Gaussian Process Modelling for Uncertainty Quantification in Convectively-Enhanced Dissolution Processes in Porous Media

D. Crevillén-García[a,*], R.D. Wilkinson[b], A.A. Shah[a], and H. Power[c]

[a]*School of Engineering, University of Warwick, Coventry CV4 7AL, UK*
[b]*School of Mathematics and Statistics, University of Sheffield, Sheffield S3 7RH, UK*
[c]*Faculty of Engineering, University of Nottingham, Nottingham NG7 2RD, UK*

Abstract

Numerical groundwater flow and dissolution models of physico-chemical processes in deep aquifers are usually subject to uncertainty in one or more of the model input parameters. This uncertainty is propagated through the equations and needs to be quantified and characterised in order to rely on the model outputs. In this paper we present a Gaussian process emulation method as a tool for performing uncertainty quantification in mathematical models for convection and dissolution processes in porous media. One of the advantages of this method is its ability to significantly reduce the computational cost of an uncertainty analysis, while yielding accurate results, compared to classical Monte Carlo methods. We apply the methodology to a model of convectively-enhanced dissolution processes occurring during carbon capture and storage. In this model, the Gaussian process methodology fails due to the presence of multiple branches of solutions emanating from a bifurcation point, i.e., two equilibrium states exist rather than one. To overcome this issue we use a classifier as a precursor to the Gaussian process emulation, after which we are able to successfully perform a full uncertainty analysis in the vicinity of the bifurcation point.

Keywords: Convectively-enhanced dissolution, Partial differential equations with random inputs, Multiple solutions and bifurcation, Gaussian process emulation and classification, Uncertainty quantification.

[*] Corresponding author. Tel: +44 (0)24 7652 8193. Fax: +44 (0)24 7641 8922. *E-mail address*: D.Crevillen-Garcia@warwick.ac.uk

1. Introduction

Geological storage of CO_2 in deep saline aquifers is a potential way of limiting greenhouse gas emissions to the atmosphere while continuing the use of fossil fuels. The major spreading and trapping mechanisms of CO_2 in geological media are subject to spatial variability due to heterogeneity of the physical and chemical properties of the medium. Heterogeneity affects the multi-phase flow properties of the CO_2-brine system and can lead to trapping of brine behind the CO_2 phase as well as increased spread of the CO_2-brine interface [8]. These heterogeneity-induced processes increase the CO_2-brine contact area and thus the dissolution efficiency of CO_2. Dissolution of CO_2 into the resident brine of the storage site causes an increase in the mixture density, which promotes the sinking of the enriched brine-CO_2 mixture, and results in an enhancement of the dissolution process due to mixing and recirculation. This process is known as convectively-enhanced dissolution (C-ED). The mixing of the denser CO_2 rich water with the reservoir water is dominated by dispersion and the interaction with spatial heterogeneity as well as buoyancy effects. The efficiency of the chemical reactions, which are controlled by mass transfer limitations and interaction with the medium, is also affected by spatial heterogeneity in the properties of the medium [17]. While heterogeneity can lead to increased spreading and mixing of waters with different chemical compositions, chemical reaction rates for heterogeneous media can be much lower than those in a laboratory setting (homogeneous conditions) [18].

To the best of our knowledge, there are only a limited number of publications in the literature (e.g., [24]) dealing with multi-phase flow and reactive transport models of CO_2 sequestration and the associated trapping process, taking into account the impact of heterogeneity on front spreading and mass transfer between high and low permeability zones of the heterogeneous medium, together with the impact of physical and chemical heterogeneity on the chemical reactions. Most of the literature on enhanced dissolution in CO_2 sequestration is concerned with the time evolution of the process and not with the analysis of the non-uniqueness of solutions, the corresponding bifurcation map and the stability of solutions. Moreover, in most of these works only molecular diffusion is considered (e.g., [44]), chemical reaction is neglected (e.g., [28]) and the porous medium is taken as homogeneous (e.g., [26, 56]). In this paper we target the problem of uncertainty quantification in the C-ED mechanism due to the medium heterogeneity, *and* taking into account dispersivity and chemical reaction, by considering possible stable (physically feasible) solutions on a bifurcation map, i.e., the long term evolution.

The impact of medium heterogeneity could be critical; for instance, for assessing the long-term fate of the geological storage of CO_2 during carbon capture and storage (CCS). One of the main aims of this paper is to develop computationally feasible methods for investigating the effects of dispersivity in a porous medium, i.e., that the complex microscopic flow in a porous medium leads to an apparent macroscopic dispersive transport, enhancing molecular diffusion. To date, the quantification of uncertainty arising from heterogeneities in rock properties and temporal fluctuations has not received much attention in large scale CO_2 sequestration modelling. The few studies conducted have used classical Monte Carlo (MC) methods and have required enormous computational resources due to the slow rate of MC convergence: The error scales according $N^{-1/2}$, where N is the number of MC samples [13].

The aforementioned uncertainties, normally represented as random inputs within the systems of partial differential equations (PDEs), may have limited impact on the outputs of interest. If this can be established, it is not necessary to incorporate them explicitly in the model. Those sources of uncertainty that do have a significant impact, must, on the other hand, be taken into account. This is a challenging problem. For instance, for the large-scale, time-dependent simulations that must be carried out when investigating a CO_2 storage site, classical MC simulation will be impractical unless considerable computing resources are available. Even if such resources are available, they could be better deployed if more efficient methods are available to solve the random-input model, e.g., investigating alternative conceptual models and models that incorporate more detailed physical phenomena.

One method for overcoming the computational resource limitations is to build a statistical surrogate model (or 'emulator') for the computer model (the 'simulator'). In this approach, an approximate mapping between the inputs and outputs is established using a supervised machine learning method (such as Gaussian process (GP) regression) based on the outputs of the simulator (training points) from a limited number of simulations. The emulator can then be used as a replacement for the full simulator in, for instance, a classical MC calculation.

A detailed review of surrogate modelling in groundwater flow modelling is provided in [5]. They split surrogates into three main categories. (a) Data-driven surrogates. These involve empirical approximations of the complex model output calibrated on a set of inputs and outputs of the complex model, for instance neural networks [60], Gaussian processes [31, 6], and polynomial chaos expansions [32]. (b) Projection-based methods. The governing equation(s) or the spatially discretised system is projected onto a low-dimensional subspace spanned by a set of orthonormal modes. The most prominent methods are proper orthogonal decomposition [36] and Krylov-subspace methods [21], both of which are used to define the approximating basis. Karhunen-Loéve (KL) expansions of input fields such as the permeability [32] can also be considered in this category. (c) Multifidelity (or hierarchical) based surrogates. These are built by simplifying the underlying physics or reducing the numerical resolution (coarse grids or relaxed error tolerances). They include multigrid methods [4] and multiscale finite element methods [30].

In addition to the comprehensive model we consider, the work proposed in this paper combines a novel GP emulation/classification with a KL decomposition to perform uncertainty quantification (UQ) in a C-ED model with random inputs, where the main source of uncertainty is considered to be the heterogeneity of the porous rock formation. The model presented here has an infinite number of stochastic degrees of freedom and will be approximated by a finite number of degrees of freedom using KL decompositions.

The appearance of multiple solutions around a bifurcation point in the C-ED model (located using arclength continuation), however, poses challenges for any emulator of the simulator outputs. GP emulation alone is found to be inaccurate. To overcome the difficulties, we develop an emulation/classification approach that allows us to label (or classify) the variety of outputs for a given input. Once the output has been classified we can apply GP emulation individually to each of the classes. A further issue is that the GP emulation training is impractical for the original high dimensional input space (the number of nodes in the numerical formulation), involving an optimisation over a very large number of hyperparameters. Thus, we exploit the properties of the KL decomposition to develop computationally efficient emulators.

The outline of this paper is as follows. In Section 2 we present the equations with random inputs for the model problem, the C-ED process. We present the numerical method used to solve the system of PDEs with random inputs. In Section 3 we describe the GP emulation methodology and its application to the model problem, including the combination GP emulation and classification to reduce the uncertainty in regions where the numerical model returns multiple outputs for a given input. In Section 4 we define the quantity of interest and present our numerical results. Concluding remarks are provided in Section 5, together with suggestions for future work.

2. Mathematical model: Surface flux in C-ED processes in porous media

In this section we present the C-ED simulator. The goal of this work is to build a statistical surrogate model for the simulator which will allow us to perform a full uncertainty analysis on the simulator outputs.

During CCS processes, the CO_2 is trapped in saline geological formations located deep underground. Once there, it reacts with minerals in the geologic formation, leading to the precipitation of a secondary carbonate mineral [26]. When the CO_2 is dissolved into the brine, the density of the resulting solution is higher than that of brine, and this density difference can engender flow instabilities that lead to the formation of CO_2 downward growing plumes with finger structures. The descending mixed fluid along the fingers will induce recirculation cells of brine fluid with an associated fluid entrainment into the fingers. The entrained brine reduces the density difference and the CO_2 concentration at the diffusive boundary layer, resulting in enhancement of the dissolution process (see e.g., [41, 56]).

In this paper, the dissolution flux will be characterized by a generalization of the Sherwood number (see, e.g., [44, 56]), which is a dimensionless measure of the convective flux across the upper boundary of the domain. We call this quantity the *surface flux* ($\mathcal{S}$). We will consider the effect of the porous rock heterogeneity on the C-ED process, in which the solute undergoes a first order chemical reaction, by investigating how $\mathcal{S}$ is affected by rock heterogeneities. Thus, the main goal of this section will be to *quantify the uncertainty in the* CO_2 *dissolution flux into the brine due to uncertainty in the rock morphology*.

The complex microscopic flow in a porous medium leads to an apparent macroscopic dispersive transport, enhancing the molecular diffusion. We model this phenomenon by an apparent dispersion tensor, $\mathbf{D}$, dependant on the local Darcy velocity of the fluid $\mathbf{u}$ (see e.g., [49, 51]) as follows:

$$\mathbf{D} = D_m\mathbb{I} + \beta_T||\mathbf{u}||\mathbb{I} + (\beta_L - \beta_T)\frac{\mathbf{u}\otimes\mathbf{u}}{||\mathbf{u}||}, \tag{1}$$

where $\otimes$ represents the tensor product, $\mathbb{I}$ is the unit (identity) tensor, D_m is the molecular diffusion coefficient of the solute in the fluid, and β_L and β_T are respectively the longitudinal and transverse dispersion coefficients, which satisfy $\beta_L \geq \beta_T \geq 0$. We use the rule-of-thumb $\beta_T = \beta_L/10$ suggested in [28], which was based on an analysis by Gelhar *et al.* [23] of measurements from different field sites; realistic ranges of values for the dispersivity coefficients β_T and β_L obtained from these field sites are provided in [23]. A detailed discussion on the influence that these coefficients have on the enhanced dissolution process is given in [47].

We consider the dissolution of a solute (CO_2) in a fluid (brine) flowing in a two-dimensional domain representing a heterogeneous, isotropic porous medium of depth 2H and length L. The spatial variable is denoted $\mathbf{x} = (x, z)$ on the domain $[0, L] \times [-H, H]$. The governing equations used to describe the C-ED process are continuity (2), Darcy's law (3) and convection-diffusion-reaction (4) (for more details see [44, 56, 58]):

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

$$\mathbf{u} = -\frac{K}{\mu}(\nabla P + \rho g\mathbf{e}_z), \tag{3}$$

$$\phi\frac{\partial C}{\partial t} + \mathbf{u}\cdot\nabla C = \phi\nabla\cdot(\mathbf{D}\nabla C) - \gamma_c C. \tag{4}$$

In these equations, $\mathbf{e}_z$ is the outward-pointing unit vector along the ordinate axis, C is the concentration of dissolved CO_2, $\mathbf{u} = (u_x, u_z)$ is the liquid velocity and P is the liquid pressure. The parameters K, μ, ϕ, γ_c and $\mathbf{g}$ are the medium permeability field, the fluid viscosity, the rock porosity, the reaction rate and acceleration due to gravity.

In the above system of governing equations, the solute locally increases the solution density of the fluid, and the linearised density of the fluid takes the form, $\rho = \rho_0 + \beta_c C$, where ρ_0 and β_c are the density of the pure fluid and the volumetric expansion coefficient. The change in density is small, which enables us to use the Boussinesq approximation [56]. Moreover, the solute undergoes a first order reaction and is converted into an inert product with no effect on the solution density. The above system of PDEs is required to satisfy the following boundary conditions:

$$\begin{gathered} C(x, z=H) = C_0, \\ u_x(0,z) = u_x(L,z) = u_z(x,\pm H) = 0, \\ \frac{\partial C}{\partial z}(x,-H) = \frac{\partial C}{\partial x}(0,z) = \frac{\partial C}{\partial x}(L,z) = 0. \end{gathered} \tag{5}$$

By representing the velocity field using a stream function formulation, $u_x = \partial\Psi/\partial z$ and $u_z = -\partial\Psi/\partial x$, where Ψ is the streamfunction, it is possible to eliminate the pressure field from the governing equations, resulting in a new set of equations for the unknown field variables (Ψ, C). The resulting set of governing equations and boundary conditions can be written in a dimensionless form by defining:

$$(x', z') = \frac{(x,z)}{H}, \quad \Psi' = \frac{\Psi\mu}{HC_0K_0\beta_c g}, \quad C' = \frac{C}{C_0}, \quad t' = \frac{tC_0K_0\beta_c\rho}{\mu\phi H}, \tag{6}$$

$$(\beta'_L, \beta'_T) = \frac{(\beta_L, \beta_T)C_0K_0\beta_c g}{D_0\mu}, \quad K' = \frac{K}{K_0}, \quad \mathcal{L} = \frac{L}{H}, \tag{7}$$

$$Ra = \frac{K_0C_0g\beta_cH}{\phi\mu D_0}, \quad Da = \frac{\gamma_c\mu H}{K_0C_0g\beta_c}. \tag{8}$$

β_T and β_L are respectively the longitudinal and transverse dispersion coefficients [58, 28]; K_0 and D_0 are reference permeability and diffusion coefficients, respectively; $\mathcal{L}$ is the aspect ratio of the domain; Ra is the Rayleigh number, related to the buoyancy driven flow; and Da is the Damkhöler number, which is the ratio of the chemical reaction rate to the mass transfer rate [56]. In terms of these dimensionless variables and numbers, the following dimensionless governing equations are obtained (where for convenience the primes have been dropped):

$$\left.\begin{aligned} &\frac{\partial}{\partial x}\left(\frac{1}{K}\frac{\partial\Psi}{\partial x}\right) + \frac{\partial}{\partial z}\left(\frac{1}{K}\frac{\partial\Psi}{\partial z}\right) + \frac{\partial C}{\partial x} = 0, \\ &\frac{\partial C}{\partial t} - \frac{\partial\Psi}{\partial z}\frac{\partial C}{\partial x} + \frac{\partial\Psi}{\partial x}\frac{\partial C}{\partial z} - \frac{1}{Ra}\left(\frac{\partial J_x}{\partial x} + \frac{\partial J_z}{\partial z}\right) + DaC = 0. \end{aligned}\right\} \quad \mathbf{x} = (x,z) \in \mathcal{R} = [0,\mathcal{L}] \times [-1,1] \tag{9}$$

The Fickian mass flux $\mathbf{J} = (J_x, J_z)$ (Scheidegger-Bear [58]) satisfies $\mathbf{J} = \mathbf{D}\nabla C$, the components of which are expressed in terms of the above dimensionless variables and numbers as follows:

$$\begin{aligned} J_x &= (1 + \beta_T||\nabla\Psi||_2)\frac{\partial C}{\partial x} + \frac{(\beta_L - \beta_T)}{||\nabla\Psi||_2}\left(\left(\frac{\partial\Psi}{\partial z}\right)^2\frac{\partial C}{\partial x} - \frac{\partial\Psi}{\partial x}\frac{\partial\Psi}{\partial z}\frac{\partial C}{\partial z}\right), \\ J_z &= (1 + \beta_T||\nabla\Psi||_2)\frac{\partial C}{\partial z} + \frac{(\beta_L - \beta_T)}{||\nabla\Psi||_2}\left(\left(\frac{\partial\Psi}{\partial x}\right)^2\frac{\partial C}{\partial z} - \frac{\partial\Psi}{\partial x}\frac{\partial\Psi}{\partial z}\frac{\partial C}{\partial x}\right), \end{aligned} \tag{10}$$

where $||\cdot||_2$ denotes the standard Euclidean norm. Finally, the corresponding dimensionless form of the boundary conditions is:

$$\begin{aligned} C(x,1) = 1, \quad \Psi(x,\pm 1) = \Psi(0,z) = \Psi(\mathcal{L},z) = 0, \\ \frac{\partial C}{\partial z}(x,-1) = \frac{\partial C}{\partial x}(0,z) = \frac{\partial C}{\partial x}(\mathcal{L},z) = 0. \end{aligned} \tag{11}$$

Note that the above conditions imply that the flow is solely a result of the density-induced instability and the concurrent flow recirculation to conserve mass.

The quantity of interest for the C-ED problem will be the surface flux $\mathcal{S}$. Integrating the second equation of (9) over $\mathcal{R}$ we obtain:

$$\int_{\mathcal{R}} \frac{\partial C}{\partial t} ds - \int_{\mathcal{R}} \left(\frac{\partial \Psi}{\partial z}\frac{\partial C}{\partial x} + \frac{\partial \Psi}{\partial x}\frac{\partial C}{\partial z} \right) ds - \int_{\mathcal{R}} \frac{1}{Ra}\left(\frac{\partial J_x}{\partial x} + \frac{\partial J_z}{\partial z} \right) ds + \int_{\mathcal{R}} DaCds = 0. \tag{12}$$

If we now define $\mathcal{M} = \int_{\mathcal{R}} Cds$, we obtain:

$$\frac{d\mathcal{M}}{dt} - \int_{\mathcal{R}} \left(\frac{\partial \Psi}{\partial z}\frac{\partial C}{\partial x} - \frac{\partial \Psi}{\partial x}\frac{\partial C}{\partial z} \right) ds = \frac{\mathcal{S}}{Ra} - \mathcal{M}Da. \tag{13}$$

Since the second integral in the left hand side is equal to zero we can re-write (13) as:

$$\frac{d\mathcal{M}}{dt} = \frac{\mathcal{S}}{Ra} - \mathcal{M}Da, \tag{14}$$

with:

$$\mathcal{S} = \int_{\mathcal{R}} \left(\frac{\partial J_x}{\partial x} + \frac{\partial J_z}{\partial z} \right) ds. \tag{15}$$

Applying the divergence theorem to the integral above, we can finally write $\mathcal{S}$ as,

$$\mathcal{S} = -\int_0^{\mathcal{L}} \left(1 + \beta_T \left| \frac{\partial \Psi}{\partial z} \right|_{z=1} \right) \left(\frac{\partial C}{\partial z} \right)_{z=1} dx. \tag{16}$$

Expression (16) above expresses the fact that the total mass flux across the upper boundary is due to the combined effect of molecular diffusion and dispersion, where the dispersion component ($\beta_T|\partial\Psi/\partial z|$) determines the enhanced dissolution effect. Equation (14) shows that the total mass of solute in the domain increases through solute injection across the upper boundary and decreases through the first order reaction. We also note that in this particular problem, the more CO_2 is absorbed through the upper boundary, the more the process is considered to be efficient.

Although the mathematical formulation of the problem was described in terms of its transient formulation, in this work we are interested in the long term behaviour of the system and consequently we will consider only non-trivial solutions of the corresponding steady state equations.

2.1. Arclength continuation for finding numerical solutions of the C-ED model

The C-ED problem depends on several scalar parameters that need to be specified before attempting to find a numerical solution, namely, Ra, Da, β_L and β_T. Once these four parameters are specified, for a given permeability field, equations (9) are solved for the unknowns (Ψ, C) to yield $\mathcal{S}$.

The system of equations admits multiple solutions. In other words, for a given permeability field, there exists more than one streamfunction and concentration field, and therefore more than one $\mathcal{S}$. Ward *et al.* [56], for instance, demonstrated the existence of multiple solutions for each given Ra in the particular case of $K \equiv 1$, $\beta_L = \beta_T = 0$, and $Da = 0.1$. The existence of multiple outputs for the same input adds an additional challenge to the search for numerical solutions other than the no-flow solution. A numerical code, based on the finite element method (FEM) and detailed further in Section 4, was used to solve the problem (9). The FEM was only able to find one solution (the no-flow) for the problem. To find different solutions, it was necessary to use additional techniques in conjunction with the FEM. In this work we used *arclength continuation theory*, as described in [12] and which is briefly introduced below.

For varying non-dimensional parameters such as Ra, numerous steady-states of (9) appear through bifurcations [12, 56, 47, 15]. The purpose of this section is to evaluate the effects that the heterogeneity of the porous medium has on steady-states solutions; in particular, on the bifurcations from the no-flow steady-state, as represented in Figure 1.

Let $\mathbf{\Phi} = (\Psi, C)$ be the state vector and $\Upsilon = (Ra, Da, \mathcal{L})$ be the vector of parameters of our model. Then, for given β_L, β_T and K, the problem (9) defines a nonlinear time-dependent problem of the form:

$$\mathbf{G}\left(\frac{\partial \mathbf{\Phi}}{\partial t}\right) = \mathbf{F}(\mathbf{\Phi}, \Upsilon)$$

where:

$$\mathbf{F}(\mathbf{\Phi}, \Upsilon) = \left(\begin{array}{c} \dfrac{\partial}{\partial x}\left(\dfrac{1}{K}\dfrac{\partial \Psi}{\partial x}\right) + \dfrac{\partial}{\partial z}\left(\dfrac{1}{K}\dfrac{\partial \Psi}{\partial z}\right) + \dfrac{\partial C}{\partial x} \\ -\dfrac{\partial \Psi}{\partial z}\dfrac{\partial C}{\partial x} + \dfrac{\partial \Psi}{\partial x}\dfrac{\partial C}{\partial z} - \dfrac{1}{Ra}\left(\dfrac{\partial J_x}{\partial x} + \dfrac{\partial J_z}{\partial z}\right) + DaC \end{array} \right)$$

and:

$$\mathbf{G}\left(\frac{\partial \mathbf{\Phi}}{\partial t}\right) = \left(\begin{array}{c} 0 \\ \dfrac{\partial C}{\partial t} \end{array} \right).$$

The study of steady-states is therefore reduced to the nonlinear problem:

$$\mathbf{F}(\mathbf{\Phi}, \Upsilon) = \mathbf{0} \tag{17}$$

to which we can apply the general bifurcation theory. We can numerically compute paths and branches of solutions of (17). For each steady-state along the paths, a linear analysis enables us to determine the linear stability (or instability) of the steady-states. Provided that the Jacobian of $\mathbf{F}$ exists, the linearization about $\mathbf{\Phi}$ (solution of (17)) of the equation for the perturbation is:

$$\mathbf{G}\left(\frac{\partial \mathbf{\Phi}'}{\partial t}\right) = \frac{\partial \mathbf{F}}{\partial \mathbf{\Phi}}(\mathbf{\Phi}, \Upsilon)\mathbf{\Phi}'.$$

From the above considerations, we can analytically compute the no-flow steady-state:

$$\Psi = 0, \quad C(z) = \frac{\cosh(\sqrt{RaDa}(z+1))}{\cosh(2\sqrt{RaDa})}, \tag{18}$$

which is a solution of (17) for all Υ. This solution corresponds to pure diffusion-reaction of the solute in the domain. The above base solution is unstable for certain parameter values and new bifurcation branches of stable steady state solutions can appear (non-trivial solutions). Under steady state conditions, the mass balance equation (14) reduces to: $\mathcal{S} = RaDa\mathcal{M}$. Since Ra and Da are parameters of the model, and recalling that $\mathcal{M} = \int_{\mathcal{R}} Cds$, this shows that $\mathcal{S}$ is directly proportional to $\mathcal{M}$. For computing $\mathcal{S}$, therefore, it is sufficient to solve equations (9) for Ψ and C, followed by the application of equation (14).

Using equation (18), the total mass of solute in $\mathcal{R}$ corresponding to the base steady solution is:

$$\mathcal{M}_0 = \int_{\mathcal{R}} C_0 \mathrm{d}\mathbf{x} = \mathcal{L}\frac{\tanh(2\sqrt{RaDa})}{\sqrt{RaDa}}, \tag{19}$$

which is independent of the permeability field, increases with the aspect ratio $\mathcal{L}$ and decreases with the magnitudes of Ra and Da. This value of $\mathcal{M}_0$ is due only to the dissolution of the solute across the upper boundary. This is not, however, the case for non-trivial solutions, for which the surface dissolution flux varies according to the values of the permeability field. In the case of non-trivial solutions, the surface dissolution flux is enhanced by the effect of the convective recirculating flow, in which the flow intensity depends on the magnitude and spatial variation of the medium permeability.

The system studied in this paper undergoes a bifurcation, namely the appearance of multiple solutions (more than one equilibrium state) for values of Ra above a critical value. At the critical value, a supercritical pitchfork bifurcation (first bifurcation) occurs and two additional branches of stable solutions appear (see Figure 1). Further bifurcations can take place for higher values of Ra, leading to more stable and unstable solutions. For a detailed analysis of the evolution of bifurcations in porous media convection the reader is referred to, e.g., [46, 56, 47, 15]. All of the stable solutions, corresponding to different bifurcation branches, are physically feasible and which one is observed in reality depends on the effects of natural perturbations occurring during the time evolution process.

Figure 1 shows a representation of the solution bifurcation map, i.e., all stable and unstable solutions, found with our numerical scheme for the C-ED problem in a homogeneous medium for different Ra and Da fixed. The reference value to characterise each computed solution was the value of the streamfunction (velocity) at the center of the domain, Ψ_c, according to the corresponding Ra. Note that for $Ra < 42.5$, i.e., below the numerically computed bifurcation point, all solutions have the same streamfunction constant value at the centre of the domain, corresponding to the no-flow solution with total dissolved mass given by $\mathcal{M}_0$ in equation (19). On the other hand, for $Ra > 42.5$ there are two different non-trivial solutions, with total dissolved mass greater than $\mathcal{M}_0$, since the base solution corresponds to the lower bound of capture given that the uptake of the solute increases as soon as convection occurs in the domain. For more details regarding the bifurcation problem, we refer the reader to [12, 56].

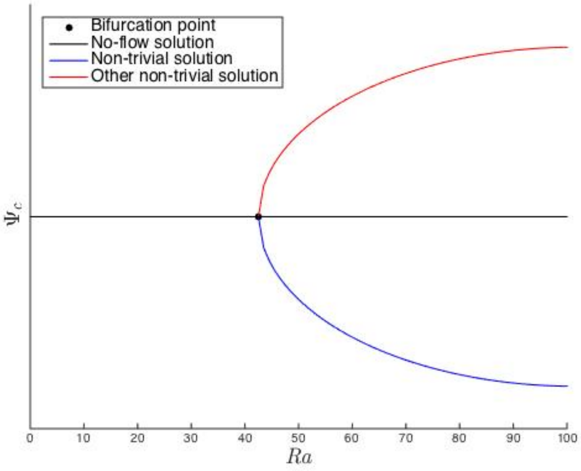

Figure 1: Bifurcation diagram with respect to Ra for a homogeneous permeability field. The black lines correspond to steady-state solutions of the streamfunction at the center of the domain. The other model parameters are $K \equiv 1$, $Da = 0.1$ and $\beta_T = \beta_L = 0$.

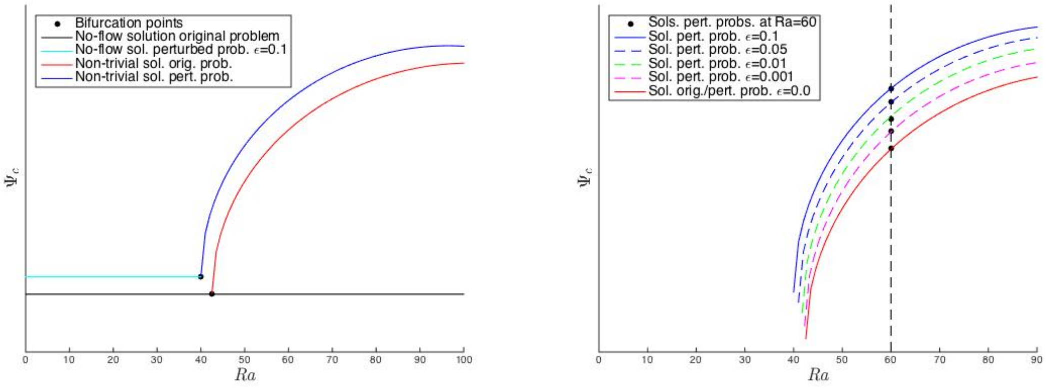

Figure 2: Branches of steady-state solutions for the original and perturbed problems according to different values of Ra. The ordinate axis represents steady-state solutions of the streamfunction at the center of the domain. The rest of parameters are $K \equiv 1$, $Da = 0.1$ and $\beta_T = \beta_L = 0$.

In practice, to compute a non-trivial solution of (9) for a given Ra using arclength continuation, we first perturb the convection-diffusion-reaction equation by an amount ϵ, i.e., we set:

$$\frac{\partial C}{\partial t} - \frac{\partial \Psi}{\partial z}\frac{\partial C}{\partial x} + \frac{\partial \Psi}{\partial x}\frac{\partial C}{\partial z} - \frac{1}{Ra}\left(\frac{\partial J_x}{\partial x} + \frac{\partial J_z}{\partial z}\right) + DaC + \epsilon = 0, \tag{20}$$

and solve the 'perturbed' problem (20) with the FE method as described earlier for an increasing sequence of Ra values ($Ra^{(j)}$, $j = 0, 1, \ldots$), where in each step $j+1$ of the iterative algorithm we set as the initial guess for the next iteration the precomputed solution for $Ra^{(j)}$ (see Figure 2). Once we have computed the solution for the ϵ-perturbed problem at the desired Ra, we reduce ϵ with the same arclength methodology until $\epsilon = 0$.

The permeability parameter appearing in our model needs to be characterised in order to be used as an input for the simulator. Bear [7], for instance, provides empirical data for the permeability and classifies several scenarios according to the data. Farthing *et al.* [22] presented results for different porous media scenarios by using different types of sands. It has been shown [11, 29, 48] that although the permeability values can exhibit large spatial variations, these variations are not entirely random but spatially correlated. Previously, such fields have been modelled using a log-normal distribution assumption [39]. In this paper we will also use a log-normal distribution to model the parameter K, i.e., in the two dimensional case we replace the conductivity tensor by a scalar valued field, the log of which is Gaussian. In the next section we describe how we model the permeability as a log normal random field and how the C-ED model presented earlier yields a system of PDEs with random coefficients.

2.2. Generation of random permeability fields

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. Given $\mathbf{x} \in \mathcal{R}$, we use $Z(\mathbf{x}, \omega)$ (or simply $Z(\mathbf{x})$) to denote a real-valued random field (RF) indexed by $\mathbf{x}$ on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. For each $\mathbf{x} \in \mathcal{R}$, $Z(\mathbf{x}, \cdot) : \Omega \to \mathbb{R}$ is a random variable, while for a fixed $\omega \in \Omega$, Z, is a deterministic function $Z(\cdot, \omega) : \mathcal{R} \to \mathbb{R}$, called a realisation or sample path of the process. We define the mean function $m(\cdot) : \mathcal{R} \to \mathbb{R}$ of a RF $Z(\mathbf{x})$ by:

$$m(\mathbf{x}) = \mathbb{E}[Z(\mathbf{x})] = \int_{\Omega} Z(\mathbf{x}) \; d\mathbb{P}(\omega),$$

and the covariance function $c(\cdot, \cdot) : \mathcal{R} \times \mathcal{R} \to \mathbb{R}$, by:

$$c(\mathbf{x}, \mathbf{x}') = \mathbb{E}\left[(Z(\mathbf{x}) - m(\mathbf{x}))(Z(\mathbf{x}') - m(\mathbf{x}')\right]. \tag{21}$$

The numerical codes used in this work require the values of the permeability at the M *nodes* obtained during the discretization of the physical domain. To generate different permeability fields according to a log Gaussian distribution, let $Z(\mathbf{x})$ be a RF with given mean function $m(\mathbf{x})$ and given covariance function $c(\mathbf{x}, \mathbf{x}')$ on the underlying space $(\Omega, \mathcal{F}, \mathbb{P})$. Then, given the set of nodes $\{\mathbf{x}_i\}_{i=1}^{M}$, the vector

$\mathbf{Z} := (Z(\mathbf{x}_1), \ldots, Z(\mathbf{x}_M))^T$ is a discrete random field. In fact, $\mathbf{Z} : \Omega \to \mathbb{R}^M$ is a multivariate random variable with mean vector and covariance matrix:

$$\mathbf{m} = (m_1, \ldots, m_M)^\intercal = \mathbb{E}[\mathbf{Z}] \in \mathbb{R}^M, \quad \mathbf{C} = \mathbb{E}[(\mathbf{Z} - \mathbf{m})(\mathbf{Z} - \mathbf{m})^\intercal] \in \mathbb{R}^{M \times M} \tag{22}$$

respectively, where:

$$m_i = \mathbb{E}[\mathbf{Z}(\mathbf{x}_i)] = m(\mathbf{x}_i), \quad C_{ij} = c(\mathbf{x}_i, \mathbf{x}_j), \quad i, j = 1, \ldots, M \tag{23}$$

Thus, for a given RF, $\mathbf{Z}$, we can set $\mathbf{K} = \exp(\mathbf{Z})$ to obtain the desired discrete permeability field [33]. Furthermore, if $\mathbf{Z}$ is chosen to be normally distributed then $\mathbf{K}$ is log normal. Note that the log Gaussian assumption is used to avoid negative (unphysical) values of the permeability.

One possible method to generate different (Gaussian distributed) $\mathbf{Z}$ utilises a Cholesky decomposition of the covariance matrix associated to the covariance function given in (21) [55]. Even for a few hundred sampling points, however, the round-off error in this method cannot be neglected due to the fact that the associated covariance matrix is likely to become extremely ill-conditioned [19]. An alternative method for simulating a Gaussian RF is the circulant embedding algorithm [20] described, e.g., in [33]. This method provides an exact simulation of a Gaussian RF, although its implementation is not straightforward. Another technique that has been used extensively to produce samples of the permeability fields is the KL expansion method (see for instance [32]).

In this paper, we use a highly efficient and accurate KL decomposition [25, 15]. This method could be inappropriate for problems in which the simulator necessitates an extremely fine discretization of the computational domain, but this does not apply to the problem considered in this paper. Conversely, the advantage of this approach is that it only requires a single eigen-decomposition of the covariance matrix, the results of which are stored and used to generate new realisations of the permeability field very cheaply. Moreover, the KL decomposition may be truncated, which leads to a lower-dimensional input space for our emulator. The main difference between the KL expansion and KL decomposition methods is that, while KL expansions provide an approximation (due to the truncation of the infinite series) of the permeability fields at all the points of the continuous domain, which can be sampled afterwards on any grid, KL decompositions provide the exact decomposition of the correlation function on a discrete grid. Since the whole eigen-decomposition is considered, variance preservation is not an issue (the total variance is preserved). We remark that KL decompositions may not be useful for models requiring either a very fine discretization of the domain or several evaluations of the permeability at different sets of grid points.

For modelling the correlation of $\mathbf{Z}$, i.e., expression (21), we use the following exponential covariance function [29, 54, 14, 13]:

$$c(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 \ \exp\left(\frac{-||\mathbf{x}_i - \mathbf{x}_j||_2}{\lambda}\right) \quad \mathbf{x}_i, \mathbf{x}_j \in \mathcal{R}, \tag{24}$$

where λ represents the correlation length and σ^2 represents the process variance. In subsurface flow applications, λ is typically chosen to be significantly smaller than the size of the computational region and also large enough to be taken into account in the numerical formulation [13]. The values of σ^2 and λ are therefore problem dependent and are discussed separately for each of the models in Section 4.

We denote by $\mathbf{C}$ the positive semi-definite covariance matrix associated to the function c, i.e., $C_{ij} = c(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{R}$. Since this covariance matrix $\mathbf{C}$ is real-valued and symmetric, it admits an eigen-decomposition [55]: $\mathbf{C} = (\Phi\Lambda^{\frac{1}{2}})(\Phi\Lambda^{\frac{1}{2}})^\intercal$, where Λ is the $M \times M$ diagonal matrix of ordered decreasing eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_M \geq 0$, and Φ is the $M \times M$ matrix whose columns $\boldsymbol{\phi}_i$, $i = 1, \ldots, M$, are the eigenvectors of C. Let $\xi_i \sim \mathcal{N}(0, 1)$, $i = 1, \ldots, M$, be independent and identically distributed (i.i.d.) random variables. We can draw samples from $\mathbf{Z} \sim \mathcal{N}(\mathbf{m}, \mathbf{C})$ using the KL decomposition of $\mathbf{Z}$ using the following [33]:

$$\mathbf{Z} = \mathbf{m} + \Phi\Lambda^{\frac{1}{2}}(\xi_1, \ldots, \xi_M)^\intercal = \mathbf{m} + \sum_{i=1}^{M} \sqrt{\lambda_i}\boldsymbol{\phi}_i\xi_i. \tag{25}$$

The discrete random permeability field is therefore given by:

$$\mathbf{K} = \exp\left(\mathbf{m} + \sum_{i=1}^{M} \sqrt{\lambda_i}\boldsymbol{\phi}_i\xi_i\right). \tag{26}$$

The terms $\xi_i \sim \mathcal{N}(0, 1)$ above will be called *KL coefficients*. With the permeability parameter K modelled as a log Gaussian random field $K = K(\mathbf{x})$ on $\mathcal{R} \times \Omega$, or the discrete form $\mathbf{K}$ given by equation (26), equations (9) become a new system of PDEs with random inputs $K(\mathbf{x})$. This system is solved for the streamfunction $\Psi(\mathbf{x})$ and the concentration $C(\mathbf{x})$, which are also, therefore, random fields. A realisation of the permeability field $\mathbf{K} \in \mathbb{R}^M$ represents possible sets of permeability values in a slice of porous rock across which we would like to study the dissolution process. Figure 3 gives two examples of the permeability field that are used later in the numerical simulations. An approximation of $\mathbf{K}$ can be obtained by restricting the expansion in (26) to the first, say, D KL coefficients, i.e., to the subspace spanned by $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_D$. In the emulator construction, this is exploited to obtain a computationally practical method, as discussed in the next section.

Let us define the random vector $\boldsymbol{\xi} \in \mathbb{R}^D$, for $D \leq M$, distributed according to $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Note that although we allow $D \leq M$, for the simulator runs to generate the training data we strictly set $D = M$, i.e., no approximation of the permeability field is made. The numerical code described in Section 2 can be regarded, respectively, as a mapping from $\mathbf{K}$ to $\mathcal{S}$ (for a log-normally distributed random vector $\mathbf{K}$). Alternatively, the representation (26) of the permeability field allows us to consider the mapping $f_s : \boldsymbol{\xi} \mapsto \mathcal{S}$ for the C-ED simulator, respectively, where $\boldsymbol{\xi} \in \mathbb{R}^D$ is defined as above. In the next section we will develop an emulator for this mapping.

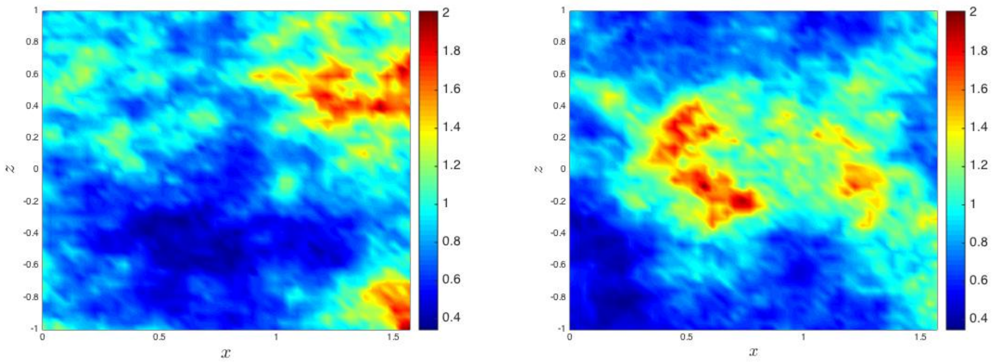

Figure 3: Two different input permeability fields generated with the KL decomposition method where the whole set of 2601 KL coefficients were retained.

3. Gaussian process emulator of the simulator output

A GP can be interpreted as a family of random variables, any finite number of which have a joint Gaussian distribution. A GP is fully specified by its mean function and covariance function [45]. A GP emulator is a statistical approximation of the simulator, in which the mapping between the inputs and outputs is learned using a limited number of simulator runs at carefully selected inputs (*design points*). Such an approximation incurs a fraction of the computational cost and can replace the simulator in an uncertainty analysis, thereby avoiding a large number of costly or even prohibitive simulator evaluations. In this section, we outline GP emulation and discuss the selection of the design points. The majority of the studies in the literature fix the covariance function *a-priori*. In this study, we will apply cross-validation (CV), as recommended in [45], to select an optimal covariance function for each of the two simulators.

3.1. GP emulation general framework

Our aim is to develop an emulator for the simulator $f_s : \boldsymbol{\xi} \mapsto \mathcal{S}$, where $\boldsymbol{\xi} \in \mathbb{R}^D$, for $D \leq M$, is distributed according to $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Note that M denotes the number of nodes in the computational domain of our model problem and is set to $M = 2,601$, as discussed further in Section 4. The GP model involves so-called 'hyperparameters' in the covariance function, to be discussed below. In the covariance structures used, each component of the input is associated with a hyperparameter and the hyperparameters are inferred from the simulator data by solving an optimisation problem. Thus, for high-dimensional input spaces (D moderately large), the GP model would be impractical. We will discuss optimal values of D for building the GP emulator in Section 4.2.

In GP emulation, assumptions about the target function are imposed by specifying a prior probability distribution over a family of possible functions. This prior distribution is then updated in the light of training data (by using Bayes' rule), which yields a posterior distribution that can be used for inference. Let us denote by $f(\cdot)$ the GP used to model $f_s(\cdot)$. The prior specification involves setting a mean function $m(\boldsymbol{\xi})$ and a covariance function $k(\boldsymbol{\xi}, \boldsymbol{\xi}')$, which are defined as: $m(\boldsymbol{\xi}) = \mathbb{E}[f(\boldsymbol{\xi})]$ and $k(\boldsymbol{\xi}, \boldsymbol{\xi}') = \mathbb{C}\mathrm{ov}(f(\boldsymbol{\xi}), f(\boldsymbol{\xi}')) = \mathbb{E}[(f(\boldsymbol{\xi}) - m(\boldsymbol{\xi}))(f(\boldsymbol{\xi}') - m(\boldsymbol{\xi}'))]$, in which in $\mathbb{E}[f(\cdot)]$ and $\mathbb{C}\mathrm{ov}(\cdot, \cdot)$ are the expectation and covariance operators on the (common) probability space underlying the family of random variables $f(\boldsymbol{\xi})$. The covariance function contains *hyperparameters*, which are collectively assigned the symbol $\boldsymbol{\theta}$. Typically (as is the case in this paper), these hyperparameters have to be inferred from the data. We denote the GP prior by:

$$f(\boldsymbol{\xi}) \sim \mathcal{GP}(m(\boldsymbol{\xi}), k(\boldsymbol{\xi}, \boldsymbol{\xi}')). \tag{27}$$

Given the GP above, we can approximate the mapping $f(\cdot)$ using a small number d of simulator runs at carefully selected design points $\{\hat{\boldsymbol{\xi}}_j\}_{j=1}^d$, where $\hat{\boldsymbol{\xi}}_j \in \mathbb{R}^D$, for some $D \leq M$. It is not required that the design points $\hat{\boldsymbol{\xi}}$ are distributed according to $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The choice of design points should be optimal in terms of learning the deterministic mapping $f(\cdot)$ with a limited number of simulator runs and is discussed in Section 3.2. To avoid numerical instabilities (ill-conditioning of the matrix system), an i.i.d. random noise $\epsilon_j \sim \mathcal{N}(0, \sigma_n^2)$, where σ_n^2 is the variance, is typically introduced into the model, i.e.:

$$y_j = f_s(\hat{\boldsymbol{\xi}}_j) + \epsilon_j, \tag{28}$$

where y_j is the noisy simulator output at the design point $\hat{\boldsymbol{\xi}}_j$. Collectively, $\{y_j\}_{j=1}^d$ are termed the *observed values* or *targets*. We can define the *design matrix* as $\mathbf{X} = [\hat{\boldsymbol{\xi}}_1, \hat{\boldsymbol{\xi}}_2, \ldots, \hat{\boldsymbol{\xi}}_d]$ and write the observed values in vector form $\mathbf{y} = (y_1, \ldots, y_d)^\intercal$. The training set is defined as the pair $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$.

A key property of GPs is that their posterior distribution, after taking into account the training data $\mathcal{D}$, is still a GP; in this case, given a prior as in expression (27) and a training set $\mathcal{D}$, we obtain a *posterior* GP process with updated mean and covariance functions. This allows us to make predictions. The variance-covariance matrix for the distribution over $\mathbf{y}$ is given by:

$$\mathbb{C}\mathrm{ov}(\mathbf{y}) = \Sigma(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}, \tag{29}$$

where the (i, j)-th entry of $\Sigma(\mathbf{X}, \mathbf{X}) \in \mathbb{R}^{d \times d}$ is given by $k(\hat{\boldsymbol{\xi}}_i, \hat{\boldsymbol{\xi}}_j)$. Predictions can be made for new input $\boldsymbol{\xi}_*$, i.e., we obtain the distribution of $f^* := f(\boldsymbol{\xi}_*)$, conditioned on the training data $\mathcal{D}$. From the joint distribution of $\mathbf{y}$ and f^*:

$$\begin{bmatrix} \mathbf{y} \\ f^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ \mu^* \end{bmatrix}, \begin{bmatrix} \Sigma(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I} & \Sigma(\boldsymbol{\xi}_*, \mathbf{X}) \\ \Sigma(\boldsymbol{\xi}_*, \mathbf{X})^\intercal & k(\boldsymbol{\xi}_*, \boldsymbol{\xi}_*) \end{bmatrix} \right), \tag{30}$$

where $\boldsymbol{\mu} = (m(\hat{\boldsymbol{\xi}}_1), \ldots, m(\hat{\boldsymbol{\xi}}_d))^\intercal$, $\mu^* = m(\boldsymbol{\xi}_*)$ and $\Sigma(\boldsymbol{\xi}_*, \mathbf{X}) = (k(\boldsymbol{\xi}_*, \hat{\boldsymbol{\xi}}_1), \ldots, k(\boldsymbol{\xi}_*, \hat{\boldsymbol{\xi}}_d))^\intercal$, the posterior distribution of f^* conditioned on $\mathcal{D}$ is given by [45]:

$$f(\boldsymbol{\xi})|\mathcal{D} \sim \mathcal{GP}(m_{\mathcal{D}}(\boldsymbol{\xi}), k_{\mathcal{D}}(\boldsymbol{\xi}, \boldsymbol{\xi}')) \tag{31}$$

where

$$m_{\mathcal{D}}(\boldsymbol{\xi}^*) := \mathbb{E}\left[f^*|\mathcal{D}, \boldsymbol{\xi}^*\right] = \mu^* + \Sigma(\boldsymbol{\xi}^*, \mathbf{X}) \left[\Sigma(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}\right]^{-1} \mathbf{y}, \tag{32}$$

and

$$k_{\mathcal{D}}(\boldsymbol{\xi}^*, \boldsymbol{\xi}^*) = k(\boldsymbol{\xi}^*, \boldsymbol{\xi}^*) - \Sigma(\boldsymbol{\xi}^*, \mathbf{X})^\intercal \left[\Sigma(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}\right]^{-1} \Sigma(\boldsymbol{\xi}^*, \mathbf{X}). \tag{33}$$

Expressions (32) and (33) provide, respectively, a prediction for the simulator output at $\boldsymbol{\xi}^*$, and the predictive variance in the output (encoding the uncertainty in the prediction). In this study, the GP emulation is implemented using the GPML MATLAB toolbox v3.4 [45].

3.2. Generation of the design points and cross validation

To design an emulator, it is desirable to use a limited number of the expensive simulator runs, with design points that cover the full range of physically reasonable input values. Design points that are too close together can lead to ill-conditioned covariance matrices. The design points are generated using sampling, i.e., a random (or psuedo random) distribution of points in a defined interval according to some distribution or rule. There are several methods of sampling the input values, the most common of which are Latin Hypercube Sampling (LHS) [35, 43] and Sobol sequence sampling [52]. In this paper, we use the latter to build our design. Sobol sequences are a family of quasi-random sequences that are designed to generate samples of multiple parameters in a highly uniform manner; consideration of the previously sampled points avoids the occurrence of clusters and gaps [10, 50].

In practical terms, we use a Sobol sequence to generate d points in $[0,1]^M$. Each of the M components of these points can be considered as possible values of the cumulative distribution function of a random variable in $\mathbb{R}$. Each of the d points are pushed component wise through the inverse cumulative distribution function of M random variables distributed according to $\mathcal{N}(0, \sigma_d^2)$, with $\sigma_d^2 > 1$, to, jointly, form $\{\hat{\xi}_j^1, \ldots, \hat{\xi}_j^M\}_{j=1}^d$. We treat the ensembles $\{\hat{\xi}_1^1, \ldots, \hat{\xi}_1^M\} \ldots \{\hat{\xi}_d^1, \ldots, \hat{\xi}_d^M\}$ as sets of KL coefficients, and for each of them compute the corresponding noisy outputs $\{y_j\}_{j=1}^d$. We then, by retaining the first D terms of each of the d ensembles above, denote the set of design points by $\{\hat{\boldsymbol{\xi}}_j\}_{j=1}^d \subset \mathbb{R}^D$, where $\hat{\boldsymbol{\xi}}_j := (\hat{\xi}_j^1, \ldots, \hat{\xi}_j^D)^\intercal$. These design points are used with the noisy outputs $\{y_j\}_{j=1}^d$ to form the training set $\mathcal{D}$.

We use $\sigma_d^2 > 1$ to ensure that the design points have a greater spread than the random variables (KL coefficients) $\boldsymbol{\xi}$ we wish to model, which are $\mathcal{N}(\mathbf{0}, \mathbf{I})$, i.e., to ensure that the tails of the target input distribution are covered with the design. During this study, alternative training sets based on training points generated according to different values of σ_d where considered, including $\hat{\boldsymbol{\xi}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, i.e., $\sigma_d^2 = 1$. The best calibrated GP emulator was obtained with a value of $\sigma_d^2 = 1.44$. Henceforth, in this paper, for both of the models studied in Section 4 we set $\sigma_d^2 = 1.44$ to form the design points in the training set.

The properties of the Sobol sequences require a sample size of 2^j where $j = 1, 2, \ldots$ [10] in order to exploit the manner in which values are spread in $[0, 1]^M$. Thus, the number of design points for each training set to be considered throughout this paper will be $2^8 = 256$ points. Once a model has been tested and selected, this number of design points can be minimised (see Figure 9) in order to reduce computational cost depending on the desired accuracy in the predictions.

For complex simulators, such as the one discussed in this paper, is often not possible to gain a high number of simulator outputs with which to test the emulator accuracy. In such cases we can use CV to estimate the emulator error. We split the training set into two disjoint sets, one of which is used for training. The performance on the remaining ('validation') set is used to estimate the prediction error, and model testing and selection are carried out using this measure. We use the leave-one-out cross-validation (LOO-CV), which consists of using all but a single data point for training, and computing the model prediction error on the omitted point. This process is repeated until all available d points have been exhausted.

CV can be used with any loss function, although the squared error loss is the most common for emulation. In this work, we will use the Dawid score (DS) introduced by Dawid and Sebastiani [16] and the mean squared error (MSE) defined, respectively, as follows [57]:

$$DS = \frac{1}{d} \sum_{j=1}^{d} \left(\frac{(y_j - m_j)^2}{s_j^2} + \log\ s_j^2 \right), \tag{34}$$

$$MSE = \frac{1}{d} \sum_{j=1}^{d} (y_j - m_j)^2. \tag{35}$$

where m_j is the predicted expected value at a given design point, $\hat{\boldsymbol{\xi}}_j$, given by (32), s_j^2 its corresponding variance, given by (33), and y_j the corresponding observed value at $\hat{\boldsymbol{\xi}}_j$.

In the following section, we will describe how we select appropriate parameters and functions for the GP emulator.

3.3. Specification of the GP emulation model

The selection of the mean and, in particular, the covariance function is crucial in a GP predictor [45]. In order for a model to be a practical tool, we need to make decisions about the details of its specification. Some properties may be easy to specify from the context of the problem, while we typically have only vague information available in regards to other aspects, e.g., length-scales or process variances. In this study, we test three different families of covariance functions. For each function, we test the emulator predictions against the observed values by using the LOO-CV method. To select the final covariance function for the GP emulator, we use the criterion that the predicted data will lie within the 95% CI in 95% of cases [27], i.e., we compute the percentage of points out of range for each emulator and then choose the model (covariance function) with the smallest number of cases outside the 95% CI. The 95% CIs are computed as in [45], i.e., by using the intervals $(m_j - 2s_j, m_j + 2s_j)$ where m_j and s_j are respectively the predictive mean (32) and square root (standard deviation) of the variance (33) for a given design point $\hat{\boldsymbol{\xi}}_j$.

The first covariance function (the most commonly used, see e.g., [59]) is the squared exponential (SE). This function is infinitely differentiable, which means that the associated GP has mean-square derivatives to all orders. The Matérn class[1] [34] is often used as an alternative for cases in which a strong smoothness assumption is deemed unrealistic [53]. The third covariance function tested was the rational quadratic (RQ), which is as an alternative to the Matérn class [34]. The two anisotropic SE and RQ covariance functions used in this study are defined as follows:

$$k_{SE}(\boldsymbol{\xi}, \boldsymbol{\xi}') = \sigma_f^2 \exp\left(-\frac{1}{2}(\boldsymbol{\xi} - \boldsymbol{\xi}')^\top \mathrm{diag}(\ell_1^{-2}, \ldots, \ell_D^{-2})(\boldsymbol{\xi} - \boldsymbol{\xi}')\right) + \sigma_n^2 \delta_{ij}, \tag{36}$$

$$k_{RQ}(\boldsymbol{\xi}, \boldsymbol{\xi}') = \sigma_f^2 \left(1 + \frac{1}{2\alpha}(\boldsymbol{\xi} - \boldsymbol{\xi}')^\top \mathrm{diag}(\ell_1^{-2}, \ldots, \ell_D^{-2})(\boldsymbol{\xi} - \boldsymbol{\xi}')\right)^{-\alpha} + \sigma_n^2 \delta_{ij}, \tag{37}$$

respectively. The use of different characteristic length scales $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_D)$ for each input implements automatic relevance determination (ARD) [40] since the inverse of the length-scale determines the relevance of each of the D inputs: If the length-scale has a very large value, the covariance will become almost independent of that input, effectively removing it from the inference [45]. The hyperparameters for each case are $\boldsymbol{\theta}_{SE} = (\sigma_f^2, \boldsymbol{\ell}, \sigma_n^2)$ and $\boldsymbol{\theta}_{RQ} = (\sigma_f^2, \boldsymbol{\ell}, \alpha, \sigma_n^2)$. To obtain estimates of the hyperparameters we maximize the negative log marginal likelihood (38) w.r.t. the hyperparameters:

$$-\log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = \frac{1}{2}\mathbf{y}^\intercal(\Sigma + \sigma_n^2 \mathbf{I})^{-1}\mathbf{y} + \frac{1}{2}\log|\Sigma + \sigma_n^2 \mathbf{I}| + \frac{n}{2}\log 2\pi. \tag{38}$$

Once the hyperparameters of the four covariances functions have been inferred from the training data, we apply the LOO-CV technique described in Section 3.2 to each of the corresponding GP models for model (covariance function) selection.

4. Numerical results

In this section we present the results obtained after using the GP emulator to perform a full UQ on the distribution of the surface flux described in Section 2. We restrict ourselves to the domain $\mathcal{R} = [0, \pi/2] \times [-1, 1] \subset \mathbb{R}^2$ and set $\mathcal{L} = \pi/2$. The model parameters are chosen to be: $Ra = 100$, $Da = 0.1$, $\beta_L = \pi/2$ and $\beta_T = \beta_L/10$. The permeability fields used as inputs in the simulator were

[1] There are a whole family of Matérn class functions and in this work we consider only the Matérn$_{3/2}$ and Matérn$_{5/2}$. For explicit expressions of these covariance functions, we refer the reader to [45].

generated from the correlation function (24) with parameter values $\lambda = 0.5$ (value within the range suggested in [13] for a similar problem) and $\sigma^2 = 0.1$ (a small variance is imposed to study how small variations in the permeability affect the quantity of interest). To be consistent with the non dimensional formulation of the equations in (9) we generate a set of log Gaussian permeability fields with pointwise mean 1, i.e., $\mathbb{E}[K(\mathbf{x}_i)]$, $\forall \mathbf{x}_i \in \mathcal{R}$. For that purpose we set $\mathbf{m} = -(\sigma^2/2)\mathbf{I}$ in (25).

Numerical approximations to the solution of (9) were computed with a H^1-conforming finite element method (FEM) [9]. The numerical approximations were evaluated on a shape-regular rectangular partition of $\mathcal{R} = [0, \pi/2] \times [-1, 1] \subset \mathbb{R}^2$ comprising 2,500 elements (i.e., $M = 2601$), employing basis functions of polynomial degree 1. All computations were performed using the AptoFEM finite element toolkit, documented in [3], together with the MUMPS linear solver [1, 2]

Figure 4 shows the three solutions corresponding to each of the three bifurcation branches shown in Figure 1 for $Ra = 60$, i.e., the simulated contours of the streamfunction Ψ (right) and the concentration C (left) for the same permeability field $\mathbf{K}$ (the left hand field in Figure 3). The middle row shows the solutions corresponding to a constant surface flux, while the top and bottom rows are the solutions corresponding to the two new bifurcation branches with a non-constant flux appearing after the critical Ra. We consider two types of solutions; what we called trivial solutions, i.e., solutions of equation (18) that lead to a constant surface flux, $\mathcal{S}_0 = 4.97$, and non-trivial solutions, i.e., solutions leading to a non-constant surface flux, $\mathcal{S} \neq \mathcal{S}_0$.

Figure 5 shows contour plots of the streamfunction and the concentration (bottom right and top right respectively) for the solution on one of the stable bifurcation branches at $Ra = 100$, corresponding to the heterogeneous permeability field $\mathbf{K}$ on the right in Figure 3. The solution on the left of Figure 5 corresponds to the homogeneous case, $K = 1$. In both cases, the base unstable solution is the same (equation (18)), with surface flux $\mathcal{S}_0 = 4.97$. The corresponding enhancement of the dissolution by convection is the difference between the fluxes for the homogeneous (5.42) and heterogeneous (5.17) cases and the flux for the base solution (4.97), i.e., in the homogeneous case the enhancement is $5.42 - 4.97 = 0.45$ and in the heterogeneous case the enhancement is $5.17 - 4.97 = 0.20$. This unexpected result, where heterogeneity reduces enhancement, is analysed in detail in [15, 47].

UQ of the CDF of $\mathcal{S}$ for the C-ED problem

We use GP emulation to perform a full UA on the C-ED problem. In this case, we estimate the uncertainty distribution of the surface flux arising from the uncertainty distribution in the input permeability. As a first attempt, we used the standard GP emulation methodology based on GP emulation discussed in Section 3 to perform a full UA of the CDF of the surface flux. However, the

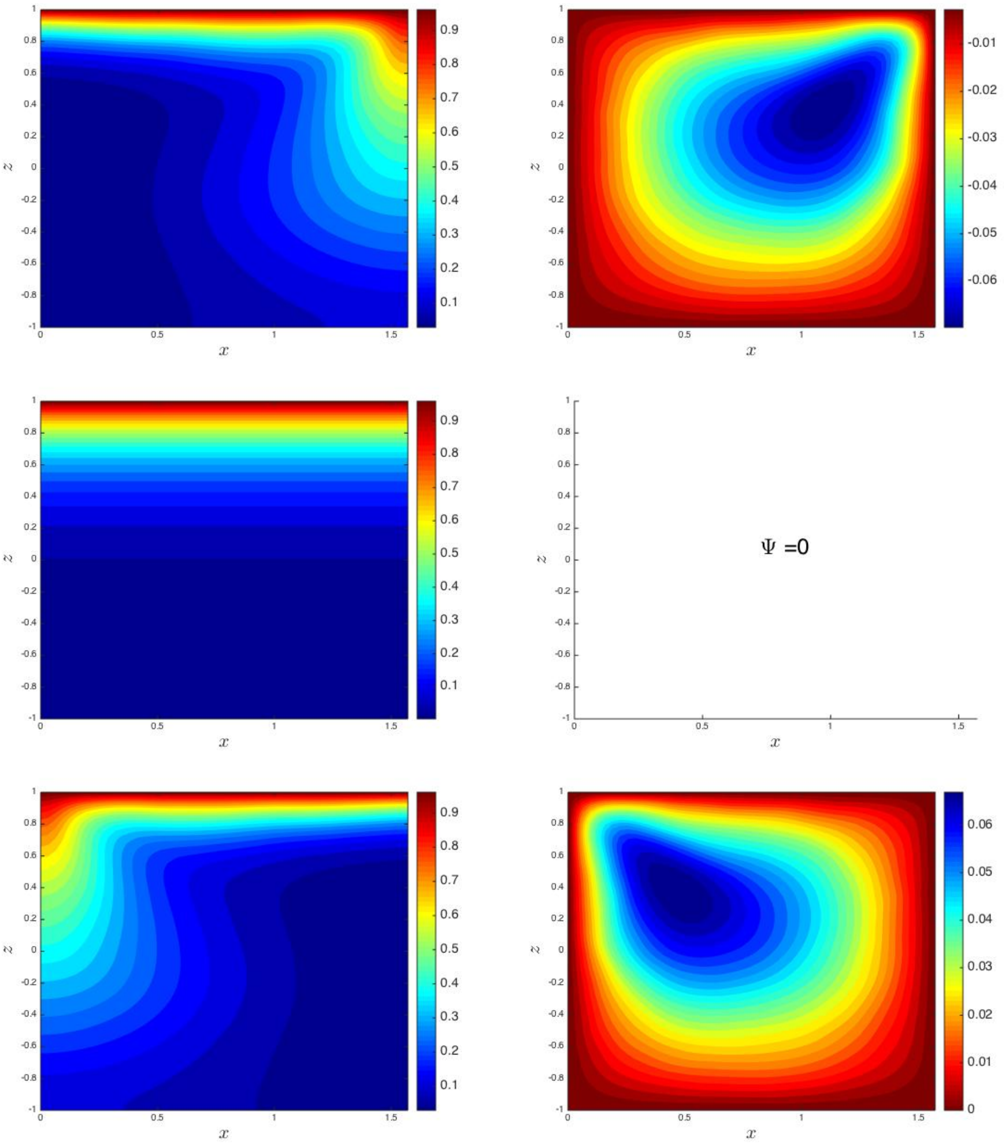

Figure 4: Example of three different solutions of problem (9) for a given heterogeneous permeability field (the left hand field in Figure 3) with parameters: $Ra = 60$, $Da = 0.1$, $\beta_L = \pi/2$ and $\beta_T = \beta_L/10$. The upper branch solution is shown in the top row of figures, the trivial solution is shown in the middle row and lower branch solution is shown in the bottom row.

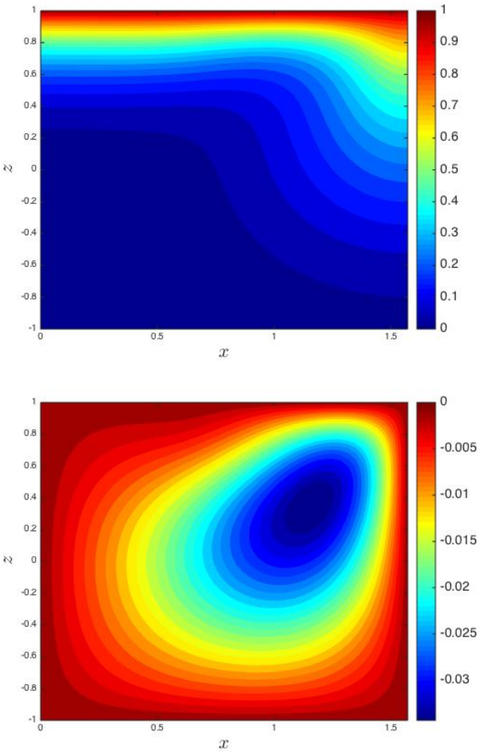

Figure 5: Concentration and streamfunction contours for both a homogeneous permeability field ($K = 1$) and a heterogeneous permeability field (the right hand field in Figure 3) for the C-ED problem with parameters: $Ra = 100$, $Da = 0.1$, $\beta_L = \pi/2$ and $\beta_T = \beta_L/10$. The homogeneous case is shown on the left and the heterogeneous case on the right. The corresponding surface fluxes for the homogeneous and heterogeneous cases, respectively, are 5.42 and 5.17.

method was not able to accurately predict the CDF around the bifurcation point (see Figure 10). To overcome this issue, we introduce a precursor to the GP emulation, which we call the *classifier*. The classifier will allow us to predict the 'class' of the solution given the input. Once the solution is classified, we can then use the same approach followed in Section 3.1 to estimate the value of the $\mathcal{S}$ using GP emulation. Before showing the UQ results, let us introduce the GP classifier first.

4.1. The GP classifier

We wish to assign an input pattern $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to one of two classes: $C_1 : \mathcal{S} = \mathcal{S}_0$ and $C_2 : \mathcal{S} \neq \mathcal{S}_0$. We will use a GP classifier, in which test predictions take the form of class probabilities. If we use the labels $y = +1$ and $y = -1$ to distinguish the two classes C_1 and C_2, respectively, we predict, for instance, $\pi(\boldsymbol{\xi}) = \Pr(y = +1|\boldsymbol{\xi})$, where π denotes the probability that an input $\boldsymbol{\xi}$ is in the class $y = +1$. Since a GP prior over functions does not restrict the output to lie in the interval $[0, 1]$, we need to "squash" the prior function (in practical terms, what we squash are the samples, drawn from the prior distribution of $f(\cdot)$). A common choice for this squashing function is the function $\gamma(z) = (1 + \exp(-z))^{-1}$, called the $logistic\ function$. The GP prior over $f(\cdot)$ induces a prior over probabilistic classifications π. We can then apply the methodology described in Section 3 to obtain the posterior mean for that $\pi(\boldsymbol{\xi})$. Thus, we use a Gaussian process in essentially the same way, except that the Gaussian likelihood function often used for emulation is inappropriate for classification. The likelihood function considered for our classification model will be the error-function (or cumulative Gaussian), and does not contain any hyperparameters; the error function of a Gaussian distribution is defined as $Erf(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^{x} e^{-t^2} dt$.

Since exact inference is only possible for a Gaussian likelihood, we need an alternative approximation inference method. Thus, we will use the Expectation Propagation (EP) algorithm [38] described in [45]. EP provides an approximations to an intractable factorized probability distribution $p(x) = \prod_i p_i(x)$, using a simpler factored form $q(x) = \prod_i q_i(x)$, where each factor $q_i(x)$ belongs to the exponential family. EP is designed to minimize the distance between the two distributions, measured by the Kullback-Leibler divergence $KL(p||q)$. Exact minimization is not feasible because it involves an expectation with respect to the original distribution $p(x)$. EP therefore uses an iterative procedure that at each step minimises the KL divergence between the (new) $q(x)$ and a distribution defined by the current $q(x)$ with one factor $q_i(x)$ replaced by the corresponding factor $p_i(x)$. The factor $q_i(x)$ is then easily updated, and the process is repeated (going through all the factors in turn) until convergence [37].

The mean and covariance functions for the GP classification model are chosen to be a mean-zero function and the SE covariance function. Let $\mathcal{D} = \{\hat{\boldsymbol{\xi}}_j, y_j\}_{j=1}^{256}$ be the training set used for the travel time simulator in Section 3.2, and let $X = \{\hat{\boldsymbol{\xi}}_j\}_{j=1}^{256}$ be the design matrix and $\mathbf{y} = \{y_j\}_{j=1}^{256}$

the vector of noisy observations. If we split the set X into two disjoint sets, X_1 and X_2, where $X_1 = \{\hat{\boldsymbol{\xi}} \in X : f_s(\hat{\boldsymbol{\xi}}) = \mathcal{S}_0\}$ and $X_2 = \{\hat{\boldsymbol{\xi}} \in X : f_s(\hat{\boldsymbol{\xi}}) \neq \mathcal{S}_0\}$, and set $\mathcal{D}_2 = \{X_2, \mathbf{y}_2\}$, where $\mathbf{y}_2$ is the set of simulator outputs $f_s(\cdot)$ with inputs in X_2, we can consider a GP emulator (as previously described), labelled GP$_2$, based on the training set $\mathcal{D}_2$. Thus, for any given input $\boldsymbol{\xi}^* \in \mathbb{R}^D$ distributed according to $\mathcal{N}(\mathbf{0}, \mathbf{I})$ we run the classifier as a first step and if the classifier labels the output as a constant $\mathcal{S}_0$, we take the emulator output as $\mathcal{S}_0$. Conversely, if the output is labelled as a non-constant $\mathcal{S}$ then we use GP$_2$ for making predictions; we will take as the predicted $\mathcal{S}$ value, $m_{\mathcal{D}_2}(\boldsymbol{\xi}^*) = \mathcal{S}$, i.e., the mean of the posterior GP built upon the training set $\mathcal{D}_2$. The entire procedure is illustrated in Figure 6.

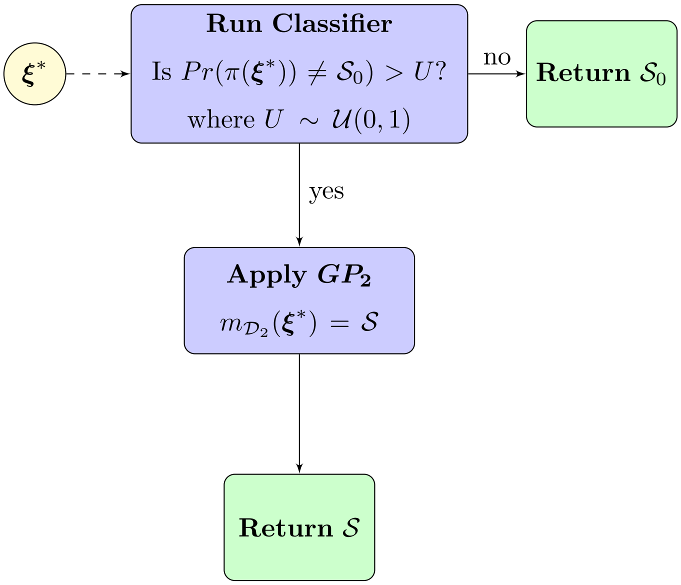
$\boldsymbol{\xi}^*$
Run Classifier
Is $Pr(\pi(\boldsymbol{\xi}^*)) \neq \mathcal{S}_0) > U$?
where $U \sim \mathcal{U}(0,1)$
no
Return $\mathcal{S}_0$
yes
Apply GP_2
$m_{\mathcal{D}_2}(\boldsymbol{\xi}^*) = \mathcal{S}$
Return $\mathcal{S}$

Figure 6: Algorithm followed for predicting the $\mathcal{S}$ for any input given by using a Gaussian process classifier. $U \sim \mathcal{U}(0,1)$ is a uniform random variable between 0 and 1 which gives randomness to the classification process.

4.2. GP emulation and classification for UQ of the CDF of $\mathcal{S}$

In this application, we considered the training set $\mathcal{D} = \{\hat{\boldsymbol{\xi}}_j, y_j\}_{j=1}^{256}$ where the points $\hat{\boldsymbol{\xi}}_j \sim \mathcal{N}(\mathbf{0}, 1.44\,\mathbf{I})$ were generated from an initial Sobol sequence of 256 points over $[0,1]^M$. After running the simulator f_s for the corresponding 256 cases, we found 64 cases leading to a constant surface flux $\mathcal{S}_0$. Thus, after removing those 64 pairs, $\{\hat{\boldsymbol{\xi}}_j, y_j\}$, form $\mathcal{D}$, the training set, $\mathcal{D}_2$, for the GP$_2$ emulator consisted of $d_2 = 192$ points.

For GP model selection, we applied the LOO-CV analysis discussed in Section 3.2 to the four covariance functions introduced in Section 3.3, namely, the SE, Matérn$_{3/2}$, Matérn$_{5/2}$, and RQ. The largest percentage (95.94%) of predictions within the bounds, among all the models, with the 95%

acceptance interval, was given by the model using the RQ covariance function. Thus, throughout this section, we use a GP emulator with a mean-zero function and the RQ covariance function. Figure 7 shows an illustration of the observed (blue) and predicted values (red), and 95% uncertainty bounds (black vertical bars) for the LOO-CV method applied to the selected GP emulator. The horizontal axis shows the first component $\hat{\xi}^1$ of each of the corresponding 192 design points $\{\hat{\boldsymbol{\xi}}_j\}_{j=1}^d \subset \mathbb{R}^D$.

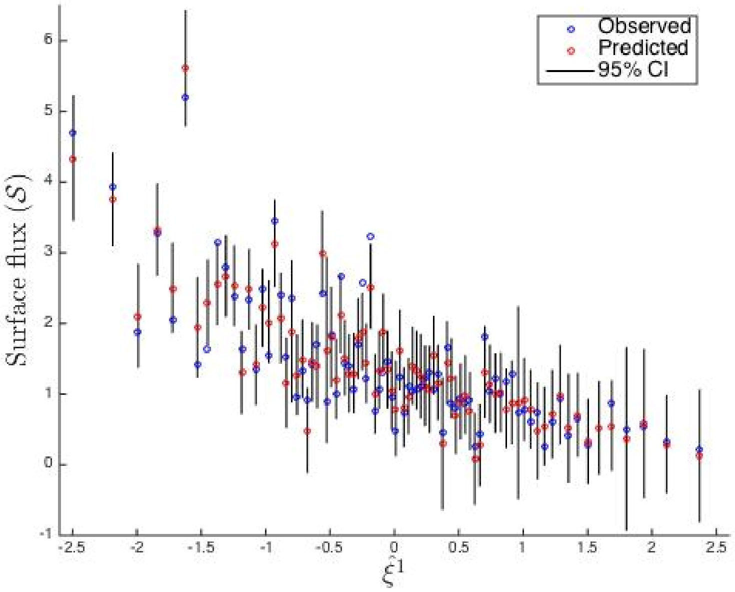

Figure 7: LOO-CV from the design of GP$_2$ formed by 192 points. 4.06% of the observed values are out of range. Vertical axis: predicted surface fluxes (red) with 95% bounds (black bars) given by the GP emulator using a mean-zero function and a RQ covariance function and observed surface fluxes (blue). Horizontal axis: first KL coefficient $\hat{\xi}^1$ of each of the corresponding 192 design points.

We also investigated a further refinement of the model in terms of the GP emulator input space, i.e., we studied the effectiveness of the GP emulator for different lower dimensional input spaces ($\mathbb{R}^D, D \leq M$), by inspecting the MSE (35) and DS (34) scores. From the training set $\mathcal{D}_2$ of $d_2 = 192$ points, we considered an ordered sequence of sets $\mathcal{D}^i$, $i = 1, 2, \ldots$, in which the first i KL coefficients were retained from the original M. For instance, $\mathcal{D}^3 = \{(\hat{\xi}_j^1, \hat{\xi}_j^2, \hat{\xi}_j^3)^\intercal, y_j\}_{j=1}^d$. Applying LOO-CV to each of the training sets in the sequence $\mathcal{D}^i$, $i = 1, \ldots$, we calculated the MSE and DS. Figure 8 shows the MSE and DS scores against the number of KL coefficients retained (or D). The plots show that after around 16 KL coefficients the scores do not vary significantly. Thus, the value of D in the GP model can be set to $D = 16$. Note that the lower dimensional input space applies only to the GP emulator $f(\cdot)$ and not to the simulator f_s, which is always a mapping from $\mathbb{R}^M$ to $\mathbb{R}$.

In addition, we studied the effect of the training point number d_2 on the accuracy of the GP em-

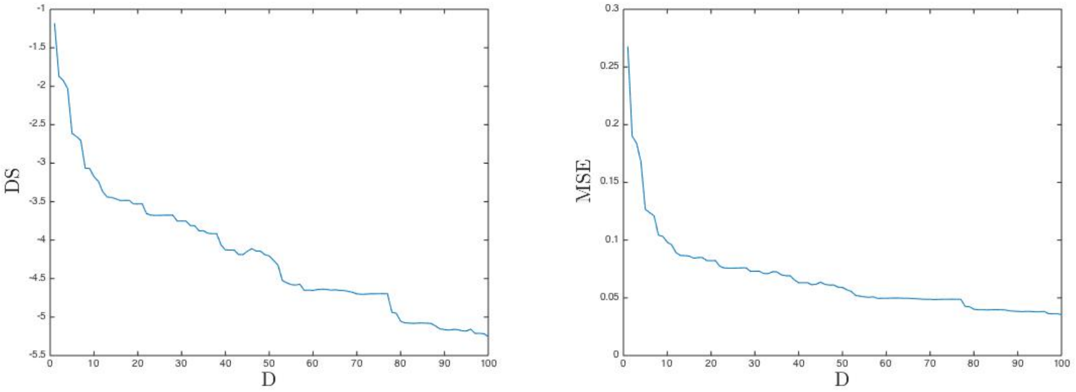

Figure 8: Two scores, DS (left) and the MSE (right), computed by using the LOO-CV method applied to a sequence of training sets with $d = 192$ design points. The GP emulator was built with a mean-zero function and a RQ covariance function. The x-axis represents the number D of KL coefficients used in each of the training sets to calculate the scores.

ulator predictions. For a sample of 10^3 different sets of $\mathcal{N}(0,1)$ KL coefficients $\{\xi^*_{1,j},\dots,\xi^*_{M,j}\}_{j=1}^{1,000}$, we used as test points the first 16 terms of each of set to form $\{\boldsymbol{\xi}^*_j \in \mathbb{R}^{16}\}_{j=1}^{1,000}$, which are, therefore, distributed according to $\mathcal{N}(\mathbf{0},\mathbf{I})$. We then computed the relative error between the corresponding true observed (simulator) surface fluxes, $\mathcal{S}_1,\dots,\mathcal{S}_{1,000}$, and the corresponding predictions, $\{\mathcal{S}^*_j = f(\boldsymbol{\xi}^*_j)\}_{j=1}^{1,000}$, using a GP emulator built with $d_2 = 2^n$ design points, $n = 1,\dots,8$. The error considered for comparisons between two vectors throughout this work will be the L^2-norm relative error unless stated otherwise. For two vector $\mathbf{x} = (x_1,\dots,x_n)$ and $\mathbf{y} = (y_1,\dots,y_n)$, we define the L^2-norm relative error between $\mathbf{x}$ and $\mathbf{y}$ as:

$$RE(\mathbf{x},\mathbf{y}) = \frac{||\mathbf{x}-\mathbf{y}||_2}{||\mathbf{x}||_2} \tag{39}$$

where $||\mathbf{x}||_2$ is Euclidean norm. Figure 9 suggests that around 64 the training points are adequate. The selection of the number of design points to keep will depend on the accuracy desired for each problem and will depends on whether the user we can afford to run the simulator a large number of times or not, and therefore retain the entire training set. For the C-ED model discussed in this paper, a single run can take over 24 hours, in which case the number of design points becomes an extremely important consideration.

Once the GP emulator specifications have been selected, the accuracy of the GP emulator predictions can be measured by generating an ensemble $\xi^*_1,\dots,\xi^*_M$ of M realizations of a $\mathcal{N}(0,1)$ random variable and running the simulator f_s to obtain the *true* observed value $\mathcal{S}$. Then we compare to the GP emulator prediction $\mathcal{S}^* := f(\boldsymbol{\xi}^*)$ for the test input $\boldsymbol{\xi}^* := (\xi^*_1,\dots,\xi^*_D)^\intercal$ formed from the first D KL coefficients of the original ensemble $\xi^*_1,\dots,\xi^*_M$. In this work, we used a sample of 1,000 test points $\{\boldsymbol{\xi}^*_j \in \mathbb{R}^{16}\}_{j=1}^{1,000}$ distributed according to $\mathcal{N}(\mathbf{0},\mathbf{I})$. After running the simulator f_s for those

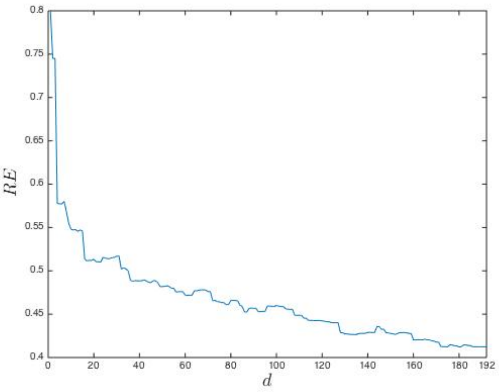

Figure 9: The relative error curve between the observed and predicted surface fluxes for 192 different designs using the RQ covariance function. The curve shows a smooth decreasing tendency and after around 160 design points the decrease in the relative error is negligible.

points, 825 out of the 1,000 led to a non-trivial solution. For those 825 we computed the relative error between the corresponding observed surface fluxes, $\{\mathcal{S}_j\}_{j=1}^{825}$, and the corresponding predictions, $\{\mathcal{S}_j^* := f(\boldsymbol{\xi}_j^*)\}_{j=1}^{825}$, using the GP$_2$ emulator. The results suggested that around 32 training points are adequate, and thus we used the training set formed by the first 32 elements of $\mathcal{D}_2$ (i.e, we retained the first 32 design points from the original 192 forming GP$_2$).

4.3. GP emulation for UQ of the CDF of $\mathcal{S}$

We approximate the cumulative distribution of $\mathcal{S}$ empirically based on the GP emulator as follows:

$$F(s) = \Pr(\mathcal{S} \leq s) = \int_{\mathbb{R}^D} \mathbb{I}\{f(\boldsymbol{\xi})|\mathcal{D} \leq s\} dG(\boldsymbol{\xi}), \tag{40}$$

where $G(\cdot)$ is the probability density function of a random vector $\boldsymbol{\xi} \in \mathbb{R}^D$, with $D = 16$, distributed according to $\mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbb{I}$ denotes the indicator function and $f(\cdot)|\mathcal{D} \sim \mathcal{GP}(m_{\mathcal{D}}, k_{\mathcal{D}})$ as in (31). We will use the approach described in [42] to derive the posterior moments of $F(\cdot)$. We need to simulate draws $F_{(i)}(\cdot)$ from the posterior distribution of $F(\cdot)$, for which we first draw a realisation of the posterior distribution $f(\cdot)|\mathcal{D}$ by drawing a large random sample of inputs $\boldsymbol{\xi}_1^*, \ldots, \boldsymbol{\xi}_N^*$ from $G(\cdot)$, for some integer N. We then form the joint distribution for those inputs as in (31), and draw random samples (denoted by $f_{(i)}$) from $f(\cdot)|\mathcal{D}$. The random samples at a given input $\boldsymbol{\xi}_j^*$ are generated by using the following formula [45]:

$$f_{(i)}(\boldsymbol{\xi}_j^*) = m_{\mathcal{D}}(\boldsymbol{\xi}_j^*) + (k_{\mathcal{D}}(\boldsymbol{\xi}_j^*, \boldsymbol{\xi}_j^*))^{1/2} \boldsymbol{\vartheta}_i \tag{41}$$

where $\boldsymbol{\vartheta}_i \sim \mathcal{N}(0, 1)$, and $m_{\mathcal{D}}(\cdot)$ and $k_{\mathcal{D}}(\cdot, \cdot)$ are the predictive mean and predictive variance given, respectively, by (32) and (33). Finally, we approximate $F_{(i)}(\cdot)$ by using the empirical cumulative distribution function:

$$F_{(i)}(s) = \frac{1}{N}\sum_{j=1}^{N} \mathbb{I}\{f_{(i)}(\boldsymbol{\xi}_j^*) \leq s\}. \tag{42}$$

If we repeat the process a large number of times (n) we can obtain a large sample of distributions $F_{(1)}(\cdot), \ldots, F_{(n)}(\cdot)$, and from this sample we can obtain any required statistic, for instance the sample mean $\bar{F}(s)$, which approximates (40) and is given by:

$$\bar{F}(s) = \frac{1}{n}\sum_{j=1}^{n} F_{(j)}(s). \tag{43}$$

Oakley and O'Hagan [42] remark that since $F(s)$ is constrained to take values between zero and one, the distribution of $F(s)$ may be skewed for low and high values of s. Hence the mean of this distribution may be a poor location summary; it may overestimate $F(s)$ at low values of s and underestimate $F(s)$ at high values of s. Consequently, the sample median might be preferred as a location summary. Now to find our uncertainty bounds we consider the corresponding quantile function. If we let p_α be the 100α percentile, such that $F(p_\alpha) = \alpha$, the distribution of p_α is given by:

$$\Pr(p_\alpha \leq t) = \Pr\{F(t) \geq \alpha\}, \tag{44}$$

where $\Pr\{F(t) \geq \alpha\}$ can be estimated using the method just described (Pr here denotes 'probability'). We can estimate p_α by its sample mean, by finding $p_{(i)\alpha}$, the 100α percentile for realisation i, $i = 1, \ldots, n$. To compute the ECDF of the surface flux with the GP emulator, we follow the procedure above and compute the mean (43), and the lower, upper and median quantiles from this distribution of ECDFs as an approximation of the true CDF of the surface flux $\mathcal{S}$.

Finally, we show the results achieved after performing a full UA for the CDF of the $\mathcal{S}$ by using the GP emulation/classification emulator. Firstly, Figure 10 demonstrates that the GP emulation methodology alone led to failure (top). Figure 10, on the other hand, shows how the posterior samples are able to predict the bifurcation around $\mathcal{S}_0 = 4.9707$ when using the combined GP emulation and classification (bottom). Figure 11 shows the $2.5th$ and $97.5th$ percentiles (dashed), the median of the predicted distribution (red) computed according with the method given in Section 4.3. For comparison purposes, an approximation of the Monte Carlo ECDF (black line) was computed from a sample of 1,000 $\mathcal{S}$ values as follows: (i) generate a large number $N = 1,000$ of different ensembles $\{\xi^*_{1,j}, \ldots, \xi^*_{M,j}\}_{j=1}^{N}$ of KL coefficients ($M = 2,601$), where each $\xi^*_{i,j}$ is distributed according to $\mathcal{N}(0,1)$; (ii) use the simulator to compute the corresponding true $\mathcal{S}_j$ for each of the ensembles; (iii) compute the ECDF, $\hat{F}$, of the set of values $\{\mathcal{S}_j\}_{j=1}^{N}$ according to:

$$\hat{F}(s) = \frac{1}{N}\sum_{j=1}^{N} \mathbb{I}_{\{\mathcal{S}_j \leq s\}}, \tag{45}$$

where $\mathbb{I}$ is the *indicator* function:

$$\mathbb{I}_{\{\tau_j \leq s\}} = \begin{cases} 1 & if \;\; \mathcal{S}_j \leq s, \\ 0 & if \;\; \mathcal{S}_j > s. \end{cases}$$

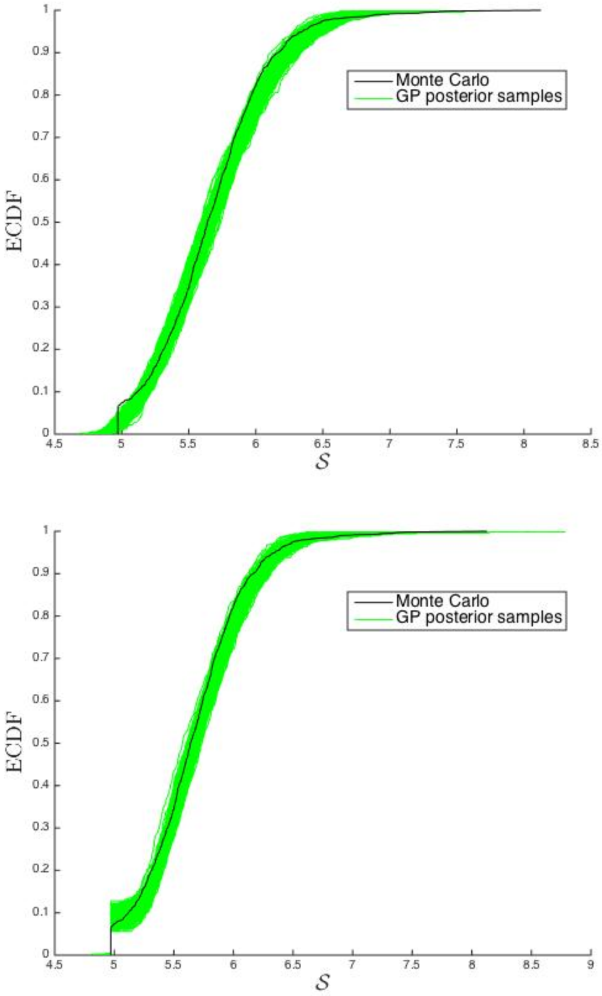

Figure 10: Predicted ECDFs (green) and 1,000 samples based MC approximation of the true $\mathcal{S}$ ECDF (black) with (bottom) and without (top) using GP classification. The number of design points were 32 and the number of KL coefficients were 16. The parameters for the E-CD problem were: $Ra = 100$, $Da = 0.1$, $\beta_L = \pi/2$ and $\beta_T = \beta_L/10$. Prior mean-zero and SE covariance functions, an Erf likelihood function and the Expectation Propagation (EP) method were chosen for the GP classifier. Prior mean-zero and RQ covariance functions were chosen for GP emulation model.

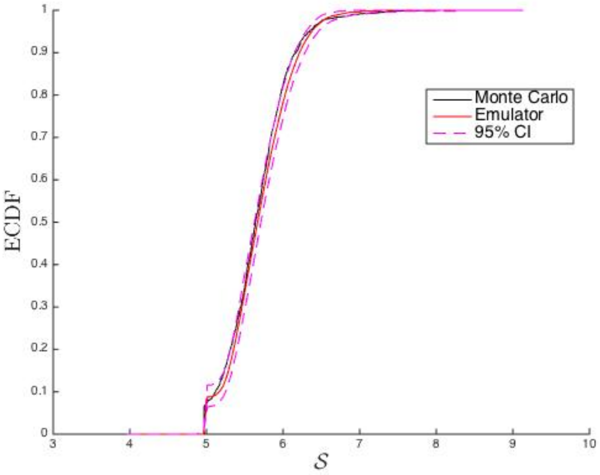

Figure 11: Full uncertainty analysis of the predicted CDF of $\mathcal{S}$ using GP classification/emulation: predicted ECDF (red), the 2.5th and 97.5th percentiles (dashed magenta). The number of design points was 32 and the number of KL coefficients was 16. The parameters for the E-CD problem were $Ra = 100$, $Da = 0.1$, $\beta_L = \pi/2$ and $\beta_T = \beta_L/10$. A prior mean-zero, a SE covariance, an *Erf* likelihood function and the EP method were used for the GP classifier. A prior mean-zero and a RQ covariance were used for the GP emulation. An approximation of the true ECDF with the MC method (black) based on 1,000 samples is also showed for reference.

5. Conclusions and further work

In this paper we developed a methodology for quantifying the uncertainty distribution of groundwater flow simulator outputs, where the uncertainty arises from the input permeability. In a C-ED model that admits multiple solutions, the standard emulation methodology was combined with a classification step in order to predict the simulator outputs around a bifurcation point. The GP classification/emulation methodology proposed in this paper could be used as a bifurcation predictor and applied to models where the user is interested in finding possible model bifurcations. We also showed that it is possible to use a much lower dimensional input space for the GP emulator, leading to a highly efficient emulation.

Acknowledgments This research was funded by the EU Panacea project, FP7, grant agreement 282900.

[1] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster. *A fully asynchronous multifrontal solver using distributed dynamic scheduling.* SIAM J. Matrix Analysis and Applications, 23(1):15-41, 2001.

[2] P. R. Amestoy, A. Guermouche, J.-Y. L'Excellent, and S. Pralet. *Hybrid scheduling for the parallel solution of linear systems.* Parallel Computing, 32(2):136-156, 2006.

[3] P. Antonietti, S. Giani, E. Hall, P. Houston, and R. Krahl. *Aptofem. Finite element software toolkit.* School of Mathematics, University of Nottingham. 2013.

[4] S. F. Ashby and R. D. Falgout. *A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations.* Nucl. Sci. Eng., 124(1), 145-159, 1996.

[5] M. J. Asher, B. F. W. Croke, A. J. Jakeman, and L. J. M. Peeters. *A review of surrogate models and their application to groundwater modeling.* Water Resour. Res., 51, 5957-5973, doi:10.1002/2015WR016967, 2015.

[6] D. A. Baú and A. S. Mayer. *Stochastic management of pump-and-treat strategies using surrogate functions.* Adv. Water Resour., 29(12), 1901-1917, 2006.

[7] J. Bear. *Dynamics of Fluids in Porous Media.* New York; London: American Elsevier, 1972.

[8] D. Bolster, M. Barahona, M. Dentz, D. Fernandez-Garcia, X. Sanchez-Vila, P. Trinchero, C. Valhondo and D. M. Tartakovsky. *Probabilistic risk analysis of groundwater remediation strategies.* Water Resour. Res. 45, W06413, 2009.

[9] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods.* Springer-Verlag New York, Inc, 2002.

[10] S. Burhenne, D. Jacob, and G. P. Henze. *Sampling based on Sobol Sequences for Monte Carlo techniques applied to building simulations.* 12th Conference of International Building Performance Simulation Association, Sydney, 14-16 November, 2011.

[11] E. Byers and D. B. Stephens. *Statistical and stochastic analyses of hydraulic conductivity and particle-size in a fluvial sand.* Soil Science Society of America Journal, 47:1072-1081, 1983.

[12] K. A. Cliffe, A. Spence, and S. J. Tavener. *The numerical analysis of bifurcation problems with application to fluid mechanics.* Acta Numerica, Cambridge University Press, 2000.

[13] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup. *Multilevel Monte Carlo Methods and Applications to Elliptic PDEs with Random Coefficients.* Comput Visual Sci 14:3-15, Springer-Verlag, 2011.

[14] N. Collier, A-L. Haji-Ali, F. Nobile, E. von Schwerin, and R. Tempone. *A Continuation Multilevel Monte Carlo algorithm.* BIT Numerical Mathematics, 2014.

[15] D. Crevillen-Garcia. *Uncertainty Quantification for Flow and Transport in Porous Media.* PhD Thesis. University of Nottingham, 2016.

[16] A. P. Dawid and P. Sebastiani. *Coherent disperson criteria for optimal experimental design.* The Annals of Statistics 27, 65(81), 1999.

[17] M. Dentz, P. Gouze, and J. Carrera. *Effective non-local reaction kinetics for transport in physically and chemically heterogeneous media.* Journal of Contaminant Hydrology, 2011a.

[18] M. Dentz, T. Le Borgne, A. Englert, and B. Bijeljic. *Reactive Transport and Mixing in Heterogeneous Media: A Brief Review.* Journal of Contaminant Hydrology, 2011b.

[19] C. R. Dietrich and G. Newsam. *A stability analysis of the geostatistical approach to aquifer identification.* Stochastic Hydrol. Hydraul., 4(3), 293-316, 1989.

[20] C. R. Dietrich and G. Newsam. *Fast and exact simulation of stationary Gaussian processes through circulant embedding of the covariance matrix.* SIAM J. SCI. COMPUT., 18(4), 1088-1107, 1997.

[21] W.S. Dunbar and A. D. Woodbury. *Application of the Lanczos algorithm to the solution of the groundwater flow equation.* Water Resour. Res., 25, 551-558, 1989.

[22] M. W. Farthing and M. A. Seyedabbasi and P. T. Imhoff and C. T. Miller. *Influence of porous media heterogeneity on nonaqueous phase liquid dissolution fingering and upscaled mass transfer.* Water Resources Research,Vol. 48, W08507, 2012.

[23] L. W. Gelhar, C. Welty, and K. R. Rehfekdt. *A critical-review of data on field-scale dispersion in aquifers.* Water Resour. Res.. 28: 1955-1974, 1992.

[24] N. Gershenzon, R. W. Ritzi, D. F. Dominic, M. R. Soltanian, E. Mehnert, and R.T. Okwen. *Influence of Small-Scale Fluvial Architecture on CO2 Trapping Processes in Deep Brine Reservoirs.* Water Resources Research, Vol.51, 10, 8240-8256, 2015.

[25] R. Ghanem and D. Spanos. *Stochastic Finite eelement: a spectral approach.* Springer, New York, 1991.

[26] K. Ghesmat, H. Hassanzadeh, and J. Abedi. *The impact of geochemistry on convective mixing in a gravitationally unstable diffusive boundary layer in porous media: CO_2 storage in saline aquifers.* J. Fluid Mech., vol. 673, pp. 480-512, Cambridge University Press, 2011.

[27] D. A. Henderson, K. J. Krishnan R. J. Boys, C. Lawless, and D. J. Wilkinson. *Bayesian Emulation and Calibration of a Stochastic Computer Model of Mitochondrial DNA Deletions in Substantia Nigra Neurons.* Journal of the American Statistical Association, 2009.

[28] J. Hidalgo and J. Carrera. *Effect of dispersion on the onset of convection during CO2 sequestration.* J. Fluid Mech. 640, 441-452, 2009.

[29] R.J. Hoeksema and P.K. Kitanidis. *Analysis of the spatial structure of properties of selected aquifers.* Water Resources Research. 21, 536-572, 1985.

[30] T. Y. Hou and X.-H. Wu. *A multiscale finite element method for elliptic problems in composite materials and porous media.* J. Comput. Phys., 134(1), 169-189, 1997.

[31] M. Kennedy and A. O'Hagan. *A Bayesian calibration of computer models (with discussion).* Journal of the Royal Statistical Society, Series B 63, 425-464, 2001.

[32] E. Laloy, B. Rogiers, J. A. Vrugt, D. Mallants, and D. Jacques. *Efficient posterior exploration of a high-dimensional groundwater model from two-stage Markov Chain Monte Carlo simulation and polynomial chaos expansion.* Water Resour. Res., 49, 2664-2682, doi:10.1002/wrcr.20226, 2013.

[33] G. J. Lord, C. E. Powell, and T. Shardlow. *An introduction to computational stochastic PDEs.* Cambridge texts in applied mathematics, 2014.

[34] B. Matérn. *Spatial Variation.* Meddelanden från Statens Skogsforskningsinstitut, 49, No.5. Almänna Förlaget, Stockholm, 1960.

[35] M. D. McKay, R.J. Beckman, and W.J. Conover. *A comparison of three methods for selecting values of input variables in the analysis of output from a computer code.* Technometrics, 21(2):239-245, 1979.

[36] J. McPhee and W. W.-G. Yeh. *Groundwater management using model reduction via empirical orthogonal functions.* J. Water Resour. Plann. Manage., 134, 161-170, 2008.

[37] T. P. Minka. *Expectation propagation for approximate bayesian inference.* In Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence, pages 362-369. Morgan Kaufmann Publishers Inc., 2001.

[38] T. P. Minka. *A Family of Algorithms for Approximate Bayesian Inference.* PhD thesis, Massachusetts Institute of Technology, 2001.

[39] A. Mondal, Y. Efendiev, B. Mallick, and A. Datta-Gupta. *Bayesian Uncertainty Quantifcation for Flows in Heterogeneous Porous Media using Reversible Jump Markov Chain Monte Carlo Methods.* Advances in Water Resources 33, 211-256, Elsevier, 2010.

[40] R. M. Neal. *Bayesian Learning for Neural Networks.* Springer, New York. Lecture Notes in Statistics 118, 1996.

[41] J. A. Neufeld, M. A. Hesse, A. Riaz, M. A. Hallworth, H. A. Tchelepi, and H. E. Huppert. *Convective dissolution of carbon dioxide in saline aquifers*. Geophys. Res. Lett., 37, L22404, 2010.

[42] J. Oakley and A. O'Hagan. *Bayesian inference for the uncertainty distribution of computer model outputs*. Biometrika, 89, 4, pp. 769-784, 2002.

[43] E. J. Pebesma and G. B. M Heuvelink. *Latin hypercube sampling of gaussian random fields*. Technometrics, 41(4):303-312, 1999.

[44] P. Ranganathan, R. Farajzadeh, H. Bruining, and P. L. J. Zitha. *Numerical Simulation of Natural Convection in Heterogeneous Porous media for CO2 Geological Storage*. Springerlink.com, 2012.

[45] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. the MIT Press, 2006.

[46] D. S. Riley and K. H. Winters. *Time-periodic convection in porous media: the evolution of hopf bifurcations with aspect ratio*. J. Fluid Mech. 223, 457-474, 1991.

[47] G. Barba Rossa, K. A. Cliffe, and H. Power. *Effects of hydrodynamic dispersion on the stability of buoyancy driven porous-media convection in the presence of first order chemical reaction*. Journal of Engineering Mathematics, 2016.

[48] D. Russo and M. Bouton. *Statistical analysis of spatial variability in unsaturated flow parameters*. Water Resources Research, 28(7):1911-1925, 1992.

[49] P. G. Saffman. *A theory of dispersion in a porous medium*. Journal of Fluid Mechanics, Cambridge University Press, 1959.

[50] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, and S. Tarantola. *Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index*. Computer Physics Communications, 181(2):259-270, 2010.

[51] A. E. Scheidegger. *General theory of dispersion in porous media*. Journal of Geophysical Research, American Geophysical Union, 1961.

[52] I. M. Sobol. *On the distribution of points in a cube and approximate evaluation of integrals*. U.S.S.R Comput. Maths. Math. Phys. 7: 86-112, 1967.

[53] M. L. Stein. *Interpolation of Spatial Data*. Springer-Verlag, New York, 1999.

[54] N. Stone. *Gaussian Process Emulators for Uncertainty Analysis in Groundwater Flow.* PhD Thesis. University of Nottingham, 2011.

[55] G. Strang. *Introduction to linear algebra.* Cambridge (MA): Wellesley-Cambridge Press, 2003.

[56] T. J. Ward and K. A. Cliffe and O. E. Jensen and H. Power. *Dissolution-driven porous-medium convection in the presence of chemical reaction.* J. Fluid Mech., vol. 747, pp. 316-349, 2014.

[57] R. D. Wilkinson, M. Vrettas, D. Cornford, and J. E. Oakley. *Quantifying simulator discrepancy in discrete-time dynamical simulators.* Vol. 16, Issue 4, pp 554-570. Journal of Agricultural, Biological, and Environmental Statistics, 2011.

[58] Y. Xie, C. T. Simmons, A. D. Werner, and H-J. G. Diersch. *Prediction and uncertainty of free convection phenomena in porous media.* Water Resources Research,Vol. 48, W02535, 2012.

[59] T. Xu and A. J. Valocchi. *A Bayesian approach to improved calibration and prediction of groundwater models with structural error.* Water Resour. Res., 51, 9290-9311, 2015.

[60] S. Yan and B. Minsker. *Optimal groundwater remediation design using an adaptive neural network genetic algorithm.* Water Resour. Res., 42, W05407, doi:10.1029/2005WR004303, 2006.