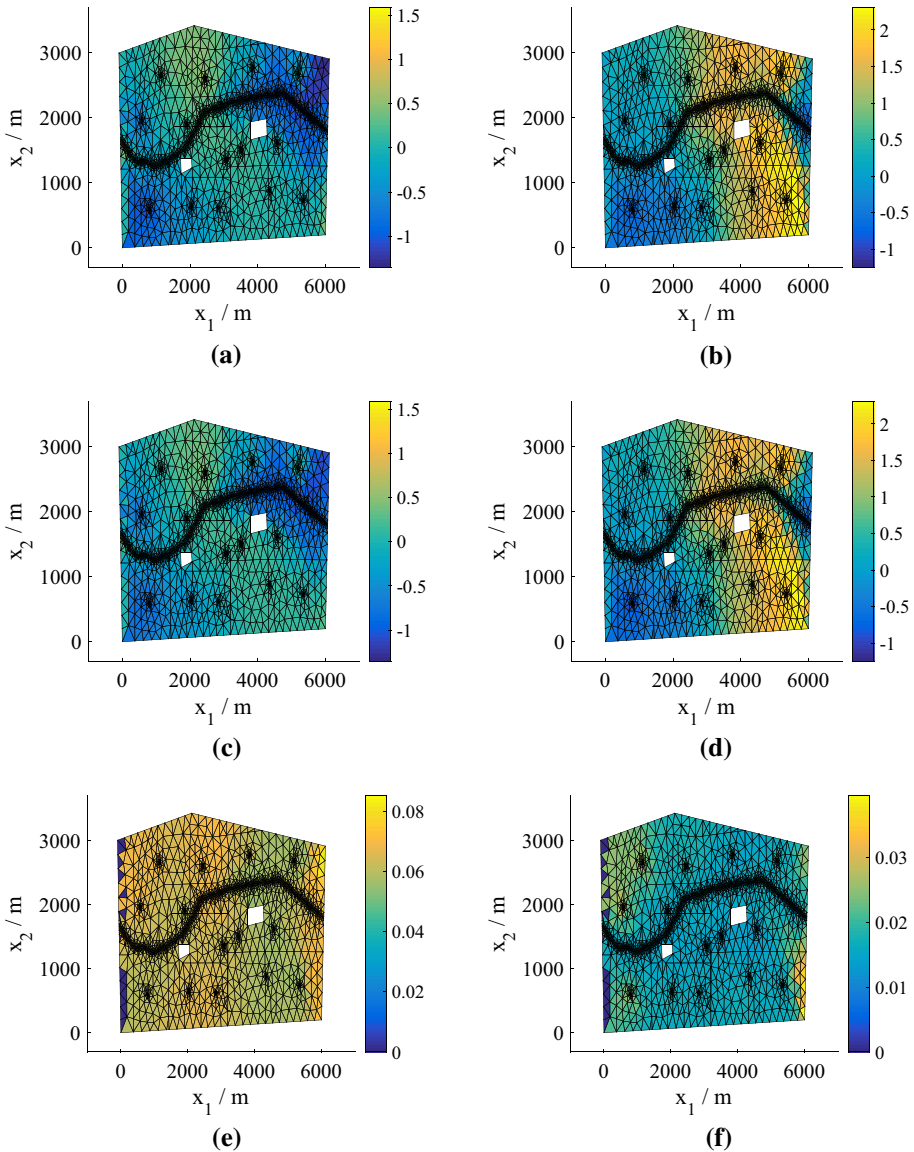


Fig. 3 The test predictive mean and standard deviation of the normalized pressure head for a test point with an error near the median of the $P = 15$ boxplot in Fig. 1 from emulators using $P = 15$, for both 25 and 75 training points (model M1). **a** True value, $N = 25$. **b** True value, $N = 75$. **c** Mean of the mean, $N = 25$. **d** Mean of the mean, $N = 75$. **e** Mean of the standard deviation, $N = 25$. **f** Mean of the standard deviation, $N = 75$

We now focus on the forward problem, in which we estimate the marginalized predictive distribution (32) using Algorithm 1. KDE is used to obtain estimates of the pdf of a feature for different predictive posterior, hyperparameter and precision samples, as previously described. The feature we choose is the pressure head at the spatial location $\mathbf{x} = (2511, 486) \in \mathcal{R}$. We plot a heat map of the pdfs in Fig. 4 for different N .

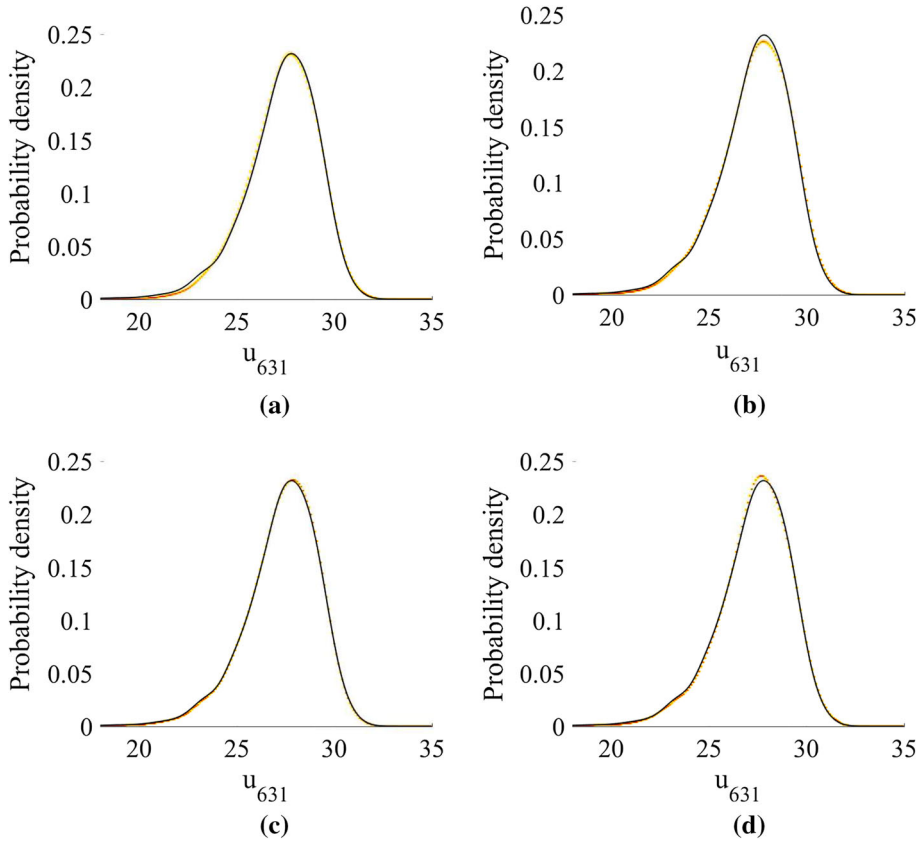


Fig. 4 The pdfs of the pressure head response at the spatial coordinate $\mathbf{x} = \mathbf{x}_{631} = (2511, 486) \in \mathcal{R}$ on the finite difference grid, obtained using kernel density estimation on $Q = 5000$ points (Model **M1**). The black line gives the MC prediction using the simulator. The contours show how the emulator predictions vary with hyperparameter, precision and predictive distribution samples. **a** 25 training points, 15 k-NN. **b** 50 training points, 15 k-NN. **c** 75 training points, 15 k-NN. **d** 100 training points, 15 k-NN

The distributions are accurately estimated for all values of N . While the predictions improve as the number of training samples N increases, the true value does not always lie within the contours. This is because: (i) as stated earlier, an increased GP predictive variance acts to smooth the density, rather than increase the width of the contours; (ii) by choosing a priori the number of neighbours we also a priori assume a *global* smoothness of the emulator; and (iii) we have a pre-image map $\hat{\mathbf{f}} : \mathcal{F} \rightarrow \mathcal{Y}$ for which the error is unknown (as with all methods), but not estimated (as with probabilistic methods).

We can find the means and standard deviations across the samples obtained for different predictive posterior, hyperparameter and precision samples using Algorithm 1. We obtain distributions over the moments of the marginalized predictive distribution (32). In Fig. 5 we plot the mean and standard deviation of the marginalized predictive mean and standard deviation for $N = 25$, with comparisons to the true values obtained by finding the mean and standard deviation across the test responses $\{\tilde{\mathbf{y}}_q^*\}_{q=1}^Q$. Even for this low number of training points the results are highly accurate.

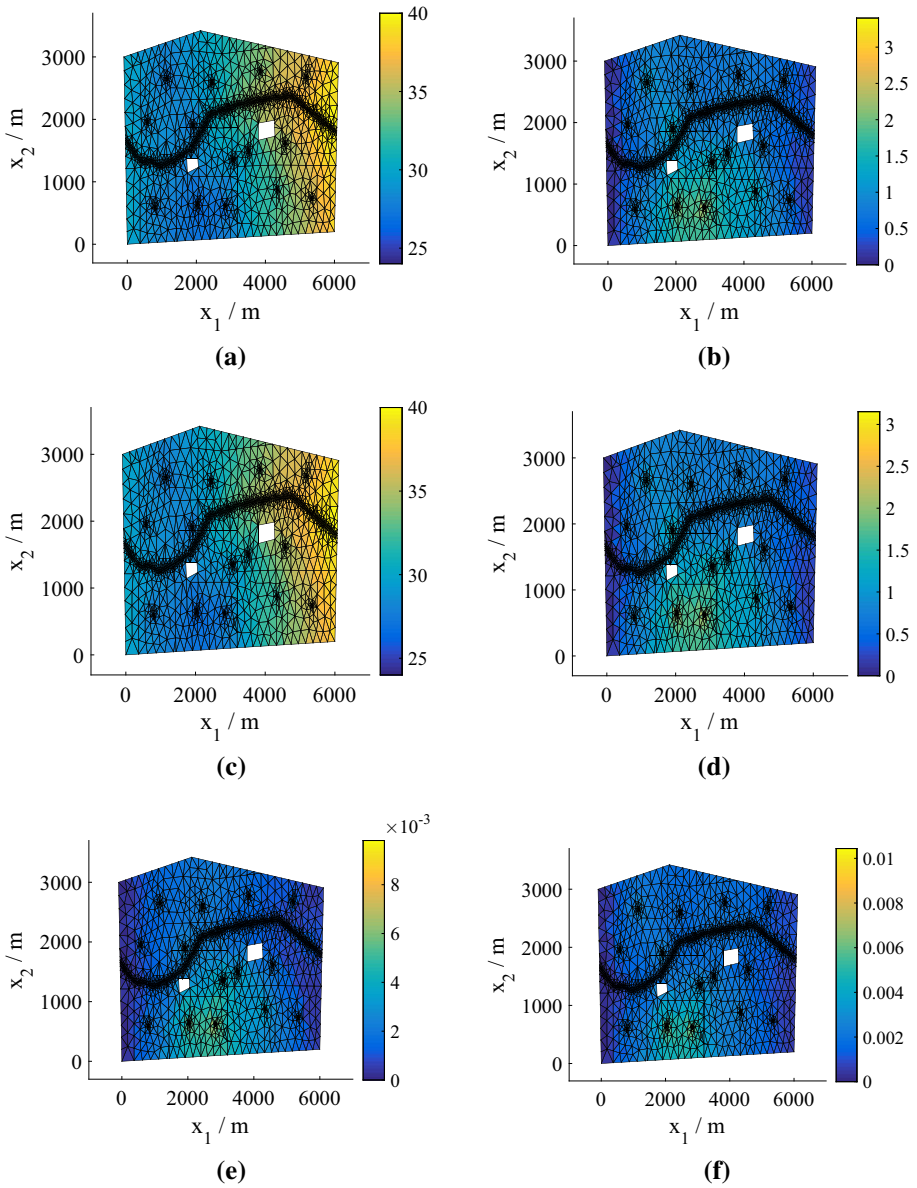


Fig. 5 Moments of the mean and standard deviation of the pressure head in Model **M1**. The emulator variation is a consequence of the hyperparameter, precision and predictive distribution samples. We have a single, parameterized realization of the manifold. **a** Monte Carlo mean. **b** Monte Carlo standard deviation. **c** Mean of the mean for 25 training points. **d** Mean of the standard deviation for 25 training points. **e** Standard deviation of the mean for 25 training points. **f** Standard deviation of the standard deviation for 25 training points

We now consider Model **M2**, in which we increase the variance of the stochastic input, while keeping the mean fixed. For this example we again set $l_1 = 2000$ m and $l_2 = 1000$ m, requiring $k_\xi = 5$. The distributions of $\{e_q\}_{q=1}^Q$ for different training set sizes N and

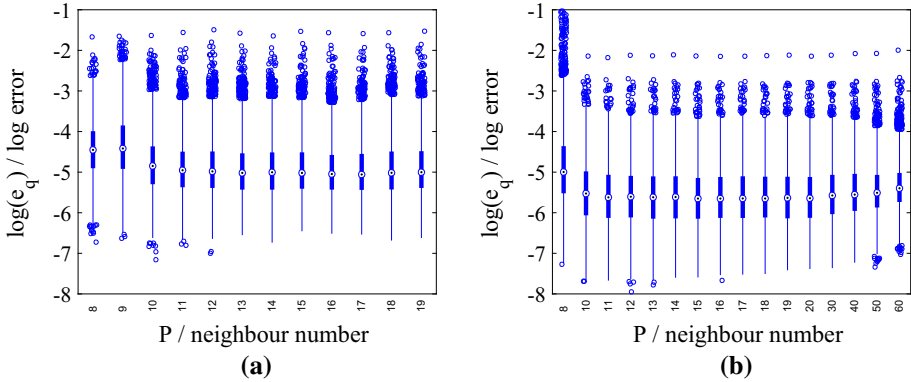


Fig. 6 Log normalized error $\ln(e_q)$ in the normalized pressure head prediction for an emulator trained on $N = 25$ and 75 points \mathbf{y}_n and tested with $Q = 5000$ test points $\tilde{\mathbf{y}}_q^*$ for different nearest neighbour numbers P (model **M2**). Predictions were obtained by averaging over hyperparameter and precision posterior samples. **a** 25 training points. **b** 75 training points

increasing P are shown in Fig. 6. We observe trends similar to those observed using Model **M1**, although the increased variance leads to larger errors at fixed N and P (higher maxima and minima). With the exception of an isolated outlier (shown later), the predictions are nevertheless accurate for $N = 75$.

The worst case (highest e_q) for $P = 15$ is shown in Fig. 7 for $N = 25$ and 75 points (see Fig. 6). As before the top row is the test (solver prediction), while the middle and bottom rows are the mean prediction and standard deviation of the prediction averaged over all hyperparameter and precision samples. The true values lie within the credible regions, although for this model a higher number of training points are required to ensure that even the worst-case predictions are accurate. Figure 8 demonstrates the quality of the predicted responses when the errors are at the median in the $P = 15$ boxplots in Fig. 6. Here, even $N = 25$ provides accurate results.

Figure 9 shows heat maps of the pdfs of the pressure head at the spatial location $\mathbf{x} = (2511, 486)$ for different N (generated using KDE) in the case of Model **M2**. Using $N = 75$ we achieve very good agreement with the MC prediction based on the simulator results (test points), although again the true value does not lie within the contours. For $N = 25$, we plot the mean and standard deviation of the marginalized predictive mean and standard deviation in Fig. 10, with a comparison to the true values obtained from $\{\tilde{\mathbf{y}}_q^*\}_{q=1}^Q$. The predictions are highly accurate. In fact, even for $N = 25$ (not shown to conserve space) the mean was very accurate and the standard deviation exhibited only slight differences from the true value.

For Model **M3** (decreased correlation lengths, high standard deviation and $k_\xi = 15$), the distributions of $\{e_q\}_{q=1}^Q$ for increasing N and P in are shown in Fig. 11. In this case it is clear that a much higher value of P ($P > 60$, giving a similar neighbourhood radius in-line with the increased sample density) is required to obtain a reasonable accuracy. For $N = 500$ and $P = 80$, there are a small (9 out of 5000) number of outliers with low accuracy, while the errors for the remaining points satisfy $\ln(e_q) < -3.25$. The worst cases (highest e_q) for $P = 70, N = 300$ and $P = 80, N = 500$ are shown in Fig. 12, and in Fig.13 we show predicted responses with errors at the medians for the same values of P and N . There are noticeable differences in the worst cases, although the qualitative agreement is very good at both values of N . For the median error cases both emulators perform extremely well.

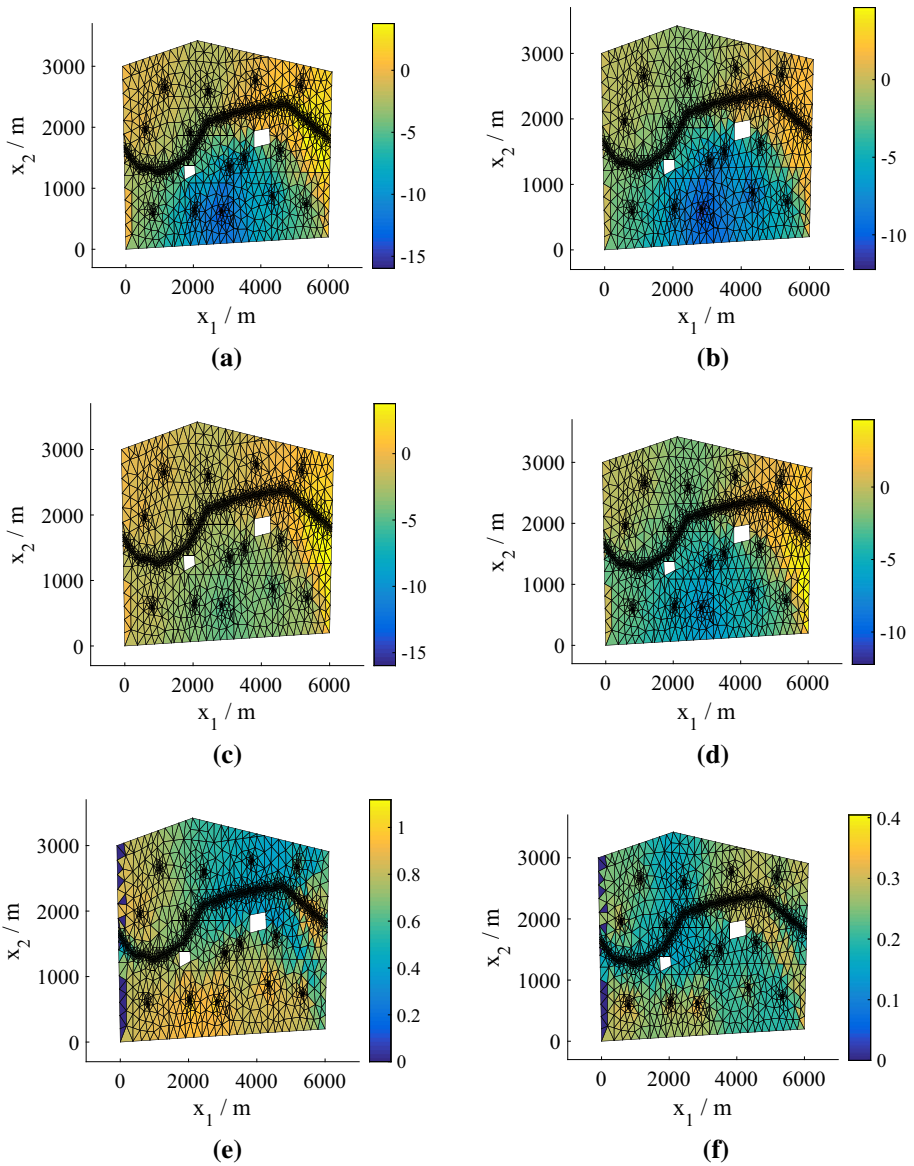


Fig. 7 The test predictive mean and standard deviation of the normalized pressure head in the case of the highest errors e_q in Fig. 6 for $P = 15$ and $N = 25$ and 75 training points (Model **M2**). **a** True value, $N = 25$. **b** True value, $N = 75$. **c** Mean of the mean, $N = 25$. **d** Mean of the mean, $N = 75$. **e** Mean of the standard deviation, $N = 25$. **f** Mean of the standard deviation, $N = 75$

In Fig. 14 we show the heat maps of the pdfs of the pressure head at $\mathbf{x} = (2511, 486)$ for different N . For both values of N there is very good agreement with the simulator result and the true value this time lies within the contours. For $N = 500$, we show the mean and standard deviation of the marginalized predictive mean and standard deviation in Fig. 15, with a comparison to the true values obtained from $\{\tilde{y}_q^*\}_{q=1}^Q$. The predictions are again highly accurate (which was also the case for $N = 300$).

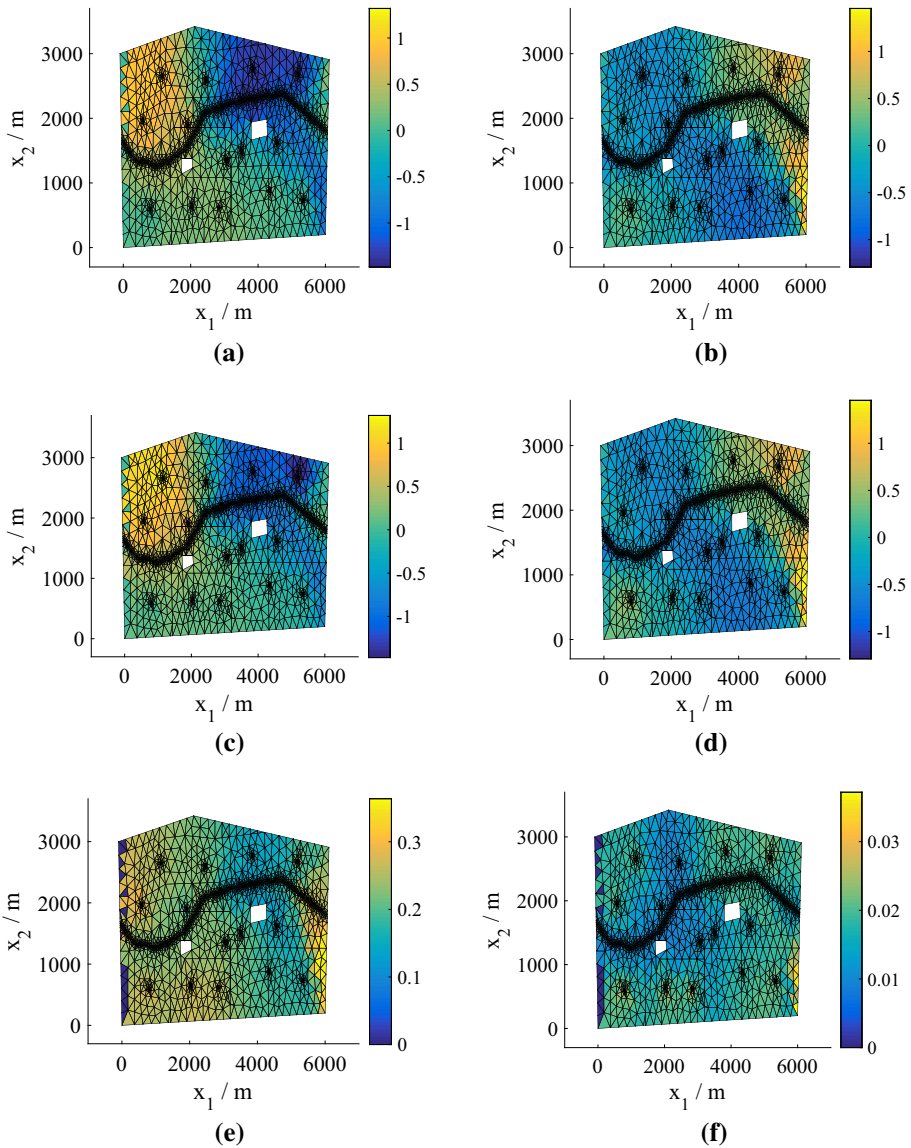


Fig. 8 The test predictive mean and standard deviation of the normalized pressure head in the case of errors e_q near the median in Fig. 6 for $P = 15$ and $N = 25$ and 75 training points (Model M2). **a** True value, $N = 25$. **b** True value, $N = 75$. **c** Mean of the mean, $N = 25$. **d** Mean of the mean, $N = 75$. **e** Mean of the standard deviation, $N = 25$. **f** Mean of the standard deviation, $N = 75$

6.2 Richards Equation: Unsaturated Flow in Porous Media

Consider a single-phase flow through a 3-d porous region $\mathcal{R} \subset \mathbb{R}^3$ containing unsaturated soil with a random permeability field. The vertical flow problem can be solved using Richards equation (Darcy's law combined with a mass balance). There are three standard forms of Richards equation: the pressure head based (h -based) form; the water content-based (θ -

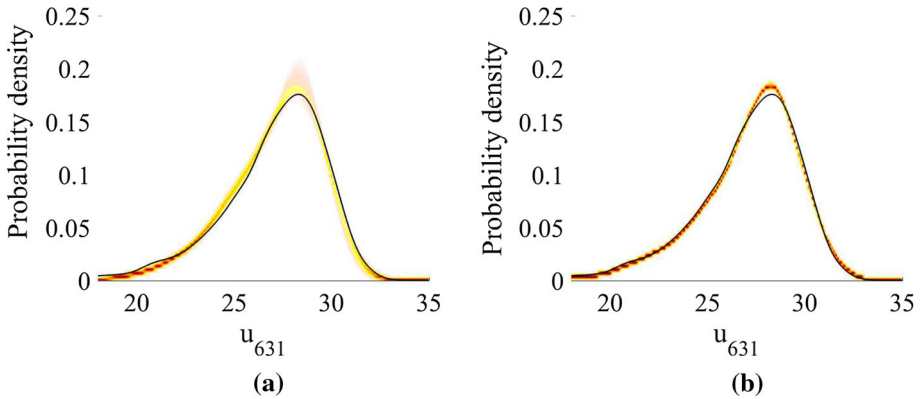


Fig. 9 The pdfs of the pressure head response at the spatial coordinate $\mathbf{x} = \mathbf{x}_{631} = (2511, 486)$ on the finite difference grid, obtained using kernel density estimation on $Q = 5000$ points (Model **M2**). The black line gives the MC prediction using the simulator. The contours show how the emulator predictions vary with hyperparameter, precision and predictive distribution samples. **a** 25 training points, 15 k-NN. **b** 75 training points, 15 k-NN

based) form; and the mixed-based form. For flow in saturated or layered soils, the h -based form is particularly appropriate (Huang et al. 1996; Shahraiyini and Ataie-Ashtiani 2011).

The h -based form with an implicit or explicit finite difference (FD) scheme has been shown to provide good accuracy, although this approach may result in high mass balance errors (Zarba et al. 1990; Huang et al. 1996). The mixed-based form, on the other hand, exhibits low mass balance errors with highly accurate predictions using a fully implicit FD scheme (Ray and Mohanty 1992; Zarba et al. 1990; Celia et al. 1987). The latest work of Shahraiyini and Ataie-Ashtiani (2011) showed that a fully implicit FD scheme with a standard chord slope (CSC) approximation (Rathfelder and Abriola 1994) not only solved the mass balance problem of the h -based form but also improved convergence. Thus, in the paper we adopt this approach, although other numerical formulations are by no means precluded. The h -based form of Richards equation can be written as follows:

$$u(h) \frac{\partial h}{\partial t} - \nabla \cdot K(h) \nabla (h + x_3) = 0, \quad (\mathbf{x}, t) \in \mathcal{R} \times (0, T], \tag{40}$$

where h is the pressure head, $u(h) = \partial\theta/\partial h$ is the specific moisture capacity, in which θ is the moisture content, $K(h)$ is the unsaturated hydraulic conductivity, and $\mathbf{x} = (x_1, x_2, x_3)^T$ is the spatial coordinate, in which x_3 is the vertical coordinate. The nonlinear functions $\theta(h)$ and $k(h)$ can take on different forms. For example, in Haverkamp et al. (1977), a least square fit to experimental data was used to derive:

$$\begin{aligned} \theta(h) &= \frac{\alpha_1(\theta_s - \theta_r)}{\alpha_1 + |h|^{\alpha_2}} + \theta_r, \\ K(h) &= K_s(\mathbf{x}) \frac{\alpha_3}{\alpha_3 + |h|^{\alpha_4}}, \end{aligned} \tag{41}$$

where θ_r and θ_s are the residual the saturated water contents, $K_s(\mathbf{x})$ is the saturated hydraulic conductivity, and $\alpha_1, \alpha_2, \alpha_3$ and α_4 are fitting parameters. We adopt the relationships (41) and use the parameter values in Haverkamp et al. (1977): $\alpha_1 = 1.611 \times 10^6$, $\alpha_2 = 3.96$, $\alpha_3 = 1.175 \times 10^6$, $\alpha_4 = 4.74$, $\theta_s = 0.287$ and $\theta_r = 0.075$. The domain \mathcal{R} is taken to be 20 cm \times 20 cm \times 20 cm. $K_s(\mathbf{x})$ is treated as a random field input with a log-normal

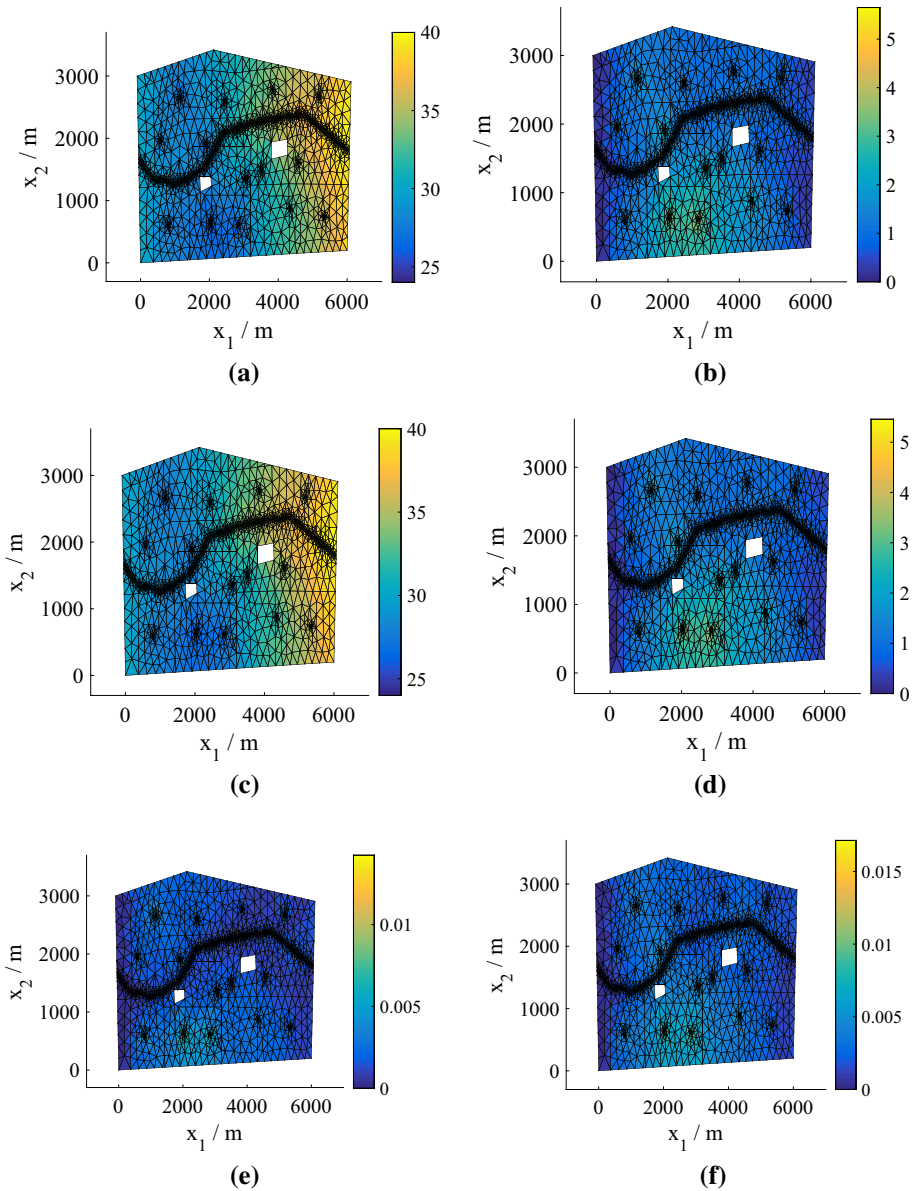


Fig. 10 Moments of the mean and standard deviation of the pressure head for $P = 15$, $N = 75$ (Model M2). The emulator variation is a consequence of the hyperparameter and predictive distribution samples. We have a single, parameterized realization of the manifold. **a** Monte Carlo mean. **b** Monte Carlo standard deviation. **c** Mean of the mean for 75 training points. **d** Mean of the standard deviation for 75 training points. **e** Standard deviation of the mean for 75 training points. **f** Standard deviation of the standard deviation for 75 training points

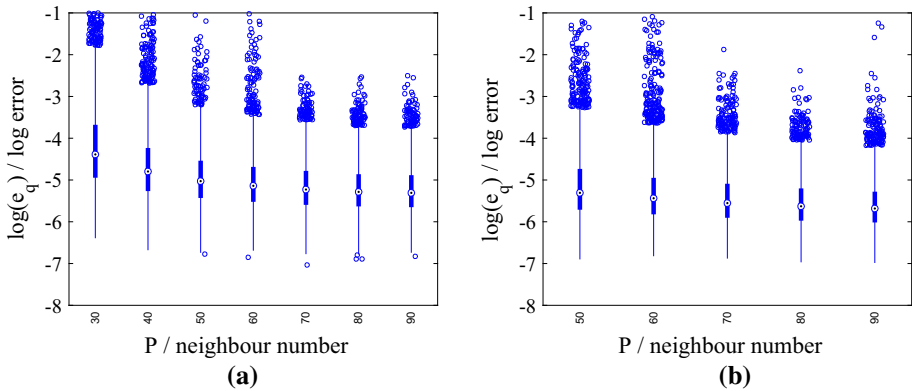


Fig. 11 Log normalized error between true and predictive mean in the normalized pressure head prediction from an emulator trained on 300 and 500 points \mathbf{y}_n and interrogated with $Q = 5000$ test points $\tilde{\mathbf{y}}_q^*$ for different nearest neighbour numbers P (Model **M3**). Predictions were obtained by averaging over hyperparameter and precision posterior samples. **a** 300 training points. **b** 500 training points

distribution ($K_s(\mathbf{x}) = \exp(Z(\mathbf{x}))$), again discretized using the Karhunen–Loève theorem. We generate realizations of a corresponding discrete random field on an $n_1 \times n_2 \times n_3$ finite difference grid ($k_y = n_1 n_2 n_3$), with grid spacings $\Delta x_1, \Delta x_2$ and Δx_3 in the directions x_1, x_2 and x_3 , respectively. The output field of interest is again the pressure head, at a fixed time T . Thus, we set $u(\mathbf{x}; K) = h(\mathbf{x}, T)$.

The boundary conditions are those used in Haverkamp et al. (1977), corresponding to laboratory experiments of infiltration in a plexiglass column packed with sand. Along the top boundary (surface) $x_3 = 20$ cm, the pressure head is maintained at $h = -20.7$ cm ($\theta = 0.267$ cm³ cm⁻³), and along the bottom boundary $x_3 = 0$ cm, it is maintained at $h = -61.5$ cm. At all other boundaries a no-flow condition is imposed: $\nabla h \cdot \mathbf{n} = 0$, where \mathbf{n} is the unit, outwardly pointing normal to the surface. The initial condition is $h(\mathbf{x}, 0) = -61.5$ cm.

The covariance function for the random field $Z(\mathbf{x})$ is again of the form:

$$c_Z(\mathbf{x}, \mathbf{x}') = \sigma_Z^2 \exp \left\{ -\frac{(x_1 - x'_1)^2}{l_1^2} - \frac{(x_2 - x'_2)^2}{l_2^2} - \frac{(x_3 - x'_3)^2}{l_3^2} \right\}, \tag{42}$$

where the l_i are correlation lengths, chosen as $l_1 = l_2 = l_3 = 7.5$ cm. The mean m_Z and variance σ_Z^2 are chosen such that the mean and standard deviation of $K(\mathbf{x})$ are 0.0094 cm s⁻¹ (Shahraiyani and Ataie-Ashtiani 2011; Haverkamp et al. 1977) and 0.00235 cm s⁻¹ (25 % of the mean), respectively. The generalized variance satisfies $\sum_{i=1}^{k_\xi} \sqrt{\lambda_i} / \sum_{i=1}^n \sqrt{\lambda_i} = 0.75$ for $k_\xi = 15$.

The training and test input samples were drawn independently: $\xi_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\xi_q \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to yield $\{\mathbf{y}_n\}_{n=1}^N$ for training and $\{\tilde{\mathbf{y}}_q^*\}_{q=1}^Q$ for testing and UQ. We set $Q = 5000$ and $N \leq 800$. As before, the manifold dimension was set to $k_z = k_\xi$. The number of neighbours P and the number of training points N were chosen as in the first example by examining the errors $e_q = \|\tilde{\mathbf{y}}_q^* - \bar{\mathbf{y}}_q^*\| / \|\tilde{\mathbf{y}}_q^*\|$ on the test set, where again $\tilde{\mathbf{y}}_q^*$ is the solver output (truth) and $\bar{\mathbf{y}}_q^*$ is emulator prediction based on the GP predictive mean (26).

Equation (40) was solved using a finite difference scheme with first-order differencing for the first-order derivatives, central differencing for the second-order derivatives and a fully

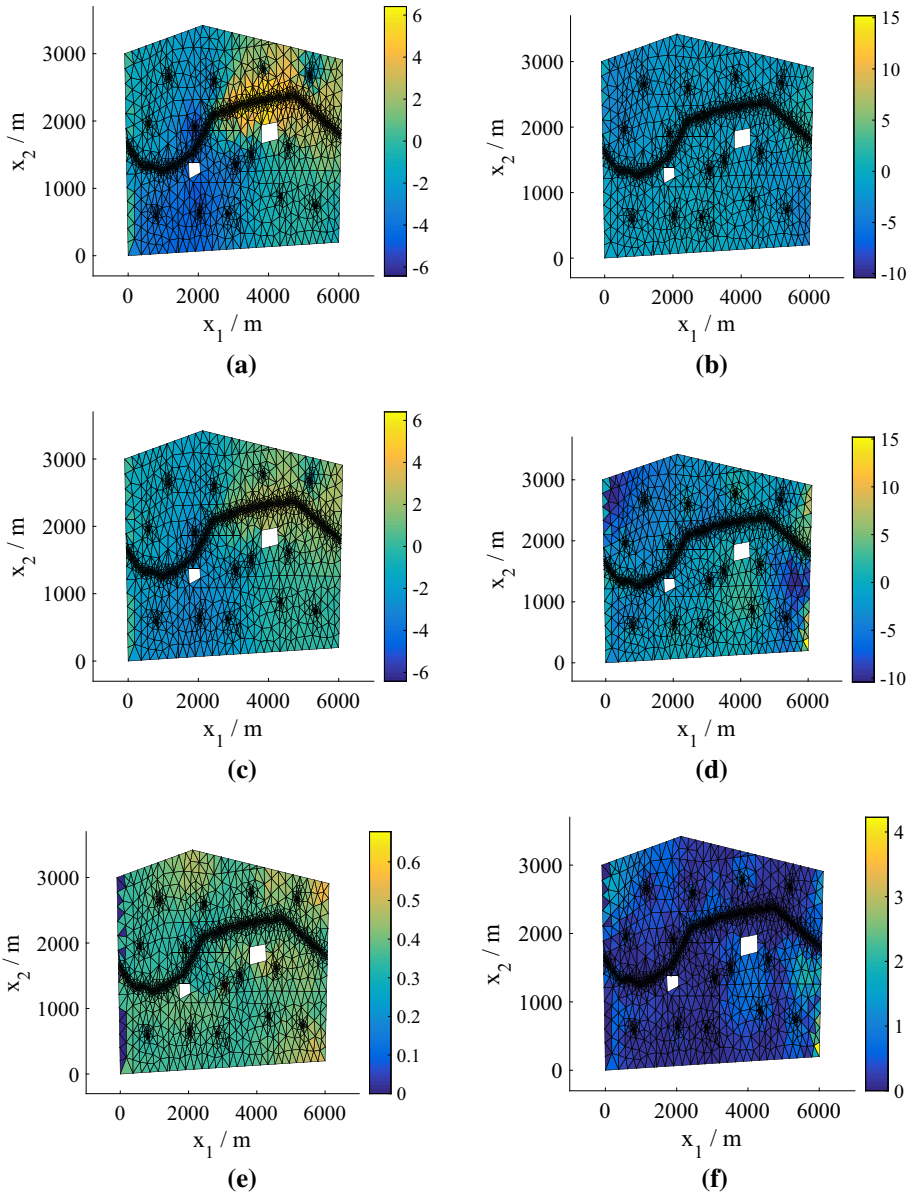


Fig. 12 The test predictive means and standard deviations for predictions of the normalized pressure head with the highest errors from emulators using $P = 70$, $N = 300$ and with $P = 80$, $N = 500$, corresponding to the relevant boxplots in Fig. 11 (Model M3). **a** True value, $N = 300$. **b** True value, $N = 500$. **c** Mean of the mean, $N = 300$. **d** Mean of the mean, $N = 500$. **e** Mean of the standard deviation, $N = 300$. **f** Mean of the standard deviation, $N = 500$

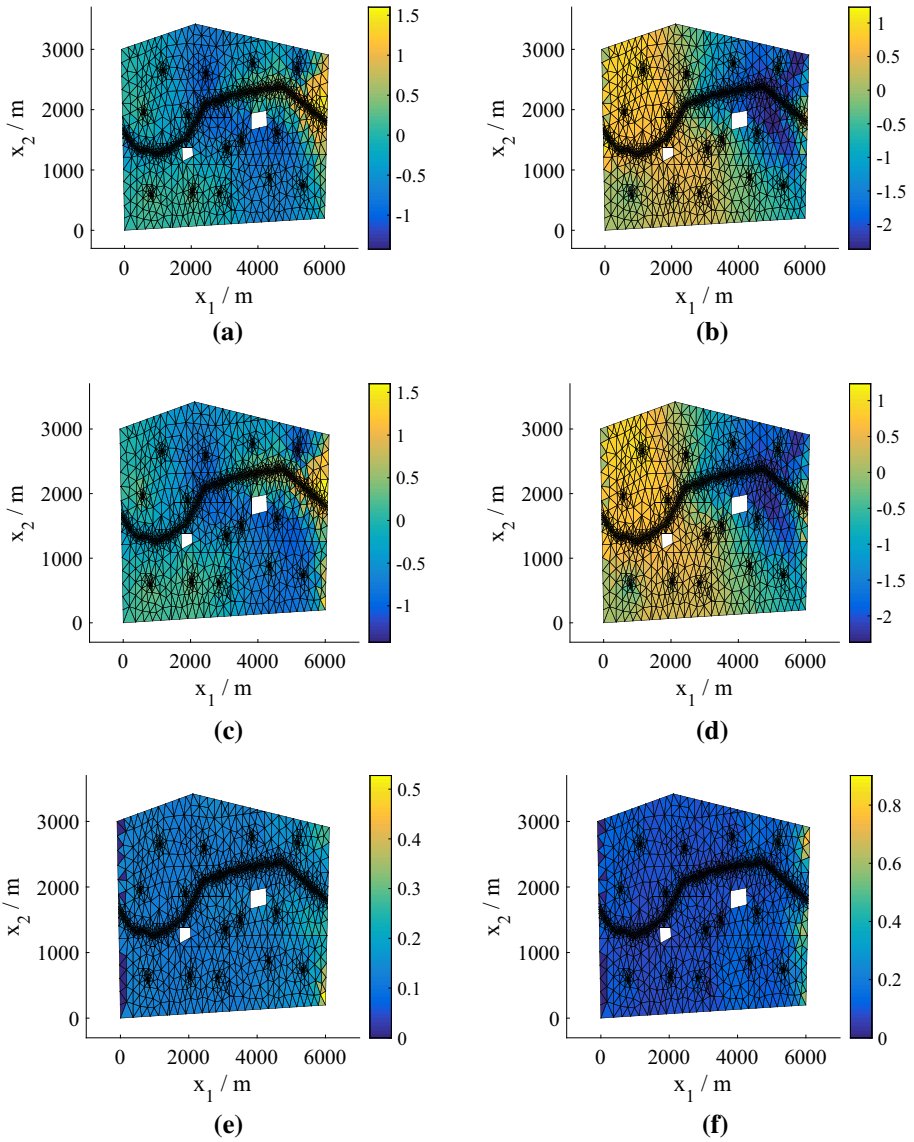


Fig. 13 The test predictive means and standard deviations for predictions of the normalized pressure head with errors at the median from emulators using $P = 70$, $N = 300$ and with $P = 80$, $N = 500$, corresponding to the relevant boxplots in Fig. 11 (Model **M3**). **a** True value, $N = 300$. **b** True value, $N = 500$. **c** Mean of the mean, $N = 300$. **d** Mean of the mean, $N = 500$. **e** Mean of the standard deviation, $N = 300$. **f** Mean of the standard deviation, $N = 500$

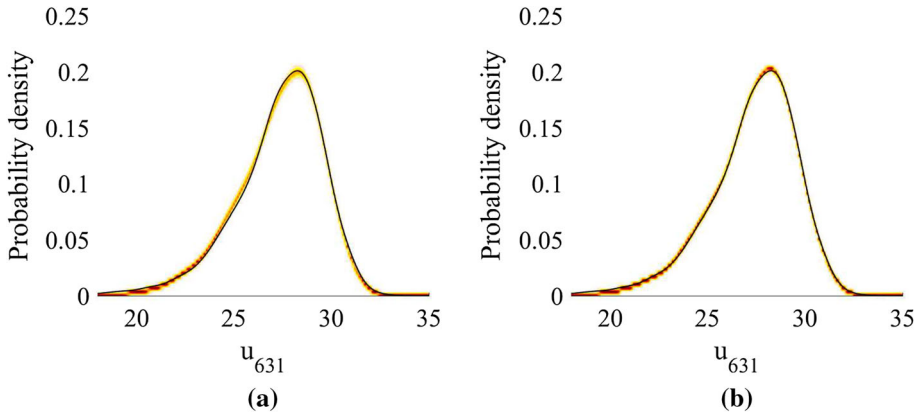


Fig. 14 The pdfs of the pressure head response at the spatial coordinate $\mathbf{x} = \mathbf{x}_{631} = (2511, 486)$ on the finite difference grid, obtained using kernel density estimation on $Q = 5000$ points (Model **M3**). The black line gives the MC prediction using the simulator. The contour shows how the emulator predictions vary with hyperparameter, precision and predictive distribution samples. **a** 300 training points, $P = 70$. **b** 500 training points, $P = 80$

implicit backward Euler time stepping scheme. A picard iteration scheme is used (Celia et al. 1990) at each time step. Details are provided in “Appendix C”.

We followed the procedure of the first example. Training point numbers below 600 led to inaccurate results. For $N = 600$, the results were reasonably accurate but to achieve good accuracy we required $N > 700$. We present the results for $N = 800$. The pressure head is normalized as in the first example in order to highlight the errors in the predictions more clearly. In Fig. 16a we plot the log normalized error $\ln(e_q)$ for an emulator trained on $N = 800$ points \mathbf{y}_n and tested with $Q = 5000$ points $\tilde{\mathbf{y}}_q^*$ for different nearest neighbour numbers $P > 20$ (averaging over hyperparameter and precision posterior samples). For $P \leq 20$ the errors were high, with the same trend as seen in the first example.

We use Algorithm 1 and KDE to obtain predictions of the pdf of a feature of the response. We choose as a feature the pressure head at the location $\mathbf{x} = (10.4, 10.4, 10.4)^T$ (grid point number 4411). The distributions are shown in Fig. 16b for $N = 800$. We can again find the means and standard deviations across predictive posterior, hyperparameter and precision samples to obtain distributions over the moments of the marginalized distribution (32). These are plotted in Fig. 17, alongside comparisons to the true values obtained from $\{\tilde{\mathbf{y}}_q^*\}_{q=1}^Q$. These results show that the emulator performs extremely well, accurately capturing both the mean and standard deviation with high precision.

7 Numerical Computation

LTSA naturally lends itself to parallelization since almost all computations are performed on each neighbourhood independently. After merging threads we need only solve an eigenvalue problem for an $N \times N$ matrix. Similarly, independent Gaussian processes across latent dimensions leads to a natural parallelization framework.

For large sample sizes and feature space dimensions saving each Q_i can become infeasible ($N \times k_y \times k_z$ elements). Similarly, for large sample and neighbourhood sizes saving f can

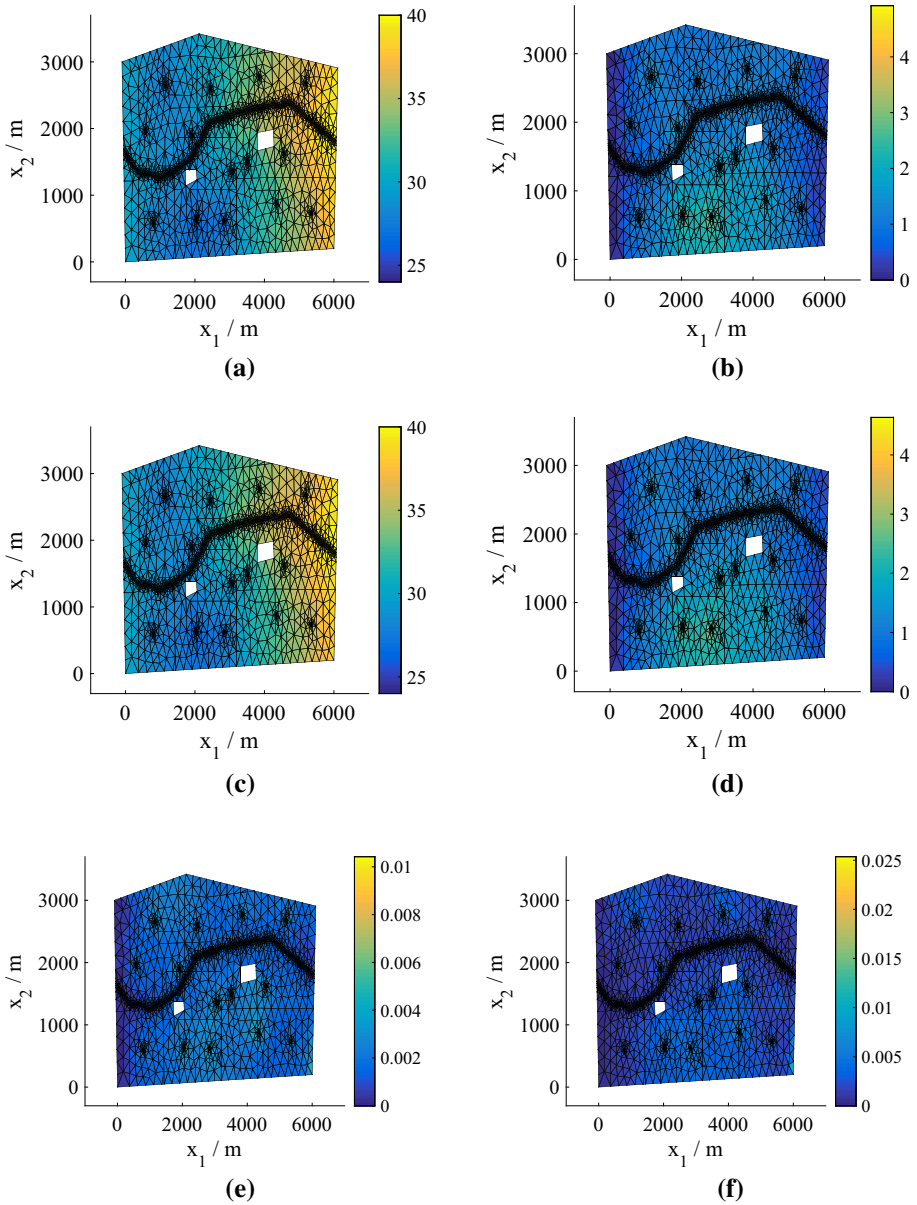


Fig. 15 Moments of the mean and standard deviation for $P = 80$, $N = 500$ of the pressure head (Model M3). The emulator variation is a consequence of the hyperparameter and predictive distribution samples. We have a single, parameterized realization of the manifold. **a** Monte Carlo mean. **b** Monte Carlo standard deviation. **c** Mean of the mean for 500 training points. **d** Mean of the standard deviation for 500 training points. **e** Standard deviation of the mean for 500 training points. **f** Standard deviation of the standard deviation for 500 training points

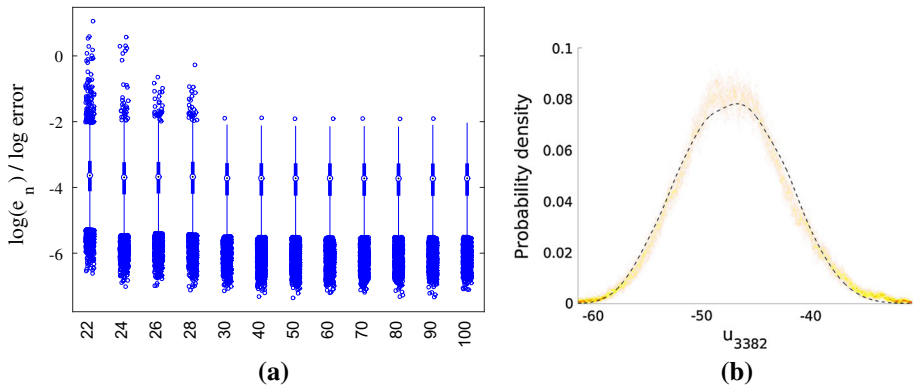


Fig. 16 **a** Log normalized error $\ln(e_q)$ for an emulator trained on $N = 800$ points \mathbf{y}_n and tested with $Q = 5000$ test points $\tilde{\mathbf{y}}_q^*$ for different nearest neighbour numbers P . Predictions were obtained by averaging over hyperparameter and precision posterior samples. **b** The pdfs of the pressure head response at the location $\mathbf{x} = (10.4, 10.4, 10.4)^T$ ($N = 800$), obtained using kernel density estimation on $\{\tilde{\mathbf{y}}_q^*\}_{q=1}^Q$. The black line gives the MC prediction using the simulator. **a** 800 training points. **b** 800 training points, 30 k-NN

become infeasible ($N \times k^2$ elements). In such cases, these tensors may be saved to file or re-calculated online.

The scalability of our approach is limited by the computational complexity of Gaussian processes $\mathcal{O}(N^3)$. However, this can be alleviated by using sparse Gaussian process regression models. These models introduce m inducing points, reducing computational complexity to $\mathcal{O}(m^2N)$. We may also use active learning to reduce the number of samples required.

8 Summary and Conclusions

In this paper we developed a new approach to the emulation of a model involving a random field input and a field output, with a focus on problems arising in groundwater flow modelling. The main challenges are the high input and output space dimensionalities, which we dealt with using a KL expansion and manifold learning, respectively. We implemented LTSA on the given outputs (training data), which allowed us to perform Bayesian inference in a low-dimensional feature space. Furthermore, we developed a framework for UQ in such problems by marginalizing over the inputs, either analytically (the mean and possibly in some cases the standard deviation) or using MC sampling.

Testing the emulation method on two examples reveals that it performs well in certain cases. When the variance of the log-normal input is high or the correlation lengths of the normal process $Z(\mathbf{x})$ are short, the accuracy suffers, as is found in all other approaches. Nevertheless, the accuracy in terms of the forward UQ problem is high even in such cases for the examples considered. (Of course, further increases in the variance and correlation lengths would eventually lead to unacceptably poor performance.)

The major drawback of the KL expansion approach (and similarly with circulant embedding) is the curse of dimensionality as the number of retained coefficients grows. Some progress can be made in this regard by using a Smolyak algorithm (Smolyak 1963) for sampling or incremental local tangent space alignment (Liu et al. 2006) combined with active learning (Settles 2012), but the gains will be limited. Our method, in common with other

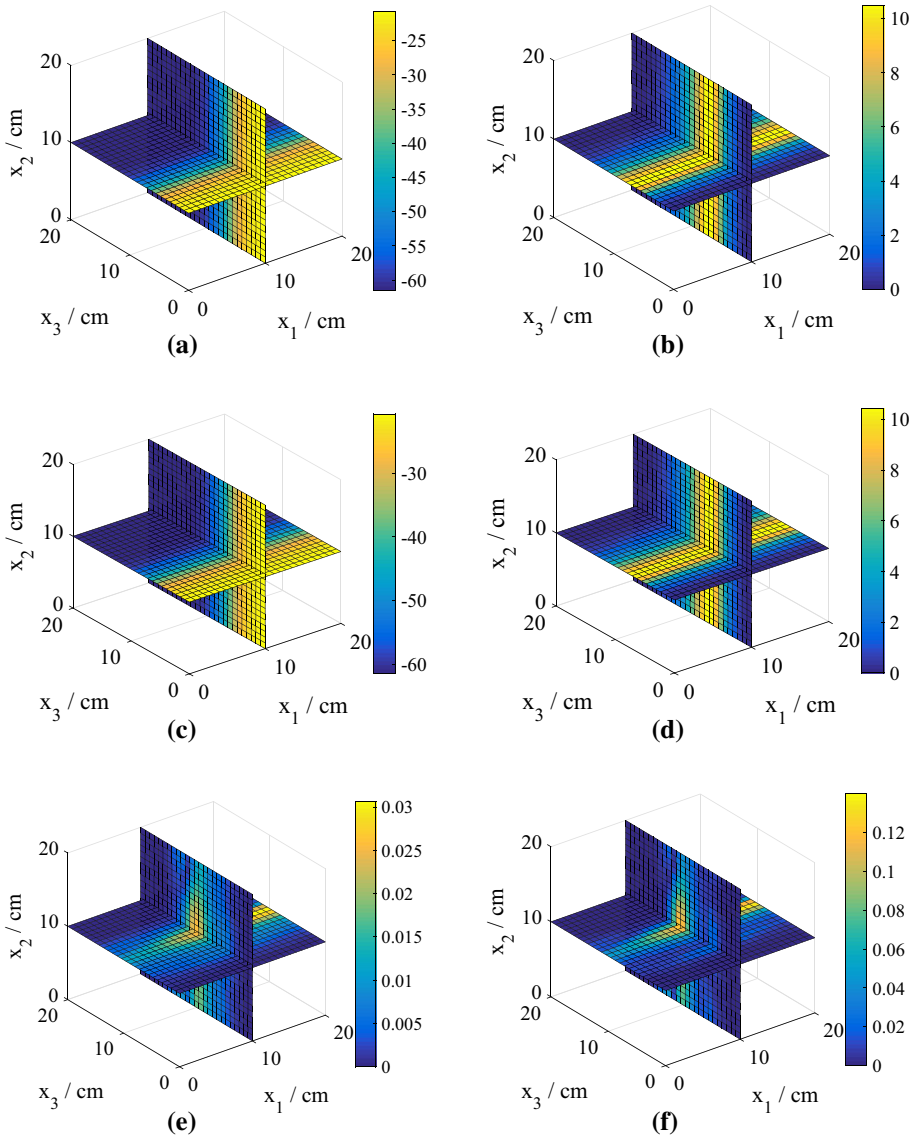


Fig. 17 Moments of the mean and standard deviation of the pressure head for $P = 30, N = 800$. Shown are the planes $x_1 = 10.4$ cm and $x_2 = 10.4$ cm. The emulator variation is a consequence of the hyperparameter and predictive distribution samples. We have a single, parameterized realization of the manifold. **a** Monte Carlo mean. **b** Monte Carlo standard deviation. **c** Mean of the mean. **d** Mean of the standard deviation. **e** Standard deviation of the mean. **f** Standard deviation of the standard deviation

methods except direct Monte Carlo or ROMs, is therefore potentially limited, given current computational resources, to problems in which the domain size is at most a few multiples of the shortest correlation length. The assumption of independence of the feature vector coordinates is also sub-optimal. Since the number of coordinates is small, however, this assumption can easily be relaxed by adopting, e.g. a convolved GP approach.

Appendix A: Moments of the Marginal Distribution Over \mathbf{z}

Focusing on the i th feature of \mathbf{z} , we find the first two moments, i.e. the mean and variance, of the marginal distribution $p(z_i|\mathcal{D}, \boldsymbol{\theta}_i, \beta_i)$. Following Girard and Murray-Smith (2003), we approximate $p(z_i|\mathcal{D}, \boldsymbol{\theta}_i, \beta_i)$ as a Gaussian with mean m and variance v :

$$p(z_i|\mathcal{D}, \boldsymbol{\theta}_i, \beta_i) = \int p(z_i|\boldsymbol{\xi}', \mathcal{D}, \boldsymbol{\theta}_i, \beta_i) p(\boldsymbol{\xi}') d\boldsymbol{\xi}' \approx \mathcal{N}(m, v). \quad (\text{A1})$$

Below we use the notation $\mathbb{E}_\chi[\cdot]$ and $\text{Var}_\chi(\cdot)$ to denote an expectation and variance operator with respect to a random variable χ , respectively. Using Fubini's theorem and the laws of total expectation and variance, the moments are then given by:

$$\begin{aligned} m &= \int z'_i p(z'_i|\mathcal{D}, \boldsymbol{\theta}_i, \beta_i) dz'_i \\ &= \int z'_i \left[\int p(z'_i|\boldsymbol{\xi}', \mathcal{D}, \boldsymbol{\theta}_i, \beta_i) p(\boldsymbol{\xi}') d\boldsymbol{\xi}' \right] dz'_i \\ &= \int \left[\int z'_i p(z'_i|\boldsymbol{\xi}', \mathcal{D}, \boldsymbol{\theta}_i, \beta_i) dz'_i \right] p(\boldsymbol{\xi}') d\boldsymbol{\xi}' \\ &= \mathbb{E}_\boldsymbol{\xi} [\mathbb{E}_{z_i} [z_i|\boldsymbol{\xi}, \mathcal{D}, \boldsymbol{\theta}_i, \beta_i]] \\ &= \mathbb{E}_\boldsymbol{\xi} [\boldsymbol{\mu}(\boldsymbol{\xi})] \\ &= \mathbb{E}_\boldsymbol{\xi} \left[\mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)^T (\mathbf{C}_i + \beta_i^{-1}\mathbf{I})^{-1} \mathbf{z}_{:,i} \right] \\ &= \mathbb{E}_\boldsymbol{\xi} [\mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)]^T (\mathbf{C}_i + \beta_i^{-1}\mathbf{I})^{-1} \mathbf{z}_{:,i} \end{aligned} \quad (\text{A2})$$

and:

$$\begin{aligned} v &= \int (z'_i)^2 p(z'_i|\mathcal{D}, \boldsymbol{\theta}_i, \beta_i) dz'_i - m^2 \\ &= \int (z'_i)^2 \left[\int p(z'_i|\boldsymbol{\xi}', \mathcal{D}, \beta_i) p(\boldsymbol{\xi}') d\boldsymbol{\xi}' \right] dz'_i - m^2 \\ &= \mathbb{E}_\boldsymbol{\xi} [\text{Var}_{z_i} (z_i|\boldsymbol{\xi}, \mathcal{D}, \boldsymbol{\theta}_i, \beta_i)] + \text{Var}_\boldsymbol{\xi} (\mathbb{E}_{z_i} [z_i|\boldsymbol{\xi}, \mathcal{D}, \boldsymbol{\theta}_i, \beta_i]) \\ &= \mathbb{E}_\boldsymbol{\xi} [\sigma^2(\boldsymbol{\xi})] + \text{Var}_\boldsymbol{\xi} (\boldsymbol{\mu}(\boldsymbol{\xi})) \\ &= \mathbb{E}_\boldsymbol{\xi} [\sigma^2(\boldsymbol{\xi})] + \mathbb{E}_\boldsymbol{\xi} [\boldsymbol{\mu}(\boldsymbol{\xi})^2] - m^2 \\ &= \mathbb{E}_\boldsymbol{\xi} \left[\mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i) - \mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)^T (\mathbf{C} + \beta_i^{-1}\mathbf{I})^{-1} \mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i) \right] \\ &\quad + \mathbb{E}_\boldsymbol{\xi} \left[\left(\mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)^T (\mathbf{C}_i + \beta_i^{-1}\mathbf{I})^{-1} \mathbf{z}_{:,i} \right)^2 \right] - m^2 \\ &= \mathbb{E}_\boldsymbol{\xi} [\mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i)] - m^2 \\ &\quad - \left[(\mathbf{C}_i + \beta_i^{-1}\mathbf{I})^{-1} - \left((\mathbf{C}_i + \beta_i^{-1}\mathbf{I}) \mathbf{z}_{:,i} \right)^2 \right] \mathbb{E}_\boldsymbol{\xi} [\mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)^T \mathbf{c}_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)]. \end{aligned} \quad (\text{A3})$$

Appendix B: Kernel Expectation

Given a squared exponential kernel and a Gaussian stochastic input distribution, we are able to analytically find the mean and variance of the marginalized latent predictive distribution. This kernel takes the form:

$$c_h(\boldsymbol{\xi}, \boldsymbol{\xi}'; \boldsymbol{\theta}_i) = s \exp\left(-\frac{1}{2} (\boldsymbol{\xi} - \boldsymbol{\xi}')^T \mathbf{A} (\boldsymbol{\xi} - \boldsymbol{\xi}')\right), \tag{B1}$$

where \mathbf{A} is a diagonal matrix whose elements are inversely proportional to the correlation lengths across input dimensions. For computational convenience, we write this covariance function in Gaussian function form with normalizing constant $a = (2\pi)^{k_\xi/2} |\mathbf{A}|^{\frac{1}{2}} s$:

$$c_h(\boldsymbol{\xi}, \boldsymbol{\xi}'; \boldsymbol{\theta}_i) = a \mathcal{N}_\xi(\boldsymbol{\xi}', \mathbf{A}). \tag{B2}$$

where the notation $\mathcal{N}_\chi(\cdot, \cdot)$ denotes a normal distribution over a random vector $\boldsymbol{\chi}$, with mean and covariance matrix given by the first and second arguments, respectively. We wish to evaluate:

$$\begin{aligned} \mathbb{E}_\xi [c_h(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i)] &= a, \\ \mathbb{E}_\xi [c_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)] &= \mathbb{E}_\xi [c_h(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i)] = a \int \mathcal{N}_\xi(\boldsymbol{\xi}, \mathbf{A}) \mathcal{N}_\xi(\boldsymbol{\mu}, \Sigma_\xi) d\boldsymbol{\xi}, \\ \mathbb{E}_\xi [c_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)^T c_h(\boldsymbol{\xi}, \boldsymbol{\Xi}; \boldsymbol{\theta}_i)] &= \mathbb{E}_\xi [c_h(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i) c_h(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i)] \\ &= a^2 \int \mathcal{N}_\xi(\boldsymbol{\xi}, \mathbf{A}) \mathcal{N}_\xi(\boldsymbol{\xi}, \mathbf{A}) \mathcal{N}_\xi(\boldsymbol{\mu}, \Sigma_\xi) d\boldsymbol{\xi}, \end{aligned} \tag{B3}$$

where $(\boldsymbol{\mu}, \Sigma_\xi)$ are the stochastic input distribution moments. The solutions can be found by using the product of Gaussians rule:

$$\begin{aligned} \mathbb{E}_\xi [c_h(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i)] &= a \mathcal{N}_\mu(\boldsymbol{\xi}, \mathbf{A} + \Sigma_\xi), \\ \mathbb{E}_\xi [c_h(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i) c_h(\boldsymbol{\xi}, \boldsymbol{\xi}; \boldsymbol{\theta}_i)] &= a^2 \mathcal{N}_\xi(\boldsymbol{\xi}, 2\mathbf{A}) \mathcal{N}_\mu\left(\boldsymbol{\xi}, \Sigma_\xi + \frac{\mathbf{A}}{2}\right). \end{aligned} \tag{B4}$$

Appendix C: Numerical Algorithm for Richards Equation

Let $\psi_{i',j',k'}^{n',m'}$ denote the value of a quantity ψ at time step n' (time $t = n' \Delta t$ for a constant time step Δt), at Picard iteration m' and at the spatial location $x_1 = i' \Delta x_1$, $x_2 = j' \Delta x_2$ and $x_3 = k' \Delta x_3$. The spatial and temporal discretizations lead to:

$$\begin{aligned} a_1 h_{i-1,j,k}^{n+1,m+1} + b h_{i,j,k}^{n+1,m+1} + c_1 h_{i+1,j,k}^{n+1,m+1} + a_2 h_{i,j-1,k}^{n+1,m+1} + c_2 h_{i,j+1,k}^{n+1,m+1} \\ + a_3 h_{i,j,k-1}^{n+1,m+1} + c_3 h_{i,j,k+1}^{n+1,m+1} = d, \end{aligned} \tag{C1}$$

which is applicable to all interior nodes (grid points), and where:

$$\begin{aligned}
 a_1 &= -\frac{k_{i,j,k}^{n+1,m} + k_{i-1,j,k}^{n+1,m}}{2\Delta x_1^2}, & a_2 &= -\frac{k_{i,j,k}^{n+1,m} + k_{i,j-1,k}^{n+1,m}}{2\Delta x_2^2}, & a_3 &= -\frac{k_{i,j,k}^{n+1,m} + k_{i,j,k-1}^{n+1,m}}{2\Delta x_3^2} \\
 b &= \frac{u_{i,j,k}^{n+1,m}}{\Delta t} + \frac{k_{i+1,j,k}^{n+1,m} + 2k_{i,j,k}^{n+1,m} + k_{i-1,j,k}^{n+1,m}}{2\Delta x_1^2} \\
 &\quad + \frac{k_{i,j+1,k}^{n+1,m} + 2k_{i,j,k}^{n+1,m} + k_{i,j-1,k}^{n+1,m}}{2\Delta x_2^2} + \frac{k_{i,j,k+1}^{n+1,m} + 2k_{i,j,k}^{n+1,m} + k_{i,j,k-1}^{n+1,m}}{2\Delta x_3^2} \\
 c_1 &= -\frac{k_{i,j,k}^{n+1,m} + k_{i+1,j,k}^{n+1,m}}{2\Delta x_1^2}, & c_2 &= -\frac{k_{i,j,k}^{n+1,m} + k_{i,j+1,k}^{n+1,m}}{2\Delta x_2^2}, & c_3 &= -\frac{k_{i,j,k}^{n+1,m} + k_{i,j,k+1}^{n+1,m}}{2\Delta x_3^2} \\
 d &= \frac{-k_{i,j,k+1}^{n+1,m} + k_{i,j,k-1}^{n+1,m}}{2\Delta x_3} + u_{i,j,k}^{n+1,m} \frac{h_{i,j,k}^n}{\Delta t}
 \end{aligned} \tag{C2}$$

The CSC approximation (Rathfelder and Abriola 1994) yields $u_{i,j,k}^{n+1,m} = (\theta_{i,j,k}^{n+1,m} - \theta_{i,j,k}^n)/(h_{i,j,k}^{n+1,m} - h_{i,j,k}^n)$. In matrix form, the system of Eq. (C1) can be written as:

$$\mathbf{A}(\mathbf{h}^{n+1,m})\mathbf{h}^{n+1,m+1} = \mathbf{a}(\mathbf{h}^{n+1,m}) \tag{C3}$$

where $\mathbf{h}^{n+1,m'} \in \mathbb{R}^{k_y}$ is a vector of values of $h_{i,j,k}^{n+1,m'}$, $i = 1, \dots, n_1$, $j = 1, \dots, n_2$, $k = 1, \dots, n_3$. $\mathbf{A} \in \mathbb{R}^{k_y \times k_y}$ and $\mathbf{a} \in \mathbb{R}^{k_y}$ depend only on values of the head at iteration m . Thus, the system (C3) is linear in $\mathbf{h}^{n+1,m+1}$. It can be solved by iterating (in m) within each time step n until convergence; that is, for each time step n , m is incremented until the residual satisfies $\|\mathbf{A}(\mathbf{h}^{n+1,m+1})\mathbf{h}^{n+1,m+1} - \mathbf{a}(\mathbf{h}^{n+1,m+1})\| < \varepsilon$ for some specified tolerance ε . In the results presented in Sect. 6.2, we use $n_1 = n_2 = n_3 = 26$ ($\Delta x_1 = \Delta x_2 = \Delta x_3 = 0.8$ cm), $\Delta t = 0.5$ s and $\varepsilon = 0.01$.

Acknowledgements CG would like to acknowledge the Warwick Centre for Predictive Modelling for a Ph.D. scholarship. AS would like to acknowledge the EPSRC, UK, for financial support (Grant No. EP/P012620/1).

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Al-Tabbaa, A., Ayotamuno, J., Martin, R.: One-dimensional solute transport in stratified sands at short travel distances. *J. Hazard. Mater.* **73**(1), 1–15 (2000)
- Aly, A.H., Peralta, R.C.: Optimal design of aquifer cleanup systems under uncertainty using a neural network and a genetic algorithm. *Water Resour. Res.* **35**(8), 2523–2532 (1999)
- Ataie-Ashtiani, B., Ketabchi, H., Rajabi, M.M.: Optimal management of a freshwater lens in a small island using surrogate models and evolutionary algorithms. *J. Hydrol. Eng.* **19**(2), 339–354 (2014)
- Babuška, I., Nobile, F., Tempone, R.: A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Numer. Anal.* **45**(3), 1005–1034 (2007)
- Barry, D., Prommer, H., Miller, C., Engesgaard, P., Brun, A., Zheng, C.: Modelling the fate of oxidisable organic contaminants in groundwater. *Adv. Water Resour.* **25**(8), 945–983 (2002)
- Bau, D.A., Mayer, A.S.: Stochastic management of pump-and-treat strategies using surrogate functions. *Adv. Water Resour.* **29**(12), 1901–1917 (2006)
- Betz, W., Papaioannou, I., Straub, D.: Numerical methods for the discretization of random fields by means of the Karhunen–Loève expansion. *Comput. Methods Appl. Mech. Eng.* **271**, 109–129 (2014)

- Bhattacharjya, R.K., Datta, B.: Optimal management of coastal aquifers using linked simulation optimization approach. *Water Resour. Manag.* **19**(3), 295–320 (2005)
- Boi, D., Stankovi, V., Gorgievski, M., Bogdanovi, G., Kovaevi, R.: Adsorption of heavy metal ions by sawdust of deciduous trees. *J. Hazard. Mater.* **171**(1), 684–692 (2009)
- Borgonovo, E., Castaings, W., Tarantola, S.: Model emulation and moment-independent sensitivity analysis: an application to environmental modelling. *Environ. Model. Softw.* **34**, 105–115 (2012)
- Celia, M.A., Ahuja, L.R., Pinder, G.F.: Orthogonal collocation and alternating-direction procedures for unsaturated flow problems. *Adv. Water Resour.* **10**(4), 178–187 (1987)
- Celia, M.A., Bouloutas, E.T., Zarba, R.L.: A general mass-conservative numerical solution for the unsaturated flow equation. *Water Resour. Res.* **26**(7), 1483–1496 (1990)
- Conti, S., O'Hagan, A.: Bayesian emulation of complex multi-output and dynamic computer models. *J. Stat. Plan. Inference* **140**(3), 640–651 (2010)
- Curran, C., Mitchell, T., Morris, M., Ylvisaker, D.: A Bayesian approach to the design and analysis of computer experiments, Tech. rep., ORNL-6498, Oak Ridge National Laboratory (1988)
- Feyen, J., Jacques, D., Timmerman, A., Vanderborght, J.: Modelling water flow and solute transport in heterogeneous soils: a review of recent approaches. *J. Agric. Eng. Res.* **70**(3), 231–256 (1998)
- Foo, K., Hameed, B.: An overview of landfill leachate treatment via activated carbon adsorption process. *J. Hazard. Mater.* **171**(1), 54–60 (2009)
- Fu, J., Gomez-Hernandez, J.J.: Uncertainty assessment and data worth in groundwater flow and mass transport modeling using a blocking Markov chain Monte Carlo method. *J. Hydrol.* **364**(3), 328–341 (2009)
- Gelhar, L.W.: Stochastic subsurface hydrology from theory to applications. *Water Resour. Res.* **22**(9S), 135S–145S (1986)
- Gelhar, L.W., Axness, C.L.: Three-dimensional stochastic analysis of macrodispersion in aquifers. *Water Resour. Res.* **19**(1), 161–180 (1983)
- Ghanem, R.G., Spanos, P.D.: *Stochastic Finite Elements: A Spectral Approach*. Springer, New York (2003)
- Girard, A., Murray-Smith, R.: Gaussian processes: prediction at a noisy input and application to iterative multiple-step ahead forecasting of time-series, pp. 546–551. *Lecture Notes in Computer Science*, Springer, Switching and Learning in Feedback Systems (2003)
- Haverkamp, R., Vauclin, M., Touma, J., Wierenga, P., Vachaud, G.: A comparison of numerical simulation models for one-dimensional infiltration. *Soil Sci. Soc. Am. J.* **41**(2), 285–294 (1977)
- Hemker, T., Fowler, K.R., Farthing, M.W., von Stryk, O.: A mixed-integer simulation-based optimization approach with surrogate functions in water resources management. *Optim. Eng.* **9**(4), 341–360 (2008)
- Herckenrath, D., Langevin, C.D., Doherty, J.: Predictive uncertainty analysis of a saltwater intrusion model using null-space Monte Carlo. *Water Resour. Res.* **47**(5), W05504 (2011)
- Higdon, D., Gattiker, J., Williams, B., Rightley, M.: Computer model calibration using high-dimensional output. *J. Am. Stat. Assoc.* **103**(482), 570–583 (2008)
- Huang, K., Mohanty, B., Van Genuchten, M.T.: A new convergence criterion for the modified Picard iteration method to solve the variably saturated flow equation. *J. Hydrol.* **178**(1–4), 69–91 (1996)
- Karatzas, G.P.: Developments on modeling of groundwater flow and contaminant transport. *Water Resour. Manag.* **31**(10), 3235–3244 (2017)
- Ketabchi, H., Ataie-Ashtiani, B.: Review: coastal groundwater optimization—advances, challenges, and practical solutions. *Hydrogeol. J.* **23**(6), 1129–1154 (2015)
- Kourakos, G., Harter, T.: Parallel simulation of groundwater non-point source pollution using algebraic multi-grid preconditioners. *Comput. Geosci.* **18**(5), 851–867 (2014)
- Kourakos, G., Mantoglou, A.: Pumping optimization of coastal aquifers based on evolutionary algorithms and surrogate modular neural network models. *Adv. Water Resour.* **32**(4), 507–521 (2009)
- Kourakos, G., Klein, F., Cortis, A., Harter, T.: A groundwater nonpoint source pollution modeling framework to evaluate long-term dynamics of pollutant exceedance probabilities in wells and other discharge locations. *Water Resour. Res.* **48**(6), W00L13 (2012)
- Kristensen, A.H., Poulsen, T.G., Mortensen, L., Moldrup, P.: Variability of soil potential for biodegradation of petroleum hydrocarbons in a heterogeneous subsurface. *J. Hazard. Mater.* **179**(1), 573–580 (2010)
- Li, H., Teng, L., Chen, W., Shen, I.-F.: Supervised learning on local tangent space. In: *Advances in Neural Networks-ISNN*, Lecture Notes in Computer Science, vol. 2005, pp. 546–551. Springer (2005)
- Liu, X., Yin, J., Feng, Z., Dong, J.: Incremental manifold learning via tangent space alignment. In: Schwenker, F., Marinai, S. (eds.) *Artificial Neural Networks in Pattern Recognition*, pp. 107–121. Springer, Berlin (2006)
- Ma, X., Zabarar, N.: An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. *J. Comput. Phys.* **228**(8), 3084–3113 (2009)
- Ma, X., Zabarar, N.: Kernel principal component analysis for stochastic input model generation. *J. Comput. Phys.* **230**(19), 7311–7331 (2011)

- Maxwell, R.M., Welty, C., Harvey, R.W.: Revisiting the cape cod bacteria injection experiment using a stochastic modeling approach. *Environ. Sci. Technol.* **41**(15), 5548–5558 (2007)
- Nobile, F., Tempone, R., Webster, C.G.: A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.* **46**(5), 2309–2345 (2008)
- O'Hagan, A., Kingman, J.F.C.: Curve fitting and optimal design for prediction. *J. R. Stat. Soc. Ser. B (Methodological)* **40**, 1–42 (1978)
- Paleologos, E.K., Avaniidou, T., Mylopoulos, N.: Stochastic analysis and prioritization of the influence of parameter uncertainty on the predicted pressure profile in heterogeneous, unsaturated soils. *J. Hazard. Mater.* **136**(1), 137–143 (2006)
- Rajabi, M.M., Ataie-Ashtiani, B., Simmons, C.T.: Polynomial chaos expansions for uncertainty propagation and moment independent sensitivity analysis of seawater intrusion simulations. *J. Hydrol.* **520**, 101–122 (2015)
- Rathfelder, K., Abriola, L.M.: Mass conservative numerical solutions of the head-based Richards equation. *Water Resour. Res.* **30**(9), 2579–2586 (1994)
- Ray, R., Mohanty, B.: Some Numerical Investigations of the Richards' Equation, ASAE Paper 92-2586. American Society of Agricultural Engineers, St Joseph (1992)
- Razavi, S., Tolson, B.A., Burn, D.H.: Review of surrogate modeling in water resources. *Water Resour. Res.* **48**(7), W07401 (2012)
- Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P.: Design and analysis of computer experiments. *Stat. Sci.* **4**(4), 409–423 (1989)
- Schfer, D., Schfer, W., Kinzelbach, W.: Simulation of reactive processes related to biodegradation in aquifers. *J. Contam. Hydrol.* **31**(1), 167–186 (1998)
- Settles, B.: Active learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **6**(1), 1–114 (2012)
- Shahraiyini, H.T., Ataie-Ashtiani, B.: Mathematical forms and numerical schemes for the solution of unsaturated flow equations. *J. Irrig. Drain. Eng.* **138**(1), 63–72 (2011)
- Simonoff, J.S.: *Smoothing Methods in Statistics*. Springer, New York (1996)
- Smolyak, S.A.: Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl. Akad. Nauk SSSR* **4**, 240–243 (1963)
- Sreekanth, J., Datta, B.: Coupled simulation-optimization model for coastal aquifer management using genetic programming-based ensemble surrogate models and multiple-realization optimization. *Water Resour. Res.* **47**(4), w04516 (2011)
- Sreekanth, J., Datta, B.: Comparative evaluation of genetic programming and neural network as potential surrogate models for coastal aquifer management. *Water Resour. Manag.* **25**(13), 3201–3218 (2011)
- Sreekanth, J., Datta, B.: Stochastic and robust multi-objective optimal management of pumping from coastal aquifers under parameter uncertainty. *Water Resour. Manag.* **28**(7), 2005–2019 (2014)
- Vomvoris, E.G., Gelhar, L.W.: Stochastic analysis of the concentration variability in a three-dimensional heterogeneous aquifer. *Water Resour. Res.* **26**(10), 2591–2602 (1990)
- Wan, X., Karniadakis, G.E.: A sharp error estimate for the fast Gauss transform. *J. Comput. Phys.* **219**(1), 7–12 (2006)
- Wei, J., Peng, H., Lin, Y.-S., Huang, Z.-M., Wang, J.-B.: Adaptive neighborhood selection for manifold learning. In: *Machine Learning and Cybernetics, 2008 International Conference on*, vol. 1, pp. 380–384. IEEE (2008)
- Wong, E.: *Stochastic Processes in Information and Dynamical Systems*. McGraw-Hill, New York (1971)
- Xing, W., Shah, A.A., Nair, P.B.: Reduced dimensional Gaussian process emulators of parametrized partial differential equations based on isomap. *Proc. R. Soc. Lond. A* **471**(2174), 20140697 (2015)
- Xing, W., Triantafyllidis, V., Shah, A., Nair, P., Zabarar, N.: Manifold learning for the emulation of spatial fields from computational models. *J. Comput. Phys.* **326**, 666–690 (2016)
- Xiu, D.: Efficient collocation approach for parametric uncertainty analysis. *Commun. Comput. Phys.* **2**(2), 293–309 (2007)
- Xiu, D., Hesthaven, J.S.: High-order collocation methods for differential equations with random inputs. *SIAM J. Sci. Comput.* **27**(3), 1118–1139 (2005)
- Xiu, D., Karniadakis, G.E.: The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.* **24**(2), 619–644 (2002)
- Zarba, R.L., Bouloutas, E., Celia, M.: General mass-conservative numerical solution for the unsaturated flow equation. *Water Resour. Res. WRERAQ* **26**(7), 1483–1496 (1990)
- Zhan, Y., Yin, J.: Robust local tangent space alignment via iterative weighted PCA. *Neurocomputing* **74**(11), 1985–1993 (2011)
- Zhang, D., Lu, Z.: An efficient, high-order perturbation approach for flow in random porous media via Karhunen–Loève and polynomial expansions. *J. Comput. Phys.* **194**(2), 773–794 (2004)

- Zhang, Z., Zha, H.: Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM J. Sci. Comput.* **26**(1), 313–338 (2004)
- Zhang, Z., Wang, J., Zha, H.: Adaptive manifold learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(2), 253–265 (2012)
- Zou, X., Zhu, Q.: Adaptive neighborhood graph for LTSA learning algorithm without freeparameter. *Int. J. Comput. Appl.* **19**(4), 28–33 (2011)