An adaptive ANOVA-based PCKF for high-dimensional nonlinear inverse modeling

Weixuan Li, Guang Lin, Dongxiao Zhang

A Sonny Astani Department of Civil and Environmental Engineering, University of Southern California, Los Angeles, CA 90089, USA
b Pacific Northwest National Laboratory, Richland, WA 99352, USA
c Department of Energy and Resources Engineering, College of Engineering, Peking University, Beijing 100871, China

A R T I C L E   I N F O

Article history:
Received 2 August 2013
Received in revised form 20 October 2013
Accepted 13 November 2013
Available online 20 November 2013

Keywords:
Uncertainty quantification
Inverse modeling
Adaptive ANOVA
Kalman filter
Polynomial chaos

A B S T R A C T

The probabilistic collocation-based Kalman filter (PCKF) is a recently developed approach for solving inverse problems. It resembles the ensemble Kalman filter (EnKF) in every aspect—except that it represents and propagates model uncertainty by polynomial chaos expansion (PCE) instead of an ensemble of model realizations. Previous studies have shown PCKF is a more efficient alternative to EnKF for many data assimilation problems. However, the accuracy and efficiency of PCKF depends on an appropriate truncation of the PCE series. Having more polynomial chaos basis functions in the expansion helps to capture uncertainty more accurately but increases computational cost. Selection of basis functions is particularly important for high-dimensional stochastic problems because the number of polynomial chaos basis functions required to represent model uncertainty grows dramatically as the number of input parameters (random dimensions) increases. In classic PCKF algorithms, the PCE basis functions are pre-set based on users' experience. Also, for sequential data assimilation problems, the basis functions kept in PCE expression remain unchanged in different Kalman filter loops, which could limit the accuracy and computational efficiency of classic PCKF algorithms. To address this issue, we present a new algorithm that adaptively selects PCE basis functions for different problems and automatically adjusts the number of basis functions in different Kalman filter loops. The algorithm is based on adaptive functional ANOVA (analysis of variance) decomposition, which approximates a high-dimensional function with the summation of a set of low-dimensional functions. Thus, instead of expanding the original model into PCE, we implement the PCE expansion on these low-dimensional functions, which is much less costly. We also propose a new adaptive criterion for ANOVA that is more suited for solving inverse problems. The new algorithm was tested with different examples and demonstrated great effectiveness in comparison with non-adaptive PCKF and EnKF algorithms.

© 2013 Elsevier Inc. All rights reserved.

1. Introduction

Numerical models are powerful tools to simulate physical systems and make predictions of the quantities of our interest with specified model parameters. However, in many real applications, model parameters are subject to uncertainty, i.e., we do not have enough direct measurements of all model parameters to determine their exact values. To validate simulation results in this situation, we need to estimate the corresponding output uncertainty/error bounds induced by the uncertain input parameters. In other words, we study how the uncertainty propagates through the model from input (parameters)
to output (simulation results). This process is referred to as the “forward uncertainty quantification” (UQ) or “forward modeling.” On the other hand, if we are not satisfied with the scale of the prediction’s uncertainty/error bounds, we could reduce the parametric uncertainty by gathering more information. In addition to direct measurements of the parameters, we often have the observations of some of the output variables from the real physical system. By calibrating the simulation results to such observations, we obtain a less uncertain estimation of the parameters, as well as a more accurate prediction of other unobservable output variables. This process is referred to as “inverse UQ” or “inverse modeling” because it infers the input variables from the output.

UQ problems are much more challenging and time consuming than a deterministic simulation because they usually require multiple simulations of the same model with different parameter values. In comparing the two types of UQ problems, forward and inverse, inverse modeling generally is more difficult. Usually, what we have at hand is a simulator that solves the deterministic forward problem. For a single input submitted in the simulator, we get a unique output. In contrast, the simulator does not generate a model “input” with a given “output.” In fact, an inverse problem usually is ill-posed. Even for a single “output” value, there may be multiple corresponding “input” values that all provide a match to the “output” value. Essentially, this is because the information contained in the observed data is insufficient to determine all of the uncertain model parameters. Hence, the parametric uncertainty usually cannot be eliminated, merely reduced.

An inverse modeling problem can be formulated generally using Bayes’ theorem, which computes the posterior distribution of model parameters from the pre-specified prior distribution, as well as the observation of the model output. Although the posterior distribution provides all of the information regarding the remaining uncertainty associated with model parameters, it cannot be analytically expressed except for a few special cases. Instead, a sample of realizations can be drawn from the posterior distribution to represent the uncertainty. For instance, the Markov chain Monte Carlo method (MCMC), based on the Metropolis–Hastings (M–H) algorithm [1] or Gibbs sampler [2], is a popular sampling algorithm particularly suited for Bayesian inversion problems. Solving inverse modeling with sampling methods such as MCMC typically requires a very large number of model evaluations. Thus, it is computationally prohibitive for large-scale models that take a long time for a single run.

A special case in which the posterior distribution can be analytically derived is linear models with the prior density of parameters and observation error both normally distributed. Under these conditions, the posterior distribution of model parameters also is normal and fully characterized by its first two statistical moments: mean and covariance. The classic algorithm to solve these kinds of inverse problems is the Kalman filter [3]. In addition, the Kalman filter usually is implemented in a recursive manner. Suppose the data (observations) become available as a time sequence. For each node on this data sequence, we solve an inverse modeling problem with a Kalman filter loop. The posterior mean and covariance of model parameters computed from the previous loop serve as the prior for the next loop. Whenever new data are collected, the parameters are updated instantly. This process is known as “sequential data assimilation.”

For nonlinear systems, the posterior distribution generally is non-Gaussian, regardless of whether or not the prior density and observation error are Gaussian. As long as the nonlinearity is not strong, the Kalman filter still yields a good approximate solution of the posterior mean and covariance. The key step in implementing the Kalman filter is to compute the “Kalman gain,” which requires knowledge of the covariance of the (prior) joint distribution of model parameters and output variables. While this covariance is easily computed for linear systems, extra computational effort is necessary to estimate it for nonlinear models. Indeed, a forward UQ problem must be solved to study how the parametric uncertainty propagates through the nonlinear system to affect the output quantities. This becomes the main difficulty and computational overhead in extending the Kalman filter to nonlinear models.

The ensemble Kalman filter (EnKF), a relatively new nonlinear variant of the Kalman filter, represents and propagates uncertainty using the Monte Carlo method [4,5]. First, a number of realizations (an ensemble) of model parameters are randomly sampled from the prior distribution. Then, for each realization, a simulation is run to predict the corresponding realization of model outputs, and the required covariance is estimated from the joint ensemble. Finally, each realization is updated by the Kalman filter to form a new ensemble that represents the posterior distribution of model parameters. Due to the simplicity in its implementation, EnKF has gained great popularity within the past two decades. Since its invention, EnKF has been applied to many different research fields and has demonstrated great effectiveness. Evensen has provided a review of the development and applications of EnKF [6]. However, like all Monte Carlo approaches, EnKF accuracy relies on a sufficiently large ensemble, which causes an enormous computational burden for large-scale computational models.

An alternative to the Monte Carlo method for studying uncertainty propagation is the polynomial chaos expansion (PCE). In this approach, the random quantities under study are expanded using a polynomial chaos basis, which are the orthogonal polynomials with respect to a set of independent random variable with known distributions [7–9]. Once the PCE representation is obtained, the statistical moments of our interest (e.g., mean and covariance of the random quantities) can be easily computed from the coefficients in front of the polynomial chaos basis.

The key—and computationally demanding—step of implementing the PCE approach is solving for the PCE coefficients. Among the different methods available to tackle this problem, such as the stochastic Galerkin projection [8] and regression method [10], the probabilistic collocation method (PCM) [11] is particularly convenient and effective. PCM constructs the PCE approximation by interpolation, where the interpolation points are called “collocation points.” Similar to Monte Carlo, PCM is a non-intrusive approach, which means it treats the model as a black box and requires only repetitive evaluations of the model at the collocation points with a deterministic simulator. To achieve high accuracy in estimating the statistical moments, which by definition are integrals over the random space, the collocation points usually are deployed the same
way as the quadrature points used in numerical integration schemes, such as Gauss quadrature points. For high-dimensional PCM, commonly used collocation schemes include sparse grid and Stroud quadrature points [12]. A discussion regarding the selection of collocation points is given by Eldred and Burkardt [13]. PCM was compared with the Monte Carlo method in previous studies and demonstrated better efficiency in solving forward UQ problems for many different models [14–17].

Similar to the EnKF, which combines the Monte Carlo method with the Kalman filter, the so-called "probabilistic collocation-based Kalman filter" or "polynomial chaos-based ensemble Kalman filter" (PCKF), which combines the PCE with the Kalman filter, was developed to solve inverse modeling problems [18–22]. PCKF resembles EnKF in almost every aspect except that it uses the PCE to represent uncertainty. While an EnKF user must decide on the size of the ensemble before running the algorithm, a PCKF user has to determine in advance the truncation of the PCE, i.e., to select the basis functions to form the PCE approximation. The trade-off is that keeping more PCE basis functions helps to capture uncertainty more accurately, but it increases the computational cost. An ideal PCE should accurately represent the model uncertainty but keep the number of basis functions as small as possible. Dealing with this issue is particularly vital in solving high-dimensional problems (i.e., stochastic models with a large number of uncertain parameters) because the total number of PCE terms can grow dramatically fast as the dimensionality increases.

In this study, we develop a new algorithm that adaptively selects active PCE basis functions for uncertainty representation in different problems and automatically adjusts the number of basis functions in different Kalman filter loops. We construct the PCE based on adaptive functional ANOVA (analysis of variance) decomposition. Functional ANOVA, also referred to as "high-dimensional model representation" (HDMR), was shown as an effective dimensionality reduction method [23,24] and was combined with PCE approximation to solve forward UQ problems [25]. Now, we extend this methodology toward solving inverse modeling problems. Because in many physical models the coupling effect of a large number of input parameters on the model output can be reduced to the coupling effect of only a few, functional ANOVA decomposition is able to approximate a high-dimensional function with the summation of a set of low-dimensional functions, known as ANOVA components. For different models, the components of the ANOVA decomposition may be adaptively calculated following some adaptive criterion, which greatly reduces the computational cost and improves efficiency [26,27]. Once an ANOVA decomposition is done, the ANOVA components (low-dimensional) can be expanded with PCE, which is much less costly than expanding the original model (high-dimensional).

This paper is organized as follows: Section 2 reviews the approach of Kalman filter and its nonlinear variants, EnKF and PCKF. In Section 3, we formulate the ANOVA-based PCKF and discuss the adaptive criteria of selecting ANOVA components. The adaptive ANOVA-based PCKF method is demonstrated in two illustrative examples and compared with EnKF and non-adaptive PCKF approaches in Section 4. Finally, a summary and conclusions are provided in Section 5.

2. Inverse modeling and Kalman filters

2.1. Problem statement

We formulate a general inverse modeling problem as follows. Suppose a physical system can be simulated with a computer model:

\[ \mathbf{d} = \mathbf{g}(\mathbf{m}). \]  

(1)

where \( \mathbf{m} \) is a vector consisting of the model parameters and \( \mathbf{d} \) is the vector of output variables that are evaluated from the model and can be checked against observations of the real system. We assume the true values of the model parameters are not known. Instead, we have a probability distribution function (pdf) \( p(\mathbf{m}) \), known as the prior pdf, that represents all of our knowledge about \( \mathbf{m} \). For instance, \( p(\mathbf{m}) \) may come from an expert’s guess or some direct, but limited, measurements on \( \mathbf{m} \). To reduce the parametric uncertainty represented by \( p(\mathbf{m}) \) and obtain a more accurate estimation of \( \mathbf{m} \), an observation \( \mathbf{d}^* \) of the real system is made. Note that even when the parameter is correctly assigned, observation \( \mathbf{d}^* \) could differ from simulation result \( \mathbf{d} \) due to some error, e.g., modeling error (simplification of physics, numerical approximation, etc.) and measurement error.

\[ \mathbf{d}^* = \mathbf{d} + \mathbf{e}_d. \]  

(2)

where \( \mathbf{e}_d \) is the error term characterized by another pdf, which usually is independent of \( p(\mathbf{m}) \). The objective of the inverse modeling is to estimate the distribution of \( \mathbf{m} \) conditioned to the observation \( \mathbf{d}^* \), known as the posterior pdf \( p(\mathbf{m}|\mathbf{d}^*) \). The posterior pdf is given by the Bayes’ theorem, although it generally cannot be explicitly and analytically derived but can be estimated with some numerical algorithm.

2.2. Kalman filter

The Kalman filter gives the exact solution of the inverse modeling problem when the following conditions are satisfied:

(1) The model \( g(\bullet) \) is linear. Thus, it can be rewritten as \( \mathbf{d} = \mathbf{G}\mathbf{m} \), where \( \mathbf{G} \) is a matrix.

(2) Both the prior density \( p(\mathbf{m}) \) and error \( \mathbf{e}_d \) follow normal distributions.
In this situation, the posterior distribution \( p(m|d^*) \) also is normal, and its mean \( \mu_{m|d^*} \) and covariance \( C_{mm|d^*} \) are given by the formulas of the Kalman filter:

\[
\begin{align*}
\mu_{m|d^*} &= \mu_m + K(d^* - \mu_d), \\
C_{mm|d^*} &= C_{mm} - KC_{dm},
\end{align*}
\]

where \( \mu_m \) and \( C_{mm} \) are the prior mean and covariance of parameter vector \( m \), respectively. \( \mu_d \) is the predicted/prior mean of \( d \). \( C_{dm} \) is the prior covariance between vectors \( d \) and \( m \).

\( K \) is a matrix named Kalman gain:

\[
K = C_{md}(C_{dd} + R)^{-1},
\]

where \( C_{md} = (C_{dm})^T \), \( C_{dd} \) is the predicted/prior covariance of vector \( d \), and \( R \) is the covariance of the error \( e_d \), which usually is pre-set based on users' knowledge of the error. Implementation of the Kalman filter requires first solving a forward UQ problem, i.e., to estimate the predicted mean \( \mu_d \), as well as the covariance matrices \( C_{dm} \) and \( C_{dd} \). They are easily solved for linear models:

\[
\begin{align*}
\mu_d &= G\mu_m, \\
C_{dm} &= GC_{mm}, \\
C_{dd} &= GC_{mm}G^T.
\end{align*}
\]

Often, the observation data become available as a sequence:

\[
d = [d^{(1)}^T, d^{(2)}^T, \ldots]^T = [(g^{(1)}(m))^T, (g^{(2)}(m))^T, \ldots]^T.
\]

In this situation, the Kalman filter may be implemented in a recursive manner. For each node \( d^{(i)} \) on the data sequence, we solve an inverse problem with Eqs. (3)–(8). The resulting posterior then serves as the prior for the next Kalman filter loop when the new data \( d^{(i+1)} \) is available.

### 2.3. Ensemble Kalman filter

The Kalman filter can be extended to inverse problems of nonlinear models for which it gives an approximate solution provided that the nonlinearity is not strong, in which case the posterior distribution still is unimodal and well described by its first two statistical moments. However, to propagate uncertainty from \( m \) to \( d \) is not a simple task when a model is nonlinear. One of the most commonly used approaches is the Monte Carlo method, which leads to a variant of the Kalman filter: the EnKF. In EnKF we initially draw a random ensemble of realizations from the prior distribution of the parameters then propagate the uncertainty by running simulations for each individual realization to generate the corresponding ensemble of output \( d \).

\[
d_i = g(m_i), \quad i = 1, 2, \ldots, N_e.
\]

The required statistical moments are estimated from the ensembles:

\[
\hat{\mu}_d = \frac{1}{N_e} \sum_{i=1}^{N_e} d_i, \\
\hat{C}_{dd} = \frac{1}{N_e - 1} \sum_{i=1}^{N_e} (d_i - \hat{\mu}_d)(d_i - \hat{\mu}_d)^T, \\
\hat{C}_{dm} = \frac{1}{N_e - 1} \sum_{i=1}^{N_e} (d_i - \hat{\mu}_d)(m_i - \hat{\mu}_m)^T.
\]

Furthermore, instead of updating the first two statistical moments, EnKF updates each realization with the equation of Kalman filter:

\[
m^{(i)}_u = m_i + K(d^{(i)}^* - g(m_i)), \quad i = 1, 2, \ldots, N_e,
\]

where the superscript “\( u \)” implies “updated.” The updated realizations form a new ensemble representing the posterior uncertainty of the parameters and serve as the prior ensemble for the next data assimilation loop.
Remark 1. The observation ensemble \( \mathbf{d}^* \) used in (14) is synthetically generated by perturbing the actually observed value according to \( \mathbf{R} \). If a single deterministic observation is used to update all of the realizations, the posterior covariance would be systematically underestimated [28]. However, generating perturbed \( \mathbf{d}^* \) would introduce additional sampling errors. Whitaker and Hamill [29] have proposed a method termed “ensemble square root filter” (EnSRF), which uses the deterministic observation but yields a consistent estimation of the posterior covariance. It updates the mean and perturbation of \( \mathbf{m} \) separately. While the mean is updated with the standard Kalman gain (5), the perturbation is updated with a modified gain:

\[
\mathbf{K} = C_{\mathbf{m}d}((\sqrt{C_{\mathbf{dd}}} + R)^{-1})^T(\sqrt{C_{\mathbf{dd}}} + R + \sqrt{R})^{-1}.
\]

In this paper, we adopt the EnSRF as the update scheme.

2.4. Probabilistic collocation Kalman filter

An intrinsic property of the Monte Carlo method is that its accuracy is guaranteed only when the ensemble size is sufficiently large, which could cause an enormous computational burden for using EnKF when the model \( g(\cdot) \) is costly to simulate. An effort to minimize the number of simulations for a given required accuracy has led to the development of another variant of the Kalman filter: PCKF. PCKF is similar to EnKF in every aspect except that it employs the PCE instead of an ensemble for uncertainty representation and propagation. By PCE, the input and output random vectors \( \mathbf{m} \) and \( \mathbf{d} \) are expressed as two truncated series:

\[
\mathbf{m}(\xi) \approx \sum_{i=0}^{Q} \mathbf{c}_i^m \psi_i(\xi), \quad (16)
\]

\[
\mathbf{d}(\xi) \approx \sum_{i=0}^{Q} \mathbf{c}_i^d \psi_i(\xi), \quad (17)
\]

where \( \xi = [\xi_1, \xi_2, \ldots, \xi_n]^T \) is a random vector comprising a set of independent random variables with given pdfs, such as normal distribution, uniform distribution, and so on. \( \psi_i(\xi) \) are the orthogonal polynomial basis with respect to \( \xi \): \( E(\psi_i(\xi)\psi_j(\xi)) = \delta_{ij} \), where \( \delta_{ij} \) is the Kronecker delta function. The first basis function \( \psi_0(\xi) = 1 \) represents the mean (deterministic) term, whereas the following terms are the perturbation (random) terms, which have zero means. Vectors \( \mathbf{c}_i \) are the deterministic coefficients. In the implementations of PCKF, PCE representation of input parameters (16) is set according to the prior distribution of \( \mathbf{m} \) (see Section 3.3.1), whereas (17) is obtained by solving a forward uncertainty propagation problem (discussed in detail in Sections 3.1 and 3.2). Note that the numbers of PCE basis functions used in (16) and (17) are not necessarily the same, but we can always extend (16) and (17) by adding the missing terms with zero coefficients so that they include the same basis.

With PCE expressions (16) and (17), the statistical moments needed for implementing the Kalman filter can be easily calculated from the coefficients:

\[
\mu_d = \sum_{i=0}^{Q} \mathbf{c}_i^d E(\psi_i(\xi)) = \mathbf{c}_0^d, \quad (18)
\]

\[
C_{dd} = E((\mathbf{d} - \mu_d)(\mathbf{d} - \mu_d)^T) = E\left(\left(\sum_{i=1}^{Q} \mathbf{c}_i^d \psi_i(\xi)\right)\left(\sum_{j=1}^{Q} \mathbf{c}_j^d \psi_j(\xi)\right)^T\right) = \sum_{i=1}^{Q} \mathbf{c}_i^d \mathbf{c}_i^{dT}, \quad (19)
\]

\[
C_{dm} = E((\mathbf{d} - \mu_d)(\mathbf{m} - \mu_m)^T) = E\left(\left(\sum_{i=1}^{Q} \mathbf{c}_i^d \psi_i(\xi)\right)\left(\sum_{j=1}^{Q} \mathbf{c}_j^m \psi_j(\xi)\right)^T\right) = \sum_{i=1}^{Q} \mathbf{c}_i^d \mathbf{c}_i^{mT}. \quad (20)
\]

Note that the orthogonal property of \( \psi_i(\xi) \) is used in Eqs. (18)–(20).

Similar to EnKF, the majority of the computational cost in PCKF is spent on propagating the uncertainty, i.e., building the PCE representation/calculation of the deterministic coefficients of output vector (17) given the PCE representation of the input vector (16). The required computational cost for this task depends on the number of PCE basis functions included in (17). Usually, this cost is much smaller than running the simulations for all realizations in EnKF if the dimensionality of the problem, i.e., the number of random variables \( \xi_i \), is relatively low. However, PCKF may lose its advantage over EnKF for relatively high-dimensional problems because the number of PCE basis functions grows very fast as the dimensionality increases. For these problems, we need a carefully designed method to select the basis functions that form the PCE approximation (17).
3. Adaptive ANOVA-based PCKF

A key step in PCKF is to determine the truncation of the PCE for the output vector, i.e., to select the PCE basis to form the approximation. Having more basis functions retained in the truncated PCE gives a better approximation but increases the computational cost. An ideal set of basis for PCE approximation should find a balance between accuracy and computational cost. Nevertheless, there were no clear guidelines for picking the PCE basis functions for PCKF. A common selection is to keep all of the polynomial chaos terms whose degree is smaller than or equal to a certain integer \( k \). This results in a total number of basis functions equal to \( (n+k)!/(n!k!) \), where \( n \) is the number of independent random variables \( \xi_j \). In most PCKF applications, setting \( k = 2 \) was sufficient to yield a good result. However, the total number of basis functions still grows very fast as \( n \) increases, which could make PCKF even more computationally demanding than EnKF for high-dimensional problems. Also, in previous applications of PCKF on sequential data assimilation problems, the PCE basis were selected in advance and remained fixed through the entire process.

Demonstrated in this paper, we develop an algorithm that improves the efficiency of PCKF by adaptively determining the appropriate PCE truncations for the specific problem under study. For sequential data assimilation problems, the algorithm also adjusts the PCE basis in different loops because the uncertainty changes as more and more information comes in.

We note that most of the terms in the PCE approximation for a high-dimensional function are those high-dimensional polynomials used to represent the coupling effects of multiple input parameters on the model output. However, in many practical problems, these effects are weak and negligible. This observation leads to the idea of functional ANOVA decomposition. Functional ANOVA is a dimensionality reduction technique widely used in forward UQ problems and can be conveniently combined with PCE approximation. It partitions a high-dimensional random function into a group of lower-dimensional components called “ANOVA components,” and the total variance of the random function is distributed among the ANOVA components. With the decomposition, the PCE terms needed for representing this group of low-dimensional random functions are much less than those needed for the original high-dimensional function. In addition, adaptive algorithms have been developed in previous studies [26,27] to select active ANOVA components for specific random functions, and this directly leads to an adaptive approach to construct PCE approximation in PCKF.

3.1. PCE approximation based on functional ANOVA decomposition

In this subsection, we introduce the functional ANOVA decomposition and explain how to construct the PCE approximation of the output vector \( \mathbf{d} \) given its ANOVA decomposition. Following the notations defined in Section 2, our model can be rewritten as a function with respect to the \( n \) random variables \( \mathbf{d} = g(\mathbf{m}(\xi)) = f(\xi) = f(\xi_1, \ldots, \xi_n) \). Functional ANOVA decomposes this function into a finite group of component functions—each of which takes a subset of \( \{\xi_1, \ldots, \xi_n\} \) as argument:

\[
f(\xi) = f_0 + \sum_{1 \leq j_1 \leq n} f_{j_1}(\xi_{j_1}) + \sum_{1 \leq j_1 < j_2 \leq n} f_{j_1,j_2}(\xi_{j_1}, \xi_{j_2}) + \cdots + f_{1,2,\ldots,n}(\xi_1, \xi_2, \ldots, \xi_n), \quad \xi \in \mathbb{R}^n,
\]

where the function \( f_{j_1,j_2,\ldots,j_t} \) with \( t \) argument variables \( \xi_{j_1}, \ldots, \xi_{j_t} \) is called a “\( t \)th-order ANOVA component.” Each component is computed by integration:

\[
f_T(\xi_T) = \int_{\mathbb{R}^{n-t}} f(\xi) \, d\mu(\xi|_S) - \sum_{S \subset T} f_S(\xi_S),
\]

where \( T = \{j_1, \ldots, j_t\}, \xi_T = (\xi_{j_1}, \ldots, \xi_{j_t}) \), \( S \) indicates every strict subset of \( T \), and \( C \) is the compliment of \( T \). Note that the computation of \( f_T \) requires first computing the lower-order components \( f_S \). So, we first compute the 0th-order component:

\[
f_0 = \int_{\mathbb{R}^n} f(\xi) \, d\mu(\xi).
\]

Then, the following 1st- and higher-order component functions are computed with (22) in a recursive manner.

Different choices of integration measure \( d\mu(\xi) \) result in different ANOVA decompositions. If \( d\mu(\xi) \) is chosen to be the same as the probability measure of \( \xi \), we have the following property:

\[
\sigma^2(f) = \sum_{1 \leq j_1 \leq n} \sigma^2(f_{j_1}) + \sum_{1 \leq j_1 < j_2 \leq n} \sigma^2(f_{j_1,j_2}) + \cdots + \sigma^2(f_{1,2,\ldots,n}),
\]

which means the total variance of \( f \) is exactly equal to the sum of the variances of all the ANOVA components. However, in practice, the integration with respect to the probability measure generally is difficult to compute. Instead, the Dirac measure is used as a convenient alternative: \( d\mu(\xi) = \delta(\xi - \mathbf{c}) \, d\xi \), where \( \mathbf{c} = (c_1, \ldots, c_n) \in \mathbb{R}^n \) is a constant point, usually chosen to be the mean of \( \xi \), and called the “anchor point.” The corresponding ANOVA decomposition is termed “anchored-ANOVA.” In anchored-ANOVA, (22) becomes:

\[
f_T(\xi_T) = f(\xi_T, \mathbf{c}_C) - \sum_{S \subset T} f_S(\xi_S).
\]
The computation of an anchored-ANOVA component requires only evaluating the model at corresponding points. For example, the first few components of anchored-ANOVA decomposition are listed as follows:

\[ f_0 = f(\mathbf{c}), \]
\[ f_1(\xi_1) = f(\xi_1, c_2, \ldots, c_n) - f_0, \]
\[ f_2(\xi_2) = f(c_1, \xi_2, c_3, \ldots, c_n) - f_0, \]
\[ f_{1,2}(\xi_1, \xi_2) = f(\xi_1, \xi_2, c_3, \ldots, c_n) - f(\xi_1) - f_2(\xi_2) - f_0. \]

In many practical problems, the variance of the function \( f \) distributes mainly on low-order ANOVA components. In other words, the coupling effect between multiple dimensions usually is small and negligible. This allows us to truncate the high-order ANOVA components in (21) and obtain an approximation of the original high-dimensional function \( f \) with its low-dimensional components:

\[ f(\xi) \approx \sum_{i=1}^{N} f_i(\xi_{T_i}). \]

With the truncated ANOVA decomposition (30), we construct the PCE for the random function by expanding each component into PCE. For a component function \( f_T(\xi_T) \), we form its polynomial chaos basis from the tensor product of the one-dimensional basis:

\[ f_{j_1,j_2,\ldots,j_v}(\xi_{j_1}, \ldots, \xi_{j_v}) \approx \sum_{i_1=0}^{v} \cdots \sum_{i_v=0}^{v} C_{i_1\cdots i_v} \phi_{i_1}(\xi_{j_1}) \cdots \phi_{i_v}(\xi_{j_v}), \]

where \( \phi_{i}(\xi) \) is the \( i \)th-degree one-dimensional polynomial chaos with respect to the random variable \( \xi \), while \( C_{i_1\cdots i_v} \) are the deterministic coefficients. For example, the first three one-dimensional polynomial chaos basis functions (including the deterministic one) for standard normal random variable are: \( \phi_0(\xi) = 1 \), \( \phi_1(\xi) = \xi \), and \( \phi_2(\xi) = (\xi^2 - 1)/\sqrt{2} \) (known as Hermite polynomials). The 1st-order ANOVA component \( f_1 \) with respect to a single standard normal random variable is expanded as follows:

\[ f_1(\xi_1) \approx \sum_{i_1=0}^{2} C_{i_1} \phi_{i_1}(\xi_1) = C_0 + C_1 \xi_1 + C_2 (\xi_1^2 - 1)/\sqrt{2}, \]

and the 2nd-order ANOVA component \( f_{1,2} \) with respect to two standard normal random variables is expanded as follows:

\[ f_{1,2}(\xi_1, \xi_2) \approx \sum_{i_1=0}^{2} \sum_{i_2=0}^{2} C_{i_1i_2} \phi_{i_1}(\xi_1) \phi_{i_2}(\xi_2) = C_{00} + C_{01} \xi_2 + C_{02} (\xi_2^2 - 1)/\sqrt{2} + C_{10} \xi_1 + C_{11} \xi_1 \xi_2 + C_{12} \xi_1 (\xi_2^2 - 1)/\sqrt{2} + C_{20} (\xi_1^2 - 1)/\sqrt{2} + C_{21} \xi_2 (\xi_1^2 - 1)/\sqrt{2} + C_{22} (\xi_2^2 - 1)/2. \]

To determine the PCE coefficients, we use PCM [11]. In PCM, we evaluate the component \( f_{j_1,j_2,\ldots,j_v}(\xi_{j_1}, \ldots, \xi_{j_v}) \) following the anchored-ANOVA procedures at a set of collocation points. The number of collocation points is chosen to be the same as the number of polynomial basis functions. By requiring the PCE approximation to be equal to the true function at these points (i.e., interpolation), one solves for the coefficients. PCM is a non-intrusive approach where we may treat the function \( f \) as a black box. The accuracy of PCM depends on the positions of collocation points. Commonly, the collocation points are positioned at the quadrature points (e.g., Gauss points), resulting in accurate computations of the statistical moments of \( f(\xi) \). Specifically, for a one-dimensional PCE with the highest polynomial degree of \( d \), the collocation points/Gauss quadrature points are the roots of the \( (d + 1) \)st polynomial chaos. For a multidimensional PCE, the collocation points are formed from the tensor product of the one-dimensional collocation points. For instance, the PCE representation of a two-dimensional component, such as (33), may be built by interpolation on a 3 by 3 tensor product of one-dimensional Gaussian quadrature points. After expanding each ANOVA component with PCE, we sum up all of the PCE terms to form the final PCE approximation for the original function (17).

With the PCE representation of an ANOVA component \( f_T \), we can easily calculate its variance \( \sigma^2(f_T) \) from the PCE coefficients. Note that the PCE representation for an ANOVA component built with PCM may include the PCE terms that, in fact, belong to lower-order ANOVA components. For example, take representation (33): \( C_{00} \) belongs to the 0th-order component \( f_0 \), and \( C_{10} \xi_1 \) belongs to the 1st-order component \( f_1 \). These terms are not necessarily zero. When the same PCE basis function appears in the PCE representations of different ANOVA components, the resulting variances of different components are not additive, e.g., \( \sigma^2(f_1 + f_{1,2}) \neq \sigma^2(f_1) + \sigma^2(f_{1,2}) \) when \( \xi_1 \) in (32) and \( \xi_1 \) in (33) are both non-zero. This is because the Dirac measure is used in anchored-ANOVA. To regain the additive property, we may rearrange and lump
the PCE terms according to their arguments. For example, we amend the PCE representation (33) for \( f_{1,2} \) by removing the term \( C_{101} \xi_1 \) and adding it to the PCE representation for \( f_1 \) (32). Other terms in (33), such as \( C_{00}, C_{01} \xi_2, C_{02} (\xi_2^2 - 1)/\sqrt{2}, \) and \( C_{20} (\xi_1^2 - 1)/\sqrt{2}, \) should be moved similarly.

The approach of forming collocation points and polynomial chaos basis using tensor product would suffer the so-called “curse of dimensionality” for high-dimensional functions because the tensor product yields a huge number of points and basis functions. Still, it works well in our algorithm if we have most high-order ANOVA components truncated and mainly compute the low-dimensional components. This truncation may be obtained for a specific function \( f \) based on some adaptive criterion, which will be discussed in detail in Section 3.2.

### 3.2. Adaptive selection of ANOVA components

As mentioned in Section 3.1, when computing the ANOVA decomposition for a specific function, we can use some criteria to decide which ANOVA components are computed and which are truncated. This results in the adaptive ANOVA decomposition.

**Remark 2.** In our context, “computing an ANOVA component” does not imply evaluating the component at every possible point with Eq. (25), but to give the PCE approximation (31) and estimate the statistical moments from the PCE coefficients.

The idea of the adaptive ANOVA is to identify the important dimensions, called “active dimensions,” and only compute the high-order components on these dimensions. In the first step of this process, we choose the 1st-order active dimensions \( i \in A_1 \) and compute their corresponding 1st-order ANOVA components:

\[
f(\xi) = f_0 + \sum_{i \in A_1} f_i(\xi_i). \tag{34}
\]

Note that this step is much easier than computing 2nd- and higher-order ANOVA components because the number of PCE basis functions, as well as the number of required model evaluations in this step, grows only linearly with the number of dimensions included in \( A_1 \). Usually, if we do not have any prior information to determine which dimensions can be safely neglected, we could afford to keep all dimensions 1st-order active. Next, we select the 2nd-order active dimensions by examining the statistical moments of the 1st-order ANOVA components. For example, we may check how the total variance of (34) is distributed to each individual dimension \( \sigma^2(f) = \sum_{i \in A_1} \sigma^2(f_i) \). The dimensions with large portions of variance are identified as important and chosen to be 2nd-order active. For each pair of the 2nd-order active dimensions \( (i, j) \in A_2 \), we compute the corresponding 2nd-order ANOVA component \( f_{i,j}(\xi_i, \xi_j) \). Similarly, we can adaptively select the 3rd- and higher-order active dimensions by examining the statistical moments of lower-order ANOVA components. However, when using ANOVA-based PKF, our numerical results suggest computing the ANOVA components no further than the 2nd-order. This is not simply because the computation of 3rd- and higher-order components is much more expensive. More importantly, the Kalman filter and its variants assume that the nonlinearity of the studied model is mild. Thus, using only the first two statistical moments is sufficient to infer the input from the observations of the output. For these models, 2nd-order ANOVA usually is capable of providing a good approximation. For those models where nonlinearity is strong—such that keeping higher-order ANOVA components is necessary—other inversion approaches, rather than Kalman filters, should be considered. This study focuses on the adaptive criteria for choosing the 2nd-order active dimensions. Three different criteria are discussed here and will be tested in Section 4 with illustrative examples.

**Remark 3.** In the following discussions on adaptive criteria, we consider the situation when there is only one observable output variable. For the general case when the model has multiple observable output variables, we apply the adaptive criteria to the ANOVA decomposition for each individual output variable. Because different output variables may be sensitive to different input dimensions, the selected ANOVA components and resulting PCE basis functions for different output variables generally are not the same. Finally, the entire output vector is represented with the union of all selected basis functions.

**Criterion 1.** Yang et al. [27] showed an adaptive criterion that assesses the “importance” of each dimension based on the variances of the 1st-order ANOVA components \( \sigma^2(f_i) \). This requires sorting the dimensions according to \( \sigma^2(f_i) \) in descending order and keeping the first \( N_2 \) in the set of 2nd-order active dimensions \( A_2 \) such that:

\[
\sum_{j=1}^{N_2} \sigma^2(f_j) \geq (1 - \alpha) \sum_{i \in A_1} \sigma^2(f_i), \tag{35}
\]

where \( \alpha \) is a proportional constant between 0 and 1. Smaller \( \alpha \) retains more dimensions in \( A_2 \). Then, we compute the 2nd-order ANOVA components for each pair \( (i, j) \in A_2 \).

This criterion determines the relative importance between different dimensions, but it does not compare the relative importance between 1st- and 2nd-order ANOVA components. For example, consider two functions \( f \) and \( g \), which have approximately the same 1st-order ANOVA components. Their difference is that the variance distributed on the 2nd-order
ANOVA components of \( f \) is comparable with its 1st-order components, whereas the 2nd-order components of \( g \) are negligible. Ideally, an adaptive criterion should compute more 2nd-order ANOVA components for \( f \) than for \( g \). However, Criterion 1 cannot distinguish these two different situations because it does not check the variances of the 2nd-order ANOVA components. To solve this issue, we propose a more sophisticated adaptive criterion.

**Criterion 2.** We propose a new adaptive criterion that compares the relative importance not only between different dimensions but also between the 1st- and 2nd-order ANOVA components. This criterion is based on the estimation of the variance distributed on the 2nd-order components that are excluded from the ANOVA decomposition. To obtain this estimation we build a regression model that can be used to approximately predict \( \sigma^2(f_{i,j}) \) for a dimension pair \((i,j)\) before actually computing \( f_{i,j} \). Specifically, we assume that \( \sigma^2(f_{i,j}) \) is related to the variances of 1st-order components \( \sigma^2(f_i) \) and \( \sigma^2(f_j) \) through the following equation

\[
\sigma^2(f_{i,j}) = \beta \sigma^2(f_i)\sigma^2(f_j) + e_{i,j},
\]

where \( \beta \) is a ratio coefficient and \( e_{i,j} \) is an error term. Eq. (36) is based on two simple assumptions which are valid for many models: (i) a dimension pair \((i,j)\) is likely to have larger coupling effect (measured by \( \sigma^2(f_{i,j}) \)) comparing with other pairs if dimensions \( i \) and \( j \) both have relatively large individual effect (measured by \( \sigma^2(f_i) \) and \( \sigma^2(f_j) \)); (ii) if both dimensions \( i \) and \( j \) have relatively small individual effect, the corresponding dimension pair \((i,j)\) is likely to have relatively small coupling effect. Based on the second assumption, we prefer Eq. (36) to an additive model such as \( \sigma^2(f_{i,j}) = \beta \sigma^2(f_i) + \sigma^2(f_j) + e_{i,j} \), or a regression model that includes a constant term: \( \sigma^2(f_{i,j}) = \alpha + \beta \sigma^2(f_i)\sigma^2(f_j) + e_{i,j} \). We point out that the regression model (36) is a simplified relation that is intended to capture only the big trend and may not accurately fit every dimension pair. Indeed it is impossible to have a general equation that accurately predicts the value of \( \sigma^2(f_{i,j}) \) for every function \( f \) from only the statistical moments of the 1st-order ANOVA components. If more properties of the function \( f \) are known in a specific problem, we may replace (36) with a better model.

With Eq. (36) we develop the new adaptive criterion as follows. After computing all 1st-order active ANOVA components, we sort all of the dimension pairs \((i,j)\) \(i,j \in A_1\) into a list according to \( \sigma^2(f_i)\sigma^2(f_j) \) in descending order. We then compute the 2nd-order ANOVA components, starting with the important pairs, which are the leading pairs in this list. After computing each 2nd-order component, we give an assessment of the error in calculating the total variance of \( f \), induced by leaving the rest of the pairs out of the ANOVA decomposition:

\[
\text{Err}^{(P)}_{\sigma_f^2} = \sum_{(i,j) \notin P} \sigma^2(f_{i,j}) \approx \beta \sum_{(i,j) \notin P} \sigma^2(f_i)\sigma^2(f_j),
\]

where \( \beta = \frac{\sum_{(i,j) \notin P} \sigma^2(f_{i,j})}{\sum_{(i,j) \in P} \sigma^2(f_i)\sigma^2(f_j)} \) and \( P \) is the set of all pairs whose 2nd-order ANOVA components have been computed. This error estimation is based on assumption (36), and the ratio coefficient \( \beta \) is estimated from the previously computed pairs. As \( P \) expands and more 2nd-order ANOVA components are computed, this error diminishes. Computing stops once this error is small enough when compared to the total variance of all the 1st- and 2nd-order ANOVA components that have been computed:

\[
\text{Err}^{(P)}_{\sigma_f^2} < \alpha \left( \sum_{i \in A_1} \sigma^2(f_i) + \sum_{(i,j) \in P} \sigma^2(f_{i,j}) + R \right).
\]
Note that larger $R$ makes the stop criterion (39) easier to achieve and thus requires a smaller number of 2nd-order ANOVA components to be calculated. Intuitively, this implies that the criterion considers the quality of the observation when running PCKF. A large error variance $R$ indicates the observed data provide little information for inferring the input parameters. Accordingly, we do not need to spend much computational effort on assimilating these data.

3.3. Additional discussions on implementation of adaptive ANOVA-based PCKF

In this section, we discuss the pre- and post-procedures in the implementation of the adaptive ANOVA-based PCKF before summarizing this algorithm in Section 3.4.

3.3.1. Representation of prior parametric uncertainty

The first step in PCKF is to provide a PCE representation of the input parameter vector $\mathbf{m}$ that is consistent with its prior statistical moments. In our algorithm, we use principal component analysis (PCA), also known as Karhunen–Loève expansion (KLE). A random parameter vector $\mathbf{m}$ with prior mean $\mu_m$ and covariance matrix $\mathbf{C}_{mm}$ can be expressed in the following form:

$$\mathbf{m} = \mu_m + \sum_{i=1}^n \sqrt{\lambda_i} \mathbf{m}_i \xi_i, \tag{40}$$

in which $\lambda_i$ and $\mathbf{m}_i$ are the $i$th eigenvalue and eigenvector of $\mathbf{C}_{mm}$, respectively. $\xi_i$ are a set of uncorrelated random variables with zero mean and unit variance. Eq. (40) is equivalent to the PCE form

$$\mathbf{m} (\xi) = \mathbf{c}_0 + \sum_{i=1}^n \mathbf{c}_i \psi_i (\xi), \tag{41}$$

where $\mathbf{c}_0 = \mu_m$, $\mathbf{c}_i = \sqrt{\lambda_i} \mathbf{m}_i$, $\psi_i (\xi) = \xi_i$, $i = 1, \ldots, n$.

PCA not only decouples the correlation—represents correlated parameters with uncorrelated random variables—but also redistributes the variability of all input parameters mainly to the first few dimensions (i.e., the dimensions with largest eigenvalues). By truncating the dimensions with negligible eigenvalues, we may reduce the dimensionality of the problem to a number less than the parameters.

Remark 4. Generally, the pdf of $\xi_i$ depends on the distribution of $\mathbf{m}$ and can be estimated numerically. In our algorithm, we simply set $\xi_i$ to be independent standard normal. It is a reasonable approximation as, in a PCKF problem, the prior distribution should be approximately normal. Also, the first two statistical moments of $\mathbf{m}$, which are all of the information considered by Kalman filter, still are accurately represented.

Remark 5. A special case arises when input parameters are uncorrelated. In such a situation, the PCA does not help to reduce the dimensionality. In fact, PCA (40) is equivalent to representing each individual parameter with a random variable.

3.3.2. Update scheme of PCKF

After obtaining the PCE representation of the output vector and computing the needed statistical moments from the PCE coefficients, we update the PCE representation of the parameter vector $\mathbf{m}$ given the observation using the Kalman filter. In this paper, we use an updated scheme of EnSRF, which does not require an artificially perturbed observation. The mean is updated with Kalman gain:

$$\mathbf{c}_{m}^{\mu u} = \mathbf{c}_{m}^{\mu} + \mathbf{K} (\mathbf{d}^* - \mathbf{c}_{d}^{\mu}). \tag{42}$$

The perturbation (all PCE terms except the first) is updated with the modified gain defined in (15), which yields a consistent estimation of the posterior covariance:

$$\sum_{j=1}^Q \mathbf{c}_{j}^{m u} \mathbf{p}_j = \sum_{j=1}^Q \mathbf{c}_{j}^{m} \mathbf{p}_j + \mathbf{K} \left( \sum_{j=1}^Q \mathbf{c}_{j}^{d} \mathbf{p}_j - \sum_{j=1}^Q \mathbf{c}_{j}^{d} \mathbf{p}_j \right). \tag{43}$$

Note that when using the EnSRF, all of the random terms of the observation are zero ($\mathbf{c}_{j}^{d} = \mathbf{0}$, $j = 1, \ldots, Q$) because perturbation is not needed. By multiplying Eq. (43) with each basis function and taking the expectation, we have the update equations for the coefficients other than the mean (first) term:

$$\mathbf{c}_{j}^{m u} = \mathbf{c}_{j}^{m} - \tilde{\mathbf{K}} \mathbf{c}_{j}^{d}, \quad j = 1, \ldots, Q. \tag{44}$$

In our algorithm, we update the PCE of parameters by directly updating its coefficients, which follows [18,20]. Another method is seen in [19], where the PCE approximation is used as a surrogate to generate a large ensemble of realizations and the posterior PCE is obtained from the updated ensemble.
The posterior mean and covariance of the parameters can be calculated from the updated PCE representation:

\[
\mu_{\mathbf{m|d}} = \mathbf{c}_0^{\mu_u},
\]

\[
\mathbf{C}_{\mathbf{mm|d}} = \sum_{j=1}^{Q} (\mathbf{c}_j^{\mu_u})(\mathbf{c}_j^{\mu_u})^T.
\]

### 3.4. Summary of adaptive ANOVA-based PCKF algorithm

The algorithm of adaptive ANOVA-based PCKF is summarized herein. The following procedures consist of a complete data assimilation loop:

- **Step 1:** Represent the parameter vector \( \mathbf{m} \) with PCE (16) according to the prior statistical moments. For a correlated input parameter, PCA (40) may be used to reduce the input dimensionality.
- **Step 2:** Compute the PCE representation (17) for the output variables \( \mathbf{d} \) with adaptive ANOVA. This includes:
  1. Compute the 0th-order ANOVA components by evaluating the model output at the anchored point (26).
  2. Compute the PCE approximations for the 1st-order ANOVA component with respect to each input dimension using the PCM method.
  3. Select the dimension pairs with an adaptive criterion and compute the corresponding 2nd-order ANOVA components using the PCM method.
  4. Rearrange all PCE terms to form the final PCE representation (17).
- **Step 3:** Estimate the required statistical moments—(18), (19), and (20)—for computing the Kalman gain (5) and the modified gain (15).
- **Step 4:** Update the PCE representation of \( \mathbf{m} \) using EnSRF. Mean (i.e., the first PCE term) is updated with standard Kalman gain (42). The perturbation (i.e., all PCE terms except the first) is updated with the modified gain (44).
- **Step 5:** Compute the posterior mean and covariance of \( \mathbf{m} \) from the updated PCE (45) and (46).

### 4. Illustrative examples

#### 4.1. Problem 1: Inversion of a differential equation with random input

Consider the following differential equation, which describes a variety of physical problems (e.g., Fourier’s law in the field of heat conduction, Fick’s law in diffusion theory, and Darcy’s law for flow in porous media):

\[
\begin{align*}
\frac{d}{dx} \left[ a(x, \xi) \frac{d}{dx} u(x, \xi) \right] &= 0, \quad 0 \leq x \leq 1, \\
u(0) &= 0, \quad u(1) = 1.
\end{align*}
\]

\( a(x, \xi) \) is the model parameter, depending on location \( x \), and \( u(x, \xi) \) is the model output. We assume that \( a(x, \xi) \) is subject to a prior uncertainty and is expressed by a random function with stationary log-normal distribution. The prior mean and covariance of \( A = \log(a) \) is \( \mathbb{E}(A(x)) = 0 \) and \( \text{Cov}(A(x_1), A(x_2)) = \sigma_A^2 \exp(-25(x_1 - x_2)^2) \), respectively. \( \sigma_A^2 \) is the prior variance of \( A \). A realization of \( A(x, \xi) \) is sampled from the prior distribution and serves as the reference parameter value. The reference output is solved from the differential equation with the reference parameter. Observations of the output are made on the reference output at 10 locations \( (x_1 = 0.05, x_2 = 0.15, \ldots, x_{10} = 0.95) \) with the observation errors assumed to be independent normal random variables \( N(0, R) \). The objective is to estimate model parameter \( A(x) \) from the observed data. The data sequence \( u_i = u(x_i) \) is assimilated with 10 Kalman filter loops recursively. We have tested our adaptive algorithm in three case studies with different prior variances and observation errors (Table 1).

#### 4.1.1. Demonstration of adaptive ANOVA-based PCKF

We first demonstrate the adaptive ANOVA-based PCKF algorithm using the initial case study. The PCE representation for parameter \( A \) is initialized using PCA with 10 random variables, which makes this a 10-dimensional inverse problem. The model output \( u \) is approximated using functional ANOVA decomposition. In Fig. 1, we plot the variances of the 10 1st-order ANOVA components of \( u_{x=0.05} \). It is apparent that some dimensions have larger contributions to the total output variance (the 3rd, 4th, and 5th dimensions), whereas other dimensions do not affect the output very much (the 8th, 9th, and 10th dimensions).
In each Kalman filter loop, the ANOVA components are selected according to the adaptive criteria described in Section 3.2 ($\alpha = 0.05$), and each ANOVA component is expanded into PCE (31) using the tensor product of the first three one-dimensional Hermite polynomial basis functions.

Fig. 2 shows the number of total PCE basis functions, which is the same as the number of required model simulations kept in different data assimilation loops. We can see that all three criteria are capable of adaptively adjusting the PCE basis in different loops. Overall, we recognize a decreasing trend in the number of PCE basis functions as the data assimilation process goes loop by loop, which means that a smaller number of PCE basis functions are sufficient to represent the remaining uncertainty.

In each loop, we calculate the means and standard deviations of the model output $u(x, \xi)$ and input $A(x, \xi)$ from their PCEs and plot them in Figs. 3 and 4, respectively. Initially, the mean of $A(x, \xi)$ is zero with standard deviation equal to $\sqrt{2}$ throughout the domain, which is consistent with our assumption of the prior distribution. Also, we see significant uncertainty associated with the initial prediction of the model output $u(x, \xi)$—except for the left and right ends $x = 0$ and $x = 1$, where they are constrained by the boundary conditions. As observations are assimilated, we see a trend of uncertainty reduction in the estimation of both parameter $A(x, \xi)$ and output $u(x, \xi)$. For the latter, where direct observations are made, our estimation converges to the reference case. For the parameter, we obtain a much more accurate posterior estimation with reduced uncertainty compared to the prior one. However, a non-negligible uncertainty still exists, even after 10 data assimilation loops. This shows the ill-posedness of this inverse problem and implies that the reference parameter cannot be exactly identified by observing $u(x, \xi)$ at the 10 locations.
4.1.2. Efficiency comparison with EnKF and non-adaptive PCKF

To check the efficiency of our algorithm, we solve the same inverse modeling problem using EnKF methods with different ensemble sizes. Furthermore, we conduct two more non-adaptive PCKF algorithms. All 10 dimensions are expanded to only the 1st-order ANOVA in the first non-adaptive algorithm and to the full 2nd-order ANOVA in the second. The estimated posterior means and standard deviations of $A(x, \xi)$ by different methods are compared in Fig. 5. The result given by EnKF with a large number of realizations (10000) is set as the benchmark. From the comparison, we see that the result given by PCKF based on 2nd-order ANOVA is able to aptly capture the posterior uncertainty and is very close to the benchmark, whereas the 1st-order ANOVA-based PCKF gives a result that apparently differs from the benchmark, as well as the reference. The adaptive ANOVA-based PCKFs offer results that also are very close to the benchmark but require much less computational cost compared with the 2nd-order ANOVA-based PCKF. Moreover, the results from the adaptive ANOVA-based PCKFs are more reliable than the EnKFs with similar computational costs. To accurately check the performances of different methods, we compute a single index that simultaneously measures the accuracies of both estimated mean $\mu_A(x)$ and variance $\sigma_A^2(x)$ of the model parameter:

$$H^2(\mu_A(x), \sigma_A^2(x)) = 1 - \frac{2\sigma_A(x)\sigma_{Ab}(x)}{\sigma_A^2(x) + \sigma_{Ab}^2(x)} \exp \left[ -\frac{1}{4} \frac{(\mu_A(x) - \mu_{Ab}(x))^2}{\sigma_A^2(x) + \sigma_{Ab}^2(x)} \right],$$

(48)

where $\mu_{Ab}(x)$ and variance $\sigma_{Ab}^2(x)$ are the benchmark. Eq. (48) is close to 1 if the estimation and benchmark have little overlap and equals 0 if the estimation is identical to the benchmark. Index $H^2$ also is known as the squared Hellinger distance between two normal distributions. We plot $H^2$ (averaged in $x \in [0, 1]$) of the estimated posterior means and variances given by different methods, including adaptive and non-adaptive PCKFs and EnKFs with different ensemble sizes, against their computational cost on a log–log plot (Fig. 6). From the plot, we recognize the convergence of EnKF toward the benchmark as the ensemble size increases. Because the result of EnKF depends on the sampling of initial realizations and has some randomness, we run EnKF 10 times for every tested ensemble size with different initial realizations and plot the maximum, average, and minimum $H^2$ of the 10 experiments. For this specific problem, the 2nd-order ANOVA-based PCKF is more accurate than the EnKF with similar computational cost, while the 1st-order ANOVA-based PCKF is worse than most of the EnKF experiments. The adaptive ANOVA-based PCKFs are more efficient than the non-adaptive PCKFs because they...
achieved relatively high accuracy with relatively small computational cost. Also, the adaptive PCKFs outperform the EnKFs with similar computational cost.

**Remark 6.** When comparing the performances of the different methods, particular attention should be noted in the differences between:

(a) The reference parameter.
(b) The posterior distribution of the parameter given by Bayesian theorem.
(c) The estimation of the parameter given by Kalman filter.

Point (b) is the solution we seek in an inverse problem. It contains all of the possible realizations, including the reference (a), that are consistent with the model and available observations. We cannot identify the reference (a) unless more information is given. Point (c) is an approximation to (b) that may be obtained with less computational cost (e.g., compared with MCMC). The difference between (b) and (c) results from applying the Kalman filter to a nonlinear model, and this error cannot be eliminated by enlarging either the ensemble size in EnKF or the PCE terms in PCKF. Hence, rather than (a) or (b), we choose the EnKF with large ensemble size as the benchmark.

### 4.1.3. Comparison between adaptive criteria

To further study and compare the different adaptive criteria, we test our algorithm on Case 2 and Case 3. In Case 2, we set a smaller prior parameter variance. Because the parameter varies in a smaller range, this makes input–output relation less nonlinear and results in smaller 2nd-order ANOVA components. This case is designed to study how the adaptive criteria respond to the change in the problem’s nonlinearity. In Case 3, we assume the observations are less accurate by setting larger observation errors. The purpose is to check how the adaptive criteria react to the change in observation quality. Table 2 shows the average number of PCE basis functions selected in the 10 loops by the three different criteria. From Table 2, we see that Criterion 3 responds to the changes in Cases 2 and 3 by retaining a smaller number of PCE basis functions compared with Case 1, which implies that fewer 2nd-order components are computed in the ANOVA decomposition. Criterion 2 reduces the number of PCE basis functions for only Case 2 but is not sensitive to the change in Case 3. Criterion 1 retains...
Fig. 5. Comparison of the parameter estimations given by EnKFs, non-adaptive PCKFs, and adaptive ANOVA-based PCKFs.

Fig. 6. Efficiency comparison between EnKFs, non-adaptive PCKFs, and adaptive ANOVA-based PCKFs using $H^2$ index (Case 1, Problem 1).

almost the same number of PCE basis functions in all three cases. This result is consistent with our expectations. Criterion 1 selects 2nd-order ANOVA components by distinguishing the important dimensions among all dimensions using 1st-order ANOVA, but it considers neither the nonlinearity of the problem nor the quality of the observations. Criterion 2, which is an improvement upon Criterion 1, examines model nonlinearity. Finally, Criterion 3 improves Criterion 2 by considering the quality of the observations.

To check the accuracies of the adaptive ANOVA-based PCKF using different criteria, we once again solve the last two cases by different KF approaches and plot the index $H^2$ of their results against the corresponding computational cost for
Table 2
Number of PCE basis functions selected by three criteria (average number of 10 data assimilation loops).

<table>
<thead>
<tr>
<th></th>
<th>Criterion 1</th>
<th>Criterion 2</th>
<th>Criterion 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>64</td>
<td>39</td>
<td>38</td>
</tr>
<tr>
<td>Case 2</td>
<td>63</td>
<td>27</td>
<td>26</td>
</tr>
<tr>
<td>Case 3</td>
<td>69</td>
<td>38</td>
<td>26</td>
</tr>
</tbody>
</table>

Fig. 7. Efficiency comparison between EnKFs, non-adaptive PCKFs, and adaptive ANOVA-based PCKFs using $H^2$ index (Case 2, Problem 1).

Fig. 8. Efficiency comparison between EnKFs, non-adaptive PCKFs, and adaptive ANOVA-based PCKFs using $H^2$ index (Case 3, Problem 1).

Case 2 and Case 3 (Fig. 7 and Fig. 8, respectively). Apparently, the accuracies of the three adaptive ANOVA-based PCKF methods are at almost the same level. Considering the computational cost, we conclude that adaptive Criterion 3 offers the most effective selection of the PCE basis. In fact, in the comparison with EnKFs, the 1st-order ANOVA-based PCKF performs much better for Cases 2 and 3 than for Case 1. This also shows that a large number of 2nd-order components are not necessary for the last two cases, and the response of reducing PCE basis functions automatically made by Criterion 3 (also by Criterion 2 in Case 2) is appropriate.

4.2. Problem 2: History matching of a reservoir model

To further test and demonstrate the adaptive ANOVA-based PCKF, we apply it to a history matching problem, which is commonly practiced in reservoir engineering. We calibrate a petroleum reservoir model to the recorded well production data. Consider a synthetic, two-dimensional, square-shaped reservoir bounded by no-flow boundaries (Fig. 9(a)). The 1200-ft by 1200-ft domain is partitioned into 40 by 40 grid blocks, and the thickness of every grid block is 30 ft. We simulate the physical process of two-phase flow (oil and water) in the reservoir. Initially, the reservoir is saturated with 75% oil and
25% water (porosity is 0.2). Four wells, two producers, and two water injectors are located at the four corners. The oil is recovered from the reservoir using the waterflooding method: two injectors are used to push oil toward the producers by injecting water, both at the constant rate of 300 stock tank barrels (STB) per day and the two production wells both produce fluids while maintaining a constant bottom hole pressure (BHP) of 3000 psi. This production process is simulated with the reservoir simulator ECLIPSE. In this process, the permeability $k(x, y)$ of the reservoir medium has significant influence on the fluid flow and determines how much of the total oil stored in the reservoir may be recovered. However, due to the heterogeneity of the permeability distribution and difficulty of precise measurement, we usually do not have complete and accurate knowledge of $k(x, y)$. In this example, $k(x, y)$ is assumed to be subject to a prior uncertainty. $Y = \log(k)$ is a stationary Gaussian random function, and its prior mean and covariance are:

$$E(Y(x, y)) = \mu_Y,$$

$$\text{Cov}(Y(x_1, y_1), Y(x_2, y_2)) = \sigma_Y^2 \exp \left[ -\left( \frac{x_1 - x_2}{\lambda_x} \right)^2 - \left( \frac{y_1 - y_2}{\lambda_y} \right)^2 \right].$$

where the mean of log permeability is $\mu_Y = 2$ (this mean is calculated when $k$ is measured in the unit of millidarcy); variance $\sigma_Y^2 = 1$; and the correlation factors in $x$ and $y$ directions, $\lambda_x$ and $\lambda_y$, are both 400 ft. A realization of $Y$ is sampled
Fig. 10. Simulated production history with error bars (before model calibration).

from the prior distribution and is assumed to be the true permeability field (Fig. 9(b)). Given the permeability field, we can predict the water saturation in different stages during the waterflooding process (Figs. 9(c) and 9(d)).

In this inverse problem, our objective is to estimate $Y$ from the recorded production data in different stages (also known as the production history). The available data include the BHP recorded at the two injection wells, as well as the water production rate (WPR) and the oil production rate (OPR) at the two production wells. The data are generated from the true permeability field and are collected every 40 days at a total of 15 time steps. The data observed at different time steps are assimilated sequentially in Kalman filter loops. The errors associated with observations are assumed to be independent normal with zero means and standard deviations being equal to 5% of the observed values.

We solve this problem with adaptive ANOVA-based PCKF using Criterion 3. The PCE representation for parameter $Y$ is initialized using PCA with 20 random variables, which results in a 20-dimensional inverse problem. Fig. 10 and Fig. 11 show the simulation results with error bars of the well production history before and after data assimilations, respectively. We observe that, due to the uncertain parameters, the simulation results in Fig. 10 do not match the observed production history and are subject to significant uncertainties. After assimilating all of the observations, we note the simulations match the production history well and the uncertainties are greatly reduced (Fig. 11). The estimated mean and posterior standard deviation of the log permeability field $Y(x, y)$ by the adaptive ANOVA-based PCKF is plotted in Fig. 12. Compared with Fig. 9(b), we see that the estimated mean is not identical to the true permeability but captures the big trend, i.e., high permeability in the upper right area and low permeability in the lower left area. From Fig. 12(b), it is apparent that the parametric uncertainty is reduced mainly in the regions near the wells, especially the production wells (note that the prior standard deviation is uniformly equal to 1). In the regions far away from the wells, the uncertainty is less reduced, indicating the production data are less sensitive to the permeability at those locations. To accurately estimate the permeability of the entire domain, other measurements would be necessary.

In Fig. 13, we plot the number of adaptively selected PCE basis functions in different data assimilation loops. The most visible trend in this plot is that the adaptive criterion increased the number of PCE basis functions from Day 180 until around Day 380. In fact, this is the early period when the production wells started producing water, known as “breakthrough.” During breakthrough, there are some radical changes in the production history, such as a sudden increase in WPR and a decrease in OPR. Having more PCE terms helps to accurately capture the model response during this period.

Finally, we compare the efficiency of the adaptive ANOVA-based PCKF with non-adaptive PCKFs and EnKFs. Again, we use the EnKF with large ensemble size (2000) as the benchmark and calculate the $H^2$ indexes of the results given by different approaches to measure their accuracy. Fig. 14 depicts the “accuracy versus computational cost” chart. Notably, the non-adaptive PCKFs are almost at the same accuracy levels with the EnKFs with comparable computational cost, but the adaptive ANOVA-based PCKF is able to achieve higher accuracy with relatively low cost, making it the most efficient option among all of the methods tested.
5. Conclusions and discussions

In this paper, we developed an adaptive ANOVA-based PCKF algorithm to solve nonlinear inverse modeling problems. Our research contributes to previous studies from the following aspects, and these improvements make the adaptive ANOVA-based PCKF a more flexible and efficient tool for inverse modeling problems when compared with classic PCKF.
(1) Our algorithm adaptively selects the PCE basis functions to represent uncertainties for different models and in different Kalman filter loops of sequential data assimilation problems, while, in classic PCKF methods, the PCE basis are pre-set by users and remain fixed in all loops.

(2) We extend the adaptive functional ANOVA algorithm, which mainly is used in solving forward uncertainty propagation problems, to solving inverse modeling problems.

(3) We propose two new criteria (Criteria 2 and 3 in Section 3.2) for adaptive functional ANOVA decomposition, which are more efficient and especially suited for using PCKF. Criterion 2 is not only able to detect the important input dimensions among all of them, but it also compares the relative importance between the low- and high-order components. Criterion 3 is a modified version of Criterion 2. It accounts for the quality of the observations and is more effective for inverse modeling problems.

We have illustrated the developed method by solving two inverse modeling problems. The new method has been demonstrated to be more efficient and reliable in comparison with non-adaptive PCKF and EnKF approaches.

Finally, there are two other points worth noting:

(1) The major computational burden in the new approach still is the model evaluations at collocation points. The extra computational cost caused by manipulating the PCE terms and judging the adaptive criteria does not make a difference in the efficiency competition between the new and old Kalman filter methods.

(2) Even with adaptive basis selection, the PCKF approach is not supposed to replace the EnKF for extremely high-dimensional problems, e.g., when the dimension is a few hundred or higher. For these problems, the computational
cost of the PCKF based only on 1st-order ANOVA decomposition still may be comparable to or exceed that required by EnKF.

Acknowledgements

The first author would like to acknowledge support from the China Scholarship Council through Grant 2009601077 and by US NSF Recovery Act (ARRA) Award ACI grant 0904754. The second author would like to acknowledge support from the US Department of Energy (DOE) Office of Sciences Advanced Scientific Computing Research Applied Mathematics program. Computations were performed using the computational resources of the National Energy Research Scientific Computing Center at Lawrence Berkeley National Laboratory, PNNL Institutional Computing cluster systems, and the William R. Wiley Environmental Molecular Sciences Laboratory (EMSL). EMSL is a DOE national scientific user facility located at PNNL. The Pacific Northwest National Laboratory is operated by Battelle for the US Department of Energy under Contract DE-AC05-76RL01830. The third author acknowledges the funding from the National Science and Technology Major Project of China through grants 2011ZX05009-006 and 2011ZX05052, as well as the National Key Technology R&D Program of China (Grant No. 2012BAC24B02).

References