

---

## *A Worked-out Example of Surrogate-based Bayesian Parameter and Field Identification Methods*

---

*Noémi Friedman,<sup>1,\*</sup> Claudia Zoccarato,<sup>2</sup> Elmar Zander<sup>3</sup> and Hermann G. Matthies<sup>3</sup>*

The aim of this chapter is to compare and present available surrogate-based methods suitable for the assimilation of measurement data of some model response in order to characterize the model's input parameters or input fields. The comparison is done through a worked-out example of assimilating seabed subsidence data for the compressibility field identification of a reservoir. Throughout the example we provide a general framework for any engineering problem where we wish to identify input parameters or fields—and by that to reduce the uncertainties of the predicted response—by assimilating observations of the output of the model. We here provide algorithms of the compared methods and link the examples to on-line accessible MATLAB<sup>®</sup> codes (see <https://ezander.github.io/ParameterAndFieldIdentification/>).

---

### 8.1 Introduction

The production of fluids from deep hydrocarbon reservoirs causes a variation of the pore pressure within the subsurface rock formation. The consequent compaction of the reservoir can be appreciated as deformation on the land or seabed surface. In case of offshore reservoirs, the vertical displacements of the seafloor can cause induced earthquakes and severe damages, such as in the well-known literature cases of the Goose Creek field south of Huston or the Groningen field in the Netherlands [208]. Thus the target is to continuously improve the accuracy of estimation of the compaction caused by the reservoir development as new information (e.g. measurements) are available.

Here, a three-dimensional (3D) finite element method (FEM) model [70] is used to describe the geomechanical behaviour of an offshore reservoir exploited for gas production. The main focus is at estimating the parameters characterising the model response and the associated uncertainty. In particular, the uniaxial vertical compressibility is considered for calibration, as it represents the most influencing parameter controlling the amount of deep compaction.

Different approaches are available to deal with the estimation of compressibility. The simplest way is by some laboratory tests for characterising the geological formations using well-logging tech-

---

<sup>1</sup> Institute for Computer Science and Control (SZTAKI), Budapest, Hungary.

<sup>2</sup> University of Padova, Italy.

<sup>3</sup> Technische Universität Braunschweig, Germany.

\* Corresponding author



niques. Here a different approach is considered, based on the inversion of seabed surface data. The idea here is to use indirect observations of seafloor subsidence to infer a heterogeneous compressibility field. In particular, a time-lapse bathymetric map of the seabed collected over the reservoir domain is used. Such method allows a relatively cheap non-destructive monitoring of the reservoir by updating the compressibility field when new measurements of the subsurface displacements are available. One way to update our initial uncertainty of the compressibility field represented by some prior distribution is to use the Bayesian posterior, an updated distribution of the compressibility field given the measured seabed displacements. This posterior is proportional to the product of the likelihood and the prior distribution of the compressibility field. The likelihood is the probability of how likely is it to measure the given values of displacements for a given realisation of the compressibility field. Due to the fact that the likelihood can not be written here with a closed form one rather uses sampling based procedures. The Metropolis-Hastings random walk can be used as our example of a sampling technique, which results in a Markov-chain Monte Carlo (MCMC) procedure. One problem is the slow convergence of Monte Carlo (MC) based methods, necessitating a large number of MC simulations; for an sufficiently accurate statistic, one is often required to run many hundred thousands of samples. This means for the evaluation of the likelihood that one needs to compute for every sample of the compressibility field the displacement field with the help of the FEM model. This can be highly demanding from a computational point of view in large-scale systems [209].

Another way to update the compressibility field is through an optimised estimator based on the conditional expectation (see Chapter 4). The task here is to find an estimator—a map from the measured data to the updated input—that minimises in a mean squared sense the difference between the prior input parameter and the estimated one computed from the measured data. The Kalman smoother or its frequently used version the Ensemble Kalman smoother (EnKF) [210] are equivalent to such approach if we restrict the estimators to be some linear function of the measured data, and the prior to be Gaussian. As it is shown in the context of the compressibility field identification, limiting the estimator to linear maps may be a too strong restriction for non-linear problems, and the assumption of Gaussian prior also limit the method. To improve the performance, non-linear estimators are also tested here for comparison.

In this study, the computational cost of all the above mentioned methods is kept here in a manageable scale by approximating the dependence of the FEM model solution, i.e., the seabed surface vertical displacements, on the compressibility field via a purely mathematical surrogate or proxy model using the so-called polynomial chaos expansion (PCE). In contrast to the FEM solver, the PCE can be evaluated with little computational cost. We generate a surrogate model for two scenarios: (I) for the case when we restrict the compressibility to be constant throughout the domain, and (II) for a stochastic spatially varying compressibility field input. The dependence of the seabed displacement on the value of the compressibility is smooth, and thus the PCE approximation can be computed by a few runs of the deterministic FEM model. However, when soil conditions are such that, a spatially varying compressibility field has to be assumed, the whole random field of the compressibility has to be updated. To enable the use of the same framework of the surrogate modelling when the random input is not a random variable but a random field, we discuss here separated representations of random fields. In this [paper](#), this discretisation of the compressibility random field is accomplished via a truncated Karhunen-Loève Expansion (KLE). Using the KLE, the compressibility field can be represented as a function of a finite number of independent random variables. With such a representation, the surrogate modelling is done in the same way as in the scalar case, but the dimension of the stochastic problem is higher, which necessitates a higher number of runs of the deterministic FEM solver for the determination of the proxy model. With

the help of the PCE surrogate model, classical stochastic inversion techniques can be applied in an efficient and straight forward manner. Using the PCE framework, the Kalman filter posterior can also be given in a PCE form, resulting in a completely sample-free inversion method [153]. Further improvements can be achieved by identifying non-linear estimators from the minimisation problem but still limiting these maps to some finite dimensional space spanned by some given basis functions. This procedure is described in detail in Chapter 4.

This chapter is organised as follows: Section 8.2 briefly describes the FEM geomechanical model, that is, the deterministic solver and its reformulation using a probabilistic framework. The PCE spectral representation of the model response, that is, the surrogate or proxy model, is explained in Section 8.3. Here, an orthogonal projection is used for the computation of a surrogate model. In the following Section 8.4, the separated representation of a random field is explained through the KLE and the POD theories. Finally, the different data assimilation methods, that is the MCMC, the PCE-based Kalman filter, and the nonlinear filters are explained and compared in Section 8.5 for the reservoir compressibility field identification. We conclude the work in Section 8.6, and also discuss an actual ongoing research outlook, and possibilities of further improvements.

---

## 8.2 Numerical modelling of seabed displacement

Modelling the geomechanical behaviour of a producing reservoir aims at computing the stress and displacements fields generated by pore-pressure changes in space and time due to reservoir activities such as fluid extraction from the subsurface. In this section, the mathematical formulation of the problem is provided, along with its extension to the stochastic dimension. This extension enables us to put forward the uncertainties of the model due to our lack of knowledge of the exact input.

### 8.2.1 The deterministic computation of seabed displacements

The geomechanical analysis of a producing gas reservoir is carried out here by solving the governing partial differential equations of poro-elasticity with the aid of a FEM model. A one-way coupling approach is implemented, assuming that the mechanics-to-fluid coupling is negligibly weak. The pore pressure increment in space and time is first solved for the fluid flow dynamics, and then used as an external source to solve the mechanical equilibrium equations. Here, we just briefly recall the basic formulation and the most important model features. Interested readers are directed to Appendix [A.1](#) for some more details, and to [70] for the whole model description. The equilibrium equations governing the phenomenon of consolidations and the boundary conditions read

$$\begin{aligned} \nabla \cdot \boldsymbol{\sigma}' - \alpha \nabla p &= \rho \mathbf{g} + \mathbf{b} && \text{in } \mathcal{D}, \\ \mathbf{u} &= \mathbf{u}_0 && \text{on } \Gamma_D, \\ \boldsymbol{\sigma} \cdot \mathbf{n} &= \mathbf{t} && \text{on } \Gamma_N, \quad \Gamma_N \cup \Gamma_D = \Gamma, \end{aligned} \tag{8.1}$$

with  $\nabla$  the Nabla operator,  $\boldsymbol{\sigma}'$  and  $\boldsymbol{\sigma}$  the effective and the total stress tensors,  $\alpha$  the Biot's coefficient,  $p$  the pore pressure,  $\rho$  the fluid density,  $\mathbf{g}$  the gravity acceleration,  $\mathbf{b}$  the external body forces, and  $\mathbf{t}$  the total force per unit surface on the Neumann boundary  $\Gamma_N$ . By deriving the weak formulation of the problem and applying a spatial discretisation of the displacement and pressure fields (see

derivation and the equation in a more detailed form in Appendix 7.1), we get a nonlinear system of equations

$$\mathcal{A}(\mathbf{a}, \mathbf{p}) = \Pi(\mathbf{a}) - \mathbf{f}(\mathbf{p}) = \mathbf{0}, \quad (8.2)$$

with the nodal displacements  $\mathbf{a}$  and the nodal pressure values  $\mathbf{p}$ . This equation can be solved for the unknown  $\mathbf{a}$  with the help of a Newton solver by initiating a solution vector  $\mathbf{a}_0$  and solving in a sequential manner the linear system of equations

$$-\mathbf{K}_T \Delta \mathbf{a}_n = \mathcal{A}(\mathbf{a}_n, \mathbf{p}) \quad (8.3)$$

for the increment of nodal displacements  $\Delta \mathbf{a}$ , and computing the new solution vector  $\mathbf{a}_{n+1} = \mathbf{a}_n + \Delta \mathbf{a}_n$ , until Equation (8.2) is satisfied with a residual error whose norm is within a certain threshold. In Equation (8.3)  $\mathbf{K}_T$  is the tangent stiffness matrix, the Jacobian of  $\Pi$ . The important thing here is that  $\mathbf{K}_T$  is scaled with the inverse of the oedometric compressibility  $c_M$  (see Equation (7.48) in Appendix 7.1). According to previous experiments, the compressibility can be described by a power law relation

$$c_M(\sigma_z) = c_{M0} \left( \frac{\sigma'_z}{f_0} \right)^{-\lambda}, \quad (8.4)$$

where  $\sigma'_z$  is the vertical effective stress,  $f_0$  is a fixed value of stress enabling a dimensionless form, and  $\lambda$  and  $c_{M0}$  are material coefficients estimated via measurements, for example, by a radioactive marker technique (RMT). The problem of the actual analysis is that no in-situ measurements for  $\lambda$  and  $c_{M0}$  were available. However, measurements were available for similar case studies, which allowed a pre-calibration of the  $c_M$  model parameters, that is  $c_{M0} = 1.0044 \cdot 10^{-2} \text{ MPa}^{-1}$  and  $\lambda = 1.1347$  with  $f_0 = 1 \text{ MPa}$ . Due to the strong compartmentalisation of the analysed fault formation, the compressibility may vary from block to block of the reservoir. In due course, a horizontally varying function  $f_{c_M}(\hat{x}, \hat{y})^1$  is introduced to scale the compressibility  $c_M^2$ , so Equation (8.4) modifies to

$$c_M(\hat{x}, \hat{y}, \sigma_z) = f_{c_M}(\hat{x}, \hat{y}) c_{M0} \left( \frac{\sigma'_z}{f_0} \right)^{-\lambda}. \quad (8.5)$$

To conclude, for a prediction of the subsidence, our task is to solve for the nodal displacements  $\mathbf{a}$  the nonlinear system of equations (8.2), where the Jacobian depends on the actual value of the stress state through the compressibility parameter  $c_M$ . Figure 8.2 shows a visualisation of this task, in the case of knowing the exact value of the scaling parameter  $f_{c_M}$ . The figure shows that given a specific pressure change, the FEM code solves for the nodal displacements  $\mathbf{a}$ , which can be mapped to the displacement field  $\mathbf{u} = [u_{\hat{x}}, u_{\hat{y}}, u_{\hat{z}}]^T$ . This can be further mapped to the seabed subsidence evaluated at all  $n_y$  points where measurements are available. To be consistent with Chapter 4, let us note that this forward model is a concrete example of the abstract solution operator  $\mathcal{G}$ , that is,

$$\mathbf{y} = \mathcal{G}(f_{c_M}(\hat{x}, \hat{y})) \quad [\mathbf{y}]_i := u_{\hat{z}}(\hat{x}_i, \hat{y}_i) \quad i = 1 \dots n_y. \quad (8.6)$$

In our numerical examples, we used the mesh shown in panel (A) of Figure 8.1 with  $n = 320,901$  mesh nodes. The  $n_y = 60$  assimilation points — where measurements are available — are shown in panels (B) and (D) of the figure. The coordinates of assimilation points were chosen such, that

<sup>1</sup>We use  $\hat{x}$ ,  $\hat{y}$  and  $\hat{z}$  letters for the spatial variables to distinguish it from the measurable response  $\mathbf{y}$  and the measurement  $\mathbf{z}$ .

<sup>2</sup>Please note, it would make sense to introduce another scaling factor, scaling the exponent  $\lambda$ , but for simplicity, for now we stick to the problem of one uncertain field.

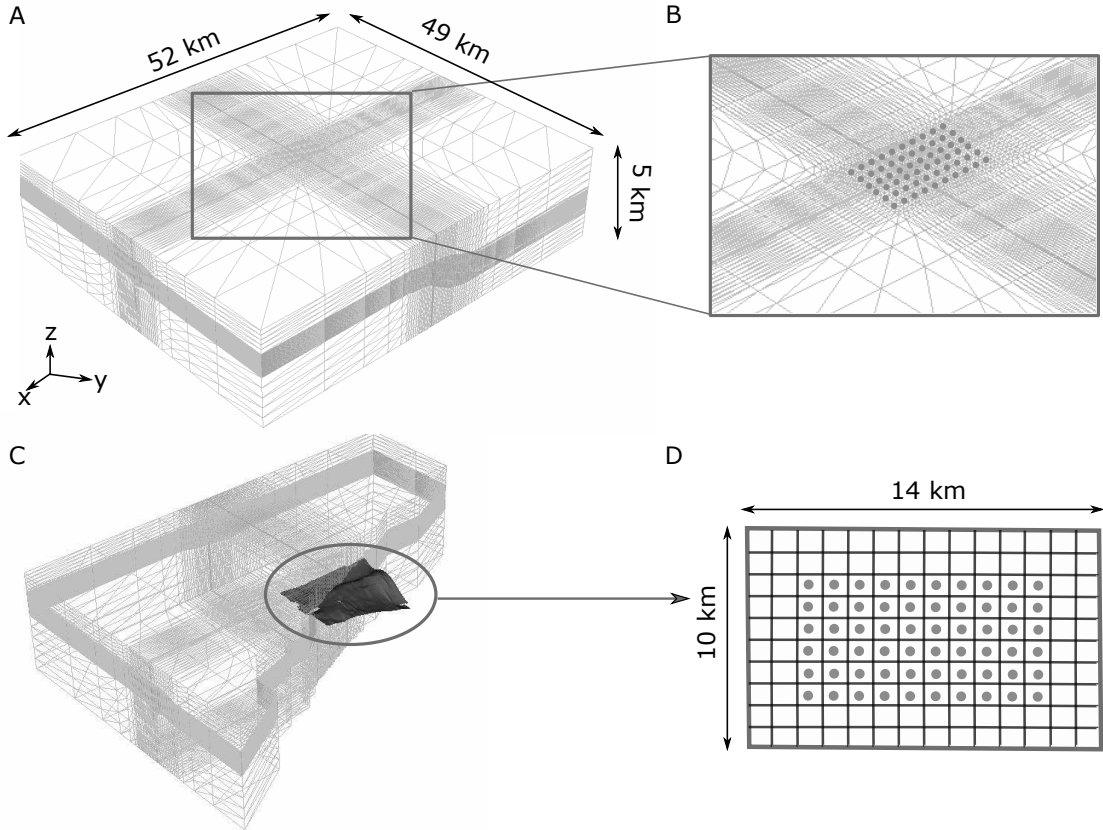


Figure 8.1: (A) Axonometric view of the FEM grid used for the geomechanical simulations, (B) zoom of the central portion of the domain with location of the  $n_y = 60$  data assimilation points on the seabed, (C) 3D section of (A) where the producing layers are highlighted, and (D) discretisation of the compressibility field by a 14x10 square cells with location of the 60-data assimilation points.

they are located at the center points of the FEM elements. The subsurface displacements of the reservoir are mainly influenced by the central portion of the spatial domain shown in panel (D). Our goal was to identify the compressibility field in this central portion  $\mathcal{D}_s$  of the domain.

### 8.2.2 Modified probabilistic formulation

Due to a lack of accurate information on the scaling factor  $f_{c_M}$ , we use the available seabed data of vertical displacements  $\mathbf{y}$  for calibrating this scaling factor in order to have a more accurate prognosis for the seabed subsidence in the future. In other words, the parameter  $f_{c_M}$  should be inferred from measurement data on the history of seabed subsidence  $\mathbf{y}$ . Unfortunately, from a given measured subsidence  $\mathbf{z}_m$  we can not give an explicit form of the parameter  $f_{c_M}$ . That would mean to invert the deterministic solver shown in Figure 8.2, or to be more exact, the operator  $\mathcal{G}$  in Equation (8.6).

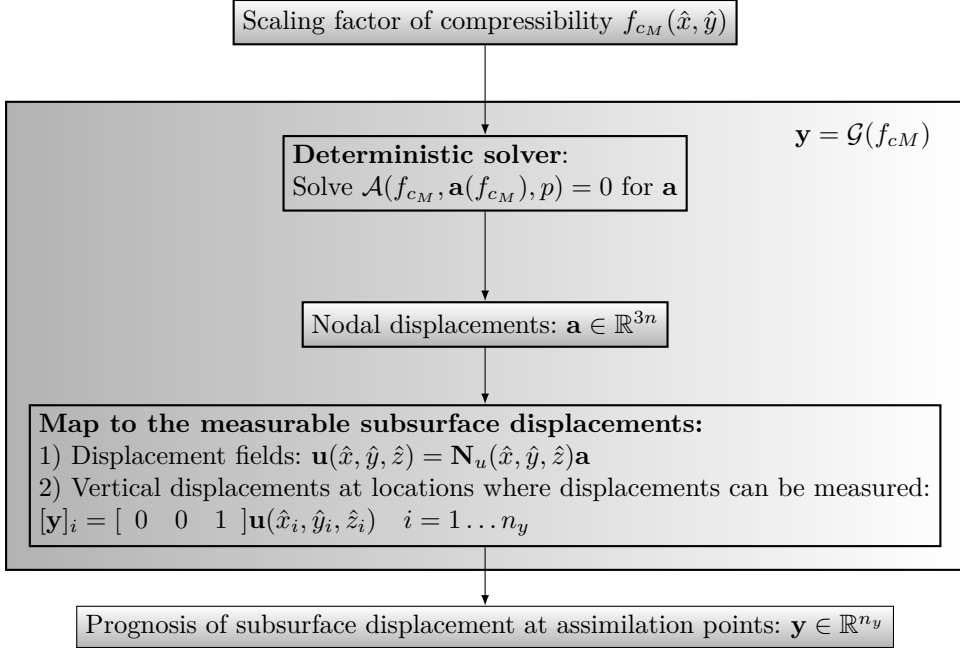


Figure 8.2: Schematic flowchart of the deterministic solver, the computation of subsidence, and the measurable expression.

$\mathcal{G}$  is typically not invertible, and hence the inverse problem is unfortunately an ill-posed one, for example, because there is no solution or no unique solution. Instead of releasing this problem by some rather ad hoc regularisation method, we follow a different path, namely, we put the whole problem in a probabilistic setting, by handling the unknown parameter  $f_{c_M}$  as a random field, that is

$$F_{c_M}(\hat{x}, \hat{y}, \omega) : \mathcal{D}_s \times \Omega \rightarrow \mathbb{R}, \quad (8.7)$$

where  $\omega^3$  is an event or a realisation from the space of all possible outcomes  $\Omega$ , and  $\mathcal{D}_s$  is the crucial part of the spatial domain shown in the (D) panel of Figure 8.1<sup>4</sup>. To represent our knowledge about the possible values of  $F_{c_M}$ , we assign to it some *a priori* distribution function  $\pi_F(f_{c_M})$ , and use the Bayesian inversion to update this *a priori* distribution using the measurement data  $\mathbf{z}_m$ .

First, let us choose the *a priori* distribution function of the scaling factor. This should be based on some professional geotechnical expertise. Careful attention has to be paid so that we do not modify the problem in such a way that we ruin the essential properties of the physics. In our case, it is important that the matrix  $\mathbf{C}$  stays positive definite, as naturally a negative compressibility would not make any sense. To keep this important property, we use the semi-bounded Lognormal distribution with support  $I_F = (0, +\infty)$ . The fact that  $\lambda$  and  $c_{M0}$  were measured from similar case

<sup>3</sup>One can think of  $\omega$  as an abstract notation showing random nature (see more in Chapter 4).

<sup>4</sup>Furthermore we assign capital letters for the random variables and random fields, and small letters for a realisation of them. For example  $F_{c_M}$  is a random variable and one realisation of it is  $F_{c_M}(\omega_i) = f_{c_M,i}$ .

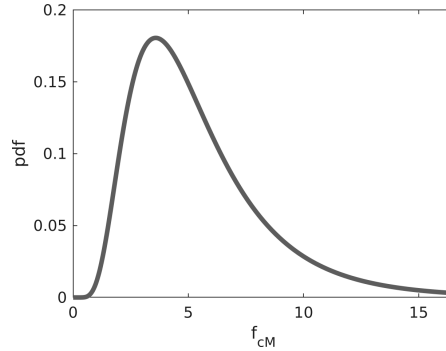


Figure 8.3: Lognormal prior probability distribution of  $f_{cm}$ .

studies would suggest a constant expected value of the scaling factor equal to one. Nevertheless, due to the experienced subsidence being much higher than what the field experts expected, we assumed a higher expected value of 5.5, that is

$$\mathbb{E}[F_{cM}] = \int_{I_F} f_{cM} \pi_F(f_{cM}) \, df_{cM} = 5.5, \tag{8.8}$$

and a variance chosen in such a way that with 99.5% probability  $f_{cM} < 10$ . Accordingly, we define the prior distribution of the scaling factor to be

$$F_{cM} \sim \ln\mathcal{N}(\mu, \sigma^2 \Sigma), \quad \mu(\hat{x}, \hat{y}) = 1.562, \quad \sigma(\hat{x}, \hat{y}) = 0.534. \tag{8.9}$$

We assumed a constant variance and mean over the spatial domain.  $\Sigma$  is the correlation function of the random field. The marginal distribution at all spatial points is shown in Figure 8.3. We consider two different scenarios:

- (I) The first, and simpler, scenario assumes that the compressibility field is spatially constant throughout the spatial domain. In such a case, the random field is fully correlated, which means that for any two points with coordinates  $(\hat{x}, \hat{y})$  and  $(\hat{x}', \hat{y}')$ , the correlation is one

$$\Sigma(\hat{x}, \hat{y}, \hat{x}', \hat{y}') = 1; \tag{8.10}$$

and thus the scaling factor can be described by one scalar random variable.

- (II) The second scenario allows the update to identify a spatially varying compressibility field. Although in this way the complexity of the model is higher, the strong compartmentalisation suggests that the compressibility may not be constant throughout the spatial domain. We assume that at the analysed height the compressibility field is smoothed out and thus its values at different spatial points are strongly correlated, and that this correlation depends only on the distance  $d$  in between the points. According to geotechnical expertise, the dependence can be

described by the Matérn covariance function<sup>5</sup>  $C_{\nu_c}(d/l_c)$

$$\Sigma(\hat{x}, \hat{y}, \hat{x}', \hat{y}') = C_{\nu_c}(d/l_c)^5, \quad \nu_c = 2, \quad l_c = 4000m, \quad (8.12)$$

where  $\nu_c$  is a non-negative parameter of the covariance function—influencing the smoothness of the field realisations,  $d$  is the distance  $\sqrt{(\hat{x} - \hat{x}')^2 + (\hat{y} - \hat{y}')^2}$ , and  $l_c$  is the correlation length. The smaller the correlation length is, the faster the covariance values decay with the distance and the wilder the realisations of the field can get.

From a mathematical point of view, it is much nicer to work with variables, which are in a vector space (so that any linear combination of the variable is also in the same space). For this purpose, we rather represent the field  $F_{cM}$  as a function of some underlying Gaussian random field. The map from the Gaussian to the Lognormal random field is straightforward:

$$F_{cM}(\Theta(\hat{x}, \hat{y}, \omega)) = e^{\mu + \sigma\Theta} \quad \Theta \sim \mathcal{N}(0, \Sigma), \quad (8.13)$$

where  $\Theta : \mathcal{D}_s \times \Omega \rightarrow \mathbb{R}$  is the underlying Gaussian field. To further simplify the description of the stochastic input, we write the Gaussian field as a function of some random variables  $\mathbf{Q}$

$$\Theta(\hat{x}, \hat{y}, \omega) = \mathcal{F}(\mathbf{Q}(\omega)). \quad (8.14)$$

This task is trivial for a spatially constant field, that is for the problem description (I). In this scenario, the field  $\Theta$  is fully correlated and thus independent of the spatial variables  $\hat{x}$  and  $\hat{y}$ , and accordingly can be written as a standard Gaussian random variable, so  $\mathbf{Q}$  is just one standard Gaussian random variable  $Q$ , and  $\mathcal{F}$  is simply the identity map

$$\Theta(\hat{x}, \hat{y}, \omega) = \mathcal{F}(Q) = Q(\omega) \quad Q \sim \mathcal{N}(0, 1). \quad (8.15)$$

In the case of a spatially varying field, the task is a bit more complicated. Fortunately, with the help of the Karhunen-Loève Expansion (KLE, see Section 8.4) we can represent any ‘decent’ stochastic field as a linear combination of the product of some square integrable spatial eigenfunctions  $r_i(\hat{x}, \hat{y})$  and Gaussian independent random variables  $X_i$  in the form

$$\Theta(\hat{x}, \hat{y}, \omega) \approx \mu_\Theta(\hat{x}, \hat{y}) + \sum_{i=1}^L \sigma_i r_i(\hat{x}, \hat{y}) X_i(\omega), \quad (8.16)$$

where  $\mu_\Theta$  is the mean of the field  $\Theta$ , which is due to Equation (8.13) is constant zero. For the properties and the numerical computation of the KLE, see Section 8.4. Setting these  $X_i$  standard Gaussian random variables to be  $Q$ , we have the desired set of input random variables and the corresponding  $\mathcal{F}$  map

$$\mathcal{F}(\mathbf{Q}) = \mu_\Theta(\hat{x}, \hat{y}) + \sum_{i=1}^L \sigma_i r_i(\hat{x}, \hat{y}) Q_i(\omega), \quad \mathbf{Q} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_L), \quad (8.17)$$

---

<sup>5</sup>The Matérn covariance function reads

$$C_{\nu_c}(d/l_c) = \sigma^2 \frac{2^{1-\nu_c}}{\Gamma(\nu_c)} \left( \sqrt{2\nu_c} \frac{d}{l_c} \right)^{\nu_c} K_{\nu_c} \left( \sqrt{2\nu_c} \frac{d}{l_c} \right) \quad (8.11)$$

where  $\Gamma$  is the gamma function, and  $K_{\nu_c}$  is the modified Bessel function of the second kind.



with  $\mathbf{I}_L$  being the identity matrix of size  $L \times L$ . Later, in the update process, instead of directly updating the  $F_{cM}$  field, we update first these random variables  $\mathbf{Q}$ , and then use the above defined maps to compute the updated scaling factor. Such a general framework enables the algorithms to be used not only for the task to update a field input, but also for the identification of a set of uncertain parameters. In such a case, the  $\mathbf{Q}$  parameters can directly be the uncertain parameters if they are Gaussian.

With the new probabilistic description of the scaling factor

$$F_{cM}(\hat{x}, \hat{y}, \omega) = e^{\mu + \sigma \mathcal{F}(\mathbf{Q}(\omega))} \quad (8.18)$$

the modified formulation of the compressibility in Equation (8.5) reads

$$C_M(\sigma'_z, \hat{x}, \hat{y}, \omega) = F_{cM}(\hat{x}, \hat{y}, \omega) c_{M0} \left( \frac{\sigma'_z}{f_0} \right)^{-\lambda}. \quad (8.19)$$

Within the probabilistic framework, the nonlinear operator given in Equation (8.2) becomes a stochastic operator, as it depends on the actual realisation of the random field  $F_{cM}$ ,

$$\mathcal{A}_s(F_{cM}(\hat{x}, \hat{y}, \omega), \mathbf{a}(F_{cM}(\hat{x}, \hat{y}, \omega)), \mathbf{p}) = 0, \quad (8.20)$$

or in a shorter form

$$\mathcal{A}_s(\omega, \mathbf{a}(\omega), \mathbf{p}) = 0. \quad (8.21)$$

Naturally, in this way the nodal displacements  $\mathbf{a}$  and accordingly, the displacement field  $U_z = U_z(\hat{x}, \hat{y}, \omega)$  are also random expressions, depending on the actual realisation  $\omega$  of the random vector  $\mathbf{Q}$ . For the assimilation of the compressibility field, we use measurements of the subsurface displacements  $\mathbf{Y} = \mathbf{Y}(\mathbf{Q}(\omega))$ , which is also a random vector. This prediction of the displacements computed by the FEM solver has to be compared with the measured data  $\mathbf{z}_m$ . However, the mathematical model is just a simplification of reality, which means that the true displacement field can differ from the predicted value, even if we know the exact compressibility field. Furthermore, our measurements are usually poisoned by some measurement errors. Supposing an additive measurement noise and modelling error, the measurement can be written as

$$\mathbf{Z}(\omega) = \mathbf{Y}(\omega) + \mathbf{E}_{FEM}(\omega) + \mathbf{E}(\omega), \quad (8.22)$$

where  $\mathbf{E}_{FEM}$  and  $\mathbf{E}$  are the random vectors of the modelling error of the FEM code and of the measurement noise, respectively. As for this chapter, we do not use real measurements, but virtual ones that we generate with the FEM model, we can ignore the modelling error, so the measurement model reads

$$\mathbf{Z} = \mathbf{Y} + \mathbf{E}. \quad (8.23)$$

In our examples, we suppose that the measurement error is a mean-free Gaussian random vector

$$\mathbf{E} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_E), \quad [\mathbf{C}_E]_{jj} = \sigma_{\epsilon,j}^2 = (0.15 \cdot [\mathbf{z}_m]_j)^2, \quad \pi_E(\boldsymbol{\epsilon}) = \prod_{j=1}^{60} \frac{1}{\sqrt{2\pi\sigma_{\epsilon,j}^2}} e^{-\frac{\epsilon_j^2}{2\sigma_{\epsilon,j}^2}}, \quad (8.24)$$

with  $\mathbf{C}_E \in \mathbb{R}^{60 \times 60}$  the measurement error covariance matrix, a diagonal matrix so that the measurement errors at the different locations are independent of each other.  $\sigma_{\epsilon,j}$  is the standard deviation of

the measurement error at the  $j^{\text{th}}$  assimilation point and is modelled proportional to the measured value of the displacements  $\mathbf{z}_m$ .  $\pi_E$  is the probability density function of the measurement noise model, and  $[\boldsymbol{\epsilon}]_j = \epsilon_j$  is a realisation of the measurement error at the  $j^{\text{th}}$  assimilation node.

For real-life applications, a modelling error is advisable to be also included in the measurement model. This error is in general a correlated random field, and its properties are not known in general. For such problems, a different error model can be included. Then the assimilation will include the identification of these parameters of the modelling error. If we rather suppose that the modelling error is just a white noise, then it can be handled together with the measurement noise model. The danger of ignoring a systematic modelling error can result in an update process that compensates a missed-out term by an improper shift of the uncertain input parameter(s).

---

### 8.3 Surrogate modelling

To mitigate the high computational cost of the different update procedures, it is advantageous to build a surrogate model which mimics the expensive FEM solver with an approximate, but computationally cheap, mathematical model. The surrogate model allows a fast evaluation of the displacements  $\mathbf{y}$  for a specific realisation  $\mathbf{q}$  of  $\mathbf{Q}$ , or indirectly for a specific realisation of the scaling factor  $F_{cM}$ . This is schematically shown in Figure 8.4. Depending on the representation of the surrogate, an evaluation of statistics and sensitivities is often also very cheap. In the following, we describe the so-called Polynomial Chaos Expansion (PCE) for the surrogate modelling.

For mathematical convenience, we represent all random expressions with some fixed ‘reference random variables’. We choose these variables to be a set of uncorrelated Gaussian random variables  $\{\Xi_i\}_{i=1}^S$  collected in a random vector  $\boldsymbol{\Xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  with distribution  $\pi_{\boldsymbol{\Xi}}$  given by

$$\pi_{\boldsymbol{\Xi}}(\boldsymbol{\xi}) = \prod_{j=1}^S \frac{1}{\sqrt{2\pi}} e^{-\frac{\xi_j^2}{2}} \quad (8.25)$$

As the variables  $\Xi_i$  are Gaussian and uncorrelated, they are independent as well. One can think of  $\boldsymbol{\Xi}$  as an orthogonal coordinate system, such as the local, reference Cartesian coordinate system in the FEM procedure, only that this one is in the parametric and not in the spatial domain. For different types of distributions of the uncertain parameters or fields, we can define functions that map the reference random variables  $\boldsymbol{\Xi}$ —also called the *germ*—to the uncertain parameters or fields of interest. In our example, the parameter  $\mathbf{Q}$  is a set of independent random variables and thus it could directly be the germ. The reason to introduce the parameter  $\mathbf{Q}$  besides the germ (see Figure 8.4) is because we will need to modify this parameter in the calibration process, by obtaining an updated map from the germ to the parameter  $\mathbf{Q}$ . A practical and efficient approach is to make a surrogate of  $\mathbf{y} = G(f_{cM}(\mathbf{Q}(\boldsymbol{\xi})))$  in the form of a linear combination of some multivariate stochastic polynomials  $\{\Phi_i(\xi_1, \xi_2, \dots, \xi_s)\}_{i=0}^{M-1}$ . Also for mathematical convenience, we choose orthogonal polynomials with respect to the Gaussian density, satisfying the orthogonality condition

$$\mathbb{E}[\Phi_i(\boldsymbol{\xi})\Phi_j(\boldsymbol{\xi})] = \int_{\mathbb{R}^n} \Phi_i(\boldsymbol{\xi})\Phi_j(\boldsymbol{\xi})\pi_{\boldsymbol{\Xi}}(\boldsymbol{\xi})d\boldsymbol{\xi} = \gamma_i\delta_{ij}, \quad (8.26)$$

where  $\delta_{ij}$  is the Kronecker delta, and  $\gamma_i$  is the squared norm of the polynomials, that is

$$\gamma_i = \mathbb{E}[\Phi_i(\boldsymbol{\xi})\Phi_i(\boldsymbol{\xi})]. \quad (8.27)$$

The polynomials orthogonal with respect to the Gaussian density  $\pi_{\Xi}$  are the so-called Hermite polynomials<sup>6</sup>. Then the measurable displacement  $\mathbf{y}$  can be written as a linear combination of these polynomials  $\Phi_i$  with coefficients  $\mathbf{v}_i$

$$\mathbf{y}_h = \sum_i \mathbf{v}_i \Phi_i = \boldsymbol{\Upsilon} \boldsymbol{\Phi}_M, \quad (8.28)$$

where in the last expression we have collected the polynomials in a vector  $\boldsymbol{\Phi}_M = [\Phi_0, \Phi_1, \dots, \Phi_{M-1}]^T$  and the PCE coefficients of the vertical displacement  $\mathbf{v}_i \in \mathbb{R}^{n_y}$  in a matrix  $\boldsymbol{\Upsilon} \in \mathbb{R}^{n_y \times M}$ , whose  $j^{\text{th}}$  row corresponds to the PCE coefficients of the displacement at the  $j^{\text{th}}$  assimilation point and its  $i^{\text{th}}$  column corresponds to the  $i^{\text{th}}$  stochastic basis function.

Setting up the surrogate model consists of two steps:

1. Represent the input random field  $f_{CM}$  as a function of the reference random variable  $\boldsymbol{\xi}$  (see Figure (8.4)).
2. Determine the surrogate model by computing the coefficients  $\mathbf{v}_i$  of the expansion of  $\mathbf{y}_h$  in Equation (8.28).

For the first task, we have already represented (Equation (8.18)) the field  $F_{CM}$  with a set of  $L$  (in the first scenario  $L = 1$ ) Gaussian random variables  $\mathbf{Q}$ . Now we only need to express  $\mathbf{Q}$  in the reference coordinate system, a set of independent standard Gaussian random variables. As in our example,  $\mathbf{Q}$  is already a set of independent random variables, for its description we need  $S = L$  reference random variables and because the elements of  $\mathbf{Q}$  have already standard Gaussian distribution, the relationship can be given now by simply providing the identity map

$$\mathbf{Q} = \mathcal{I}(\boldsymbol{\Xi}) = \boldsymbol{\Xi}. \quad (8.29)$$

The second task, the computation of the coefficients  $\mathbf{v}_i$  can be done by different approaches, which are usually classified into so-called intrusive and non-intrusive methods. Intrusive methods can sometimes lead to more stable approximations, but may need modifications of the deterministic solver and are thus harder to implement. Non-intrusive methods use the deterministic solver as it is, doing only pointwise evaluations. Interested readers are directed to [179] for a detailed description of these methods. Here, we choose a relatively straightforward non-intrusive approach, the orthogonal projection, which will be described in the next section.

### 8.3.1 Computation of the surrogate by orthogonal projection

The target here is to find a best approximation of  $\mathbf{y}(\boldsymbol{\xi})$  in the form of a linear combination of the stochastic polynomials  $\Phi_i(\boldsymbol{\xi})$ , where ‘best’ is meant in the least squares sense. Therefore, we want

---

<sup>6</sup>For non-Gaussian random vector  $\mathbf{Q}$  it is also possible to use a set of non-Gaussian reference random variables, and thus the orthogonal polynomials are not Hermite polynomials but ones that are orthogonal with respect to the probability density of these reference variables [192], in this case the approximation is termed ‘Generalized Polynomial Chaos Expansion’ (gPCE). Because some of the later described update methods are restrained to the update of Gaussian random variables we require here the input parameter  $\mathbf{Q}$  to be Gaussian.

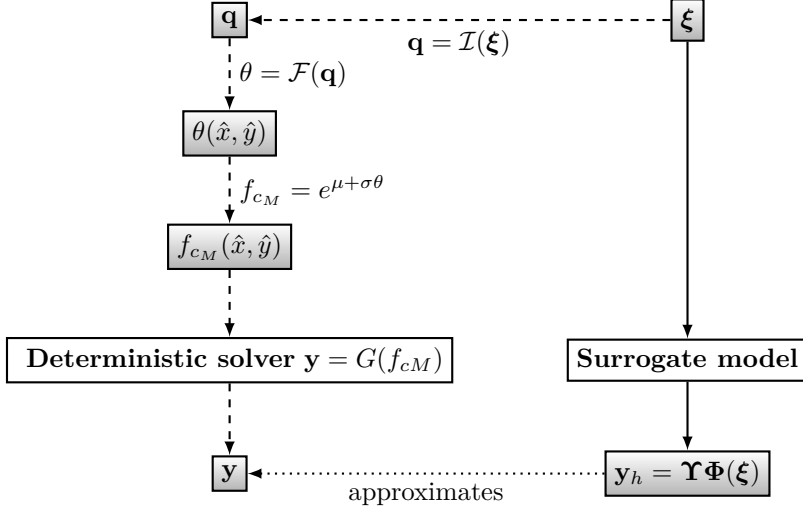


Figure 8.4: Replacing the deterministic solver by a PCE surrogate model: the dashed line shows the path without a surrogate model and the continuous line shows the one when a surrogate model is used.  $f_{c_M}$  is the random field input,  $\theta$  is the underlying Gaussian field,  $\mathbf{q}$  is a Gaussian random vector representing the field  $\theta$ ,  $\boldsymbol{\xi}$  is the germ of the surrogate, a set of independent standard Gaussian random variables serving as the reference coordinate system of the stochastic dimension,  $\mathbf{y}$  is the measurable response, and  $\mathbf{y}_h$  is its surrogate approximation.

to choose the coefficients such that the expression

$$\mathbb{E} [\epsilon_h^2] = \mathbb{E} [(\mathbf{y} - \mathbf{y}_h)^2] = \int_{\mathbb{R}^s} (\mathbf{y}(\boldsymbol{\xi}) - \mathbf{y}_h(\boldsymbol{\xi}))^2 \pi_{\Xi}(\boldsymbol{\xi}) \, d\boldsymbol{\xi} \quad (8.30)$$

is minimised. It is derived in Appendix 4 that the minimisation problem leads to the following equation for the coefficients

$$\mathbf{v}_i = \frac{1}{\gamma_i} \mathbb{E} [\mathbf{y} \Phi_i] = \frac{1}{\gamma_i} \int_{\mathbb{R}^s} \mathbf{y}(\boldsymbol{\xi}) \Phi_i(\boldsymbol{\xi}) \pi_{\Xi}(\boldsymbol{\xi}) \, d\boldsymbol{\xi}. \quad (8.31)$$

Unfortunately, this integral expression can not be computed in closed form, as the dependence of  $\mathbf{y}$  on  $\boldsymbol{\xi}$  is through the simulation and cannot be explicitly given. However, since we can compute the displacement  $\mathbf{y}$  at specific values of  $\boldsymbol{\xi}$  by the FEM solver, the integral expression can be evaluated numerically by the quadrature rule

$$\mathbf{v}_i \approx \frac{1}{\gamma_i} \sum_{j=1}^N w_j \mathbf{y}(\boldsymbol{\xi}_j) \Phi_i(\boldsymbol{\xi}_j) \quad (8.32)$$

where  $N$  is the number of integration points and  $\boldsymbol{\xi}_j$  and  $w_j$  are the points and weights of the integration rule. For an overview of the integration rules for stochastic and high-dimensional problems,

see example [179]. For the reservoir problem here, we used mostly sparse Smolyak grids with Gauss-Hermite quadrature rules. To sum up, the algorithm to compute the coefficients of the surrogate model of  $\mathbf{y}$  is as follows:

**Algorithm 11: Computation of the gPCE coefficients by orthogonal projection**

1. Define the prior distribution of the input field  $F_{cM}(\hat{x}, \hat{y})$  and determine the map from the set of independent reference random variables  $\boldsymbol{\xi} = [\xi_1, \xi_2, \dots, \xi_L]^T$  to  $f_{cM}$ .
2. Specify the orthogonal basis polynomials  $\{\Phi_i\}_{i=0}^{M-1}$  used for the expansion (see Appendix 2.1).
3. Compute the squared norms  $\gamma_i$  of the basis polynomials (see Appendix 2.2 for the norms of univariate polynomials and Appendix 2.1 for multivariate polynomials).
4. Get the integration points  $\boldsymbol{\xi}_j$  and weights  $w_j$  for all  $j = 1 \dots N$  from a suitable integration rule.
5. Map the integration points  $\boldsymbol{\xi}_j$  to the uncertain field  $f_{cM,j}(\hat{x}, \hat{y})$  by the map defined in point 1.
6. Compute the measurable response  $\mathbf{y}$  by the deterministic solver with the field  $f_{cM,j}(\hat{x}, \hat{y})$  for all  $j = 1 \dots N$ .
7. Evaluate all basis functions  $\{\Phi_i\}_{i=0}^{(M-1)}$  at all the integration points  $\{\boldsymbol{\xi}_j\}_{j=1}^N$ .
8. Compute the gPCE coefficients from  $\mathbf{v}_i \approx \frac{1}{\gamma_i} \sum_{j=1}^N w_j \mathbf{y}(\boldsymbol{\xi}_j) \Phi_i(\boldsymbol{\xi}_j)$  and collect them in the matrix  $\mathbf{Y} = [\mathbf{v}_0, \dots, \mathbf{v}_{M-1}]$ .

The gPCE of  $\mathbf{y}$  then reads  $\mathbf{y}_h(\boldsymbol{\xi}) = \mathbf{Y}\boldsymbol{\Phi}_M(\boldsymbol{\xi})$ , where  $\boldsymbol{\Phi}_M$  is a vector collecting the basis functions defined in point 2, or the same in function of the  $\mathbf{Q}$  input parameter using Equation (8.29):

$$\mathbf{y}_h(\boldsymbol{\xi}) = \mathbf{Y}\boldsymbol{\Phi}_M(\mathcal{I}^{-1}(\mathbf{q})) \tag{8.33}$$

**Example 1: Surrogate model of seabed displacement by orthogonal projection.** Here, we present an example for determining a PCE surrogate model for one nodal displacement computed by orthogonal projection. The example is developed for the univariate case, that is, for the problem where we assume a homogeneous scaling factor. (See Examples 3 and 4 when the input is a random field.)

1. **Define the prior distribution of the random field  $F_{cM}$  and determine the map from the germ to  $F_{cM}$ .**

The prior distribution of  $F_{cM}(\hat{x}, \hat{y}, \omega)$  is defined in Equation (8.9). The variables  $\theta$  and  $Q$  and the maps in between are defined in Equations (8.15), (8.18), and (8.29). As is shown there, for

the homogeneous scenario,  $q = \xi$  is just one standard Gaussian random variable. The map from the germ to the input field is defined by

$$f_{cM}(\xi) = e^{\mu + \sigma\xi} = e^{1.562 + 0.534\xi}. \quad (8.34)$$

## 2. Specify the orthogonal basis polynomials used for the expansion of $\mathbf{y}$ .

As we have only one variable  $\xi$ , we use univariate polynomials, and as we have Gaussian germ, we use the Hermite ones. We choose here a polynomial basis of maximum degree three and generate the polynomials using the three-term recurrence relation (see Appendix 2.1). The vector of PCE basis functions is then

$$\Phi_4 = \begin{bmatrix} H_0(\xi) \\ H_1(\xi) \\ H_2(\xi) \\ H_3(\xi) \end{bmatrix} = \begin{bmatrix} 1 \\ \xi \\ \xi^2 - 1 \\ \xi^3 - 3\xi \end{bmatrix}. \quad (8.35)$$

## 3. Compute the squared norms $h_i$ of the polynomials.

The norms of the polynomials can be also computed easily from the sequences of the three-term recurrence relation (see Appendix 2.2). For our Hermite polynomials the norms read

$$\gamma_i = h_i = i! \quad \gamma_0 = h_0 = 1 \quad \gamma_1 = h_1 = 1 \quad \gamma_2 = h_2 = 2 \quad \gamma_3 = h_3 = 6. \quad (8.36)$$

## 4. Get the integration points and weights.

We approximate the integral (8.31) by numerical integration. As the variable  $\xi$  is Gaussian, we use the Gauss-Hermite quadrature rules [71]. The integration points and the weights can be computed from an eigenvector problem given in Appendix 2.3. If we assume that  $y_k(\xi)$  can be well approximated by polynomials of maximum degree  $d = 3$ , then the  $y_k(\xi)\Phi_i(\xi)$  term can be approximated by a polynomial of maximum degree  $2d = 6$ . Polynomials of maximum degree  $2N - 1$  can be integrated exactly by an  $N$  point Gauss integration rule. Accordingly, we choose the number of points such that  $2N - 1$  is bigger or equal to  $2d = 6$ , that is, we go with a four-point rule  $N = d + 1 = 4$ , corresponding to the roots of the  $\Phi_4(\xi)$  polynomial. The integration points and weights for the four-point Gauss-Hermite rule are

$$\begin{array}{cccc} \xi_1 = -2.3344 & \xi_2 = -0.7420 & \xi_3 = 0.7420 & \xi_4 = 2.3344, \\ w_1 = 0.0459 & w_2 = 0.4541 & w_3 = 0.4541 & w_4 = 0.0459. \end{array} \quad (8.37)$$

## 5. Map the integration points $\xi_j$ to $f_{cM,j}$ .

The  $\xi_j$  integration points are mapped to  $f_{cM,j}(\hat{x}, \hat{y})$  by Equation (8.34)

$$f_{cM,1} = 1.3694 \quad f_{cM,2} = 3.2073 \quad f_{cM,3} = 7.0885 \quad f_{cM,4} = 16.6019. \quad (8.38)$$

## 6. Compute the measurable response.

This step is the computationally expensive one. Now we have four different homogeneous scaling factors. We have to call the deterministic solver  $N = 4$  times to get the nodal displacements for the different scaling values. Here, we only show the results for one specific nodal displacement

( $k = 25$ , which corresponds to the node in the third row from the bottom and the fifth node from the left of the nodes shown by the blue dots in the D panel of Figure 8.1).

$$y_k(\xi_1) = -0.0015 \quad y_k(\xi_2) = -0.0038 \quad y_k(\xi_3) = -0.0087 \quad y_k(\xi_4) = -0.0206. \quad (8.39)$$

The numbers are in meters and the negative sign means a downward displacement.

### 7. Evaluate the basis functions at all integration points.

The basis functions evaluated at the integration points are

$$\Phi_4(\xi_1) = \begin{bmatrix} 1.0000 \\ -2.3344 \\ 4.4495 \\ -5.7181 \end{bmatrix} \quad \Phi_4(\xi_2) = \begin{bmatrix} 1.0000 \\ -0.7420 \\ -0.4495 \\ 1.8174 \end{bmatrix} \quad \Phi_4(\xi_3) = \begin{bmatrix} 1.0000 \\ 0.7420 \\ -0.4495 \\ -1.8174 \end{bmatrix} \quad \Phi_4(\xi_4) = \begin{bmatrix} 1.0000 \\ 2.3344 \\ 4.4495 \\ 5.7181 \end{bmatrix}. \quad (8.40)$$

### 8. Compute the PCE coefficients.

The PCE coefficients of the 25<sup>th</sup> node which are written in the 25<sup>th</sup> row of the  $\mathbf{Y}$  coefficient matrix

$$\mathbf{Y} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ -0.0067 & -0.0037 & -0.0010 & -0.0002 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}. \quad (8.41)$$

As the response surface – the dependence of the  $\mathbf{y}$  displacement on  $\xi$  – is a smooth function, the absolute values of the PCE coefficients decay fast.

#### 8.3.2 Computation of statistics

Statistics like the mean and the variance of the displacements can be computed cheaply from the surrogate model, due to the advantageous orthogonality property of the basis. The mean of the displacements  $\mathbf{Y}$ , for example, can be computed directly from the PCE coefficients corresponding to the zeroth polynomial  $\Phi_0 = 1$  as

$$\begin{aligned} \mathbb{E}[\mathbf{Y}] &\approx \mathbb{E}[\mathbf{Y}_h] = \mathbb{E} \left[ \sum_{i=0}^{M-1} \mathbf{v}_i \Phi_i(\boldsymbol{\Xi}) \right] \\ &= \sum_{i=0}^{M-1} \mathbf{v}_i \mathbb{E}[\Phi_i(\boldsymbol{\Xi})] = \mathbf{v}_0, \end{aligned} \quad (8.42)$$

because of  $\mathbb{E}[\Phi_0] = 1$  and  $\mathbb{E}[\Phi_{i \neq 0}] = 0$  due to the orthogonality condition (8.26). The autocovariance of  $\mathbf{Y}$  can be computed from the rest of the coefficients by

$$\begin{aligned} \text{cov}[\mathbf{Y}, \mathbf{Y}] &\approx \text{cov}[\mathbf{Y}_h, \mathbf{y}_h] = \mathbb{E} \left[ \left( \sum_{i=0}^{M-1} \mathbf{v}_i \Phi_i(\boldsymbol{\Xi}) - \mathbb{E}[\mathbf{y}_h] \right) \left( \sum_{i=0}^{M-1} \mathbf{v}_i \Phi_i(\boldsymbol{\Xi}) - \mathbb{E}[\mathbf{y}_h] \right)^T \right] \\ &= \sum_{i=1}^{M-1} \mathbf{v}_i \mathbf{v}_i^T \gamma_i, \end{aligned} \quad (8.43)$$

again because of the orthogonality condition (8.26). Suppose, the random variable  $\mathbf{Q}$  is also given with a PCE form in the same PCE basis

$$\mathbf{Q}_h = \sum_{i=0}^{M-1} \hat{\mathbf{Q}}_i \Phi_i(\Xi). \quad (8.44)$$

Then the covariance  $\text{cov}[\mathbf{Q}, \mathbf{Y}]$  can be approximated by

$$\text{cov}[\mathbf{Q}_h, \mathbf{Y}_h] = \sum_{i=1}^{M-1} \hat{\mathbf{q}}_i \mathbf{v}_i^T \gamma_i \quad (8.45)$$

**Example 2: Statistics from PCE expansion.** In this example, we show how to compute mean and variance of the  $k$ th nodal displacement resulting from the prior uncertainties of the scaling factor. We use the PCE coefficients derived in Example 1 for the computation. The mean is computed from Equation (8.42), which is directly the zeroth coefficient

$$\mathbb{E}(y_k) = v_{k,0} = v_{0k} = -0.0067. \quad (8.46)$$

The variance of the displacement can be computed from Equation (8.43), from the square of the rest of the coefficients

$$\text{var}[y_{h,25}, y_{h,25}] = \sum_{i=1}^3 v_{25,i}^2 h_i = (-0.0037)^2 \cdot 1 + (-0.0010)^2 \cdot 2 + (-0.0002)^2 \cdot 6 = 1.5678e-05. \quad (8.47)$$

The displacement at the 25<sup>th</sup> assimilation point can also be computed at any value of the  $f_{c_M}$  scaling factor. One computation of the nodal displacements take hours, but evaluation of the surrogate model doesn't even take seconds. The PCE of  $y_{25}$  is

$$y_{h,25}(\xi) = [-0.0067 \quad -0.0037 \quad -0.0010 \quad -0.0002] \begin{bmatrix} 1 \\ \xi \\ \xi^2 - 1 \\ \xi^3 - 3\xi \end{bmatrix}. \quad (8.48)$$

For example, when  $f_{c_M} = 4.7681$ , the nodal displacement can be computed from

$$\xi = \frac{\ln(4.7681) - \mu}{\sigma} = 0, \quad (8.49)$$

$$y_{h,k}(0) = [-0.0067 \quad -0.0037 \quad -0.0010 \quad -0.0002] \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix} = -0.0057. \quad (8.50)$$

### 8.3.3 Validating surrogate models

Often, prior information on the smoothness of the response  $y$  as a function of  $\xi$  is not available, which makes it difficult to come up with a good idea of which PCE basis one should use, and by which method the coefficients should be computed. For this reason, it is always recommended to carry



Table 8.1: Validation of different degree surrogate models with the normed relative error (values are in percentage) computed on  $N_v = 10$  quasi-Monte Carlo validation points.

PCE degree $d$	1	2	3	4	5
Number of solver calls $N$	2	3	4	5	6
Normed rel. error $\ \epsilon_{rh}\ _2$	115.48	26.32	5.21	1.47	1.16

out some kind of validation of the computed PCE approximation. Here, we use a set of validation points  $\{\xi_j\}_{j=1}^{N_v}$  sampled from the distribution of  $\Xi$  using a quasi-random sampling method, in this case the Halton-sequence. First, the responses are computed by the FEM solver and then the error is evaluated at these validation points. The relative averaged squared error in the  $k$ th spatial node

$$\epsilon_{rh,k}^2 = \frac{\sum_{j=1}^{N_v} ([\mathbf{y}(\xi_j)]_k - [\mathbf{y}_h(\xi_j)]_k)^2}{\sum_{j=1}^{N_v} ([\mathbf{y}(\xi_j)]_k)^2}, \tag{8.51}$$

is then evaluated and compared to decide which surrogate model to use for the inverse method. Table 8.1 shows the  $\|\epsilon_{rh}\|_2$  normed relative errors with proxy models of different degree polynomial bases. In the later identification process, we used polynomial degree five, which gives a sufficiently accurate surrogate model.

## 8.4 Efficient representation of random fields

Random fields are a collection of infinitely many (correlated) random variables; one at every point of the field. Unfortunately, it is practically infeasible to work with infinitely many variables. We can handle, however, the fields with their so-called separated representations which represent the field as a sum of products of (often uncorrelated) random variables and pure spatial functions. Two prominent examples, the so-called Karhunen-Loève Expansion (KLE) and the proper orthogonal decomposition (POD) shall be examined in the following.

### 8.4.1 Karhunen-Loève Expansion (KLE)

When the soil conditions are treated as spatially varying, the scaling factor becomes a random field. As that consists of uncountably many random variables – one at each position  $(\hat{x}, \hat{y})$  – we cannot use it directly in a computation. However, given certain smoothness of the random field, it can be represented as a countable series of products of spatial functions times scalar random variables with decreasing magnitude, such that this series can be truncated after finitely many terms with diminishing error. Such a representation can be given using the Karhunen-Loève Expansion (KLE), which expands the Gaussian stochastic field into a series

$$\Theta(\hat{x}, \hat{y}, \omega) = \mu_{\Theta}(\hat{x}, \hat{y}) + \underbrace{\sum_{i=1}^{\infty} \sigma_i r_i(\hat{x}, \hat{y}) X_i(\omega)}_{\tilde{\Theta}}, \tag{8.52}$$

where the  $r_i(\hat{x}, \hat{y})$  are orthogonal, square integrable spatial functions, the  $X_i(\omega)$  are independent standard Gaussian random variables, and the  $\sigma_i$  are scaling factors.  $\mu_\Theta$  is the mean field and  $\tilde{\Theta}$  is the fluctuating part of the stochastic field.

The KLE spatial functions  $r_i$  have a structure which is typical for the specific random field and are optimal for the representation of the field in the sense that the truncated expansion minimizes the mean squared error and maximizes the captured variance. It is shown in Appendix 3 that when the problem is discretised in the spatial domain, and the functions  $r_i$  are written as a linear combination

$$r_i(\hat{x}, \hat{y}) \approx r_{h,i}(\hat{x}, \hat{y}) = \sum_{j=1}^n \Psi_j(\hat{x}, \hat{y}) v_{ji} = \mathbf{\Psi} \mathbf{V} \quad (8.53)$$

of some  $n$  given spatial basis functions that fulfill the Kronecker delta property  $\Psi_j(\hat{x}_k, \hat{y}_l) = \delta_{kl}$  (e.g. the FEM nodal basis functions) then these optimal spatial functions can be found by solving the generalized eigenvalue problem (see Equation (186) in Appendix 3)

$$\mathbf{G} \mathbf{C} \mathbf{G} \mathbf{v}_i = \underbrace{\lambda_i}_{\sigma_i^2} \mathbf{G} \mathbf{v}_i. \quad (8.54)$$

Here,  $\mathbf{G}$  is the Gramian matrix of the basis functions (also often called the mass matrix),  $\mathbf{C}$  is the covariance matrix, and  $\lambda_i$  and  $\mathbf{v}_i$  are the generalized eigenvalues and eigenvectors (see Appendix 3). Once Equation (8.54) is solved for  $\lambda_i$  and  $\mathbf{v}_i$ , the Gaussian  $\Theta$  field can be approximated by

$$\begin{aligned} \Theta(\hat{x}, \hat{y}, \omega) &\approx \mu_\Theta + \sum_{i=1}^L \sigma_i r_i(\hat{x}, \hat{y}) X_i(\omega) = \mu_\Theta + \sum_{i=1}^L \sigma_i \sum_{k=1}^n \Psi_k(\hat{x}, \hat{y}) v_{ki} X_i(\omega) \\ &= \mu_\Theta + \mathbf{\Psi}(\hat{x}, \hat{y}) \mathbf{V} \mathbf{S} \mathbf{X}(\omega), \end{aligned} \quad (8.55)$$

where  $L \leq n$  is the truncated number of the eigenvectors and  $\mathbf{S}$  is an  $L$  by  $L$  diagonal matrix with the  $\sigma_i = \sqrt{\lambda_i}$  values. In a more detailed form, the second part of the equation can be expressed as

$$\begin{aligned} \mathbf{\Psi}(\hat{x}, \hat{y}) \mathbf{V} \mathbf{S} \mathbf{X}(\omega) &= \quad (8.56) \\ &= \begin{bmatrix} \Psi_1(\hat{x}, \hat{y}) & \Psi_2(\hat{x}, \hat{y}) & \cdots & \Psi_n(\hat{x}, \hat{y}) \end{bmatrix} \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1L} \\ v_{21} & v_{22} & \cdots & v_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1} & v_{n2} & \cdots & v_{nL} \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_L \end{bmatrix} \begin{bmatrix} X_1(\omega) \\ X_2(\omega) \\ \vdots \\ X_L(\omega) \end{bmatrix}. \end{aligned}$$

#### Algorithm 12: Computation of the truncated KLE of a random field

1. **Choose a spatial mesh** with nodes  $(\hat{x}_j, \hat{y}_j)$  for  $j = 1 \dots n$  and a corresponding nodal basis  $\{\Psi_j(\hat{x}, \hat{y})\}_{j=1}^n$  and collect the functions in a row vector  $\mathbf{\Psi}$ .
2. **Compute the  $n \times n$  Gramian matrix  $\mathbf{G}$**  of the nodal basis with elements  $[\mathbf{G}]_{ij} = \int_{\mathcal{D}} \Psi_i(\hat{x}, \hat{y}) \Psi_j(\hat{x}, \hat{y}) dx dy$ .

3. **Compute the  $n \times n$  covariance matrix  $\mathbf{C}$**  from the covariance function,  $[\mathbf{C}]_{ij} = \text{Cov}_{\Theta}(\hat{x}_i, \hat{y}_i, \hat{x}_j, \hat{y}_j) = \Sigma(\hat{x}_i, \hat{y}_i, \hat{x}_j, \hat{y}_j) = C_{\nu_c}(\sqrt{(\hat{x}_i - \hat{x}_j)^2 + (\hat{y}_i - \hat{y}_j)^2})$ .
4. **Solve the generalized eigenvalue problem  $\mathbf{GCG}\mathbf{v}_i = \lambda_i\mathbf{G}\mathbf{v}_i$**  for  $i = 1 \dots n$  for the eigenvectors  $\mathbf{v}_i$  and the eigenvalues  $\lambda_i = \sigma_i^2$ . If  $n$  is very high, one may directly compute a truncated number of eigenvectors and eigenvalues.
5. **Truncate the expansion** by checking the captured variance with different number of eigenfunctions. The relative captured covariance with  $L$  eigenfunctions can be computed by

$$\rho_L = \frac{\sum_{i=1}^L \lambda_i}{\sum_{j=1}^n \lambda_j}.$$

Collect the  $L$  eigenvectors in the columns of the matrix  $\mathbf{V}$  and the  $L$  values  $\sigma_i$  into the diagonal of the matrix  $\mathbf{S}$ .

The separated representation of  $\theta(x, y, \omega)$  evaluated at the KLE mesh nodes is given by  $\mu_{\theta} + \mathbf{V}\mathbf{S}\mathbf{X}(\omega)$ , with  $\mathbf{X}$  being a vector of  $L$  independent standard Gaussian random variables. The complete field of  $\theta$  is given by

$$\theta(\hat{x}, \hat{y}, \omega) = \mu_{\theta}(\hat{x}, \hat{y}) + \mathbf{\Psi}(\hat{x}, \hat{y})\mathbf{V}\mathbf{S}\mathbf{X}(\omega).$$

### 8.4.2 Proper Orthogonal Decomposition (POD)

Another approach to represent the random field is by the so-called Proper Orthogonal decomposition (POD). Let  $\mathbf{T}$  be the random vector whose  $i^{\text{th}}$  element corresponds to the random field  $\Theta$  taken at the  $i^{\text{th}}$  node of the spatial mesh, that is,  $[\mathbf{T}(\omega)]_i = \Theta(\hat{x}_i, \hat{y}_i, \omega)$  for all nodes  $(\hat{x}_i, \hat{y}_i)$ . Then this random vector can be written in the form

$$\begin{aligned} \mathbf{T}(\omega) &= \mu_{\mathbf{T}} + \tilde{\mathbf{T}}(\omega) \\ &= \mu_{\mathbf{T}} + \mathbf{V}\mathbf{S}\mathbf{W}(\omega) \\ &= \mu_{\mathbf{T}} + \sum_i \sigma_i \mathbf{v}_i W_i(\omega), \end{aligned} \tag{8.57}$$

where  $\mathbf{S}$  is a diagonal matrix with  $[\mathbf{S}]_{ii} = \sigma_i$ ,  $\mathbf{V}$  an orthogonal matrix and  $\mathbf{W}(\omega)$  a vector of uncorrelated random variables, that is,  $\mathbb{E}[W_i(\omega)W_j(\omega)] = \delta_{ij}$ . The orthogonal matrix  $\mathbf{V}$  can be computed from the eigenvalue decomposition of the covariance matrix of  $\mathbf{T}$ , namely

$$\mathbf{C}_T = \mathbb{E}[\tilde{\mathbf{T}}\tilde{\mathbf{T}}^T] = \mathbf{V}\mathbf{S}\underbrace{\mathbb{E}[\mathbf{W}(\omega)^T\mathbf{W}(\omega)]}_{=\mathbf{I}}\mathbf{S}\mathbf{V}^T = \mathbf{V}\mathbf{S}^2\mathbf{V}^T = \sum_i \sigma_i^2 \mathbf{v}_i \mathbf{v}_i^T. \tag{8.58}$$

The POD then can be computed by solving the eigenvector problem

$$\mathbf{C}_T = \mathbf{V}\mathbf{S}^2\mathbf{V}^T \quad (8.59)$$

$$\mathbf{C}_T\mathbf{V} = \mathbf{V}\mathbf{S}^2\underbrace{\mathbf{V}^T\mathbf{V}}_{=\mathbf{I}} \quad (8.60)$$

or in a different form

$$\mathbf{C}_T\mathbf{v}_i = \sigma_i^2\mathbf{v}_i = \lambda_i\mathbf{v}_i \quad (8.61)$$

Note, that the POD in expression (8.57) can be regarded as the discretised form of the KLE Equation (8.52). The vectors  $\mathbf{v}_i$  represent the nodal values of some field-typical spatial functions and the  $W_i$  represent independent Gaussian random variables, given that  $\Theta$  is a Gaussian random field. If the same spatial nodes with coordinates  $\{\hat{x}_i, \hat{y}_i\}_{i=1}^n$  are chosen as the one in the KLE, then the covariance matrix  $\mathbf{C}_T$  matches the matrix  $\mathbf{C}$  introduced in the KLE discretization. In this way, Equations (8.53) and (8.61) are almost the same, and are identical when the chosen spatial functions are orthonormal, so when the Gramian  $\mathbf{G}$  is the identity matrix. In this case, the eigenvalues  $\lambda_i$  and eigenvectors  $\mathbf{v}_i$  are the same as the one computed from the POD approach. In general, the nodal basis is not orthogonal, but if the nodal points are distributed uniformly, then the results from the two computations give very similar results.

#### Algorithm 13: Computation of the POD of a discretised random field

1. **Choose a spatial mesh** with nodes  $(\hat{x}_j, \hat{y}_j)$  for  $j = 1 \dots n$ .
2. **Compute the  $n \times n$  covariance matrix  $\mathbf{C}$**  from the covariance function,  $[\mathbf{C}]_{ij} = \text{Cov}_{\Theta}(\hat{x}_i, \hat{y}_i, \hat{x}_j, \hat{y}_j) = \Sigma(\hat{x}_i, \hat{y}_i, \hat{x}_j, \hat{y}_j) = C_{\nu_c}(\sqrt{(\hat{x}_i - \hat{x}_j)^2 + (\hat{y}_i - \hat{y}_j)^2})$
3. **Solve the eigenvalue problem  $\mathbf{C}\mathbf{v}_i = \lambda_i\mathbf{v}_i$**  for  $i = 1 \dots n$  for the eigenvectors  $\mathbf{v}_i$  and the eigenvalues  $\lambda_i = \sigma_i^2$ . If  $n$  is very high, it may be more efficient to compute only a limited number of eigenvectors and eigenvalues.
4. **Truncate the expansion** by checking the captured variance with a different number of eigenfunctions. The relative captured energy with  $L$  eigenfunctions can be computed by

$$\rho_L = \frac{\sum_{i=1}^L \lambda_i}{\sum_{j=1}^n \lambda_j}.$$

The POD of the field is written in the form  $\Theta(\hat{x}_j, \hat{y}_j, \omega) = \mu_{\Theta}(\hat{x}_j, \hat{y}_j) + [\mathbf{V}\mathbf{S}\mathbf{X}(\omega)]_j$  with  $\mathbf{X}$  being a vector of  $L$  independent standard Gaussian random variables.

**Example 3: Separated representation of the scaling factor.** Let the prior distribution of  $F_{cM}$  be defined by Equation (8.9) and (8.12). First, we write a separated representation of the underlying Gaussian field  $\Theta \sim \mathcal{N}(0, \Sigma)$ , which is later mapped to the lognormal field  $F_{cM}$  by Equation (8.13). The representation is computed by the following steps:

**1. Define the spatial mesh.**

The FEM mesh with quadratic elements of the domain  $\mathcal{D}_s$  is shown with a light grey color in Figure 8.5. For the KLE mesh, we used the center points of the quadratic elements of the FEM mesh. We extended the mesh with some additional outer nodes, because the KL Expansion has sometimes poor accuracy at the boundary. The resulting 192 nodes were connected with triangular elements, which is shown with black lines in Figure 8.5. On the mesh, we defined piecewise linear basis functions  $\{\Psi_j(\hat{x}, \hat{y})\}_{j=1}^{192}$ , taking the value one on the  $j^{\text{th}}$  node and zero on every other node. These functions are collected in a horizontal vector  $\Psi$ .

**2. Compute the Gramian.**

The Gramian matrix  $\mathbf{G}$  has size  $192 \times 192$ , but only seven diagonals have non-zero elements due to the nodal basis used.

**3. Compute covariance matrix.**

The covariance matrix  $\mathbf{C}$  has also size  $192 \times 192$ . The element at row  $i$  and column  $j$  is computed by

$$[\mathbf{C}]_{ij} = \text{Cov}_{\Theta}(\hat{x}_i, \hat{y}_i, \hat{x}_j, \hat{y}_j) = C_{\nu_c} \left( \sqrt{(\hat{x}_i - \hat{x}_j)^2 + (\hat{y}_i - \hat{y}_j)^2} / l_c \right) \quad (8.62)$$

where the function  $C_{\nu}$  is given in Equation (8.11).

**4. Solve the generalized eigenvalue problem.**

After solving Equation (8.54) for  $\lambda_i$  and  $\mathbf{v}_i$ , the eigenfunctions can be computed by  $r_i(\hat{x}, \hat{y}) = \Psi(\hat{x}, \hat{y})\mathbf{v}_i$ . The first nine eigenfunctions are shown in the right panel of Figure 8.6.

**5. Truncate the expansion.**

The left panel of Figure 8.6 shows the relative captured variance  $\rho_L$  using different truncations of the expansion. We decided to keep  $L = 11$  eigenfunctions with which more than 93% of the total variance is captured.

Now we can write the field  $\Theta$  with the expansion  $\Theta = \sum_{i=1}^{11} \sigma_i X_i r_i(\hat{x}, \hat{y})$ , where the  $X_i$  are 11 independent standard Gaussian random variables and  $\sigma_i = \sqrt{\lambda_i}$ .

**6. Map the random field  $\Theta$  to  $F_{cM}$ .**

Following Equation (8.13) and using the separated representation of  $\Theta$ , the map from the reference Gaussian random variable  $\mathbf{X}$  to  $F_{cM}$  is given by

$$F_{cM}(\hat{x}, \hat{y}, \mathbf{X}) = e^{1.562 + 0.534 \cdot \left( \sum_{i=1}^L \sigma_i r_i(\hat{x}, \hat{y}) X_i \right)} \quad (8.63)$$

**Example 4: Surrogate modelling of seabed displacement with random fields.** Once we have the separated representation of the input field Equation (8.63), the forward model and the derivation of a surrogate model can be done the same way as in the scalar case. The only difference is that we have more than one germ now, meaning that the dimension of problem grew to  $L = 11$ . The steps to build the surrogate model (following Algorithm 11 to generate a gPCE model) are described below.

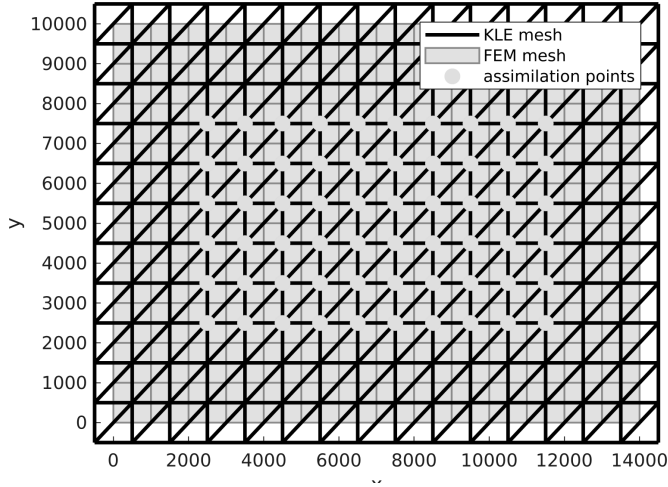


Figure 8.5: Triangular KLE mesh used for the separated representation of the scaling factor (with bold lines), the central part of the quadrilateral FEM mesh (with grey lines, see B panel of Figure 8.1), and the assimilation points (with gray dots).

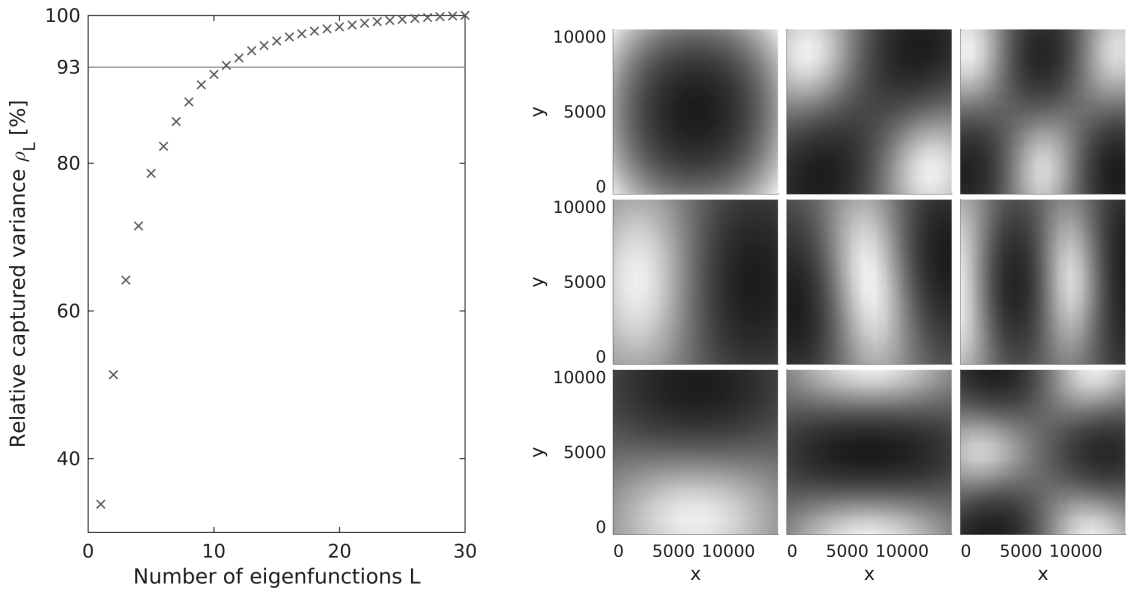


Figure 8.6: Relative captured variance  $\rho_L$  for different numbers of eigenfunctions  $L$  and the desired 93% level (to the left); the first nine eigenfunctions  $r_1$  to  $r_9$  (to the right).

1. **Define the prior distribution of the random field  $F_{cM}$  and determine the map from  $\xi$  to  $f_{cM}$ .**

This was mainly done in the previous example, see Equation (8.63) for the map from  $\mathbf{X}$  to  $F_{cM}$ . Choosing the  $\mathbf{Q}$  parameters to be  $\mathbf{X}$  and using the identity map (8.29) connecting the reference parameters to the input parameters  $\mathbf{Q}$  the map becomes<sup>7</sup>

$$f_{cM}(\hat{x}, \hat{y}, \xi) = e^{1.562+0.534 \cdot (\sum_{i=1}^L \sigma_{ir_i}(\hat{x}, \hat{y}) \xi_i)}. \tag{8.64}$$

2. **Specify the orthogonal basis polynomials used for the expansion of  $\mathbf{y}$ .**

As we chose the number of eigenfunctions to be  $L = 11$ , we need eleven independent Gaussian random variables to describe the random input and thus  $\Xi$  is a random vector of eleven variables. Accordingly, the basis functions  $\Phi_i$  are multivariate polynomials expressed as products of the univariate Hermite polynomials. The multivariate polynomials can be defined by the so-called multi-indices (see more in Appendix 2.1 and the table below). For polynomials of maximum degree  $d = 1$ , the multi-index set and the corresponding basis are shown below.

$i$	multi-index				basis functions
	$\xi_1$	$\xi_2$	$\dots$	$\xi_L$	
	$\alpha = i_1,$	$i_2$	$\dots$	$i_L)$	$\Phi_i$
0	0	0	$\dots$	0	$\Phi_0 = H_0(\xi_1)H_0(\xi_2) \dots H_0(\xi_L) = 1 \cdot 1 \dots 1 = 1$
1	1	0	$\dots$	0	$\Phi_1 = H_1(\xi_1)H_0(\xi_2) \dots H_0(\xi_L) = \xi_1 \cdot 1 \dots 1 = \xi_1$
2	0	1	$\dots$	0	$\Phi_2 = H_0(\xi_1)H_1(\xi_2) \dots H_0(\xi_L) = 1 \cdot \xi_2 \dots 1 = \xi_2$
$\vdots$			$\ddots$		$\vdots$
$L + 1$	0	0	$\dots$	1	$\Phi_{L+1} = H_0(\xi_1)H_0(\xi_2) \dots H_1(\xi_L) = 1 \cdot 1 \dots \xi_L = \xi_L$

3. **Compute the squared norms  $\gamma_i$  of the basis polynomials.**

In our case, we have effectively only univariate polynomials, for which we have defined the norms already. As the norm of  $H_0(\xi_i)$  and also of  $H_1(\xi_i)$  is one, all twelve polynomials have norm one (see Appendix 2.2).

4. **Get the integration points and weights.**

The full tensor integration rule would be too expensive here to compute because of the high dimension of the stochastic space  $L = 11$ . If we took the tensor product of the two-point univariate rules, that would give  $N = 2^L = 2048$  integration points, which means the deterministic solver would have to be run 2,048 times, which is prohibitive in our case. Instead, we use the Smolyak sparse integration grid (see e.g. [179]), which adds up to only twenty-three points.

5. **Map the integration points  $\xi_j$  to  $f_{cM,j}(\hat{x}, \hat{y})$ .**

We have twenty-three samples of the reference random vector, which results in twenty-three field realisations of the scaling factor  $F_{cM}$  (see Figure 8.7).  $F_{cM}$  scales the compressibility field in the FEM code. As the FEM computation supposes a constant compressibility field within the FEM elements, we need to extract one value of the scaling factor  $f_{cM}$  for each quadrilateral FEM element. Since the centers of the elements coincide with the KLE nodes, and it is enough to possess the values of  $f_{cM}$  at these nodes, we do not need the basis functions  $\Psi_j$  for the

---

<sup>7</sup>When coding, it is practical to write Equation (8.29),  $\mathbf{q} = \mathcal{I}(\xi)$ , in a PCE form; this will also be convenient later in the update process.

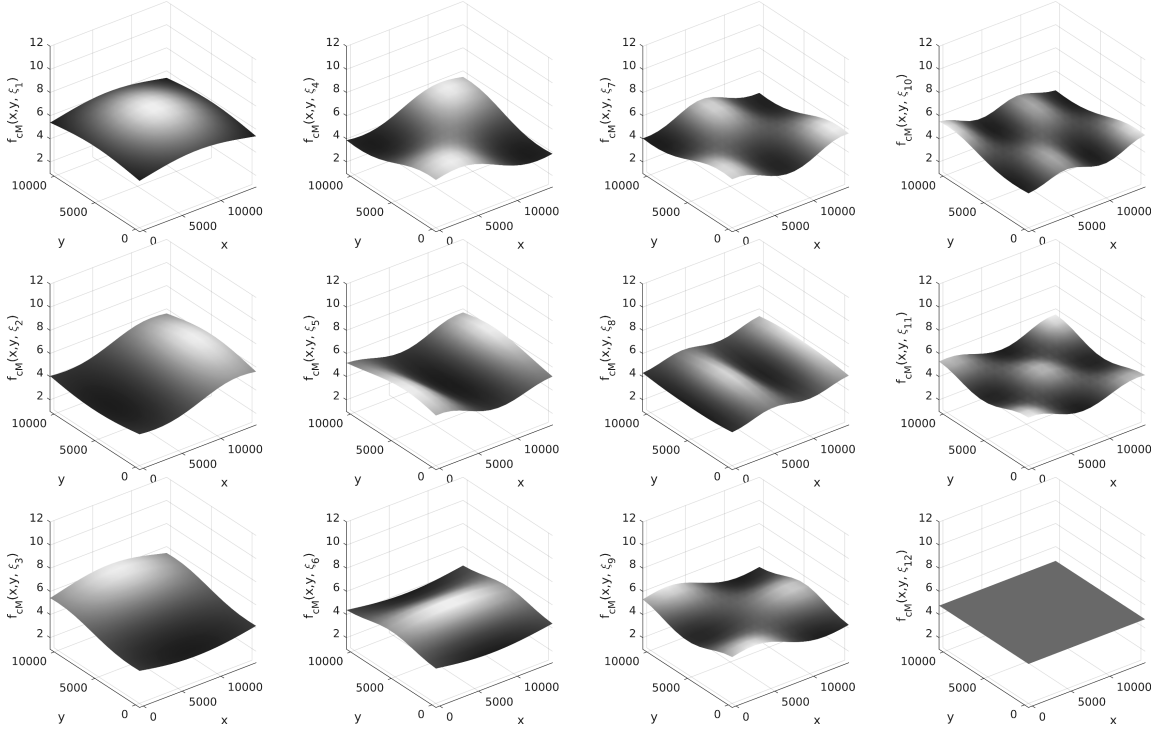


Figure 8.7: Realisations of the random field  $f_{cM,j}$  computed at the first twelve integration points  $\xi_j$ .

computation. The compressibility field at the  $k$ th KLE node with coordinates  $(\hat{x}_k, \hat{y}_k)$  is given by

$$f_{cM}(\hat{x}_k, \hat{y}_k, \xi) = e^{1.562+0.534(\sum_{i=1}^L \sigma_i [\mathbf{v}_i]_k [\xi]_i)}, \quad (8.66)$$

or the values of the scaling factor for all KLE nodes at the  $j^{th}$  integration point

$$\mathbf{f}_{cM}(\xi_j) = e^{1.562+0.534\mathbf{V}\mathbf{S}\xi_j}. \quad (8.67)$$

#### 6. Compute the measurable response.

For the computation, we only need the  $\mathbf{f}_{cM}$  values that correspond to the FEM center points (see Figure 8.5) and thus we drop the ones corresponding to the extended nodes. These selected values are used for setting the scaling factor scaling the compressibility value for each FEM element. We compute the measurable responses  $\mathbf{y}(f_{cM}(\xi_j))$  (by the map given in point 1) for all integration points  $\xi_j$ . This step involves running the FEM solver with the twenty-three different compressibility fields.

#### 7. Evaluate the basis functions at all integration points.

Here the twelve basis functions are evaluated at the twenty-three integration points.



8. Compute the PCE coefficients.

Similar to the scalar case, we compute the PCE coefficients from  $\mathbf{v}_i \approx \frac{1}{\gamma_i} \sum_{j=1}^N w_j \mathbf{y}(\boldsymbol{\xi}_j) \Phi_i(\boldsymbol{\xi}_j)$  and collect them in the matrix  $\boldsymbol{\Upsilon} = [\mathbf{v}_0, \dots, \mathbf{v}_{M-1}]$ .

## 8.5 Identification of the compressibility field

### 8.5.1 Bayes' Theorem

Having samples from similar soil conditions, we possess some knowledge about what values the scaling factor of the compressibility field  $F_{c_M}$  could possibly take. Unfortunately, as we do not know the exact value of this scaling factor, we represent this ‘belief’ of ours with probabilities. So, we model the scaling factor with a random variable/field whose prior probability distribution mirrors our ‘belief’. When we have the opportunity to learn more about the scaling factor of the compressibility field from new measurements—in this case from measuring the displacement field—then our task is to change our belief in the simplest way so that it does not contradict the newly received information. The mathematical formulation of ‘updating’ such a ‘belief’ based on some evidence is expressed with the conditional probabilities of Bayes’ Theorem (refer to Chapter 1 for more insight about Bayes’ Theorem). Bayes’ Theorem states that the updated (or assimilated or posterior) probability distribution of the scaling factor given a certain measurement  $\mathbf{z}_m$  of the displacements at the assimilation points is

$$\underbrace{\pi_{F|\mathbf{z}_m}(f_{c_M})}_{\text{posterior}} = \frac{\overbrace{\pi_{\mathbf{z}_m|F}(f_{c_M})}^{\text{likelihood}} \overbrace{\pi_F(f_{c_M})}^{\text{prior}}}{\underbrace{\int_{I_F} \pi_{\mathbf{z}_m|F}(f_{c_M}) \pi_F(f_{c_M}) df_{c_M}}_{\text{evidence}}} = \frac{L(f_{c_M}) \pi_F(f_{c_M})}{\zeta}, \tag{8.68}$$

where

- $\pi_{F|\mathbf{z}_m}$  is the posterior distribution function of the scaling factor, representing our ‘changed belief’ about the scaling factor after assimilating the measurement  $\mathbf{z}_m$ ,
- $L(f_{c_M}) = \pi_{\mathbf{z}_m|F}$  is the likelihood, expressing how likely it is to observe the measured value  $\mathbf{z}_m$  given a certain value of the scaling factor,
- $\pi_F$  is the prior distribution function of the scaling factor representing our ‘belief’ prior to learning about the new evidence, and
- $\zeta = \int_{I_F} \pi_{\mathbf{z}_m|F} \pi_F df_{c_M}$  is the evidence, the normalising factor assuring that the new posterior distribution function integrates to one.

For simplicity’s sake, we use here the above derived separated representation of the random field and we update in this case the distribution of the random variables  $\mathbf{Q}$  instead of updating directly the random field. For the input random vector  $\mathbf{Q}$ , Bayes’ Theorem becomes

$$\pi_{\mathbf{Q}|\mathbf{z}_m}(\mathbf{q}) = \frac{\pi_{\mathbf{z}_m|\mathbf{Q}}(\mathbf{q}) \pi_{\mathbf{Q}}(\mathbf{q})}{\int_{I_{\mathbf{Q}}} \pi_{\mathbf{z}_m|\mathbf{Q}}(\mathbf{q}) \pi_{\mathbf{Q}}(\mathbf{q}) d\mathbf{q}} = \frac{L(\mathbf{q}) \pi_{\mathbf{Q}}(\mathbf{q})}{\eta}. \tag{8.69}$$

In Bayes' Theorem, the challenging part is to compute the evidence and the likelihood. The likelihood is a function of  $\mathbf{q}$  and it expresses how likely it is to measure the observed displacement values  $\mathbf{z}_m$  given a certain  $\mathbf{q}$  value of the input variables. Our measurements are usually contaminated by measurement errors. Supposing that the error is additive, the measurement becomes

$$\mathbf{z}_m = \mathbf{y}(\mathbf{q}_m) + \boldsymbol{\epsilon}, \quad (8.70)$$

where  $\boldsymbol{\epsilon}$  is one realisation of the random variables  $\mathbf{E}$  of the measurement noise with probability density  $\pi_E$ , and  $\mathbf{q}_m$  is the 'true' value of the input variable(s). The likelihood can be computed from evaluating the probability density of the measurement error  $\pi_E$  at  $\boldsymbol{\epsilon} = \mathbf{z}_m - \mathbf{y}(\mathbf{q})$ , that is,

$$L(\mathbf{q}) = \pi_E(\mathbf{z}_m - \mathbf{y}(\mathbf{q})). \quad (8.71)$$

### 8.5.2 Sampling-based procedures—the MCMC method

Unfortunately, the posterior can not be written in general in an explicit, closed form. That is why the simplest way is to go with some sampling-based method: Instead of having a closed form of the updated density, we generate samples of the updated scaling factor and compute statistics of the posterior density via those samples.

The most commonly known algorithm to sample from the posterior distribution is probably the Metropolis-Hastings random walk algorithm, presented as Algorithm 1 in Chapter 1, which belongs to the group of Markov-Chain Monte Carlo (MCMC) methods. The basic idea of the Metropolis-Hastings algorithm is to generate samples by a random walk in the parameter space, constructed such that the stationary distribution of this process equals the target distribution (in this case the Bayes posterior)[150]. That is, after an initial transient phase, taking samples from the random walk is equivalent to directly sampling from the posterior.

The random walk is governed by the so-called proposal density. The proposal density is a conditional distribution that selects a new point for consideration depending on which point of the parameter space we are actually standing at. Theoretically, the algorithm converges for almost any possible proposal density; however, in practice the speed of convergence depends crucially on how well and problem-adapted it was chosen. There are many sophisticated and adaptive ways of choosing this density [83, 7], the simplest (and most-frequently used) being to choose a multivariate Gaussian density  $\mathcal{N}(\mathbf{q}, \sigma_p^2 \mathbf{I})$  centered around the current point  $\mathbf{q}$ .

To assure that we really sample from the posterior distribution, not all the proposed steps are accepted. For symmetric proposal densities, the probability with which we accept the new point is governed by the ratio of the posterior density at the new point and that of the current point. If the posterior probability at this new value of the parameter is higher than at the previous point, the step will be accepted. If the posterior probability is lower, the new step may be rejected. The lower the ratio of the two probabilities is, the new step as a sample is accepted with the lower probability.

At every point only the numerator of Equation (8.69) has to be evaluated, since the denominator is a constant value and only the ratio of the probabilities matters in determining whether the new point will be accepted or not. This is a great advantage of the method, since the denominator, that is, the evidence, is usually hard to compute due to the integral involved.

In a nutshell, the basic algorithm to generate  $N$  samples from the posterior is formulated below:

**Algorithm 14: Sampling from posterior distribution by MCMC**

1. **Choose the starting point**  $\mathbf{q}_0$  of the random walk, for example, the mean values of the prior distribution of  $\mathbf{Q}$ ,  $\mathbf{q}_0 = \mathbb{E}[\mathbf{Q}]$ .
2. **Choose proposal density**  $\pi_{\mathbf{Q}'|\mathbf{q}}(\mathbf{q}')$ , where  $\mathbf{q}'$  is the proposed new value of the reference parameter, and  $\pi_{\mathbf{Q}'|\mathbf{q}}(\mathbf{q})$  is the conditional probability of proposing a new value of the parameter  $\mathbf{q}'$  given the actual value of the parameter  $\mathbf{q}$ .
3. For  $i = 0 \dots N$  repeat:
  - (a) **Draw one  $\mathbf{q}'_i$  sample from the proposal density**  $\pi_{\mathbf{Q}'|\mathbf{q}_i}(\mathbf{q}')$ .
  - (b) Decide whether the proposed new point  $\mathbf{q}'_i$  can be accepted:
    - i. **Compute the acceptance ratio**

$$\begin{aligned} \alpha &= \min \left\{ 1, \frac{\pi_{\mathbf{Q}|\mathbf{z}_m}(\mathbf{q}'_i) \pi_{\mathbf{Q}'|\mathbf{q}_i}(\mathbf{q}'_i)}{\pi_{\mathbf{Q}|\mathbf{z}_m}(\mathbf{q}_i) \pi_{\mathbf{Q}'|\mathbf{q}'_i}(\mathbf{q}_i)} \right\} \\ &= \min \left\{ 1, \frac{L(\mathbf{q}'_i) \pi_{\mathbf{Q}}(\mathbf{q}'_i) \pi_{\mathbf{Q}'|\mathbf{q}_i}(\mathbf{q}'_i)}{L(\mathbf{q}_i) \pi_{\mathbf{Q}}(\mathbf{q}_i) \pi_{\mathbf{Q}'|\mathbf{q}'_i}(\mathbf{q}_i)} \right\}. \end{aligned}$$

Where  $\pi_{\mathbf{Q}'|\mathbf{q}'_i}(\mathbf{q}_i)$  is the probability of proposing the actual value of the parameter  $\mathbf{q}_i$  when we stand at the proposed value  $\mathbf{q}'_i$ . When the proposal distribution is symmetric, the ratio simplifies to the ratio of the posterior density at the proposed point and the one at the actual point:

$$\alpha = \min \left\{ 1, \frac{L(\mathbf{q}'_i) \pi_{\mathbf{Q}}(\mathbf{q}'_i)}{L(\mathbf{q}_i) \pi_{\mathbf{Q}}(\mathbf{q}_i)} \right\} \quad (8.72)$$

- ii. **Draw a sample  $u$  from the uniform distribution**  $\mathcal{U}[0, 1]$ .
- iii. **If  $u \leq \alpha$ , accept the new point  $\mathbf{q}'_i$ , make the step to a new point  $\mathbf{q}_{i+1} = \mathbf{q}'_i$ . Otherwise reject the new point, stay at the previous location  $\mathbf{q}_{i+1} = \mathbf{q}_i$ .**

The classical method creates one chain from the random walk. The coordinates of the chain are the samples. It is common to drop samples from the beginning of the chain, the so-called burn-in period, because those depend too much on the chosen starting point  $\mathbf{q}_0$ . The algorithm can be slightly modified by computing several chains in parallel.

As described above, the time-consuming part of the procedure is the computation of the likelihood. In the MCMC computation, the likelihood has to be evaluated at every step of the random walk to compute the acceptance ratio  $\alpha$  from Equation (8.72). For that, at each new value of  $\mathbf{q}$ , Equation (8.71) has to be evaluated, which involves the computation of a prognosis of nodal dis-

placements  $\mathbf{y}$  via the deterministic simulation. For a computationally expensive model, this may not be affordable. However, with the already determined PCE surrogate model of the nodal displacements, the evaluation of the likelihood is computationally very cheap.

**Example 5: Updating the homogeneous scaling factor by MCMC.** For testing the Bayesian inversion scheme, a synthetic measurement  $\mathbf{z}_m$  was used. First, the deterministic code was run with a chosen ‘true’ value of the scaling factor  $f_{c_M,m} = 1.89$  to generate the measurement. At this value of the scaling factor we have  $\theta_m = (\ln(f_{c_M}) - \mu)/\sigma = (\ln(1.89) - 1.562)/0.534 = -1.733$ , and thus also  $\xi_m = q_m = -1.733$ . We assumed the measurement error model given by Equation (8.24). The synthetic measurements are generated by computing  $\mathbf{y}(f_{c_M,m})$  and adding a randomly drawn sample of some ‘true’  $\mathbf{E}_m$  measurement error model with slightly lower standard deviation  $\sigma_{em,j} = 0.1 \cdot [\mathbf{y}_m]_j$  than that of  $\mathbf{E}$ . The random walk could be done directly in the space of the scaling factor  $F_{c_M}$ , but for keeping the general framework, here we carry out the random walk for the input parameter  $Q$ .

1. As starting point of the random walk we chose the prior mean, that is,

$$q_0 = \mathbb{E}[Q] = 0. \quad (8.73)$$

2. A Gaussian proposal density  $\mathcal{N}(q, \sigma_p^2)$  centered at the last step  $q$  was used with  $\sigma_p = 0.2$ , which has 30% of the variance of the prior parameter  $Q$ .
3. For the random walk, we need to define the likelihood function. Instead of the computationally expensive FEM solver, we evaluate  $\mathbf{y}$  using its PCE approximation in Equation (8.33) computed as in Example 1 but with polynomial degree five

$$\begin{aligned} L(q) &= \det(2\pi\mathbf{C}_E)^{-1/2} e^{-\frac{1}{2}(\mathbf{z}_m - \mathbf{Y}(q))^T \mathbf{C}_E^{-1} (\mathbf{z}_m - \mathbf{Y}(q))} \\ &\approx \det(2\pi\mathbf{C}_E)^{-1/2} e^{-\frac{1}{2}(\mathbf{z}_m - \mathbf{r}\Phi(\mathcal{I}^{-1}(q)))^T \mathbf{C}_E^{-1} (\mathbf{z}_m - \mathbf{r}\Phi(\mathcal{I}^{-1}(q)))}. \end{aligned} \quad (8.74)$$

The posterior distribution of  $Q$  is then defined by the product of the likelihood and the prior

$$\pi_Q(q) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{q^2}{2\sigma^2}}. \quad (8.75)$$

4. We ran Algorithm 14 with the preceding inputs which produced the MCMC chain as shown in the left panel of Figure 8.8. In the zoomed out beginning part shown in the right panel of the figure, one can see that at the beginning of the chain, the walk is stuck for a while at the starting point because of the many non-accepted proposals. This is due to the prior mean being far from the posterior mean. As the walk goes towards the posterior zone, the chain starts to look like a white noise. These samples after the transitional burn-in period were stored and mapped to the scaling factor. The posterior density and its mean were estimated by the probability density of these samples (see Figure 8.9).

**Example 6: Updating the random field by MCMC.** For the second scenario, another synthetic measurement  $\mathbf{z}_m$  was created by adding random samples of  $\mathbf{E}_m$  to the displacements  $\mathbf{y}$  computed by the FEM solver with a ‘true’ spatially varying  $f_{c_M}$  field (see left upper and left bottom panels of Figure 8.12).

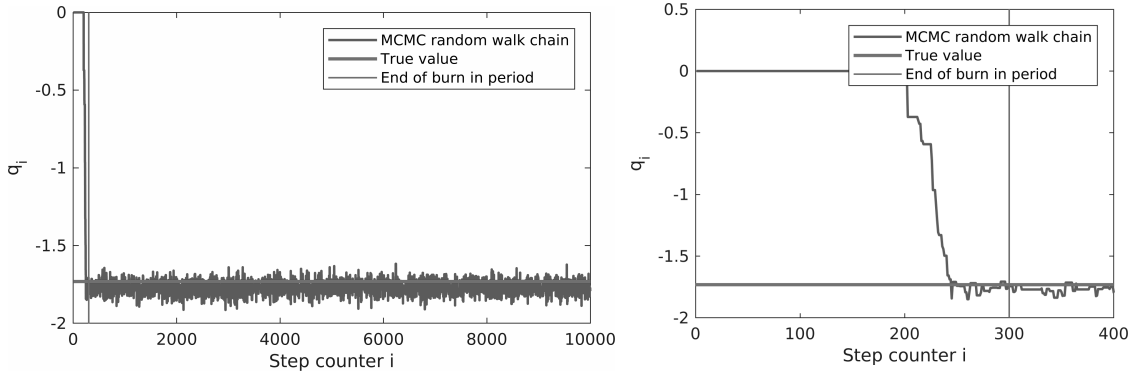


Figure 8.8: The posterior samples  $\{q_i\}_{i=1}^{10000}$  from the MCMC random walk chain (to the left) and zoomed out part of it (to the right) to show the selected transitional burn-in period.

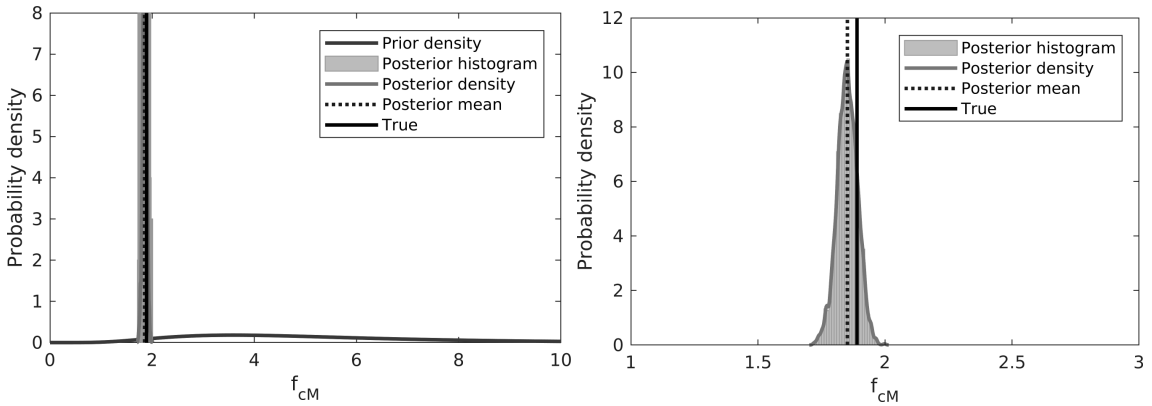


Figure 8.9: MCMC update results: the prior probability density and the posterior density estimate of  $F_{cM}$  with the mean of the posterior and the ‘true’ value.

The measurement error model and the likelihood are the same as in the scalar case, the only difference being that the random walk is now taking place in a higher dimensional space ( $L = 11$ ), the number of eigenfunctions we chose to represent the random field with, and thus also the number of independent standard Gaussian random variables. For evaluating the likelihood, the PCE derived in Example 4 was used. The random chain of the first four input parameters are shown on Figure 8.10. The posterior domain shrinks as the scatter plot of the posterior samples of the left panel of Figure 8.11 shows. The right panel of Figure 8.11 shows the 90% confidence region and the mean of the prior and the posterior distribution. At the points where no measurement data was available (at the boundaries of the domain, specially at the corner points), the uncertainties stay pretty high. However, in the critical region, the updated mean gives a good estimated value of the mean of the scaling field. Figure 8.12 shows together the ‘true’ field  $f_{cM,m}$  and the mean posterior estimate as

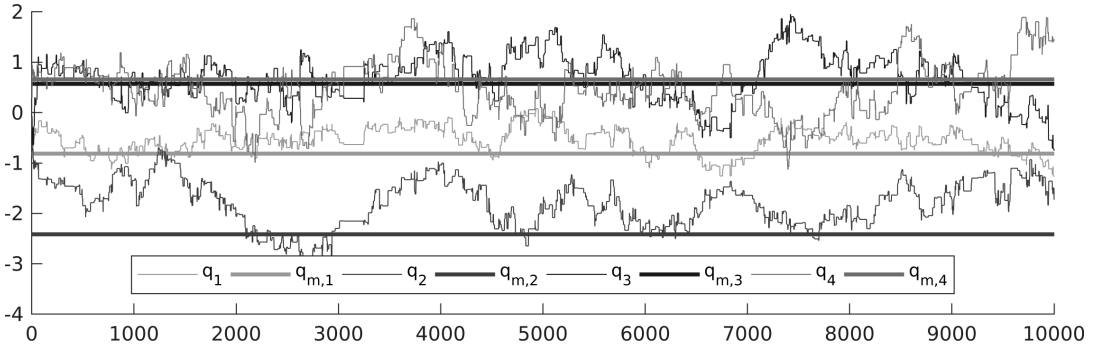


Figure 8.10: Random chain of the first four elements of  $\mathbf{q}$  (right).

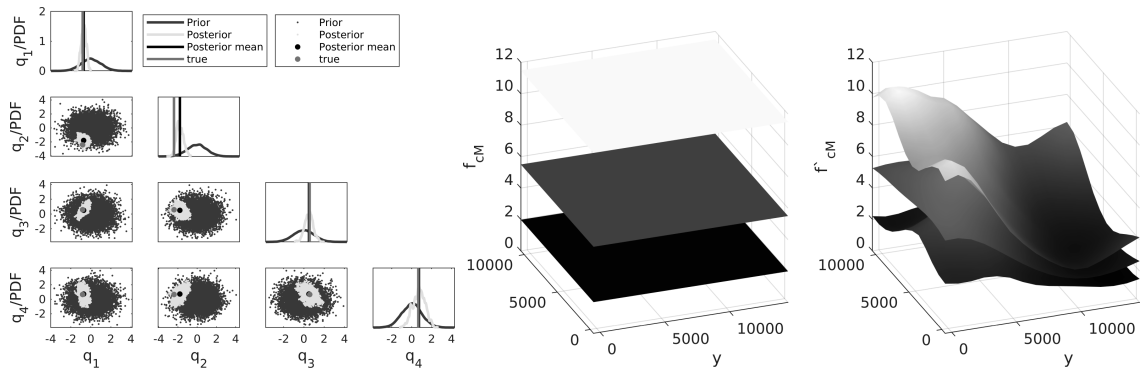


Figure 8.11: MCMC update results: scatter plot of the samples of the prior and the posterior first four input parameter  $\mathbf{q}$  and their PDFs (to the left), mean and 90% confidence region of the prior (in the middle), and the posterior (to the right).

well as the standard deviation of the posterior and the deviation of the mean estimate from  $f_{cM,m}$ . It can be seen, that in the area of the assimilation points, the standard deviation of the posterior reduced significantly compared to the standard deviation of the prior ( $\sigma_f = 3.16$ ). An exception is the region where no significant displacements were measured. Here the posterior standard deviation stays somewhat higher as these measurements could not contribute much to the assimilation process. Though only few eigenmodes were used, and we also kept the PCE representation of  $\mathbf{y}$  linear, the Bayesian posterior mean captures pretty well the ‘true’ field at the assimilation area. It is important to highlight that for this result, the inversion cost only twenty-three deterministic solver calls.

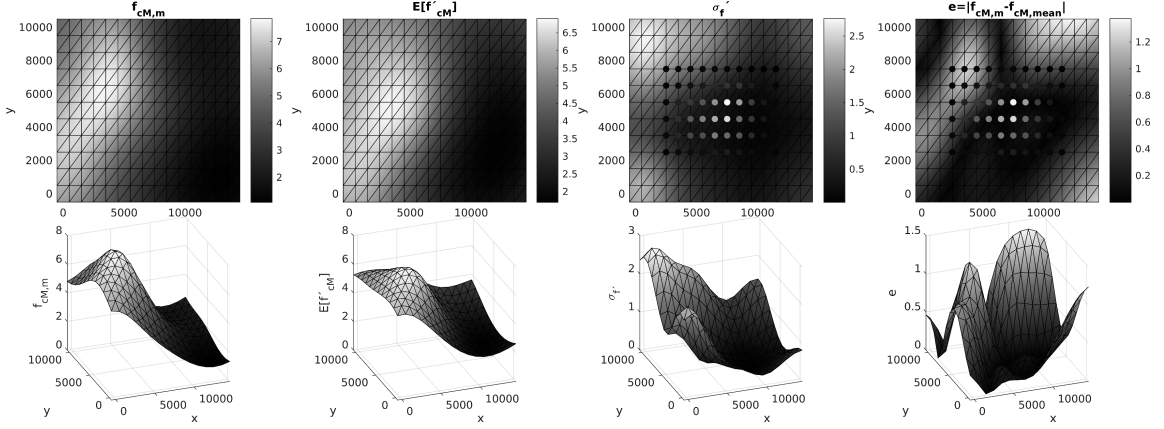


Figure 8.12: Result of the update by MCMC: 2D (on the top) and 3D view (on the bottom) of the ‘true’  $f_{cM}$  field, from which the measurement was generated from (to the left), mean of the posterior of  $f_{cM}$  computed from the MCMC samples (second column), standard deviation of the posterior  $f_{cM}$  field computed from the MCMC samples (third column) and the deviation of the mean posterior from the ‘true’ field (to the right). The upper right figures also show the sixty assimilation points used for the update, with colors showing the magnitude of the measured displacement.

### 8.5.3 The Kalman filter and its modified versions

#### 8.5.3.1 The Kalman filter

The Kalman filter was first proposed as a state estimation technique, but it can be also used for parameter estimation when the parameter is Gaussian and the estimate can be given with a linear function of the data. The filter is in detail explained in [64]. The main assumption of the Kalman filtering is that the map from the parameter  $\mathbf{Q}$  to the measurable can be written in a linear form

$$\mathbf{Y}(\mathbf{Q}) = \mathbf{H}\mathbf{Q}, \quad (8.76)$$

where  $\mathbf{H}$  is a matrix representing the linear map. Assuming Gaussian distribution for the measurement error  $\mathbf{E}$ , the likelihood reads

$$L(\mathbf{q}) = \det(2\pi\mathbf{C}_E)^{-1/2} e^{-\frac{1}{2}(\mathbf{z}_m - \mathbf{H}\mathbf{q})^T \mathbf{C}_E^{-1} (\mathbf{z}_m - \mathbf{H}\mathbf{q})}. \quad (8.77)$$

The Bayes’ Theorem states that the posterior distribution of  $\mathbf{Q}$  is

$$\pi_{Q'}(\mathbf{q}) = \frac{L(\mathbf{q})\pi_Q(\mathbf{q})}{\int_{I_Q} L(\mathbf{q})\pi_Q(\mathbf{q})}. \quad (8.78)$$

If we assume a Gaussian prior distribution for the parameter  $\mathbf{Q}$

$$\pi_Q(\mathbf{q}) = \det(2\pi\mathbf{C}_Q)^{-1/2} e^{-\frac{1}{2}(\mathbf{q} - \boldsymbol{\mu}_Q)^T \mathbf{C}_Q^{-1} (\mathbf{q} - \boldsymbol{\mu}_Q)} \quad (8.79)$$

with mean  $\boldsymbol{\mu}_Q$  and covariance  $\mathbf{C}_Q$ , then it can be shown that this posterior distribution corresponds to a Gaussian distribution

$$\pi_{Q'}(\mathbf{q}) = \det(2\pi\mathbf{C}_{Q'})^{-1/2} e^{-\frac{1}{2}(\mathbf{q}-\boldsymbol{\mu}_{Q'})^T \mathbf{C}_{Q'}^{-1}(\mathbf{q}-\boldsymbol{\mu}_{Q'})}, \quad (8.80)$$

with

$$\boldsymbol{\mu}_{Q'} = \mathbb{E}[\mathbf{Q}'] = \boldsymbol{\mu}_Q + \mathbf{K}(\mathbf{z}_m - \mathbf{H}\boldsymbol{\mu}_Q), \quad \mathbf{C}_{Q'} = \text{cov}[\mathbf{Q}', \mathbf{Q}'] = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{C}_Q, \quad (8.81)$$

where  $\mathbf{C}_{Q'}$  is the covariance of the posterior random variable and  $\mathbf{K}$  is the Kalman gain

$$\mathbf{K} = \mathbf{C}_Q \mathbf{H}^T (\mathbf{H}\mathbf{C}_Q \mathbf{H}^T + \mathbf{C}_E)^{-1} \quad (8.82)$$

$$\mathbf{K} = \mathbf{C}_{QY} (\mathbf{C}_Y + \mathbf{C}_E)^{-1} \quad (8.83)$$

with  $\mathbf{C}_{QY}$  and  $\mathbf{C}_Y$  being the covariances  $\text{cov}[\mathbf{Q}, \mathbf{Y}]$  and  $\text{cov}[\mathbf{Y}, \mathbf{Y}]$ . The random variable

$$\mathbf{Q}' = \mathbf{Q} + \mathbf{K}(\mathbf{z}_m - \mathbf{Z}) \quad (8.84)$$

can also be shown to be Gaussian and has mean  $\boldsymbol{\mu}_{Q'}$  and covariance  $\mathbf{C}_{Q'}$ . Gaussian posteriors are completely described by their first two moments, the mean and the variance, thus Equation (8.84) gives the exact posterior random variable when  $\mathbf{Q}$  is Gaussian and the map from  $\mathbf{Q}$  to  $\mathbf{Y}$  is linear.

### 8.5.3.2 The ensemble Kalman filter

The main problem is that in our case—and usually in many engineering problems—the measurable  $\mathbf{Y}$  is not linearly depending on  $\mathbf{Q}$ , so the map can not be given simply by a transformation matrix  $\mathbf{H}$ . Actually, in our case the operator

$$\mathbf{y}(\mathbf{q}) = \mathcal{G}(f_{cM}(\mathcal{F}(\mathbf{q}))) = \mathcal{H}(\mathbf{q}) \quad (8.85)$$

was only introduced in an abstract way, and is not even explicitly given. However, many variations of the Kalman filter were proposed that are applicable for non-linear problems. An example is the Ensemble Kalman filter (EnKF), which keeps the form of the filter given in Equation (8.84), but represents the random variables  $\mathbf{Q}'$ ,  $\mathbf{Q}$ , and  $\mathbf{Z}$  by MC samples

$$\mathbf{q}'(\boldsymbol{\xi}_j) = \mathbf{q}(\boldsymbol{\xi}_j) + \mathbf{K}(\mathbf{z}_m - \mathbf{z}(\boldsymbol{\xi}_j, \boldsymbol{\eta}_j)) \quad \forall j = 1 \dots N, \quad (8.86)$$

where  $N$  is the number of samples,  $\mathbf{z}(\boldsymbol{\xi}_j, \boldsymbol{\eta}_j) = \mathcal{H}(\mathbf{q}(\boldsymbol{\xi}_j)) + \boldsymbol{\epsilon}(\boldsymbol{\eta}_j)$ , with  $\boldsymbol{\epsilon}(\boldsymbol{\eta}_j)$  a sample from the measurement noise model  $\mathbf{E}$ , and  $\boldsymbol{\eta}$  the germ of the measurement error model. The Kalman gain is computed using (8.83), but approximating the covariances from MC samples

$$\mathbf{C}_Y = \frac{1}{N-1} \sum_{j=1}^N (\mathbf{y}(\boldsymbol{\xi}_j) - \bar{\mathbf{y}}) (\mathbf{y}(\boldsymbol{\xi}_j) - \bar{\mathbf{y}})^T, \quad (8.87)$$

$$\mathbf{C}_{QY} = \frac{1}{N-1} \sum_{j=1}^N (\mathbf{q}(\boldsymbol{\xi}_j) - \bar{\mathbf{q}}) (\mathbf{y}(\boldsymbol{\xi}_j) - \bar{\mathbf{y}})^T, \quad (8.88)$$

$$\mathbf{C}_E = \frac{1}{N-1} \sum_{j=1}^N (\boldsymbol{\epsilon}(\boldsymbol{\eta}_j) - \bar{\boldsymbol{\epsilon}}) (\boldsymbol{\epsilon}(\boldsymbol{\eta}_j) - \bar{\boldsymbol{\epsilon}})^T, \quad (8.89)$$



where the bar above a variable means mean of the samples, for example

$$\bar{\mathbf{q}} = \frac{1}{N} \sum_{j=1}^N \mathbf{q}(\boldsymbol{\xi}_j). \quad (8.90)$$

The ensemble Kalman filter is also applicable for (slightly) non-linear models, but as all MC procedures, necessitates a very large number of samples  $N$ , involving  $N$  evaluations of the deterministic solver to compute all responses  $\mathbf{y}(\boldsymbol{\xi}_j)$  from the samples  $\mathbf{q}(\boldsymbol{\xi}_j)$  of the input random variable. However, with the help of our surrogate model, the evaluation can be done with much less computational effort.

The algorithm of the EnKF using a PCE surrogate model is as follows:

#### Algorithm 15: Parameter update by the ensemble Kalman filter

1. Generate  $N$  samples of the random variables  $\mathbf{Q}$  and  $\mathbf{E}$ :  $\mathbf{q}(\boldsymbol{\xi}_j)$ ,  $\boldsymbol{\epsilon}(\boldsymbol{\eta}_j)$   $j = 1 \dots N$ .
2. Compute the  $\mathbf{y}_h$  displacements using the surrogate model  $\mathbf{y}_h(\boldsymbol{\xi}_j) = \sum_{i=0}^{M-1} \mathbf{v}_i \Phi_i(\boldsymbol{\xi}_j)$  for all  $j = 1 \dots N$ .
3. Compute the Kalman-gain:
  - (a) Compute the sample covariances  $\mathbf{C}_Y$ ,  $\mathbf{C}_{QY}$ , and  $\mathbf{C}_E$  using Equations:

$$\mathbf{C}_Y = \frac{1}{N-1} \sum_{j=1}^N (\mathbf{y}(\boldsymbol{\xi}_j) - \bar{\mathbf{y}}) (\mathbf{y}(\boldsymbol{\xi}_j) - \bar{\mathbf{y}})^T,$$

$$\mathbf{C}_{QY} = \frac{1}{N-1} \sum_{j=1}^N (\mathbf{q}(\boldsymbol{\xi}_j) - \bar{\mathbf{q}}) (\mathbf{y}(\boldsymbol{\xi}_j) - \bar{\mathbf{y}})^T,$$

$$\mathbf{C}_E = \frac{1}{N-1} \sum_{j=1}^N (\boldsymbol{\epsilon}(\boldsymbol{\eta}_j) - \bar{\boldsymbol{\epsilon}}) (\boldsymbol{\epsilon}(\boldsymbol{\eta}_j) - \bar{\boldsymbol{\epsilon}})^T.$$

- (b) Compute samples of the measurement model by adding to the predicted  $\mathbf{y}_i$  samples the samples of the measurement error;  $\mathbf{z}(\boldsymbol{\xi}_j, \boldsymbol{\eta}_j) = \mathbf{y}(\boldsymbol{\xi}_j) + \boldsymbol{\epsilon}(\boldsymbol{\eta}_j)$ .
  - (c) Compute the Kalman gain  $\mathbf{K} = \mathbf{C}_{QY}(\mathbf{C}_Y + \mathbf{C}_E)^{-1}$ .
4. Compute samples of the posterior using

$$\mathbf{q}'(\boldsymbol{\xi}_j) = \mathbf{q}(\boldsymbol{\xi}_j) + \mathbf{K}(\mathbf{z}_m - \mathbf{z}(\boldsymbol{\xi}_j, \boldsymbol{\eta}_j)) \quad \forall j = 1 \dots N.$$

5. Map the samples to the scaling factor  $f_{cM}$ .

The advantage of this approach is that it is much faster than doing the random walk of the MCMC, and we do not have the problem of getting stuck with the random walk, or with decisions such as how to choose the proposal density or how long the burn-in period should be. However, as later shown in the examples, the linear filter still has its limits when used for nonlinear problems.

### 8.5.3.3 The PCE-based Kalman filter

With the help of the PCE approximation, one can compute the (linear) Kalman filter in an even more straightforward way. This is done purely using the PCE algebra for the updating. The method is explained in detail in [153]. The main idea is that we can represent all the random variables used for the Kalman filtering in a PCE form. Then the filtering is done by updating the PCE coefficients of  $\mathbf{Q}$  – the map from the germ to the input parameter – without any sampling. This method is a complete sample-free update procedure. Let us write the input parameter, the error and the measurable response in a PCE format

$$\mathbf{q} = \sum_{\alpha=0}^L \mathbf{q}_\alpha \Phi_\alpha^{(Q)}(\boldsymbol{\xi}) \quad \boldsymbol{\epsilon} = \sum_{\alpha=0}^{n_y} \mathbf{e}_\alpha \Phi_\alpha^{(E)}(\boldsymbol{\eta}) \quad \mathbf{y} = \sum_{\alpha=0}^{M-1} \mathbf{v}_\alpha \Phi_\alpha(\boldsymbol{\xi}). \quad (8.91)$$

The first map is just the PCE of the map  $\mathcal{I}(\boldsymbol{\xi})$ , which is now the identity map, but in general could be any one-to-one map.

The given PCE representations of the three different random vectors are defined each in their own suitable PCE basis. Our first task is to bring the above given PCEs in one unified basis, combining all the Polynomial Chaos basis functions together into one joint basis (see Example 7). Furthermore, we suppose that the measurement errors  $\mathbf{E}$  are independent from the random input parameters  $\mathbf{Q}$  and so are  $\boldsymbol{\xi}$  and  $\boldsymbol{\eta}$ . The new PCE representations in the extended basis then take the form

$$\mathbf{q} = \sum_{\alpha=0}^{n_c} \hat{\mathbf{q}}_\alpha \hat{\Phi}_\alpha(\boldsymbol{\xi}, \boldsymbol{\eta}) = \hat{\mathbf{Q}}\hat{\Phi} \quad \boldsymbol{\epsilon} = \sum_{\alpha=0}^{n_c} \hat{\mathbf{e}}_\alpha \hat{\Phi}_\alpha(\boldsymbol{\xi}, \boldsymbol{\eta}) = \hat{\mathbf{E}}\hat{\Phi} \quad \mathbf{y} = \sum_{\alpha=0}^{n_c} \hat{\mathbf{v}}_\alpha \hat{\Phi}_\alpha(\boldsymbol{\xi}, \boldsymbol{\eta}) = \hat{\mathbf{Y}}\hat{\Phi}. \quad (8.92)$$

The new set of basis functions  $\hat{\Phi}_\alpha$  have germs  $\boldsymbol{\xi}$  and  $\boldsymbol{\eta}$ . Similarly, we can also write the deterministic values  $\mathbf{z}_m$  in a PCE format

$$\mathbf{z}_m = \sum_{\alpha=0}^{n_c} \hat{\mathbf{z}}_\alpha \hat{\Phi}_\alpha(\boldsymbol{\xi}, \boldsymbol{\eta}) = \hat{\mathbf{Z}}\hat{\Phi}, \quad (8.93)$$

where the first coefficient  $\hat{\mathbf{z}}_0$  corresponding to the polynomial  $\hat{\Phi}_0(\boldsymbol{\xi}, \boldsymbol{\eta}) = 1$  is directly the vector  $\mathbf{z}_m$  and the rest of the coefficients are zero.

With the help of the new, extended PCE expansions, we can rewrite the filtering step (8.84) as

$$\hat{\mathbf{Q}}'\hat{\Phi} = \hat{\mathbf{Q}}\hat{\Phi} + \mathbf{K} \left( \hat{\mathbf{Z}}\hat{\Phi} - \left( \hat{\mathbf{Y}}\hat{\Phi} + \hat{\mathbf{E}}\hat{\Phi} \right) \right), \quad (8.94)$$

where  $\hat{\mathbf{Q}}'$  is the matrix collecting the new PCE coefficients of the updated random variable  $\mathbf{Q}'$ . It can be easily seen from the above equation that these new coefficients can be determined from

$$\hat{\mathbf{Q}}' = \hat{\mathbf{Q}} + \mathbf{K}(\hat{\mathbf{Z}} - \hat{\mathbf{Y}} - \hat{\mathbf{E}}). \quad (8.95)$$

The covariances for the Kalman gain can be computed without any samples from the unified PCE coefficients following Equation (8.45).

**Algorithm 16: Parameter identification by the PCE-based Kalman filter**

1. Write input parameter  $\mathbf{Q}$ , measurement error model  $\mathbf{E}$ , and the response  $\mathbf{Y}$  in a PCE form:

$$\mathbf{q} = \sum_{\alpha=0}^L \mathbf{q}_\alpha \Phi_\alpha^{(Q)}(\boldsymbol{\xi}) \quad \boldsymbol{\epsilon} = \sum_{\alpha=0}^{n_y} \mathbf{e}_\alpha \Phi_\alpha^{(E)}(\boldsymbol{\eta}) \quad \mathbf{y} = \sum_{\alpha=0}^{M-1} \mathbf{v}_\alpha \Phi_\alpha(\boldsymbol{\xi}).$$

2. Generate a combined PCE basis  $\hat{\Phi}_\alpha(\boldsymbol{\xi}, \boldsymbol{\eta})$  and compute the  $\hat{\gamma}_\alpha$  squared norm of each basis function.
3. Rewrite the PCE coefficients in this extended basis; write the measured displacements  $\mathbf{z}_m$  also in a PCE using the extended basis:

$$\mathbf{q} = \sum_{\alpha=0}^{n_c} \hat{\mathbf{q}}_\alpha \hat{\Phi}_\alpha(\boldsymbol{\xi}, \boldsymbol{\eta}) \quad \boldsymbol{\epsilon} = \sum_{\alpha=0}^{n_c} \hat{\mathbf{e}}_\alpha \hat{\Phi}_\alpha(\boldsymbol{\xi}, \boldsymbol{\eta}) \quad \mathbf{y} = \sum_{\alpha=0}^{n_c} \hat{\mathbf{v}}_\alpha \hat{\Phi}_\alpha(\boldsymbol{\xi}, \boldsymbol{\eta}) \quad \mathbf{z}_m = \sum_{\alpha=0}^{n_c} \hat{\mathbf{z}}_\alpha \hat{\Phi}_\alpha(\boldsymbol{\xi}, \boldsymbol{\eta}).$$

4. Compute the Kalman gain.

- (a) Compute the covariances  $\mathbf{C}_Y$ ,  $\mathbf{C}_{QY}$ , and  $\mathbf{C}_E$  from the PCE coefficients:

$$\mathbf{C}_Y = \sum_{\alpha=1}^{n_c} \mathbf{v}_\alpha \mathbf{v}_\alpha^T \hat{\gamma}_\alpha \quad \mathbf{C}_{QY} = \sum_{\alpha=1}^{n_c} \hat{\mathbf{q}}_\alpha \hat{\mathbf{v}}_\alpha^T \hat{\gamma}_\alpha \quad \mathbf{C}_E = \sum_{\alpha=1}^{n_c} \hat{\mathbf{e}}_\alpha \hat{\mathbf{e}}_\alpha^T \hat{\gamma}_\alpha.$$

- (b) Compute the Kalman gain:

$$\mathbf{K} = \mathbf{C}_{QY} (\mathbf{C}_Y + \mathbf{C}_E)^{-1}.$$

5. Compute the PCE coefficients  $\hat{\mathbf{q}}'_\alpha$  of the updated input random variables  $\mathbf{Q}'$ :

$$\hat{\mathbf{q}}'_\alpha = \hat{\mathbf{q}}_\alpha + \mathbf{K}(\hat{\mathbf{z}}_\alpha - \hat{\mathbf{v}}_\alpha - \hat{\mathbf{e}}_\alpha).$$

The PCE of the updated reference random is

$$\mathbf{q}' = \sum_{\alpha=0}^{n_c} \hat{\mathbf{q}}'_\alpha \hat{\Phi}_\alpha(\boldsymbol{\xi}, \boldsymbol{\eta}).$$

**Example 7: Updating the scalar scaling factor by the PCE-based Kalman filter.** Here we present how to do the Kalman filtering for the scalar scaling factor of the compressibility field.

### 1. Write $Q$ , $\mathbf{E}$ , and $\mathbf{Y}$ in a PCE form.

For scenario I,  $Q$  is a scalar valued random variable and so is the Gaussian germ  $\Xi$ . The map  $\mathcal{I}$  from the germ to the parameter is the identity map (see Equation (8.29)), whose PCE is

$$q = [0 \quad 1] \begin{bmatrix} H_0(\xi) \\ H_1(\xi) \end{bmatrix} = [0 \quad 1] \begin{bmatrix} 1 \\ \xi \end{bmatrix}. \quad (8.96)$$

The measurement error model  $\mathbf{E}$  is a vector valued random variable with  $n_y = 60$  random elements – corresponding to the sixty assimilation points. The distribution of  $\mathbf{E}$  is given in Equation (8.24). For representing the sixty independent Gaussian random variables given in this error model, we need a germ  $\boldsymbol{\eta}$  with sixty independent standard Gaussians. The map from the germ  $\boldsymbol{\eta}$  to  $\mathbf{E}$  is given by scaling the germ  $\boldsymbol{\eta}$  by the standard deviation of the measurement errors. The scaling in a PCE form is given by

$$\boldsymbol{\epsilon} = \begin{bmatrix} 0 & \sigma_{\epsilon,1} & 0 & \cdots & 0 \\ 0 & 0 & \sigma_{\epsilon,2} & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_{\epsilon,60} \end{bmatrix} \begin{bmatrix} H_0 \\ H_1(\eta_1) \\ H_1(\eta_2) \\ \vdots \\ H_1(\eta_{60}) \end{bmatrix} = \begin{bmatrix} 0 & \sigma_{\epsilon,1} & 0 & \cdots & 0 \\ 0 & 0 & \sigma_{\epsilon,2} & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_{\epsilon,60} \end{bmatrix} \begin{bmatrix} 1 \\ \eta_1 \\ \eta_2 \\ \vdots \\ \eta_{60} \end{bmatrix}. \quad (8.97)$$

We use the PCE approximation of  $\mathbf{y}$  computed in Example 1, but with polynomial degree five, which reads

$$\mathbf{y} = \boldsymbol{\Upsilon} [1 \quad \xi \quad \xi^2 - 1 \quad \xi^3 - 3\xi \quad \xi^4 - 6\xi^2 + 3 \quad \xi^5 - 10\xi^3 + 15\xi]^T \quad \boldsymbol{\Upsilon} \in \mathbb{R}^{60 \times 6}. \quad (8.98)$$

### 2. Generate a combined PCE basis.

The new, combined basis containing all PCE polynomials used for the expansion of  $Q$ ,  $\mathbf{E}$ , and  $\mathbf{Y}$  becomes

$$\hat{\boldsymbol{\Phi}} = [1 \quad \xi \quad \xi^2 - 1 \quad \xi^3 - 3\xi \quad \xi^4 - 6\xi^2 + 3 \quad \xi^5 - 10\xi^3 + 15\xi \quad \eta_1 \quad \eta_2 \quad \dots \quad \eta_{60}]^T. \quad (8.99)$$

### 3. Rewrite the PCE coefficients in the extended basis.

The coefficients collected in a matrix reads

$$\begin{aligned} \hat{Q} &= [0 \quad 1 \quad 0 \quad \dots \quad 0], & \hat{Q} &\in \mathbb{R}^{1 \times 66}, \\ \hat{\mathbf{E}} &= \begin{bmatrix} 0 & 0 & \dots & 0 & \sigma_{\epsilon,1} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & \sigma_{\epsilon,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & \sigma_{\epsilon,60} \end{bmatrix}, & \hat{\mathbf{E}} &\in \mathbb{R}^{60 \times 66}, \\ \hat{\mathbf{Y}} &= \begin{bmatrix} \nu_{1,0} & \nu_{1,1} & \dots & \nu_{1,5} & 0 & 0 & \dots & 0 \\ \nu_{2,0} & \nu_{2,1} & \dots & \nu_{2,5} & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \nu_{60,0} & \nu_{60,1} & \dots & \nu_{60,5} & 0 & 0 & \dots & 0 \end{bmatrix}, & \hat{\mathbf{Y}} &\in \mathbb{R}^{60 \times 66}, \end{aligned} \quad (8.100)$$

with  $\sigma_{\epsilon,j}$  the standard deviation of the measurement error at the  $j^{\text{th}}$  assimilation point and  $v_{i,j}$  the elements  $[\mathbf{Y}]_{i,j}$ . The matrix of PCE coefficients of the measurement  $\mathbf{z}_m$  reads

$$\hat{\mathbf{Z}} = \begin{bmatrix} z_{m,1} & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ z_{m,2} & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ z_{m,60} & 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \quad \hat{\mathbf{Z}} \in \mathbb{R}^{60 \times 66}. \quad (8.101)$$

4. Compute the Kalman gain.

The covariance matrices  $\mathbf{C}_Y$  and  $\mathbf{C}_E$  can be computed from the original PCE expansions, but for the computation of  $\mathbf{C}_{QY}$ ,  $\mathbf{Q}$  and  $\mathbf{Y}$  needs to be described in the same basis.

5. Compute the coefficients of the updated input variables  $\mathbf{Q}'$ .

The PCE of the updated reference random variable is given in the extended basis, which means that its dimension is the sum of the size  $L$  (in Example 1) of the input random variables  $\mathbf{Q}$  and the dimension  $n_y$  of the germ  $\boldsymbol{\eta}$  (in the example 60) needed for the description of the measurement error  $\mathbf{E}$ , that is, the updated input parameters are given in stochastic dimension  $L+n_y = 61$ . If new measurements are available, the update can be repeated, but the new updated variable is then given in a new space extended with further stochastic dimensions. For example, when we receive a new data with measured displacements at the same sixty assimilation points, the stochastic dimension of the updated input random variable will be  $L + n_y + n_y = 121$ .

At this point, we have a PCE expansion of the posterior input parameter  $\mathbf{Q}'$ . What we are interested in though is the posterior distribution of the scaling factor  $f_{CM}$ . The posterior scaling factor can be given now in an analytical form because we have a PCE of the updated input parameter  $\mathbf{Q}'$  and an explicit map from the parameter to the scaling factor. If a probability density plot is desired, we can sample from  $\mathbf{Q}'$  and map all the samples to the scaling factor. Figure 8.13 shows the updated density of the scaling factor computed by the PCE-based Kalman filter. One can see that the filter somewhat overestimates the scaling factor, which is due to the filter being linear.

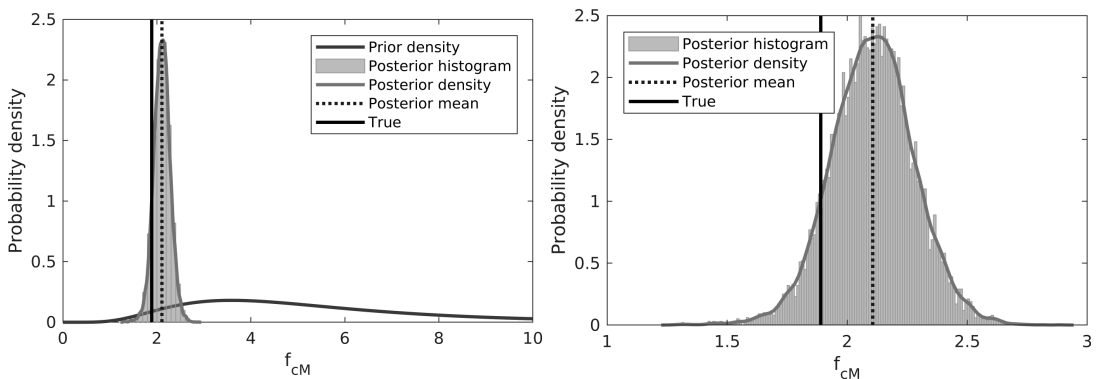


Figure 8.13: PCE based Kalman filter update results: the prior probability density and the posterior density estimate of  $F_{CM}$  with the mean of the posterior and the ‘true’ value.

**Example 8: Updating the spatially varying random field by the PCE-based Kalman filter.** Here we present how to do the Kalman filtering for the scaling factor of the compressibility field when no homogeneity conditions are assumed.

1. **Represent  $\mathbf{Q}$ ,  $\mathbf{E}$ , and  $\mathbf{Y}$  by a PCE.**

We use the truncated KLE expansion of the field  $\theta$  computed in Example 3, which represented the field by  $L = 11$  independent, standard Gaussian random variables, which we use for  $\mathbf{Q}$ . The PCE expansion of  $\mathbf{Q}$  is

$$\mathbf{q} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} H_0 \\ H_1(\xi_1) \\ H_1(\xi_2) \\ \vdots \\ H_1(\xi_L) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} 1 \\ \xi_1 \\ \xi_2 \\ \vdots \\ \xi_L \end{bmatrix}. \quad (8.102)$$

The PCE of the measurement error  $\mathbf{E}$  is the same as in the scalar case in the previous example. For the PCE expansion of  $\mathbf{Y}$  we use the results in Example 4

$$\mathbf{y} = \mathbf{\Upsilon} [1 \quad \xi_1 \quad \xi_2 \quad \dots \quad \xi_L]^T, \quad (8.103)$$

which is given in a simple linear PCE basis.

2. **Generate a combined PCE basis.**

The extended basis is

$$\hat{\Phi} = [1 \quad \xi_1 \quad \xi_2 \quad \dots \quad \xi_L \quad \eta_1 \quad \eta_2 \quad \dots \quad \eta_{60}]^T. \quad (8.104)$$

3. **Rewrite the PCE coefficients in the extended basis.**

The coefficients in the extended basis become

$$\hat{\mathbf{Q}} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \end{bmatrix} \quad \hat{\mathbf{E}} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & \sigma_{\epsilon,1} & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & \sigma_{\epsilon,2} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & \sigma_{\epsilon,60} \end{bmatrix}$$

$$\hat{\mathbf{\Upsilon}} = \begin{bmatrix} \nu_{1,0} & \nu_{1,1} & \nu_{1,2} & \dots & \nu_{1,L} & 0 & 0 & \dots & 0 \\ \nu_{2,0} & \nu_{2,1} & \nu_{2,2} & \dots & \nu_{2,L} & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \nu_{60,0} & \nu_{60,1} & \nu_{60,2} & \dots & \nu_{60,L} & 0 & 0 & \dots & 0 \end{bmatrix}$$

$$\hat{\mathbf{Z}} = \begin{bmatrix} z_{m,1} & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ z_{m,2} & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ z_{m,60} & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{bmatrix}. \quad (8.105)$$

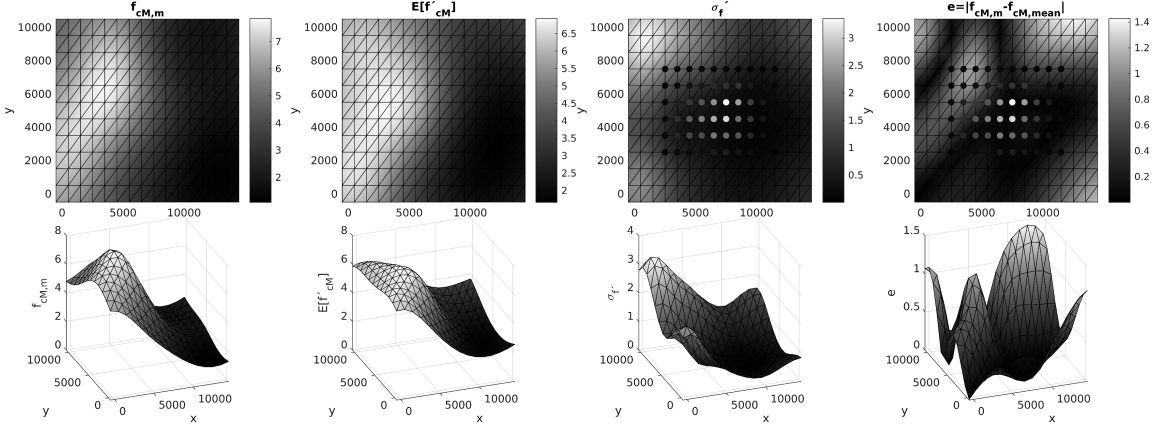


Figure 8.14: Result of the PCE-based Kalman filter field update: 2D and 3D view of the ‘true’ scaling factor (to the left), the mean (second column from left), and the standard deviation (third column from left) of the posterior scaling factor and the deviation of the mean from the ‘true’ (to the right). See explanation of the dots in Figure 8.12.

#### 4. Compute the coefficients of the updated input variables $\mathbf{Q}'$ .

The Kalman gain and the coefficients are then computed in the same way as in the previous example.

To have samples of the scaling factor, one first needs to sample from the updated PCE of  $\mathbf{Q}'$  (first by sampling from both germs,  $\boldsymbol{\xi}$  and  $\boldsymbol{\eta}$ , that is, from the 11 + 60 independent standard Gaussians, then by evaluating the polynomials at those sample points and multiplying it with the coefficients of  $\hat{\mathbf{Q}}'$ ). These samples are then mapped to the scaling factor by

$$\mathbf{f}_{cM,j} = e^{1.562+0.534\mathbf{V}\mathbf{S}\mathbf{q}'_j} \quad (8.106)$$

where  $\mathbf{V}$  and  $\mathbf{S}$  are computed in Example 3. The updated mean field of the scaling factor together with the ‘true’ field, the standard deviation of the posterior, and the deviation of the mean from the ‘true’ are shown in Figure 8.14. The result is very similar to the one with the MCMC, but the computation was much less demanding in terms of runtime here.

#### 8.5.4 Non-linear filters

It is shown for the PCE-based Kalman filter that the use of a functional approximation of the measurement model allows a functional approximation of the posterior. This method is much easier to compute than the sampling based MCMC approach. The Kalman filter applies a linear estimator to obtain the posterior. As presented before, the linearity of the estimator has its limitations. In an attempt to overcome this problem, it is shown in Chapter 4, how a non-linear filter can be built based on the conditional expectation, which can be seen as an extension to the Kalman filter. Here, we draft only the main ideas, to be able to implement it for the reservoir characterization task. The idea is to find for all  $L$  components  $Q_i$  of the parameter  $\mathbf{Q}$  an optimal estimator  $\varphi_i : \mathbb{R}^{n_y} \rightarrow \mathbb{R}$

minimizing the mean square error

$$\mathbb{E}[\|Q_i(\boldsymbol{\xi}) - \varphi_i(\mathbf{Z}(\boldsymbol{\xi}, \boldsymbol{\eta}))\|_2^2]. \quad (8.107)$$

Furthermore, we restrict this estimator to be in some finite dimensional space by writing it as a linear combination of some given basis functions  $\Psi_j$

$$\varphi_i = \sum_j \hat{\varphi}_{ij} \Psi_j(\mathbf{Z}). \quad (8.108)$$

The task is then to find the coefficients  $\hat{\varphi}_j$  from the minimisation of (8.107). As described in Chapter 4, the minimisation leads to the linear system

$$\mathbf{A} \hat{\boldsymbol{\varphi}}_i = \mathbf{b}_i, \quad (8.109)$$

where  $\hat{\boldsymbol{\varphi}}$  is the vector of coefficients, and the matrix  $\mathbf{A}$  and the vector  $\mathbf{b}_i$  are the integrals

$$[\mathbf{A}]_{ij} = \mathbb{E}[\psi_i(\mathbf{Z})\psi_j(\mathbf{Z})] = \int \psi_i(\mathbf{Z}(\boldsymbol{\xi}, \boldsymbol{\eta}))\psi_j(\mathbf{Z}(\boldsymbol{\xi}, \boldsymbol{\eta}))\pi_{\Xi}(\boldsymbol{\xi})\pi_E(\boldsymbol{\eta})d\boldsymbol{\xi}d\boldsymbol{\eta}, \quad (8.110)$$

$$[\mathbf{b}_i]_j = \mathbb{E}[Q_i\psi_j] = \int Q_i(\boldsymbol{\xi})\psi_j(\mathbf{Z}(\boldsymbol{\xi}, \boldsymbol{\eta}))\pi_{\Xi}(\boldsymbol{\xi})\pi_E(\boldsymbol{\eta})d\boldsymbol{\xi}d\boldsymbol{\eta}. \quad (8.111)$$

which can be evaluated with the help of numerical integration. Once the coefficients are computed from the system of Equations (8.109), the updated components of  $\mathbf{Q}$  can be written with the help of the MMSE filter by

$$Q'_i = Q_i(\boldsymbol{\xi}) - \varphi_i(\mathbf{Z}(\boldsymbol{\xi}, \boldsymbol{\eta})) + \varphi_i(\mathbf{z}_m). \quad (8.112)$$

The updated  $Q'_i$  would have the correct conditional mean if in Equation (8.108) an infinite series were taken, so that  $\varphi_i$  could be any measurable function. With the finite expansion, this is only approximately true.

The algorithm to compute an update of the  $L$  dimensional  $\mathbf{Q}$  parameter using the  $n_y$  dimensional measurement  $\mathbf{z}_m$  is as follows:

#### Algorithm 17: Parameter identification by the MMSE

1. Write input parameter  $\mathbf{Q}$ , measurement error model  $\mathbf{E}$ , and the response  $\mathbf{Y}$  in a PCE form

$$\mathbf{q} = \sum_{\alpha=0}^L \mathbf{q}_{\alpha} \Phi_{\alpha}^{(Q)}(\boldsymbol{\xi}) \quad \boldsymbol{\epsilon} = \sum_{\alpha=0}^{n_y} \mathbf{e}_{\alpha} \Phi_{\alpha}^{(E)}(\boldsymbol{\eta}) \quad \mathbf{y} = \sum_{\alpha=0}^{M-1} \mathbf{v}_{\alpha} \Phi_{\alpha}(\boldsymbol{\xi}).$$

2. Generate a combined PCE basis  $\hat{\Phi}_{\alpha}(\boldsymbol{\xi}, \boldsymbol{\eta})$ , and introduce the combined germ vector

$$\boldsymbol{\zeta} = [\boldsymbol{\xi} \quad \boldsymbol{\eta}]^T.$$



3. Rewrite the PCE coefficients in this extended basis

$$\mathbf{q} = \sum_{\alpha=0}^{n_c} \hat{\mathbf{q}}_{\alpha} \hat{\Phi}_{\alpha}(\zeta) \quad \boldsymbol{\epsilon} = \sum_{\alpha=0}^{n_c} \hat{\mathbf{e}}_{\alpha} \hat{\Phi}_{\alpha}(\zeta) \quad \mathbf{y} = \sum_{\alpha=0}^{n_c} \hat{\mathbf{v}}_{\alpha} \hat{\Phi}_{\alpha}(\zeta).$$

The measurement model then becomes

$$\mathbf{z}(\zeta) = \boldsymbol{\epsilon} + \mathbf{y} = \sum_{\alpha=0}^{n_c} \hat{\mathbf{e}}_{\alpha} \hat{\Phi}_{\alpha}(\zeta) + \sum_{\alpha=0}^{n_c} \hat{\mathbf{v}}_{\alpha} \hat{\Phi}_{\alpha}(\zeta) = \sum_{\alpha=0}^{n_c} (\hat{\mathbf{e}}_{\alpha} + \hat{\mathbf{v}}_{\alpha}) \hat{\Phi}_{\alpha}(\zeta).$$

4. Determine the basis for the approximation of the estimators  $\varphi_i$  (e.g. polynomials of maximum total degree  $p_{\varphi}$ ) by determining the basis functions  $\Psi_j(\mathbf{z})$  collected in a vector  $\boldsymbol{\Psi}$ .

5. Compute the elements of the matrix  $\mathbf{A}$ :

- (a) Determine the integration points  $\zeta_k$  and weights  $w_k$ ;
- (b) Evaluate the measurement model at the integration points  $\mathbf{z}_k = \mathbf{z}(\zeta_k)$ , and compute the elements of the matrix  $\mathbf{A}$  by

$$[\mathbf{A}]_{ij} \approx \sum_k i(\mathbf{z}_k) \psi_j(\mathbf{z}_k) w_k.$$

6. Compute the elements of vectors  $\mathbf{b}_i$ :

- (a) Determine integration points  $\zeta_l$  and weights  $w_l$  (which could be different from the ones we used in the computation of  $\mathbf{A}$ );
- (b) Evaluate the measurement model and all input parameters  $Q_i$  at the integration points, that is,  $\mathbf{z}_l = \mathbf{z}(\zeta_l)$ ,  $q_{i,l} = q_i(\zeta_l)$ , and compute the elements of vectors  $\mathbf{b}_i$  for all  $i = 1 \dots L$  by

$$[\mathbf{b}_i]_j \approx \sum_l q_{i,l} \psi_j(\mathbf{z}_l) w_l.$$

7. Solve the system of equations  $\mathbf{A} [\hat{\varphi}_1 \dots \hat{\varphi}_L] = [\mathbf{b}_1 \dots \mathbf{b}_L]$  for all coefficients  $\hat{\varphi}_i$  of the estimator.

The estimators are then given by

$$\varphi_i = \sum_j \hat{\varphi}_{ij} \Psi_j(\mathbf{Z}), \quad i = 1 \dots L.$$

The best estimate of  $\mathbf{Q}$  for all components of the input parameter is

$$q'_i = \varphi_i(\mathbf{z}_m) = \sum_j \hat{\varphi}_{ij} \Psi_j(\mathbf{z}_m) = \hat{\varphi}_i^T \boldsymbol{\Psi}(\mathbf{z}_m).$$

The components of the updated reference parameter reads  $Q'_i(\boldsymbol{\xi}, \boldsymbol{\eta}) = Q_i(\boldsymbol{\xi}) - \varphi_i(\mathbf{Z}(\boldsymbol{\xi}, \boldsymbol{\eta})) + q'_i$ .

One of the problematic parts of the method is the computation of integrals (8.110) and (8.111). For Gaussian  $\boldsymbol{\xi}$  and  $\boldsymbol{\eta}$ , the Gauss-Hermite quadrature can be used for the integration. As explained in Chapter 4 for the computation of matrix  $\mathbf{A}$ , an integration rule of order  $p_y p_\varphi + 1$  will suffice. Here,  $p_y$  is the maximum total degree of the PCE of  $\mathbf{y}$ , and  $p_\varphi$  is the degree of the estimator. For the computation of  $\mathbf{b}_i$ , a minimum order  $(p_x + p_y p_\varphi + 1)/2$  should be used. When the dimension  $L + n_y$  of the integration is too high, a sparse grid can be used to keep the number of points in a computationally affordable regime. If such a rule makes the computations too expensive, a MC or QMC integration rule could also be applied.

A big problem is that the dimension even for small problems can get very high. This results in a high dimensional integration demanding high computational cost or questionable accuracy. To release the burden of dimensionality, we propose here a low-rank approximation of the measurement model  $\mathbf{Z}$  by using the POD basis of the response  $\mathbf{Y}(\omega)$ . The POD approximation of the fluctuating part of  $\mathbf{Y}$  following Section 8.4.2 is given by

$$\tilde{\mathbf{Y}}(\omega) \approx \mathbf{V}_Y \mathbf{S}_Y \mathbf{Y}_r(\omega), \quad (8.113)$$

where

$$\mathbf{Y}_r = \mathbf{S}_Y^{-1} \mathbf{V}_Y^T \tilde{\mathbf{Y}} = \mathbf{P} \tilde{\mathbf{Y}}, \quad (8.114)$$

is the reduced model of the response field,  $\mathbf{P} = \mathbf{S}_Y^{-1} \mathbf{V}_Y^T$  is the projection matrix,  $\mathbf{V}_Y$  is the matrix collecting the eigenvectors  $\mathbf{v}_{y,i}$  of the covariance matrix  $\mathbf{C}_Y$  and  $\mathbf{S}_y$  is a diagonal matrix with the squared eigenvalues  $\lambda_{Y,i}$ . The eigenvalues and the eigenvectors are computed from

$$\mathbf{C}_Y \mathbf{v}_{y,i} = \lambda_{Y,i} \mathbf{v}_{y,i}. \quad (8.115)$$

$\mathbf{C}_Y$  is an  $n_y \times n_y$  matrix, thus we have at most  $n_y$  eigenvectors. We ignore the vectors with indices which are not contributing to the covariance of  $\mathbf{Y}$  more than some percentage  $\delta_\sigma$ . We use this eigenbasis to represent the measurement model, so we project the fluctuating part of the measurement model  $\tilde{\mathbf{Z}}$  onto this reduced basis spanned by the spatial vectors typical for the response, that is

$$\mathbf{Z}_r = \mathbf{P} \tilde{\mathbf{Z}} = \mathbf{P}(\mathbf{Z} - \mathbb{E}[\mathbf{Z}]) = \mathbf{P} \underbrace{(\mathbf{Y} - \mathbb{E}[\mathbf{Y}])}_{\tilde{\mathbf{Y}}} + \underbrace{\mathbf{P}\mathbf{E}}_{=\mathbf{E}_r} = \mathbf{Y}_r + \mathbf{E}_r, \quad (8.116)$$

where in the last but one equality we supposed a mean-free measurement error model  $\mathbf{E}$ . The transformed error  $\mathbf{E}_r = \mathbf{P}\mathbf{E}$  is not uncorrelated anymore, and its covariance matrix is

$$\mathbf{C}_{E_r} = \text{Cov}(\mathbf{P}\mathbf{E}) = \mathbf{P}\mathbf{C}_E\mathbf{P}^T. \quad (8.117)$$

This correlated error model  $\mathbf{E}_r$  can be represented in a smaller stochastic space—by a germ  $\boldsymbol{\eta}_r$  with much lower stochastic dimension  $R$  than  $n_y$ —namely, the projected error can be written by a simple linear transformation of the germ,

$$\boldsymbol{\epsilon}_r = \mathbf{L}\boldsymbol{\eta}_r, \quad (8.118)$$

where  $\mathbf{L}$  is the lower triangular matrix from the Cholesky decomposition of the covariance matrix  $\mathbf{C}_{E_r}$ , that is

$$\mathbf{C}_{E_r} = \mathbf{L}\mathbf{L}^T. \quad (8.119)$$

The projected error can be also given in a PCE form

$$\epsilon_r = \sum_{\beta=1}^R \mathbf{l}_\beta \Phi^{(E_r)}(\boldsymbol{\eta}_r) = \mathbf{L}\boldsymbol{\Phi}^{(E_r)}, \quad (8.120)$$

where  $\mathbf{l}_\beta$  are the columns of the matrix  $\mathbf{L}$  and  $\Phi^{(E_r)}$  are the linear univariate Hermite polynomials. Plugging in the PCE approximations of  $\mathbf{E}_r$  and  $\mathbf{Y}$  into Equation (8.116), the fluctuating part of the measurement model is approximated by

$$\tilde{\mathbf{Z}}_r \approx \mathbf{P} \left( \sum_{\alpha=0}^{M-1} \mathbf{v}_\alpha \Phi_\alpha(\boldsymbol{\xi}) - \mathbb{E}[\mathbf{Y}] \right) + \sum_{\beta=1}^R \mathbf{l}_\beta \Phi^{(E_r)}(\boldsymbol{\eta}_r) = \mathbf{P}\tilde{\boldsymbol{\Upsilon}}\boldsymbol{\Phi} + \mathbf{L}\boldsymbol{\Phi}^{(E_r)}, \quad (8.121)$$

where the  $\tilde{\boldsymbol{\Upsilon}}$  is the matrix  $\boldsymbol{\Upsilon}$  with the zeroed out first column for subtraction of the mean. This latest description has the advantage that the dimension of the germ shrunk from  $L + n_y$  to  $L + R$ . If the response  $\mathbf{Y}(\boldsymbol{\xi})$  is smooth, only a few number of eigenmodes  $R$  already give a good approximation. The optimal estimator will then be a function of the projected error model and can be computed by solving the linear system in Equation (8.109), where in the computation of matrix  $\mathbf{A}$  and the right hand side  $\mathbf{b}$  (see Equations (8.110) and (8.111)) the measurement model  $\mathbf{Z}(\boldsymbol{\xi}, \boldsymbol{\eta})$  must be replaced by the projected one  $\mathbf{Z}_r(\boldsymbol{\xi}, \boldsymbol{\eta}_r)$ . Once the optimal maps  $\phi_i$  are computed, the best estimate of  $Q_i$  can be computed by plugging in the projected measurement  $\mathbf{z}_m$  into the optimal estimator

$$q'_m = \phi_i(\mathbf{P}(\mathbf{z}_m - \mathbb{E}[\mathbf{Y}])). \quad (8.122)$$

Using the MMSE filter, the posterior input parameter becomes

$$Q'_i = Q_i(\boldsymbol{\xi}) - \phi_i(\mathbf{Z}_r(\boldsymbol{\xi}, \boldsymbol{\eta}_r)) + q'_m. \quad (8.123)$$

We only present here examples of updating the spatially not varying scaling factor. When we drop the assumption of the scalar scaling factor, the field update by MMSE is done the same way, and the update gives very similar result to the one computed by MCMC or the Kalman filter (see Figure 8.17). This is normal due to the fact that we used a linear PCE of the response  $\mathbf{Y}$ , and so the limitation is more in the PCE approximation of the response. As the response surface is slightly non-linear, we could improve the accuracy of the MCMC update and also the one of a non-linear estimator by a higher degree approximation of the response, but that would greatly increase the cost of the surrogate modelling.

**Example 9: Computation of the low rank representation of the measurement model for the MMSE update.** Consider the first scenario, where  $Q$  is a scalar valued random variable, that is,  $L = 1$  and a PCE of  $\mathbf{Y}$  of maximum polynomial degree  $p_y = 5$ . When we aim at finding a quadratic estimator, that is,  $p_\varphi = 2$  by the MMSE estimator, for the computation of the matrix  $\mathbf{A}$  by (8.110) we would need to generate integration points from the  $p_y p_\varphi + 1 = 11$ -point univariate rule, which results in  $11^{(L+n_y)} = 11^{61} = 3.349 \cdot 10^{63}$  points using a full tensor grid. Instead of replacing the integration rule with a Smolyak sparse grid or a QMC grid, we use the low-rank approximation of the measurement model as explained above.

1. **Compute the covariance matrix  $\mathbf{C}_Y$ .**

The  $60 \times 60$  covariance matrix was determined from the PCE coefficients using Equation (8.43).

2. **Compute the eigenvectors and eigenvalues of  $\mathbf{C}_Y$  and truncate.**

From the eigenvalues  $\{\lambda_i\}_{i=1}^{n_y}$ , it can be shown that more than 99% of the total variance of  $\mathbf{Y}$  can be represented by using only  $R = 3$  modes out of the 60, that is

$$\frac{\sum_{i=1}^3 \lambda_i}{\sum_{j=1}^{n_y} \lambda_j} > 0.99. \quad (8.124)$$

We collect the first three computed eigenvectors  $\{\mathbf{v}_{y,i}\}_{i=1}^3$  in the columns of the matrix  $\mathbf{V}_y$ , and the first three eigenvectors in the diagonal of the matrix  $\mathbf{S}$  and set their product to be the projection matrix  $\mathbf{P} = \mathbf{S}_y^{-1} \mathbf{V}_y^T$ .

3. **Compute the PCE of the projected measurement error  $\mathbf{E}_r$ .**

The covariance matrix of the measurement error model  $\mathbf{C}_E$  is a  $60 \times 60$  diagonal matrix whose diagonal entries are the variances of the error at the sixty assimilation points. The  $3 \times 3$  sized covariance matrix of the projected measurement error  $\mathbf{C}_{E_r}$  is computed by Equation (8.117). For the PCE of the projected measurement error  $\mathbf{E}_r$ , we first carry out the Cholesky decomposition (8.119) of  $\mathbf{C}_{E_r}$  which gives us a lower triangular matrix  $\mathbf{L}$  of size  $R \times R$ . The new, projected measurement error can be represented in PCE form by

$$\mathbf{E}_r = \mathbf{L} \Phi^{(E_r)} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} \eta_{r,1} \\ \eta_{r,2} \\ \eta_{r,3} \end{bmatrix}. \quad (8.125)$$

4. **Determine the PCE of the projected measurable response  $\mathbf{Y}_r$ .**

The computation of the PCE coefficients  $\Upsilon$  was performed as in Example 1, but here we used polynomial degree  $p_y = 5$ . In order to get only the fluctuating part  $\tilde{\mathbf{Y}}$  of  $\mathbf{Y}$ , the PCE coefficients corresponding to the basis function  $\Phi_0 = 1$  are set to zero. The coefficients  $\tilde{\Upsilon}$  of  $\tilde{\mathbf{Y}}$  are then projected onto the main three eigenvectors by

$$\Upsilon_r = \mathbf{P} \tilde{\Upsilon}. \quad (8.126)$$

The PCE of the projected model  $\mathbf{Y}_r$  becomes

$$\mathbf{Y}_r = \mathbf{P} \tilde{\Upsilon} \Phi(\xi) = \Upsilon_r \Phi(\xi). \quad (8.127)$$

5. **Compute the projected measurement  $\mathbf{z}_{r,m}$**

$$\mathbf{z}_{r,m} = \mathbf{P}(\mathbf{z}_m - \mathbb{E}[\mathbf{Y}]). \quad (8.128)$$

**Example 10: Test low-rank approximation on the MCMC update.** The low-rank approximation was first tested by using it in an MCMC computation. The procedure is the same as in Example 5, the only difference being that we use  $\mathbf{z}_{r,m}$  instead of  $\mathbf{z}_m$ , and  $\mathbf{Y}_r$  instead of the response  $\mathbf{Y}$ . The likelihood is computed from the multivariate Gaussian distribution of the projected error by

$$L(q) = \det(2\pi \mathbf{C}_{E_r})^{-1/2} e^{-\frac{1}{2}(\mathbf{z}_{r,m} - \mathbf{Y}_r(\mathcal{I}^{-1}q))^T \mathbf{C}_{E_r}^{-1}(\mathbf{z}_{r,m} - \mathbf{Y}_r(\mathcal{I}^{-1}q))}. \quad (8.129)$$

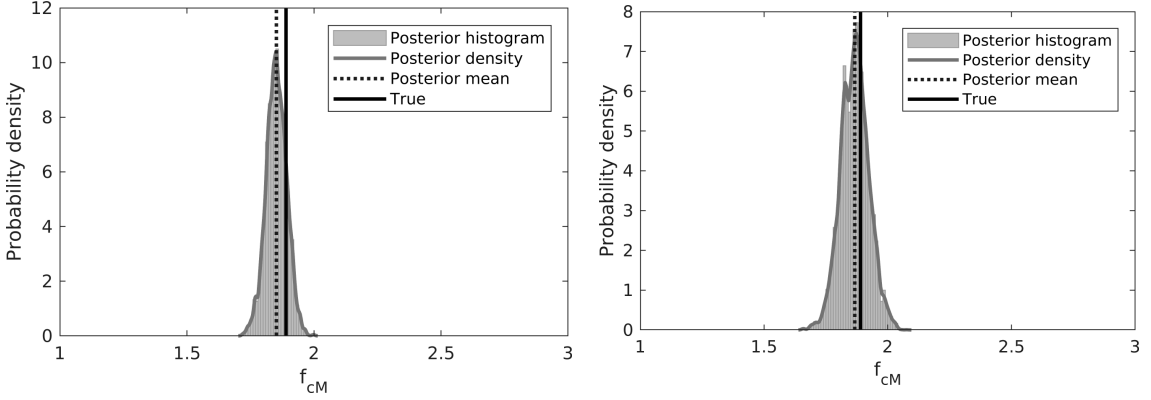


Figure 8.15: Testing low-rank approximation on MCMC update: the posterior density estimate of  $F_{cM}$  without (to the left) and with low-rank approximation (to the right).

Results of the update are shown in Figure 8.15. Cutting the stochastic dimension of the measurement model from  $n_y = 60$  to  $R = 3$  did not influence significantly the results of the MCMC update.

**Example 10: Updating the scalar scaling factor by a quadratic MMSE filter.** Here we present how to apply the MMSE filter to the reservoir characterization.

### 1. Write $Q$ , $E$ , and $Y$ in a PCE form.

The PCE expansions would be the same as the ones given in the first point of Example 7, but we use here the projected response  $\mathbf{Y}_r$  and the projected error  $\mathbf{E}_r$  whose PCE representation are explained in Example 9 and given in Equations (8.127) and (8.125).

### 2. Generate a combined PCE basis.

The PCE basis becomes

$$\hat{\Phi} = [1 \quad \xi \quad \xi^2 - 1 \quad \xi^3 - 3\xi \quad \xi^4 - 6\xi^2 + 3 \quad \xi^5 - 10\xi^3 + 15\xi \quad \eta_{r,1} \quad \eta_{r,2} \quad \eta_{r,3}]^T \quad (8.130)$$

with the combined germ

$$\zeta = [\xi \quad \eta_{r,1} \quad \eta_{r,2} \quad \eta_{r,3}]. \quad (8.131)$$

### 3. Rewrite the coefficients in the extended basis.

The coefficients in the extended basis are assembled in the same manner as in Example 7. Due to the low-rank representation, the modified sizes of the coefficient matrices of  $Q$ ,  $\mathbf{Y}_r$ , and  $\mathbf{E}_r$  are

$$\hat{Q} \in \mathbb{R}^{1 \times 9}, \quad \hat{\mathbf{Y}}_r \in \mathbb{R}^{3 \times 9}, \quad \hat{\mathbf{E}}_r \in \mathbb{R}^{3 \times 9}. \quad (8.132)$$

The PCE coefficients of the measurement model  $\mathbf{Z}_r$  in this extended basis are simply given by adding the coefficients of the response and the error, that is,

$$\hat{\mathbf{Z}}_r = \hat{\mathbf{Y}}_r + \hat{\mathbf{E}}_r. \quad (8.133)$$

#### 4. Chose approximating basis for the estimator.

We restrict the estimator  $\varphi$  to linear combinations of products of the monomials. We tried different degree estimators. The basis of maximum total degree two example is

$$\Psi = [1 \quad z_{r,1} \quad z_{r,2} \quad z_{r,3} \quad z_{r,1}^2 \quad z_{r,1}z_{r,2} \quad z_{r,1}z_{r,3} \quad z_{r,2}^2 \quad z_{r,2}z_{r,3} \quad z_{r,3}^2]^T, \quad (8.134)$$

thus, 10 basis functions of the projected measurement model  $\mathbf{z}_r$ <sup>8</sup>. Without the projection, there would be 1,891 basis functions.

#### 5. Compute matrix $\mathbf{A}$ and vector $\mathbf{b}$ .

We used the same integration rule for computing  $\mathbf{A}$  and  $\mathbf{b}$ . The tensor product of the one dimensional  $n = \max\{(p_y p_\varphi + 1), (1 + p_y p_\varphi + 1)/2\} = \max\{(5 \cdot 2 + 1), (1 + 5 \cdot 2 + 1)/2\} = 11$  point Gauss quadrature rule was used. This sums up to  $11^4 = 14,641$  integration points  $\zeta_k$ .

For the computation of the matrix  $\mathbf{A}$  and vector  $\mathbf{b}$ , we evaluate the measurement model  $\mathbf{Z}$  and the  $Q$  parameter at these integration points  $\zeta_k$  by using their PCE approximation in the extended basis.

#### 6. Solve the system of equation to determine the estimator.

The coefficients of the estimator  $\hat{\varphi}$  can be computed by solving the system of equations  $\mathbf{A}\hat{\varphi} = \mathbf{b}$ , where  $\varphi$  is a vector of 10 coefficients corresponding to the 10  $\Psi_i$  basis functions given above. The estimator is given by

$$\varphi = \sum_j \hat{\varphi}_j \Psi_j(\mathbf{z}_r). \quad (8.135)$$

The best estimate of  $Q$  is computed by plugging in the projected data  $\mathbf{z}_{r,m}$  computed in Equation (8.128) into the estimator

$$q'_m = \varphi(\mathbf{z}_{r,m}). \quad (8.136)$$

The updated  $Q'$  is

$$Q'(\zeta) = Q(\zeta) - \varphi(\mathbf{Z}_r(\zeta)) + q'_m. \quad (8.137)$$

To generate samples of  $Q'$ , we first sample from the germ of four identical and independent standard Gaussians ( $\zeta = [\xi, \eta_{r,1}, \eta_{r,2}, \eta_{r,3}]^T$ ), then we compute the input parameter and the projected measurement model from their PCE representation given in the extended basis. Then the above expression can be evaluated for any samples of the germ  $\zeta$ . By further mapping the values of  $q'$  to  $f_{cM}$ , we can estimate the posterior distribution, which is shown in Figure 8.16 using different degree estimators ( $p_\varphi = 1, 2, 3$ ).

---

<sup>8</sup>Please note, monomials are typically bad for approximation, but with such a low degree estimator it is applicable.

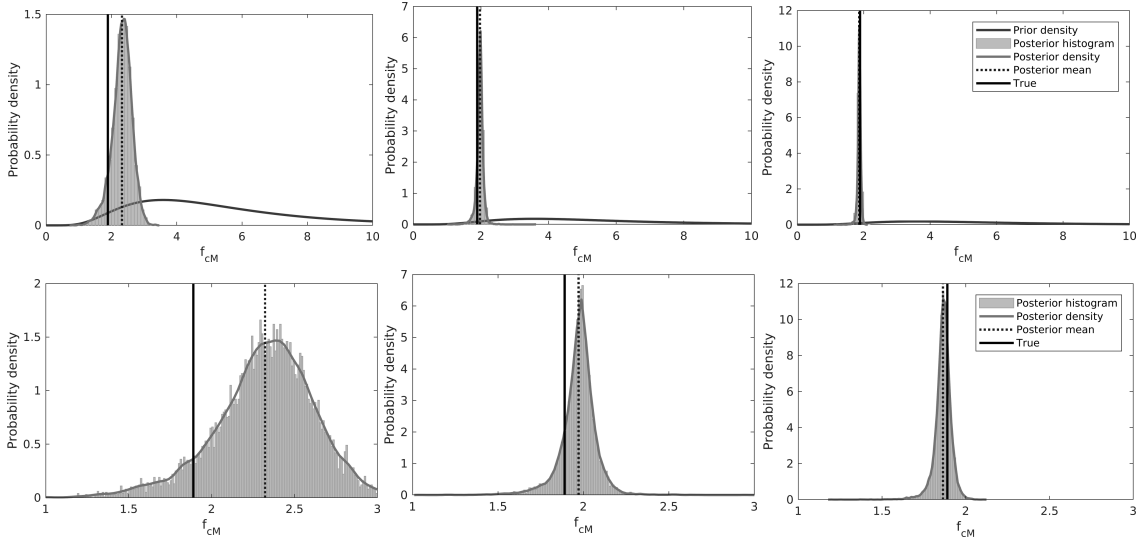


Figure 8.16: Update results of the MMSE: the prior probability density and the posterior density estimate of  $F_{cM}$  using different degree estimators (linear estimator to the left, degree two in the middle column and degree three to the right). In the bottom row the density is zoomed to the posterior.

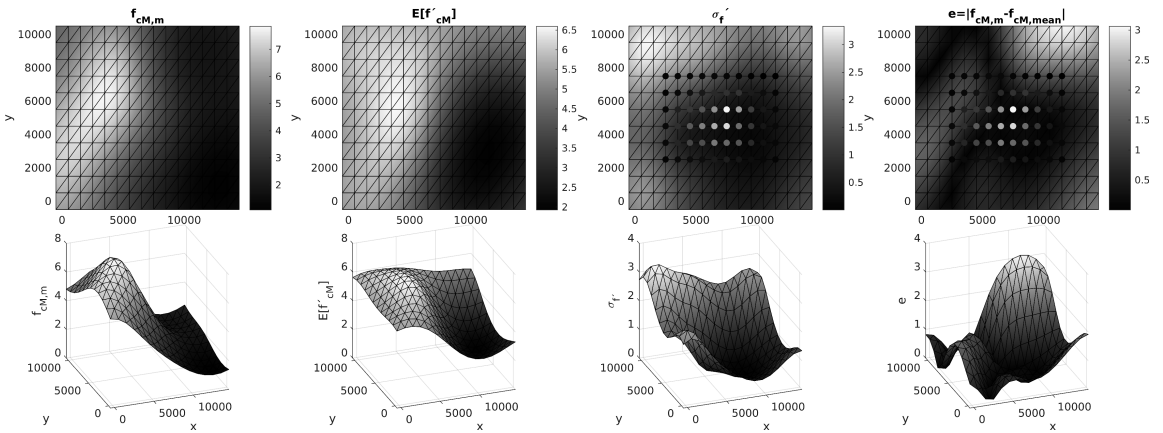


Figure 8.17: Result of the MMSE field update using a linear estimator and a low rank approximation of the measurement model, projecting to  $R = 10$  eigenvectors: 2D and 3D view of the ‘true’ scaling factor (to the left), the mean (second column from left), and the standard deviation (third column from left) of the posterior scaling factor and the deviation of the mean from the ‘true’ (to the right). See the explanation of the dots in Figure 8.12.

## 8.6 Summary, conclusion, and outlook

In this chapter, we presented a general framework for parameter and field identification using different techniques, but all of which are based on a PCE surrogate model of the response. The identification is presented on a gas reservoir problem, where we wished to identify the scaling factor  $F_{c_M}$  of the compressibility field by inference from measurement of seabed displacements. We provided examples for two different scenarios, one assuming a spatially not varying, scalar scaling factor and one dropping the homogeneity condition. A summary of the used identification methods is as follows:

- The Markov Chain Monte Carlo (MCMC) is based on a Metropolis-Hastings random walk, which generates samples of the posterior random field of the input parameter  $F_{c_M}$ . Though the method is converging very slowly, the computational expense was largely reduced by computing the likelihood at every random step from the PCE surrogate model instead of the deterministic solver.
- The Ensemble Kalman filter (EnKF) generates samples of the posterior  $F_{c_M}$  with the help of a linear estimator. First, an ensemble of the measurable response is created by sampling from the PCE surrogate model. The ensemble is then mapped by the linear estimator of the Kalman filter. The matrix of the Kalman gain is computed from the ensemble.
- The sample-free PCE-based Kalman-Filter (PCEKF) is similar to the EnKF method, but the Kalman-gain is computed from the PCE coefficients and the updated parameter  $F_{c_M}$  is determined in a PCE form, leading to an analytical form of the posterior scaling factor. Sampling was used here only for plotting the posterior density.
- The Minimum Mean Square Estimator (MMSE) also gives a functional representation of the updated parameter, but not necessarily in a PCE form. Here, the map from the measurement model to the updated parameter is not restricted to linear maps but to a finite dimensional space of polynomials and is determined by minimizing the minimum mean squared error. Here, different degree polynomials were tested ( $p = 1, 2, 3$ ).

The results for the posterior mean and variance and the deviation of the posterior from its ‘true’ value  $f_{c_M,m}$ —from which the synthetic measurement was generated—for the first scenario are shown in Table 8.2. The MCMC update method can be seen as a reference solution representing the Bayesian posterior distribution. We used 10,000 samples to get sufficiently accurate statistics of the posterior. The MCMC update method was also carried out using the low-rank representation of the measurement model to be able to see the error originating from projecting to the POD basis. This was important, because the MMSE method was only feasible with the low-rank approximation. It is shown in the examples that the Kalman filter can largely speed up the computation of the inverse method, but it only gives the exact posterior distribution if the input variable is Gaussian and the map to the measurement model is linear. For slightly non-linear problems, the PCE-based Kalman filter is a fast and efficient method to get an approximate posterior of the input variables. The MMSE method gives a good extension of the Kalman filter, and it is shown in the last example how the increased non-linearity of the filter can improve the update.



Table 8.2: Performance of the different PCE-based update methods: posterior mean  $\mathbb{E}[f'_{cM}]$ , the deviation of the mean from the true value  $f_{cM,m}$ , and the posterior variance  $\text{var}(f'_{cM})$ .

	MCMC	EnKF	PCEKF	MCMC LR	MMSE $p = 1$	MMSE $p = 2$	MMSE $p = 3$
low-rank	no	no	no	yes	yes	yes	yes
$\mathbb{E}[f'_{cM}]$	1.8521	2.1159	2.1066	1.8679	2.3235	1.9715	1.8647
$ \mathbb{E}[f'_{cM}] - f_{cM,m} $	0.0379	0.2259	0.2161	0.0221	0.4336	0.0814	0.0253
$\text{var}(f'_{cM})$	0.0016	0.0384	0.0303	0.0033	0.0907	0.0109	0.0026

For the identification of the spatially varying field, the same framework is used by separating the random field’s spatial structure from the stochastic one. This way the random field can be represented by a set of random variables. For the surrogate modelling in this scenario, we limited the PCE to a linear one to keep the computational cost affordable. This limitation constraints somewhat the update procedure and does not emphasize that much the difference between the different update methods. Even this linear PCE of the response resulted in a sufficiently accurate estimation of the spatially varying scaling factor, capturing well the spatial structure and the magnitude of the scaling factor. For the quality of the identification the location of the measurement points plays a crucial role. Obviously, in the region where no measurements are available, the scaling factor can not be updated, and also assimilation points with a measured response that are not sensitive to the scaling factor do not contribute much to the identification.

To conclude, we have shown how a PCE surrogate model of the response can enable different Bayesian techniques to update the model’s input parameter. The methods were all tested on the slightly non-linear model of the gas reservoir characterization, where the sample-free PCEKF gave a good rough estimate of the posterior with very low computational cost. The performance of the update could be further improved by using the non-linear filter of the MMSE method. It should be mentioned though that the computational cost of the MMSE method can easily grow very high, and we kept it in a manageable scale here by using a low-rank approximation of the measurement model. Another problem with the PCEKF and the MMSE methods could be that the conditional expectation is not so informative when the Bayesian posterior is multimodal, which can happen for strongly non-linear problems.



# Appendices



## † Appendix A: FEM computation of seabed displacements

The governing equilibrium equations to compute the seabed displacements read

$$\nabla \cdot \boldsymbol{\sigma}' - \alpha \nabla p = \rho \mathbf{g} + \mathbf{b}. \quad (138)$$

For a saturated porous volume  $\mathcal{D}$  with surface  $\Sigma$ , the equilibrium is prescribed minimizing the total potential energy by the virtual work theorem. By using the hypothesis of quasi-static conditions and neglecting the inertial forces, the work theorem formulation reads

$$\int_D \delta \boldsymbol{\epsilon} : (\boldsymbol{\sigma}' - \alpha p \mathbf{1}) dD = \int_D \delta \mathbf{u} \cdot \mathbf{b} dD + \int_{\Gamma_N} \delta \mathbf{u} \cdot \mathbf{t} d\Sigma, \quad (139)$$

where  $\boldsymbol{\epsilon}$  is the (small) strain tensor,  $\mathbf{u} = [ u_x(\hat{x}, \hat{y}, \hat{z}) \quad u_y(\hat{x}, \hat{y}, \hat{z}) \quad u_z(\hat{x}, \hat{y}, \hat{z}) ]^T$  is the displacement vector,  $\mathbf{1}$  the rank-2 identity tensor, and  $\mathbf{t}$  represents the total external forces acting per unit surface. The symbol  $\delta$  indicates the virtual variables. The integral formulation of equation (139) must be supplemented with appropriate Dirichlet and Neumann boundary conditions

$$\begin{aligned} \mathbf{u} &= \mathbf{u}_0 && \text{on } \Gamma_D, \\ \boldsymbol{\sigma} \cdot \mathbf{n} &= \mathbf{t} && \text{on } \Gamma_N, \quad \Gamma_N \cup \Gamma_D = \Gamma, \end{aligned} \quad (140)$$

in which  $\boldsymbol{\sigma}$  is the total stress. The FEM discretisation of Equation (139) is only briefly explained here. Interested readers are directed to [98, 173] for further details. Here, we introduce a set of basis functions collected in a row vector

$$\mathbf{N}_{\mathbf{u}} = \begin{bmatrix} N_{u1} & 0 & 0 & N_{u2} & 0 & 0 & \cdots & N_{un} & 0 & 0 \\ 0 & N_{u1} & 0 & 0 & N_{u2} & 0 & \cdots & 0 & N_{un} & 0 \\ 0 & 0 & N_{u1} & 0 & 0 & N_{u2} & \cdots & 0 & 0 & N_{un} \end{bmatrix}, \quad (141)$$

and restrict our solution space and the space of virtual displacements to the span of these basis functions. Accordingly, the virtual and real displacements and strains are approximated by

$$\begin{aligned} \delta \mathbf{u} &\approx \mathbf{N}_u \delta \mathbf{a} & \delta \boldsymbol{\epsilon} &\approx \mathbf{B} \delta \mathbf{a} \\ \mathbf{u} &\approx \mathbf{N}_u \mathbf{a} & \boldsymbol{\epsilon} &\approx \mathbf{B} \mathbf{a}, \end{aligned} \quad (142)$$

where the elements of  $\mathbf{a}$  are the unknown nodal displacements and  $\mathbf{B} = L \mathbf{N}_u$  with  $L$ , the differential operator defining the kinematic equation

$$\boldsymbol{\epsilon} \approx L \mathbf{u} \quad \text{or} \quad \delta \boldsymbol{\epsilon} \approx L \delta \mathbf{u}. \quad (143)$$

The fluid pore pressure  $p$  is also discretised by approximating it with the ansatz

$$p \approx \mathbf{N}_p \mathbf{p}, \quad (144)$$

where  $\mathbf{p}$  are the nodal component vectors of the pore pressure and  $\mathbf{N}_p$  is the vector of another set of basis functions. First, rearranging Equation (139) and then plugging in Equations (142)

through (144) into Equation (139) and differentiating with respect to the virtual variables, yields the equation:

$$\underbrace{\int_{\mathcal{D}} \mathbf{B}^T \hat{\boldsymbol{\sigma}}' dD}_{\Pi} = \underbrace{\int_{\mathcal{D}} \alpha \mathbf{B}^T \hat{\mathbf{N}}_p^T \mathbf{p} dD}_{\mathbf{f}} + \int_{\Gamma_N} \mathbf{N}_u^T \mathbf{t} d\Gamma, \quad (145)$$

where  $\hat{(\cdot)}$  is the Voigt notation of  $(\cdot)$ . In this case, a non-linear constitutive model is implemented leading to a nonlinear form of  $\Pi$ . Thus, the system of equations in (145) can be solved by a Newton-like method with the Jacobian matrix  $\mathbf{K}_T$  computed as

$$\mathbf{K}_T = \frac{d\Pi}{d\mathbf{a}} = \frac{d}{d\mathbf{a}} \int_{\mathcal{D}} \mathbf{B}^T \hat{\boldsymbol{\sigma}}'(\mathbf{a}) dD = \int_{\mathcal{D}} \mathbf{B}^T \frac{d\hat{\boldsymbol{\sigma}}'(\mathbf{a})}{d\mathbf{a}} dD = \int_{\mathcal{D}} \mathbf{B}^T \mathbf{C} \mathbf{B} dD. \quad (146)$$

The tangent constitutive tensor  $\mathbf{C}$  governs stress-strain relationship

$$d\hat{\boldsymbol{\sigma}}' = \mathbf{C} d\hat{\boldsymbol{\varepsilon}}. \quad (147)$$

For an isotropic porous material, the tangent operator  $\mathbf{C}$  reads

$$\mathbf{C} = c_M^{-1} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix}, \quad (148)$$

where  $\nu$  is the Poisson's moduli,  $c_M = [(1+\nu)(1-2\nu)]/[E(1-\nu)]$  is the oedometric compressibility and  $E$  is Young's modulus, and the tangent stiffness  $\mathbf{K}_T$  in Equation (146) depends linearly on  $c_M - 1$ . Following some experimental results, the constitutive model depends on the stress state of the seabed. The Poisson's moduli can be fixed to a constant value but the oedometric compressibility and thus the Young's modulus was observed to depend on the vertical component of the effective stress,  $\sigma'_z$  by a power relation

$$c_M(\sigma_z) = c_{M0} \left( \frac{\sigma'_z}{f_0} \right)^{-\lambda}, \quad (149)$$

where  $f_0$  is a fixed value of stress enabling a dimensionless form, and  $\lambda$  and  $c_{M0}$  are material coefficients estimated via measurements. For more detail about this experimental constitutive law, the reader is directed to [173].

## 2 Appendix B: Hermite polynomials

### 2.1 Generation of Hermite Polynomials

The univariate Hermite polynomials  $H$  are orthogonal polynomials with respect to the standard Gaussian distribution [71], that is they satisfy the orthogonality condition

$$\mathbb{E}[H_i(\xi)H_j(\xi)] = \int_{\mathbb{R}} H_i(\xi)H_j(\xi)\pi_{\Xi}(\xi)d\xi = h_i\delta_{ij}, \tag{150}$$

where  $\delta_{ij}$  is the Kroenecker delta,  $\pi_{\Xi}(\xi)$  is the standard Gaussian distribution

$$\pi_{\Xi}(\xi) = \frac{1}{\sqrt{2\pi}}e^{-\frac{\xi^2}{2}}, \tag{151}$$

and  $h_i$  is the squared norm of the polynomials, that is

$$h_i = \mathbb{E}[H_i(\xi)H_i(\xi)] = i!. \tag{152}$$

The Hermite polynomials can be generated by starting with the monomial basis  $\{1, \xi, \xi^2, \xi^3 \dots\}$  and determine the Hermite polynomials by the Gram-Schmidt orthogonalization. However, for numerical purposes it is more convenient to use the three term recurrence formula [71]

$$H_{n+1}(\xi) = (A_n\xi + B_n)H_n(\xi) - C_nH_{n-1}(\xi), \quad n \geq 0, \tag{153}$$

where the first two polynomials are set to

$$\begin{aligned} H_{-1} &= 0 \\ H_0 &= 1. \end{aligned}$$

For Hermite polynomials, the generating sequences are  $A_n = 1$ ,  $B_n = 0$  and  $C_n = n$ . For example, the first few Hermite polynomials are

$$H_0 = 1 \quad H_1 = \xi \quad H_2 = \xi^2 - 1 \quad H_3 = \xi^3 - 3\xi \quad H_4 = \xi^4 - 6\xi^2 + 3. \tag{154}$$

The generation of multivariate polynomials from the univariate Hermite polynomials is straightforward. The polynomials are generated by taking products of the univariate Hermite polynomials. The easiest is to generate the products by the so-called multi-index set, tuples of indices  $\alpha = (i_1, i_2, \dots, i_j, \dots)$  defining which indexed univariate polynomials are to be multiplied together. As an example, the multi-index set for polynomials with two variables of max total degree 3 and the

corresponding Hermite polynomials read

index	multi-	index	Deg. of polyn.	The $\alpha$ th multiv. polyn.
$i$	$\alpha = (i_1 \quad i_2)$		$ \alpha  = \sum_j i_j$	$\Phi_i = \Phi_\alpha = H_{i_1}(\xi_1)H_{i_2}(\xi_2)$
0	0	0	0	$\Phi_0 = H_0(\xi_1)H_0(\xi_2) = 1 \cdot 1 = 1$
1	1	0	1	$\Phi_1 = H_1(\xi_1)H_0(\xi_2) = \xi_1 \cdot 1 = \xi_1$
2	0	1	1	$\Phi_2 = H_0(\xi_1)H_1(\xi_2) = 1 \cdot \xi_2 = \xi_2$
3	2	0	2	$\Phi_3 = H_2(\xi_1)H_0(\xi_2) = (\xi_1^2 - 1) \cdot 1 = \xi_1^2 - 1$
4	1	1	2	$\Phi_4 = H_1(\xi_1)H_1(\xi_2) = \xi_1\xi_2$
5	0	2	2	$\Phi_5 = H_0(\xi_1)H_2(\xi_2) = 1 \cdot (\xi_2^2 - 1) = \xi_2^2 - 1$
6	3	0	3	$\Phi_6 = H_3(\xi_1)H_0(\xi_2) = (\xi_1^3 - 3\xi_1) \cdot 1 = \xi_1^3 - 3\xi_1$
7	2	1	3	$\Phi_7 = H_2(\xi_1)H_1(\xi_2) = (\xi_1^2 - 1) \cdot \xi_2 = \xi_1^2\xi_2 - \xi_2$
8	1	2	3	$\Phi_8 = H_1(\xi_1)H_2(\xi_2) = \xi_1 \cdot (\xi_2^2 - 1) = \xi_1\xi_2^2 - \xi_1$
9	0	3	3	$\Phi_9 = H_0(\xi_1)H_3(\xi_2) = 1 \cdot (\xi_2^3 - 3\xi_2) = \xi_2^3 - 3\xi_2$

The above tuples were generated in such a way that their sum is less or equal to three, corresponding to a maximal polynomial degree of three. The multivariate polynomials are still orthogonal with respect to the underlying multivariate Gaussian measure, that is, the orthogonality condition

$$\mathbb{E}[\Phi_i(\boldsymbol{\xi})\Phi_j(\boldsymbol{\xi})] = \int_{\mathbb{R}^n} \Phi_i(\boldsymbol{\xi})\Phi_j(\boldsymbol{\xi})\pi_{\Xi}(\boldsymbol{\xi})d\boldsymbol{\xi} = \gamma_i\delta_{ij}, \tag{156}$$

holds for multivariate polynomials. In the above equation,  $\gamma_i$  is the squared norm of the multivariate polynomial with linear index  $i$ . To see why the orthogonality still holds, it is easier to index the multivariate polynomials with the multi-index set. As an example to show that, let us express the expected value of the product of two second variate Hermite polynomials with multi-indices  $\alpha = (k, \ell)$  and  $\beta = (m, n)$ :

$$\begin{aligned} \mathbb{E}[\Phi_\alpha(\boldsymbol{\xi})\Phi_\beta(\boldsymbol{\xi})] &= \int_{\mathbb{R}} \int_{\mathbb{R}} \underbrace{\Phi_{(k,\ell)}(\xi_1, \xi_2)}_{H_k(\xi_1)H_\ell(\xi_2)} \underbrace{\Phi_{(m,n)}(\xi_1, \xi_2)}_{H_m(\xi_1)H_n(\xi_2)} \pi_{\Xi_1}(\xi_1)\pi_{\Xi_2}(\xi_2)d\xi_1d\xi_2 \\ &= \int_{\mathbb{R}} \int_{\mathbb{R}} H_k(\xi_1)H_\ell(\xi_2)H_m(\xi_1)H_n(\xi_2)\pi_{\Xi_1}(\xi_1)\pi_{\Xi_2}(\xi_2)d\xi_1d\xi_2 \\ &= \underbrace{\int_{\mathbb{R}} H_k(\xi_1)H_m(\xi_1)\pi_{\Xi_1}(\xi_1)d\xi_1}_{h_k\delta_{km}} \underbrace{\int_{\mathbb{R}} H_\ell(\xi_2)H_n(\xi_2)\pi_{\Xi_2}(\xi_2)d\xi_2}_{h_\ell\delta_{\ell n}} \\ &= h_k h_\ell \delta_{km} \delta_{\ell n}. \end{aligned} \tag{157}$$

The squared norm  $\gamma_i$  of the multivariate polynomial  $\Phi_i$  with linear index  $i$  corresponding to the multi-index  $\alpha = (i_1, i_2, \dots)$  can be computed from the product of the squared norms of the univariate polynomials that generated the multivariate polynomial, that is

$$\gamma_i = \mathbb{E}[\Phi_\alpha(\boldsymbol{\xi})\Phi_\alpha(\boldsymbol{\xi})] = h_\alpha = \prod_k h_{i_k} = i_1!i_2! \dots \tag{158}$$

To conclude, the multivariate polynomials are generated from the product of univariate polynomials. When these univariate polynomials are orthogonal, the multivariate polynomials are also orthogonal and their squared norms can be computed from the product of the squared norms of the univariate polynomials.



### 2.2 Calculation of the norms

The squared norms of orthogonal univariate polynomials can be computed from the  $A_n$  and  $C_n$  sequences [71]

$$h_n = \mathbb{E}[H_n^2(\xi)] = h_0 C_1 C_2 \dots C_n \frac{A_n}{A_0} = n!, \tag{159}$$

where

$$h_0 = \mathbb{E}[H_0^2(\xi)] = \mathbb{E}[1] = 1. \tag{160}$$

For the underlying Gaussian measure, for the Hermite polynomials this is simply  $h_n = n!$ . As shown above in Equation (157), the squared norms of the multivariate Hermite polynomials can be computed from the product of squared norms of the univariate polynomials they were generated from.

### 2.3 Quadrature points and weights

The integration points and the weights can be also computed from the recurrence formula in Equation (153) but by first rewriting it to a form, which generates the monic version of the polynomials

$$\phi_{n+1}(\xi) = \left( 1\xi + \underbrace{\frac{B_n}{A_n}}_{\alpha_n} \right) \phi_n(\xi) - \underbrace{\frac{C_n}{A_n A_{n-1}}}_{\beta_n} \phi_{n-1}(\xi), \quad n \geq 0. \tag{161}$$

According to the Golub-Welsch algorithm [71], the integration points and integration weights can be calculated from the eigenvalues and the eigenvectors of the Jacobi matrix

$$J = \begin{bmatrix} \alpha_0 & \sqrt{\beta_1} & \sqrt{\beta_2} & & \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_1} & & \\ & \sqrt{\beta_2} & \alpha_2 & \sqrt{\beta_2} & \\ & & \ddots & \ddots & \ddots \end{bmatrix} \tag{162}$$

from the eigenvalue problem

$$Jv_i = \lambda_i v_i \quad \|v_i\| = 1. \tag{163}$$

The integration weights are given by the first component squared of this eigenvector

$$w_i = (v_{i,1})^2 \tag{164}$$

and the integration points by:

$$x_i = \lambda_i. \tag{165}$$

### 3 Appendix C: Galerkin solution of the Karhunen Loève eigenfunction problem

Here the target is to find an optimal expansion for the mean-free random field  $\tilde{\Theta}$  in the form

$$\tilde{\Theta}(\hat{x}, \hat{y}, \omega) = \sum_{i=1}^{\infty} \sigma_i X_i(\omega) r_i(\hat{x}, \hat{y}). \quad (166)$$

Let's suppose, that we already have an orthogonal set of basis functions,

$$\int_{\mathcal{D}} r_i(\hat{x}, \hat{y}) r_j(\hat{x}, \hat{y}) d\hat{x} d\hat{y} = \gamma_i \delta_{ij} \quad (167)$$

with squared norms  $\gamma_i$

$$\gamma_i = \int_{\mathcal{D}} r_i(\hat{x}, \hat{y}) r_i(\hat{x}, \hat{y}) d\hat{x} d\hat{y}. \quad (168)$$

The  $\sigma_i X_i$  coefficients of the expansion (166) can be computed by minimizing the squared error

$$\int_{\mathcal{D}} \left( \tilde{\Theta}(\hat{x}, \hat{y}, \omega) - \sum_{i=1}^{\infty} \sigma_i X_i(\omega) r_i(\hat{x}, \hat{y}) \right)^2 d\hat{x} d\hat{y}. \quad (169)$$

As in the orthogonal projection used in the stochastic space for computing the PCE coefficients, we can proceed here similarly, but in the spatial domain using the inner product

$$\langle a(\hat{x}, \hat{y}), b(\hat{x}, \hat{y}) \rangle_{L_2(\mathcal{D})} = \int_{\mathcal{D}} a(\hat{x}, \hat{y}) b(\hat{x}, \hat{y}) d\hat{x} d\hat{y}. \quad (170)$$

Then, the coefficient of the expansion can be computed from

$$\sigma_i X_i = \frac{\int_{\mathcal{D}} \tilde{\Theta}(\hat{x}, \hat{y}, \omega) r_i(\hat{x}, \hat{y}) d\hat{x} d\hat{y}}{\gamma_i}. \quad (171)$$

However, we would like that we use here a special set of  $r_i$  basis functions, the one that maximises the captured variance for a certain truncation of the expansion. Thus, we determine the spatial functions by the maximisation problem

$$\sum_i \mathbb{E}[(\sigma_i X_i)^2] = \sum_i \mathbb{E} \left[ \frac{(\int_{\mathcal{D}} \tilde{\Theta}(\hat{x}, \hat{y}, \omega) r_i(\hat{x}, \hat{y}) d\hat{x} d\hat{y})^2}{(\int_{\mathcal{D}} r_i(\hat{x}, \hat{y}) r_i(\hat{x}, \hat{y}) d\hat{x} d\hat{y})^2} \right] = \sum_i \sigma_i^2 \rightarrow \max. \quad (172)$$

In the above term, the expected value of the nominator can be rewritten in terms of the covariance function

$$\mathbb{E} \left[ \left( \int_{\mathcal{D}} \Theta(\hat{x}, \hat{y}, \omega) r_i(\hat{x}, \hat{y}) d\hat{x} d\hat{y} \right)^2 \right] = \int_{\mathcal{D}} \int_{\mathcal{D}} \underbrace{\mathbb{E}[\theta(\hat{x}, \hat{y}, \omega) \theta(\hat{x}', \hat{y}', \omega)]}_{=\text{Cov}_{\Theta}} r_i(\hat{x}', \hat{y}') r_i(\hat{x}, \hat{y}) d\hat{x}' d\hat{y}' d\hat{x} d\hat{y}, \quad (173)$$

which can be shown by the following equality

$$\int_{\mathcal{D}} \int_{\mathcal{D}} \underbrace{\mathbb{E}[\theta(\hat{x}, \hat{y}, \omega) \theta(\hat{x}', \hat{y}', \omega)]}_{=\text{Cov}_{\Theta}} r_i(\hat{x}', \hat{y}') r_i(\hat{x}, \hat{y}) d\hat{x}' d\hat{y}' d\hat{x} d\hat{y} = \quad (174)$$

$$= \mathbb{E} \left[ \int_{\mathcal{D}} \theta(\hat{x}, \hat{y}, \omega) r_i(\hat{x}, \hat{y}) d\hat{x} d\hat{y} \int_{\mathcal{D}} \theta(\hat{x}', \hat{y}', \omega) r_i(\hat{x}', \hat{y}') d\hat{x}' d\hat{y}' \right]. \quad (175)$$

Then the term to be maximised is

$$\frac{\langle \int_{\mathcal{D}} \text{Cov}_{\Theta}(\hat{x}, \hat{y}, \hat{x}', \hat{y}') r_i(\hat{x}', \hat{y}') d\hat{x}' d\hat{y}', r_i(\hat{x}, \hat{y}) \rangle_{L2(\mathcal{D})}}{\langle r_i(\hat{x}, \hat{y}), r_i(\hat{x}, \hat{y}) \rangle_{L2(\mathcal{D})}}.$$

It can be shown that the maximisation problem is equivalent to solving the eigenvalue problem

$$\int_{\mathcal{D}} \text{Cov}_{\Theta}(\hat{x}, \hat{y}, \hat{x}', \hat{y}') r_i(\hat{x}', \hat{y}') d\hat{x}' d\hat{y}' = \sigma_i^2 r_i(\hat{x}, \hat{y}) \quad (176)$$

for the  $r_i(\hat{x}, \hat{y})$  eigenfunctions and for the  $\lambda_i = \sigma_i^2$  eigenvalues. As the  $r_i(\hat{x}, \hat{y})$  eigenfunctions are in an infinite dimensional space, for a numerical solution one has to discretise this on some finite dimensional space by approximating  $r_i(\hat{x}, \hat{y})$  by a finite linear combination of some fixed spatial basis functions  $\Psi_j(\hat{x}, \hat{y})$ , like, for example, the FEM piecewise linear basis functions used for the FEM computation. We restrict these basis function to be some nodal basis with the Kronecker property

$$\psi_i(x_j) = \delta_{ij}. \quad (177)$$

The approximation of the eigenfunctions reads

$$r_i(x) \approx r_{h,i}(x) = \sum_{j=1}^n \Psi_j(\hat{x}, \hat{y}) v_{ji} = \mathbf{\Psi} \mathbf{V}, \quad (178)$$

where  $\mathbf{\Psi}$  is a row vector with the fixed spatial  $\Psi_i(\hat{x}, \hat{y})$  basis functions and  $\mathbf{V}$  is a matrix with the  $v_{ji}$  coefficients of the expansion. The covariance can be also discretised with the help of the  $\Psi_i$  nodal basis in the form

$$\text{Cov}_{\Theta}(\hat{x}, \hat{y}, \hat{x}', \hat{y}') \approx \text{Cov}_{h,\Theta}(\hat{x}, \hat{y}, \hat{x}', \hat{y}') = \sum_{l,m} \underbrace{\text{Cov}_{\Theta}(x_l, y_l, x_m, y_m)}_{=C_{lm}} \Psi_l(\hat{x}, \hat{y}) \Psi_m(\hat{x}', \hat{y}') \quad (179)$$

$$= C_{lm} \Psi_l(\hat{x}, \hat{y}) \Psi_m(\hat{x}', \hat{y}'), \quad (180)$$

where  $C_{lm}$  are the elements of the covariance matrix computed from the discrete spatial points. To solve the eigenproblem (176), we formulate it in a weak form, using the Galerkin method, that is by requiring the residual

$$\int_{\mathcal{D}} \text{Cov}_{h,\bar{\Theta}}(\hat{x}, \hat{y}, \hat{x}', \hat{y}') r_{h,i}(\hat{x}', \hat{y}') d\hat{x}' d\hat{y}' - \lambda_i r_{h,i}(\hat{x}, \hat{y}) \quad (181)$$

to be orthogonal to the approximating subspace, or in other words orthogonal to all the spatial basis functions  $\Psi_j$ . This orthogonality condition reads

$$\left\langle \Psi_j(\hat{x}, \hat{y}), \int_{\mathcal{D}} \text{Cov}_{h,\bar{\Theta}}(\hat{x}, \hat{y}, \hat{x}', \hat{y}') r_{h,i}(\hat{x}', \hat{y}') d\hat{x}' d\hat{y}' - \lambda_i r_{h,i}(\hat{x}, \hat{y}) \right\rangle_{L_2(\mathcal{D})} = 0 \quad j = 1 \dots n \quad (182)$$

$$\left\langle \Psi_j(\hat{x}, \hat{y}), \int_{\mathcal{D}} \text{Cov}_{h,\bar{\Theta}}(\hat{x}, \hat{y}, \hat{x}', \hat{y}') r_{h,i}(\hat{x}', \hat{y}') d\hat{x}' d\hat{y}' \right\rangle_{L_2(\mathcal{D})} = \left\langle \Psi_j(\hat{x}, \hat{y}), \lambda_i r_{h,i}(\hat{x}, \hat{y}) \right\rangle_{L_2(\mathcal{D})} \quad j = 1 \dots n. \quad (183)$$

Using the discretised approximations (8.53) and (179), the left side of the orthogonality condition (183) modifies to

$$\begin{aligned} & \left\langle \Psi_j(\hat{x}, \hat{y}), \int_{\mathcal{D}} \text{Cov}_{h,\bar{\Theta}}(\hat{x}, \hat{y}, \hat{x}', \hat{y}') r_{h,i}(\hat{x}', \hat{y}') d\hat{x}' d\hat{y}' \right\rangle_{L_2(\mathcal{D})} = \\ & = \int_{\mathcal{D}} \Psi_j(\hat{x}, \hat{y}) \int_{\mathcal{D}} \text{Cov}_{h,\bar{\Theta}}(\hat{x}, \hat{y}, \hat{x}', \hat{y}') r_{h,i}(\hat{x}', \hat{y}') d\hat{x}' d\hat{y}' d\hat{x} d\hat{y} \\ & = \int_{\mathcal{D}} \int_{\mathcal{D}} \Psi_j(\hat{x}, \hat{y}) \text{Cov}_{h,\bar{\Theta}}(\hat{x}, \hat{y}, \hat{x}', \hat{y}') r_{h,i}(\hat{x}', \hat{y}') d\hat{x}' d\hat{y}' d\hat{x} d\hat{y} \\ & = \int_{\mathcal{D}} \int_{\mathcal{D}} \Psi_j(\hat{x}, \hat{y}) \sum_{l,m} C_{lm} \Psi_l(\hat{x}, \hat{y}) \Psi_m(\hat{x}', \hat{y}') \sum_k v_{ki} \Psi_k(\hat{x}', \hat{y}') d\hat{x}' d\hat{y}' d\hat{x} d\hat{y} \\ & = \sum_{l,m,k} C_{lm} v_{ki} \int_{\mathcal{D}} \int_{\mathcal{D}} \Psi_j(\hat{x}, \hat{y}) \Psi_l(\hat{x}, \hat{y}) \Psi_m(\hat{x}', \hat{y}') \Psi_k(\hat{x}', \hat{y}') d\hat{x}' d\hat{y}' d\hat{x} d\hat{y} \\ & = \sum_{l,m,k} C_{lm} v_{ki} \underbrace{\int_{\mathcal{D}} \Psi_j(\hat{x}, \hat{y}) \Psi_l(\hat{x}, \hat{y}) d\hat{x} d\hat{y}}_{G_{jl}} \underbrace{\int_{\mathcal{D}} \Psi_m(\hat{x}', \hat{y}') \Psi_k(\hat{x}', \hat{y}') d\hat{x}' d\hat{y}'}_{G_{mk}} \\ & = \sum_{l,m,k} C_{lm} v_{ki} G_{jl} G_{mk} = \sum G_{jl} C_{lm} G_{mk} v_{ki} \\ & = \mathbf{GCGv}_i, \end{aligned} \quad (184)$$

where  $\mathbf{C}$  is the covariance matrix and  $\mathbf{G}$  is the Gramian matrix of the  $\Psi_i$  nodal basis and  $\mathbf{v}_i$  is the  $i$ th column of the  $\mathbf{V}$  matrix. Using the approximation (8.53), the right hand side of the Equation (183) reads

$$\begin{aligned} \left\langle \Psi_j(\hat{x}, \hat{y}), \lambda_i r_{h,i}(\hat{x}, \hat{y}) \right\rangle_{L_2(\mathcal{D})} &= \int_{\mathcal{D}} \Psi_j(\hat{x}, \hat{y}) \lambda_i \sum_k v_{ik} \Psi_k(\hat{x}, \hat{y}) d\hat{x} d\hat{y} \\ &= \lambda_i \sum_k v_{ik} \underbrace{\int_{\mathcal{D}} \Psi_j(\hat{x}, \hat{y}) \Psi_k(\hat{x}, \hat{y}) d\hat{x} d\hat{y}}_{G_{jk}} \\ &= \lambda_i \mathbf{G} \mathbf{v}_i. \end{aligned} \tag{185}$$

Using the above two forms, the discretised weak form of the eigenvalue problem (8.52) boils down to the discrete generalized eigenvalue problem

$$\mathbf{G} \mathbf{C} \mathbf{G} \mathbf{v}_i = \underbrace{\lambda_i}_{\sigma_i^2} \mathbf{G} \mathbf{v}_i. \tag{186}$$

#### 4 Appendix D: Computation of the PCE Coefficients by Orthogonal projection

In the following, we show how to approximate the response  $\mathbf{y}$ , a function of some random variable  $\boldsymbol{\xi}$  with a linear combination of the orthogonal polynomials  $\Phi_1(\boldsymbol{\xi}), \Phi_2(\boldsymbol{\xi}), \dots$ , such that the approximation  $\mathbf{y}_h$  minimizes

$$\|\boldsymbol{\epsilon}_h\|_{L_2(\Omega)}^2 = \mathbb{E} [\boldsymbol{\epsilon}_h^2] = \mathbb{E} [(\mathbf{y} - \mathbf{y}_h)^2] = \int_{\mathbb{R}^s} (\mathbf{y}(\boldsymbol{\xi}) - \mathbf{y}_h(\boldsymbol{\xi}))^2 \pi_{\Xi}(\boldsymbol{\xi}) d\boldsymbol{\xi}. \tag{187}$$

The above quadratic form is induced by the norm generated from the inner product defined as

$$\langle u, v \rangle_{L_2(\Omega)} = \mathbb{E} [uv] = \int_{\mathbb{R}^s} uv \pi_{\Xi} d\boldsymbol{\xi}, \tag{188}$$

thus, the norm can be written with the help of this inner product

$$\|u\|_{L_2(\Omega)} = \sqrt{\langle u, u \rangle_{L_2(\Omega)}} \tag{189}$$

and the value to be minimized is the squared induced norm, that is,

$$\mathbb{E} [\boldsymbol{\epsilon}_h^2] = \langle \boldsymbol{\epsilon}_h, \boldsymbol{\epsilon}_h \rangle. \tag{190}$$

The task is to approximate the function  $\mathbf{y}$ , lying in some infinite dimensional space, in a subspace, a finite dimensional space spanned by the given orthogonal polynomials. This problem is similar to approximating let's say a three-dimensional vector in a plane. The solution to this later task is to project orthogonally the vector to the approximating plane, which will minimize the projection error, the distance between the original vector and the projected one. Similarly, in the functional space, one can do the same thing, only we need to define a proper inner product, with which we can define orthogonality. By orthogonally projecting to the subspace, the norm generated from the inner product of the error is minimized.

Returning to our original task, it is easy to show that the minimum of the quadratic form is attained when the approximation error is orthogonal to each of the stochastic polynomials. This orthogonality is written by

$$\langle (\mathbf{y} - \mathbf{y}_h), \Phi_j \rangle = 0 \quad \forall j = 0 \dots (M - 1). \tag{191}$$

Plugging in the approximation (8.28) into this orthogonality equation, rearranging it and using the linearity of inner products and the orthogonality of the PCE polynomials in Equation (191), the  $\mathbf{v}_i$  coefficients can be expressed

$$\left\langle \sum_{i=0}^{M-1} \mathbf{v}_i \Phi_i, \Phi_j \right\rangle = 0 \quad \forall j = 0 \dots M - 1 \tag{192}$$

$$\langle \mathbf{y}, \Phi_j \rangle = \sum_{i=0}^{M-1} \mathbf{v}_i \underbrace{\langle \Phi_i, \Phi_j \rangle}_{\gamma_i \delta_{ij}} \quad \forall j = 0 \dots M - 1 \tag{193}$$

$$\mathbf{v}_i = \frac{1}{\gamma_i} \langle \mathbf{y}, \Phi_i \rangle = \frac{1}{\gamma_i} \mathbb{E} [\mathbf{y} \Phi_i] = \frac{1}{\gamma_i} \int_{\mathbb{R}^s} \mathbf{y}(\boldsymbol{\xi}) \Phi_i(\boldsymbol{\xi}) \pi_{\Xi}(\boldsymbol{\xi}) d\boldsymbol{\xi}. \tag{194}$$