Journal of Computational Physics 228 (2009) 5454–5469

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A generalized polynomial chaos based ensemble Kalman filter with high accuracy [☆]

Jia Li, Dongbin Xiu [*]

Department of Mathematics, Purdue University, West Lafayette, IN 47907, USA

ARTICLE INFO

Article history:
Received 8 August 2008
Received in revised form 15 April 2009
Accepted 16 April 2009
Available online 3 May 2009

Keywords:
Kalman filter
Data assimilation
Polynomial chaos
Uncertainty quantification

ABSTRACT

As one of the most adopted sequential data assimilation methods in many areas, especially those involving complex nonlinear dynamics, the ensemble Kalman filter (EnKF) has been under extensive investigation regarding its properties and efficiency. Compared to other variants of the Kalman filter (KF), EnKF is straightforward to implement, as it employs random ensembles to represent solution states. This, however, introduces sampling errors that affect the accuracy of EnKF in a negative manner. Though sampling errors can be easily reduced by using a large number of samples, in practice this is undesirable as each ensemble member is a solution of the system of state equations and can be time consuming to compute for large-scale problems. In this paper we present an efficient EnKF implementation via generalized polynomial chaos (gPC) expansion. The key ingredients of the proposed approach involve (1) solving the system of stochastic state equations via the gPC methodology to gain efficiency; and (2) sampling the gPC approximation of the stochastic solution with an arbitrarily large number of samples, at virtually no additional computational cost, to drastically reduce the sampling errors. The resulting algorithm thus achieves a high accuracy at reduced computational cost, compared to the classical implementations of EnKF. Numerical examples are provided to verify the convergence property and accuracy improvement of the new algorithm. We also prove that for linear systems with Gaussian noise, the first-order gPC Kalman filter method is equivalent to the exact Kalman filter.

© 2009 Elsevier Inc. All rights reserved.

1. Introduction

Data assimilation addresses the problem of producing useful simulation predictions based on imperfect model equations and measurements. It has been used extensively in atmospheric and oceanic applications and other geoscience areas, and beyond. The most widely adopted approach is Kalman filter [22,5], which is optimal for linear systems of state equations associated with Gaussian modeling and observation errors. However, for nonlinear systems the Kalman filter requires a linearization or a closure model of the state equations, resulting in the extended Kalman filter (for example, [11,19]), which may introduce significant error into the scheme. Furthermore, both the Kalman filter (KF) and the extended Kalman filter (EKF) require calculations of the evolution of the covariance function of the state variables. Although the covariance function provides a good estimate of uncertainty in the solutions, its storage and manipulation can be highly inefficient for systems with large dimensions of the state variables.

The ensemble Kalman filter (EnKF), first proposed by Evensen in [6] and later developed in [3] and many more work, has become popular in a wide variety of application areas. EnKF addresses the problem associated with linearization and

[☆] This research is in part supported by NSF CAREER Award DMS-0645035, AFOSR FA9550-08-1-0353, and DOE DE-FC52-08NA28617.

[*] Corresponding author. Tel.: +1 765 496 2846.
E-mail address: dxiu@math.purdue.edu (D. Xiu).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.04.029

efficiency by using ensemble representation of solution states. Sets of ensemble realizations are generated using Monte Carlo sampling for the initial state, model noise and measurement noise. Ensemble members are then forwarded in time by solving the (nonlinear) state equations and are analyzed by an approximate Kalman filter scheme. In doing so, EnKF avoids linearization of the model equations. The ensemble covariance is used as an approximation of the true covariance, thus avoiding explicit evolution and storage of the covariance as well. Since its introduction, several variations of EnKF have appeared to gain computational efficiency. See, for example, extensive reviews in [8,10].

The obvious source of numerical errors of EnKF stems from sampling, which includes sampling of the state variables and of the measurement. Such sampling errors can have a notable impact on the effectiveness of EnKF. In fact, a numerical error estimate was conducted in [23], and the result indicates that more frequent data assimilation by EnKF does not intuitively lead to a more accurate estimate of the true states due to the accumulation of sampling errors. Efforts have been devoted to design more efficient EnKF schemes by reducing the sampling errors. In particular, ensemble square-root filter (EnSRF) [26,2,1,9] employs a deterministic update of the forecast model states without generating measurement noises numerically and thus eliminates the errors induced by sampling the measurement. However, to reduce errors in sampling the model states, there are not many effective approaches except to increase the ensemble size. See, for example, [17], for discussions on various options such as localization. The relatively slow convergence rate of Monte Carlo sampling implies that in order to effectively reduce the sampling error, a large number of realizations are required. This is undesirable in practice as each realization requires a solution of the governing model equations and can be time consuming to compute for large-scale complex systems. As a result, a trade-off between efficiency and accuracy exists when one implements EnKF (or EnSRF) in practice.

A method to reduce sampling errors for model states was proposed in [23]. It employs a set of optimal cubature rules in place of the Monte Carlo sampling and can be quite efficient. This is similar to the earlier work on unscented Kalman filter (UKF) [20,21] and Gauss–Hermite quadrature filter [18]. However, the numerical accuracy of such methods can not be easily refined without incurring additional computational cost, and this can limit its effectiveness for highly complex systems. In this paper we present a numerical strategy for EnKF based on generalized polynomial chaos (gPC). The gPC, first systematically presented in [31], is an extension of the classical polynomial chaos theory pioneered by Ghanem [13,12] and has been successful for stochastic computations. In gPC, stochastic quantities are expressed as convergent polynomial series of input random variables, and efficient numerical schemes (stochastic Galerkin or stochastic collocation) can be constructed accordingly. Here we construct a set of efficient algorithms based on the gPC expansion and the EnSRF scheme. The key ingredients of the proposed approach involve (1) solving the system of stochastic state equations via the gPC-based numerical methods (stochastic Galerkin or stochastic collocation) to gain efficiency; (2) sampling the gPC approximation of the stochastic solution with an arbitrarily large number of samples, at virtually no additional computational cost, to drastically reduce sampling errors; (3) combining with the EnSRF strategy to eliminate errors in sampling the measurement. The resulting algorithm thus achieves a high accuracy at reduced computational cost, compared to the classical implementations of EnKF/EnSRF. For the linear system of equations with Gaussian noise, it can be shown that the first-order gPC filter is equivalent to the Kalman filter. We remark that although the new gPC filer can significantly reduce sampling errors of EnKF/EnSRF, it inherits the same fundamental assumptions, such as Gaussian noise from the Kalman filter. In other words, if one views all versions of EnKF and EnSRF as numerical approximations of the Kalman filter, then the gPC filter is another approximation that offers (much) smaller numerical errors.

The rest of the paper is arranged as follows: a brief review of KF, EnKF and EnSRF is in Section 2. The gPC methods are introduced in Section 3, where fast solvers for forecast state equations are in Section 3.1 and the new filtering scheme is in Section 3.2. Numerical examples are presented in Section 4 to examine the properties of the gPC–EnSRF and to demonstrate its efficiency. Conclusions and comments are in Section 5.

2. Data assimilation and Kalman filter

In this section we briefly review the idea and main properties of the Kalman filter (KF) and ensemble Kalman filter (EnKF) for data assimilation. The exposition will be made in the context of nonlinear system of ordinary differential equations, as we follow the traditional approach by focusing on time evolution of the system.

2.1. Data assimilation

Let $\mathbf{u}^f \in \mathbb{R}^m, m \geqslant 1$, be a vector of *forecast* state variables (denoted by the superscript *f*) that are modeled by the following system:

$$\frac{d\mathbf{u}^f}{dt} = f(t, \mathbf{u}^f), \quad t \in (0, T], \tag{1}$$

$$\mathbf{u}^f(0) = \mathbf{u}_0, \tag{2}$$

with $T > 0$. The model (1) and (2) is obviously not a perfect model for true physics and the forecast may not represent the true state variables, $\mathbf{u}^t \in \mathbb{R}^m$, sufficiently well. If a set of *measurements* $\mathbf{d} \in \mathbb{R}^\ell, \ell \geqslant 1$, are available as

$$\mathbf{d} = \mathbf{H}\mathbf{u}^t + \epsilon, \tag{3}$$

where $\mathbf{H}:\mathbb{R}^m\to\mathbb{R}^\ell$ is a measurement operator relating the *true state* variables $\mathbf{u}^t$ and the observation vector $\mathbf{d}\in\mathbb{R}^\ell$, and $\epsilon\in\mathbb{R}^\ell$ is measurement error. Note the measurement operator can be nonlinear, although it is written here in a linear fashion by following the traditional exposition of the (ensemble) Kalman filter. Also the characterization of true state variables $\mathbf{u}^t$ can be highly nontrivial in practice. Here we assume they are well defined variables with dimension m.

The objective of data assimilation is to construct an optimal estimate of the true state, the *analyzed state* vector denoted as $\mathbf{u}^a\in\mathbb{R}^m$, based on the forecast $\mathbf{u}^f$ and the observation $\mathbf{d}$. Note it is possible to add a noise term in (1) as a model for the modeling error. Here we restrict ourselves to the deterministic model (1).

2.2. Kalman filter

The Kalman filter is a sequential data assimilation method that consists of two stages at each time level – a forecast stage where the system (1) and (2) is solved, and an analysis stage where the analyzed state $\mathbf{u}^a$ is obtained.

Let $\mathbf{P}^f\in\mathbb{R}^{m\times m}$ be the covariance matrix of the forecast solution $\mathbf{u}^f$. The analyzed solution $\mathbf{u}^a$ in the standard KF is determined as a combination of the forecast solution $\mathbf{u}^f$ and the measurement $\mathbf{d}$ in the following manner,

$$\mathbf{u}^a=\mathbf{u}^f+\mathbf{K}(\mathbf{d}-\mathbf{H}\mathbf{u}^f), \tag{4}$$

where $\mathbf{K}$ is the so-called *Kalman gain matrix* defined as

$$\mathbf{K}=\mathbf{P}^f\mathbf{H}^T(\mathbf{H}\mathbf{P}^f\mathbf{H}^T+\mathbf{R})^{-1}. \tag{5}$$

Here the superscript T denotes the matrix transpose, and $\mathbf{R}\in\mathbb{R}^{\ell\times\ell}$ is the covariance of the measurement error ϵ. The covariance function of the analyzed state $\mathbf{u}^a, \mathbf{P}^a\in\mathbb{R}^{m\times m}$, is then obtained by

$$\mathbf{P}^a=(\mathbf{I}-\mathbf{K}\mathbf{H})\mathbf{P}^f(\mathbf{I}-\mathbf{K}\mathbf{H})^T+\mathbf{K}\mathbf{R}\mathbf{K}^T=(\mathbf{I}-\mathbf{K}\mathbf{H})\mathbf{P}^f, \tag{6}$$

where $\mathbf{I}$ is the identity matrix.

When the system (1) is linear, the KF can be applied in a straightforward manner, as equations for the evolution of the solution covariance can be derived. For nonlinear systems, explicit derivation of the equations for the covariance function is not possible. Subsequently, the extended Kalman filter (EKF), which employs either linearization of the model equation (1) or some closure approximation, is developed. The applicability of the EKF is, however, limited due to approximation errors by the linearization or closure assumption. Furthermore, in practical applications, forwarding the covariance functions (6) in time requires an explicit storage and computation of $\mathbf{P}^f$, which scales as $O(m^2)$ and can be inefficient when the dimension of the model states, m, is large.

2.3. Ensemble Kalman filter

The ensemble Kalman filter (EnKF) overcomes the limitations of the Kalman filter (or the extended Kalman filter) by using an ensemble approximation of the random state solutions.

Let

$$(\mathbf{u}^f)_i,\quad i=1,\ldots,M,\quad M>1, \tag{7}$$

be an ensemble of the forecast state variables $\mathbf{u}^f$, where each ensemble member is indexed by the subscript $i=1,\ldots,M$, and obtained by solving the full nonlinear system (1). The analysis step for the EnKF consists of the following update performed on each of the model state ensemble members

$$(\mathbf{u}^a)_i=(\mathbf{u}^f)_i+\mathbf{K}_e((\mathbf{d})_i-\mathbf{H}(\mathbf{u}^f)_i),\quad i=1,\ldots,M, \tag{8}$$

where

$$\mathbf{K}_e=\mathbf{P}_e^f\mathbf{H}^T(\mathbf{H}\mathbf{P}_e^f\mathbf{H}^T+\mathbf{R}_e)^{-1} \tag{9}$$

is the *ensemble Kalman gain matrix*. Here

$$\begin{aligned}\mathbf{P}_e^f &\triangleq \overline{(\mathbf{u}^f-\bar{\mathbf{u}}^f)(\mathbf{u}^f-\bar{\mathbf{u}}^f)^T}\simeq\mathbf{P}^f,\\ \mathbf{P}_e^a &\triangleq \overline{(\mathbf{u}^a-\bar{\mathbf{u}}^a)(\mathbf{u}^a-\bar{\mathbf{u}}^a)^T}\simeq\mathbf{P}^a,\end{aligned} \tag{10}$$

are the approximate forecast covariance and analysis covariance, respectively, obtained by using statistical averages of the solution ensemble (denoted by the overbar), and $\mathbf{R}_e=\overline{\epsilon\epsilon^T}\simeq\mathbf{R}$ is the approximate observation error covariance. Therefore, the covariance functions are approximated by ensemble averages and are not needed to be forwarded in time explicitly.

In its original setting, cf. [6,3], the observations are treated as random variables and an ensemble of observations are generated, based on the covariance matrix $\mathbf{R}$. Though straightforward to implement, this approach introduces a sampling error in the Kalman gain matrix and subsequently affects the accuracy. An alternative, called the ensemble square-root filter (EnSRF), was introduced to eliminate the error in sampling the observations. This is achieved by constructing the analysis scheme without perturbing the measurements. Various versions of EnSRF have been proposed. See, for example, [2,1,26,9]. Here we briefly review the method developed in [26].

The forecast and analyzed states can be written as follows:

$$(\mathbf{u}^f)_i = \bar{\mathbf{u}}^f + (\mathbf{u}^f)'_i, \quad (\mathbf{u}^a)_i = \bar{\mathbf{u}}^a + (\mathbf{u}^a)'_i, \quad i = 1, \ldots, M, \tag{11}$$

where $\bar{\mathbf{u}}^f$ and $\bar{\mathbf{u}}^a$ denote the mean of the forecast and the analyzed states, and $(\mathbf{u}^f)'_i$ and $(\mathbf{u}^a)'_i$ are the corresponding deviations from their mean.

In the analysis step of EnSRF, the ensemble mean and the deviations are updated separately.

$$\bar{\mathbf{u}}^a = \bar{\mathbf{u}}^f + \mathbf{K}_e(\mathbf{d} - \mathbf{H}\bar{\mathbf{u}}^f), \tag{12}$$

$$(\mathbf{u}^a)'_i = (\mathbf{u}^f)'_i - \widetilde{\mathbf{K}}_e\mathbf{H}(\mathbf{u}^f)'_i, \quad i = 1, \ldots, M, \tag{13}$$

where $\mathbf{K}_e$ is the ensemble Kalman gain matrix (9), and

$$\widetilde{\mathbf{K}}_e = \mathbf{P}^f_e\mathbf{H}^T\left(\left(\sqrt{\mathbf{H}\mathbf{P}^f_e\mathbf{H}^T + \mathbf{R}}\right)^{-1}\right)^T\left(\sqrt{\mathbf{H}\mathbf{P}^f_e\mathbf{H}^T + \mathbf{R}} + \sqrt{\mathbf{R}}\right)^{-1}, \tag{14}$$

which is obtained by satisfying the equation

$$(\mathbf{I} - \widetilde{\mathbf{K}}_e\mathbf{H})\mathbf{P}^f_e(\mathbf{I} - \mathbf{H}^T\widetilde{\mathbf{K}}^T_e) = (\mathbf{I} - \widetilde{\mathbf{K}}_e\mathbf{H})\mathbf{P}^f_e, \tag{15}$$

so that the resulting covariance of the analysis states matches the theoretical covariance $\mathbf{P}^a$ from the KF. It is obvious that the EnSRF does not require explicit ensemble representation of the measurement $\mathbf{d}$ in the analysis scheme and eliminates the corresponding sampling error. However, sampling errors for the state variables are still present.

2.4. Error bound of EnKF

The major contribution of numerical errors for EnKF is made by sampling. To understand the impact of numerical errors, we here cite an error bound of EnKF derived in [23]. Let $t_1 < t_2 < \cdots$ be discrete time instances at which data arrive sequentially and assimilation is made. Without loss of generality let us assume they are uniformly distributed with a constant step size $\Delta T = t_k - t_{k-1}, \forall k > 1$. Let E_n be the numerical error of the EnKF, that is, the difference between the EnKF results and the exact KF results measured in a proper norm (note this is not the difference between the EnKF results and the true states,) at time level $t_n, n \geqslant 1$, then the following bound holds,

$$E_n \leqslant \left(E_0 + \sum_{k=1}^{n} e_k\right)\exp(\Lambda \cdot t_n), \tag{16}$$

where E_0 is the error of sampling the initial state, e_k is the local error at time level $t_k, 1 \leqslant k \leqslant n$, and $\Lambda > 0$ is a constant. The local error scales as

$$e_k \sim O(\Delta t^p, \sigma M^{-\alpha}), \quad \Delta t \to 0, \quad M \to \infty, \tag{17}$$

where $O(\Delta t^p)$ denotes the numerical integration error in time induced by solving (1) and (2) with a time step Δt and a temporal integration order $p \geqslant 1, \sigma > 0$ is the noise level of the measurement that scales with the standard deviation of the measurement noise, M is the size of the ensemble, and $\alpha > 0$ is the convergence rate of the sampling scheme. For Monte Carlo sampling, $\alpha = 1/2$. In most cases, this sampling error dominates. A notable result is that the constant Λ depends on the size of the assimilation step in an inverse manner, i.e., $\Lambda \propto \Delta T^{-1}$. This implies that more frequent data assimilation by the EnKF can magnify the numerical errors. Since more frequent assimilation is always desirable (whenever data are available) for better estimate of the true state, it is imperative to keep the numerical errors, particularly the sampling errors, of the EnKF under control. Although the sampling errors can be easily reduced by increasing the ensemble size, in practice this can significantly increase the computational burden, especially for large-scale problems.

3. gPC-based ensemble Kalman filter

In this section we present an ensemble Kalman filter algorithm using the methodology of generalized polynomial chaos (gPC). The gPC framework is presented first; we then discuss how to construct a set of highly accurate EnKF methods based on the gPC expansion.

3.1. Solution of the forecast state by gPC

In the Kalman filter, the modeling error in the system (1) and (2) is typically assumed to be in the initial condition (2) which is modeled as a random quantity. That is, (2) becomes

$$\mathbf{u}^f(0) = \mathbf{u}_0(Z), \quad Z \in \mathbb{R}^n, \quad n \geqslant 1, \tag{18}$$

where $Z = (Z_1, \ldots, Z_n)$ is a set of independent random variables parameterizing the random initial condition with probability density function $\rho(z) : \mathbb{R}^n \to \mathbb{R}^+ = \prod_{k=1}^n \rho^{(k)}(z_k)$. Here $\rho^{(k)}(z_k)$ is the probability distribution of $Z_k, k = 1, \ldots, n$. Subsequently, the forecast state variables become stochastic variables and can be parameterized by the same set of random variables, i.e.,

$$\mathbf{u}^f \triangleq \mathbf{u}^f(t, Z) : [0, T] \times \mathbb{R}^n \to \mathbb{R}^m.$$

An Nth-order generalized polynomial chaos (gPC) expansion to the solution of (1) and (2), $\mathbf{u}^f(t, Z)$, takes the following form, for any $t \in [0, T]$,

$$\mathbf{u}_N(t, Z) = \sum_{|\mathbf{i}|=0}^{N} \hat{\mathbf{u}}_{\mathbf{i}}(t) \Phi_{\mathbf{i}}(Z), \tag{19}$$

where $\mathbf{i} = (i_1, \ldots, i_n) \in \mathbb{N}_0^n$ is a multi-index with $|\mathbf{i}| = i_1 + \cdots + i_n$, and

$$\Phi_{\mathbf{i}}(Z) = \prod_{k=1}^{n} \phi_{i_k}(Z_k), \quad |\mathbf{i}| \leqslant N,$$

are n-variate orthogonal polynomial basis functions constructed as products of the univariate polynomials $\phi_{i_k}(Z_k)$. Here $\phi_{i_k}(Z_k)$ are the i_kth-order orthogonal polynomials in the Z_k dimension satisfying

$$\mathbb{E}_k[\phi_m(Z_k)\phi_n(Z_k)] \triangleq \int \phi_m(z_k)\phi_n(z_k)\rho^{(k)}(z_k)dz_k = \delta_{mn}, \quad 0 \leqslant m, n \leqslant N, \tag{20}$$

where δ_{mn} is the Kronecker delta function and the polynomials are normalized. Therefore, $\{\Phi_{\mathbf{i}}(Z)\}_{|\mathbf{i}| \leqslant N}$ are n-variate orthonormal polynomials of total degree up to N such that

$$\mathbb{E}[\Phi_{\mathbf{i}}(Z)\Phi_{\mathbf{j}}(Z)] \triangleq \int \Phi_{\mathbf{i}}(z)\Phi_{\mathbf{j}}(z)\rho(z)dz = \delta_{\mathbf{ij}}, \tag{21}$$

where $\delta_{\mathbf{ij}} = \prod_{k=1}^n \delta_{i_k j_k}$. The total number of basis functions is

$$\binom{N+n}{n}. \tag{22}$$

The expansion coefficients in (19) can be obtained by an orthogonal projection,

$$\hat{\mathbf{u}}_{\mathbf{i}}(t) = \mathbb{E}[\mathbf{u}^f(t, Z)\Phi_{\mathbf{i}}(Z)] = \int \mathbf{u}^f(t, z)\Phi_{\mathbf{i}}(z)\rho(z)dz, \quad \forall |\mathbf{i}| \leqslant N. \tag{23}$$

Classical approximation theory guarantees that this is the best approximation in the linear space of n-variate polynomials of degree up to N in the mean-square sense.

3.1.1. *Stochastic Galerkin and collocation methods*

In practice, the projection for the expansion coefficients (23) is not available as it requires knowledge of the solution. Two often used approaches to numerically approximate the coefficients are the stochastic Galerkin (SG) method and the stochastic collocation (SC) method. The stochastic Galerkin approach seeks an approximate gPC solution in the similar form of (19), i.e., for any $t \in [0, T]$,

$$\mathbf{v}_N(t, Z) = \sum_{|\mathbf{i}|=0}^{N} \hat{\mathbf{v}}_{\mathbf{i}}(t)\Phi_{\mathbf{i}}(Z). \tag{24}$$

The expansion coefficients $\{\hat{\mathbf{v}}_{\mathbf{i}}\}$ are obtained by satisfying (1) and (2) in the following weak form, for all $|\mathbf{k}| \leqslant N$,

$$\frac{d\hat{\mathbf{v}}_{\mathbf{k}}}{dt} = \mathbb{E}[f(t, \mathbf{v}_N)\Phi_{\mathbf{k}}], \quad t \in (0, T], \tag{25}$$

$$\hat{\mathbf{v}}_{\mathbf{k}}(0) = \mathbb{E}[\mathbf{u}_0\Phi_{\mathbf{k}}]. \tag{26}$$

The resulting equations are a set of (usually coupled) *deterministic* equations for $\{\hat{\mathbf{v}}_{\mathbf{k}}\}$, and standard numerical techniques can be applied.

Another approach is to employ the pseudo-spectral stochastic collocation approach [28]. Here we again seek an approximate solution in the form of the gPC expansion (19), i.e., for any $t \in [0, T]$,

$$\mathbf{w}_N(t, Z) = \sum_{|\mathbf{i}|=0}^{N} \hat{\mathbf{w}}_{\mathbf{i}}(t)\Phi_{\mathbf{i}}(Z), \tag{27}$$

where the expansion coefficients are determined as

$$\hat{\mathbf{w}}_{\mathbf{i}}(t) = \sum_{j=1}^{Q} \mathbf{u}^f(t, Z^{(j)})\Phi_{\mathbf{i}}(Z^{(j)})\alpha^{(j)}, \quad \forall|\mathbf{i}| \leqslant N. \tag{28}$$

Here $\{Z^{(j)}, \alpha^{(j)}\}_{j=1}^{Q}$ are a set of nodes and weights, and $\mathbf{u}^f(t, Z^{(j)})$ is the deterministic solution of (1) with fixed $Z^{(j)}$. The nodes and weights should be chosen from a cubature rule such that

$$\hat{\mathbf{w}}_{\mathbf{i}}(t) \approx \mathbb{E}[\mathbf{u}^f(t, Z)\Phi_{\mathbf{i}}(Z)] = \hat{\mathbf{u}}_{\mathbf{i}}(t), \quad \forall|\mathbf{i}| \leqslant N, \tag{29}$$

where the last equality follows from (23). Subsequently (27) becomes an approximation of the exact gPC expansion (19). The difference between the two is caused by the integration error from (29) and is termed "aliasing error" in [28], following similar terminology from the classical deterministic spectral methods (cf. [14,4,16]).

We also remark that the original development of stochastic collocation methods utilizes multivariate Lagrange interpolation technique [30]. This approach, however, is not amenable to the data assimilation work we undertake here. Therefore, we will focus on the pseudo-spectral stochastic collocation approach [28].

3.1.2. Summary of gPC-based methods

In summary, all gPC-based methods seek to approximate the stochastic solution of (1) and (2) in the form of (19), where the expansion coefficients are obtained approximately via either a Galerkin approach, (24), or a collocation approach, (27). Depending on the probability distribution of the random variables Z, different orthogonal polynomials can be employed for better performance [31]. Whenever the solution is relatively smooth in the random space, the gPC methods exhibit fast convergence and can be significantly more efficient than the traditional methods such as the Monte Carlo sampling. For an extensive review of the gPC-based numerical methods, see [29].

3.2. Solution of the analyzed state by gPC and EnKF

Let

$$\mathbf{u}_N^f(t, Z) = \sum_{|\mathbf{i}|=0}^{N} \hat{\mathbf{u}}_{\mathbf{i}}^f(t)\Phi_{\mathbf{i}}(Z) \tag{30}$$

denote the gPC solution to the forecast equation (1) and (2) with sufficiently high accuracy, where the expansion coefficients $\hat{\mathbf{u}}_{\mathbf{i}}^f(t)$ can be either the $\hat{\mathbf{v}}_{\mathbf{i}}(t)$ obtained by the stochastic Galerkin procedure (24) or the $\hat{\mathbf{w}}_{\mathbf{i}}(t)$ obtained by the stochastic collocation procedure (27).

In addition to offering efficient solvers for the forecast solution, as discussed in the previous section, another (often overlooked) advantage of the gPC expansion is that it provides an analytical representation of the solution in term of the random inputs. All statistical information about $\mathbf{u}_N^f$ can be obtained analytically, or with minimum computational effort. For example, the mean and covariance are

$$\bar{\mathbf{u}}_N^f = \hat{\mathbf{u}}_{\mathbf{0}}^f, \quad \mathbf{P}_N^f = \sum_{0<|\mathbf{i}|\leqslant N} \left[\hat{\mathbf{u}}_{\mathbf{i}}^f (\hat{\mathbf{u}}_{\mathbf{i}}^f)^T\right], \tag{31}$$

respectively. And they can be used as accurate approximations of the exact mean and covariance of the forecast solution $\mathbf{u}^f$. Furthermore, one can generate an ensemble of solution realizations by sampling the random variables Z in (30). This procedure involves nothing but polynomial evaluations and thus generating ensemble with arbitrarily large number of samples does not require any computations of the original governing equations (1) and (2). Let

$$(\mathbf{u}_N^f)_i = \sum_{|\mathbf{k}|=0}^{N} \hat{\mathbf{u}}_{\mathbf{k}}^f(t)\Phi_{\mathbf{k}}((Z)_i) \quad i = 1, \ldots, M, \quad M \gg 1, \tag{32}$$

be an ensemble of the forecast solution realizations with size M, where $(Z)_i, i = 1, \ldots, M$, are Monte Carlo samples of the random vector Z. Equipped with the knowledge of the solution statistics, particularly the mean and covariance from (31), we can apply the EnKF scheme (8) to obtain analyzed states. Here we employ the EnSRF approach, primarily because of the elimination of error in sampling the measurement. Following the procedure in Section 2.3, the gPC forecast and analyzed states are split into the mean and deviation parts:

$$(\mathbf{u}_N^f)_i = \bar{\mathbf{u}}_N^f + (\mathbf{u}_N^f)_i', \quad (\mathbf{u}_N^a)_i = \bar{\mathbf{u}}_N^a + (\mathbf{u}_N^a)_i', \quad i = 1, \ldots, M, \tag{33}$$

and updated separately as

$$\bar{\mathbf{u}}_N^a = \bar{\mathbf{u}}_N^f + \mathbf{K}_N(\mathbf{d} - \mathbf{H}\bar{\mathbf{u}}_N^f), \tag{34}$$

$$(\mathbf{u}_N^a)_i' = (\mathbf{u}_N^f)_i' - \widetilde{\mathbf{K}}_N\mathbf{H}(\mathbf{u}_N^f)_i', \quad i = 1, \ldots, M, \tag{35}$$

where $\mathbf{K}_N$ is the *gPC Kalman gain matrix* defined as

$$\mathbf{K}_N = \mathbf{P}_N^f \mathbf{H}^T (\mathbf{H}\mathbf{P}_N^f \mathbf{H}^T + \mathbf{R})^{-1}, \tag{36}$$

which approximates the Kalman gain matrix (5), and

$$\widetilde{\mathbf{K}}_N = \mathbf{P}_N^f \mathbf{H}^T \left(\left(\sqrt{\mathbf{H}\mathbf{P}_N^f \mathbf{H}^T + \mathbf{R}} \right)^{-1} \right)^T \left(\sqrt{\mathbf{H}\mathbf{P}_N^f \mathbf{H}^T + \mathbf{R}} + \sqrt{\mathbf{R}} \right)^{-1}, \tag{37}$$

which is obtained by requiring

$$(\mathbf{I} - \widetilde{\mathbf{K}}_N \mathbf{H}) \mathbf{P}_N^f (\mathbf{I} - \mathbf{H}^T \widetilde{\mathbf{K}}_N^T) = (\mathbf{I} - \widetilde{\mathbf{K}}_N \mathbf{H}) \mathbf{P}_N^f. \tag{38}$$

3.3. Algorithms

Here we present two versions of the aforementioned gPC-based EnSRF method in detail, one based on the stochastic Galerkin method and the other on the stochastic collocation method. In the following, we assume observation data arrives sequentially in time at time level $t_1 < t_2 < \cdots$, at which data assimilation is made.

3.3.1. Stochastic Galerkin based gPC–EnSRF

Here the Nth degree gPC solutions of the forecast and analyzed variables are expressed as

$$\mathbf{u}_N^f(t, Z) = \sum_{|\mathbf{i}|=0}^{N} \hat{\mathbf{v}}_{\mathbf{i}}^f(t) \Phi_{\mathbf{i}}(Z), \quad \mathbf{u}_N^a(t, Z) = \sum_{|\mathbf{i}|=0}^{N} \hat{\mathbf{v}}_{\mathbf{i}}^a(t) \Phi_{\mathbf{i}}(Z). \tag{39}$$

1. **Initialization**. At time $t = 0$, let $\mathbf{u}_N^a(0, Z) = \sum \hat{\mathbf{v}}_{\mathbf{i}}^a(0) \Phi_{\mathbf{i}}(Z)$ be the gPC approximation of the initial state (18), where the coefficients $\hat{\mathbf{v}}_{\mathbf{i}}^a(0) = \mathbb{E}[\mathbf{u}_0(Z) \Phi_{\mathbf{i}}(Z)]$.
2. **Forecast**.
   - At time t_{n-1}, let $\{(\hat{\mathbf{v}}_{\mathbf{i}}^a(t_{n-1})\}$ be the expansion coefficients for the gPC analyzed state estimates. For each $\mathbf{i}$ such that $|\mathbf{i}| \leqslant N$, we solve the system of (1) by the stochastic Galerkin scheme (25) with initial condition $\{(\hat{\mathbf{v}}_{\mathbf{i}}^a(t_{n-1})\}$ at t_{n-1} and advance to time level t_n to obtain $\{\hat{\mathbf{v}}_{\mathbf{i}}^f(t_n)\}$ for the forecast coefficients.
   - Construct Nth-order gPC approximation of the forecast solution

$$\mathbf{u}_N^f(t_n, Z) = \sum_{|\mathbf{i}|=0}^{N} \hat{\mathbf{v}}_{\mathbf{i}}^f(t_n) \Phi_{\mathbf{i}}(Z). \tag{40}$$

3. **Analysis**.
   - Evaluate the statistics of the forecast state solution such as the mean and covariance by (31). Evaluate the gPC Kalman gain matrix (36) and (37).
   - Generate a large ensemble of forecast state realizations $(\mathbf{u}_N^f(t_n))_i = \mathbf{u}_N^f(t_n, (Z)_i), i = 1, \ldots, M$, by sampling the random variables Z in the gPC solution (40) with ensemble size $M \gg 1$. Update each member of the ensemble by the EnSRF procedure (34) and (35) and obtain the ensemble of analyzed state $\{(\mathbf{u}_N^a(t_n))_i\}_{i=1}^M$.
   - Evaluate the expansion coefficients for the analyzed state by averaging

$$\hat{\mathbf{v}}_{\mathbf{i}}^a(t_n) = \mathbb{E}[\mathbf{u}^a(t_n, Z) \Phi_{\mathbf{i}}(Z)] \approx \frac{1}{M} \sum_{i=1}^{M} (\mathbf{u}_N^a(t_n))_i \Phi_{\mathbf{i}}((Z)_i). \tag{41}$$

   - Return to Step 2. Advance in time till the final time is reached.

Note the averaging procedure (41) for approximating the gPC coefficients introduces sampling errors, which can be very small because we can employ an arbitrarily large number of samples in the analysis step. Again, the computational cost of generating an arbitrarily large number of samples requires nothing but sampling of the polynomial expression of (40) with a large number of random "seeds" in Z. Hence this cost is minimal because it does not require any simulations of the governing system of equations.

3.3.2. Stochastic collocation based gPC–EnSRF

Here the gPC solutions of the forecast and analyzed variables are expressed as

$$\mathbf{u}_N^f(t, Z) = \sum_{|\mathbf{i}|=0}^{N} \hat{\mathbf{w}}_{\mathbf{i}}^f(t) \Phi_{\mathbf{i}}(Z), \quad \mathbf{u}_N^a(t, Z) = \sum_{|\mathbf{i}|=0}^{N} \hat{\mathbf{w}}_{\mathbf{i}}^a(t) \Phi_{\mathbf{i}}(Z). \tag{42}$$

1. **Initialization**. Choose a proper cubature rule with nodes and weights $\{Z^{(j)}, \alpha^{(j)}\}_{j=1}^Q$, where $Q \geqslant 1$ is the total number of nodes. At time $t = 0$, let $\{(\mathbf{u}^a(0))_j\}_{j=1}^Q = \{\mathbf{u}_0(Z^{(j)})\}_{j=1}^Q$ be the nodal values of the initial condition (18).

2. **Forecast**.
   - At time t_{n-1}, let $\{(\mathbf{u}^a(t_{n-1}))_j\}_{j=1}^Q = \{\mathbf{u}^a(t_{n-1}, Z^{(j)})\}_{j=1}^Q$ be the analyzed state estimates on the nodes $\{Z^{(j)}\}_{j=1}^Q$. For each $j = 1, \ldots, Q$, we solve the system of equations (1) with fixed $Z^{(j)}$ and initial condition $(\mathbf{u}^a(t_{n-1}))_j$ at t_{n-1} and advance to time level t_n to obtain the forecast solution at t_n, $(\mathbf{u}^f(t_n))_j$.
   - Construct Nth-order pseudo-spectral gPC approximation of the forecast solution

$$\mathbf{u}_N^f(t_n, Z) = \sum_{|\mathbf{i}|=0}^{N} \hat{\mathbf{w}}_{\mathbf{i}}^f(t_n)\Phi_{\mathbf{i}}(Z), \tag{43}$$

where the coefficients are

$$\hat{\mathbf{w}}_{\mathbf{i}}^f(t_n) = \sum_{j=1}^{Q} (\mathbf{u}^f(t_n))_j \Phi_{\mathbf{i}}(Z^{(j)})\alpha^{(j)}, \quad \forall |\mathbf{i}| \leqslant N. \tag{44}$$

3. **Analysis**.
   - Evaluate the statistics of the forecast state solution such as the mean and covariance (31). Evaluate the gPC Kalman gain matrix (36) and (37).
   - Generate a large ensemble of forecast state realizations $(\mathbf{u}_N^f(t_n))_i = \mathbf{u}_N^f(t_n, (Z)_i), i = 1, \ldots, M$, by sampling the random variables Z in the gPC solution (43) with ensemble size $M \gg 1$. Update each member of the ensemble by the EnSRF procedure (34) and (35) and obtain the ensemble of analyzed state $\{(\mathbf{u}_N^a(t_n))_i\}_{i=1}^M$.
   - Evaluate the analyzed state at the cubature nodes to obtain $(\mathbf{u}^a(t_n))_j = \mathbf{u}^a(t_n, Z^{(j)})$ for $j = 1, \ldots, Q$. A general procedure to achieve this is accomplished by first evaluating the gPC coefficients of the analyzed state via averaging

$$\hat{\mathbf{w}}_{\mathbf{i}}^a(t_n) = \mathbb{E}[\mathbf{u}^a(t_n, Z)\Phi_{\mathbf{i}}(Z)] \approx \frac{1}{M}\sum_{i=1}^{M} (\mathbf{u}_N^a(t_n))_i \Phi_{\mathbf{i}}((Z)_i), \tag{45}$$

and then constructing the gPC expansion for the analyzed state

$$\mathbf{u}_N^a(t_n, Z) = \sum_{|\mathbf{i}|=0}^{N} \hat{\mathbf{w}}_{\mathbf{i}}^a(t_n)\Phi_{\mathbf{i}}(Z), \tag{46}$$

and evaluating the expression at the nodes $Z^{(j)}, \ j = 1, \ldots, Q$.
   - Return to Step 2. Advance in time till the final time is reached.

Note the objective of the third step in the Analysis step is to evaluate the values of $\mathbf{u}^a(t_n)$ at the cubature nodes $\{Z^{(j)}\}_{j=1}^Q$, given the values of the $\mathbf{u}^a(t_n)$ at the large number of random nodes $\{(Z)_i\}_{i=1}^M$, where typically $Q \ll M$. It is possible to achieve the goal by using a multivariate interpolation scheme, without using (45) and (46). The interpolation approach can be effective when the dimension of the random space, n, is low, e.g. less than four.

3.4. Discussions

3.4.1. Efficiency and accuracy

In the stochastic Galerkin based algorithm, the key quantities are the gPC expansion coefficients $\{\hat{\mathbf{v}}_{\mathbf{i}}(t), |\mathbf{i}| \leqslant N\}$, which are propagated by the forecast equations and updated by the EnSRF scheme. In the stochastic collocation based algorithm, the key quantities are the nodal values of the gPC solution at the chosen cubature nodes, $\{\mathbf{u}(t, Z^{(j)}), j = 1, \ldots, Q\}$. For accurate approximation of the state variables and solution of the forecast system of equations (1), the total number of Galerkin equations or collocation equations can be significantly smaller than that required by traditional stochastic solvers such as Monte Carlo sampling, provided that the number of random variables n is small or moderately large. Such efficiency gain has been well documented in the literature (cf. [13,31]).

Furthermore, the present algorithms (both Galerkin based and collocation based) allow accurate EnSRF update at the analysis step, because the explicit gPC expression allows one to generate ensembles with arbitrarily large size. Such ensemble generation requires only algebraic evaluations that can be implemented without incurring notable computational cost and results in much reduced sampling errors for the state variables.

3.4.2. Choice of algorithms

For practical problems involving highly nonlinear system of equations, the stochastic collocation based algorithm is preferred, primarily due to its ease of implementation and ability to handle nonlinearity. However, it should be noted that stochastic collocation method suffers from aliasing error. Whenever possible, the stochastic Galerkin based method offers better accuracy. More discussions about Galerkin and collocation can be found in [29].

It is worth noting that the gPC collocation based filter is in a way similar to the unscented Kalman filter (UKF) [20,21] and Gauss–Hermite quadrature filter [18]. However, the key and unique feature of the gPC filter is in the construction of the gPC polynomial expression (40), which allows one to generate an arbitrarily large number of samples in the update step and thus significantly reduces sampling errors.

3.4.3. Equivalence to Kalman filter

When the system of state Eqs. (1) and (2) is linear and with Gaussian noise, the Kalman filter is optimal and relatively easy to implement. In this case, the optimal gPC basis functions are the Hermite polynomials [13,31]. It is straightforward to show that the first-order stochastic Galerkin implementation of the gPC Kalman filter is exact, in the sense that it is equivalent to the Kalman filter. This is expressed in the following theorem, whose proof is included in Appendix.

Theorem 1. *Assume the forecast system of equations is linear*

$$\frac{d\mathbf{u}^f}{dt} = \mathbf{A}(t)\mathbf{u}^f + \mathbf{g}(t), \tag{47}$$

$$\mathbf{u}^f(t_0) = \mathbf{u}_0(Z), \tag{48}$$

where the initial condition $\mathbf{u}_0(Z)$ has a Gaussian distribution, and measurement (3) also has a Gaussian distribution. Let $\mathbf{u}_1^f(t)$ be the first-order gPC Galerkin solution, using Hermite polynomials, to (47) and (48), and $\mathbf{K}_1$ be the corresponding gPC Kalman gain matrix defined in (36). Then the analyzed state obtained by the first-order gPC Kalman filter

$$\mathbf{u}_1^a = \mathbf{u}_1^f + \mathbf{K}_1(\mathbf{d} - \mathbf{H}\mathbf{u}_1^f)$$

is equivalent to the analyzed state $\mathbf{u}^a$ obtained by the exact Kalman filter (4).

For general nonlinear system of equations with possibly non-Gaussian noise, it is reasonable to assume that as the order of the gPC approximation N and the ensemble size M increase, the approximation error should decay and the gPC–EnSRF algorithms would converge. Rigorous analysis of such convergence and error estimate are beyond the scope of this paper and will be reported in future work. Again we emphasize the convergence here refers to the convergence of the gPC–EnSRF (or EnSRF) solutions to the Kalman filter solutions, not to the true states.

4. Numerical examples

In this section we provide numerical examples to examine the numerical properties and efficiency of the gPC-based EnSRF methods. The first example is a nonlinear scalar equation with a univariate random input; the second one is a linear scalar equation with a multivariate random input; and the third one is the Lorenz equations, a nonlinear system with a multivariate random input. In all examples the modeling noise is in the initial conditions and is Gaussian, and we adopt the stochastic collocation based algorithm, with the Hermite polynomials as the gPC basis. The focus is on the convergence and accuracy of the methods. Throughout this section, we consider "error" as the difference between the numerical results produced by the EnSRF or gPC EnSRF and the "exact" solution of the Kalman filter (if available). Therefore, the discrepancy between the assimilation result and the "true" state, which is often dominated by the linear Gaussian assumption made in the Kalman filter, is not considered.

4.1. Nonlinear population equation

Here we consider the following population equation:

$$\frac{du^f}{dt} = -r\left(1 - \frac{u^f}{A}\right)u^f, \quad u^f(0) = u_0, \tag{49}$$

where r and A are positive real parameters. The solution of (49) is sensitive to the initial values. If $u_0^f > A$, the solution will grow exponentially; if $0 < u_0^f < A$, the solution will converge to 0.

We fix $r = 1$ and $A = 2$, and consider the solution in the time interval $t \in [0, 1]$. A true state (unavailable to the simulation) is constructed by adding a Wiener process with a Gaussian distribution of $0.2 \times \mathcal{N}(0, t)$ to the solution of (49) with initial condition $u_0 = 2.1$. Measurements are then made every $\Delta T = 0.1$ time unit on the true state, with the measurement error following $\mathcal{N}(0, 0.1^2)$. The behavior of the gPC–EnSRF can be seen in Fig. 1. The analyzed state (dash-dotted line) can quickly deviate from the true state (solid line). However, when observation data (circles) arrives, the analyzed state can track the true state much more closely. This simulation is conducted by a gPC expansion of eighth-order ($N = 8$), with $Q = 10$ Hermite quadrature points in the stochastic collocation and $M = 10^5$ realizations in the analysis step.

Next we examine the convergence properties of gPC–EnSRF. We employ a "well-resolved" simulation result, based on a tenth-order gPC expansion, $N = 10$, with $Q = 20$ quadrature points in the collocation scheme and $M = 10^6$ ensemble realizations in the analysis step, and consider it as the "exact" solution. We then compare the error convergence of the numerical results obtained with lower resolution. In Fig. 2(a), the error convergence with respect to the order of gPC expansion (N) is shown, while the other parameters (Q and M) are fixed at the well-resolved level. The fast convergence of error, in fact exponential convergence, can be clearly observed. In Fig. 2(b), the error convergence with respect to the number of quadrature points (Q) in the gPC collocation is shown, with N and M fixed at the well-resolved level. Again we observe very fast con-

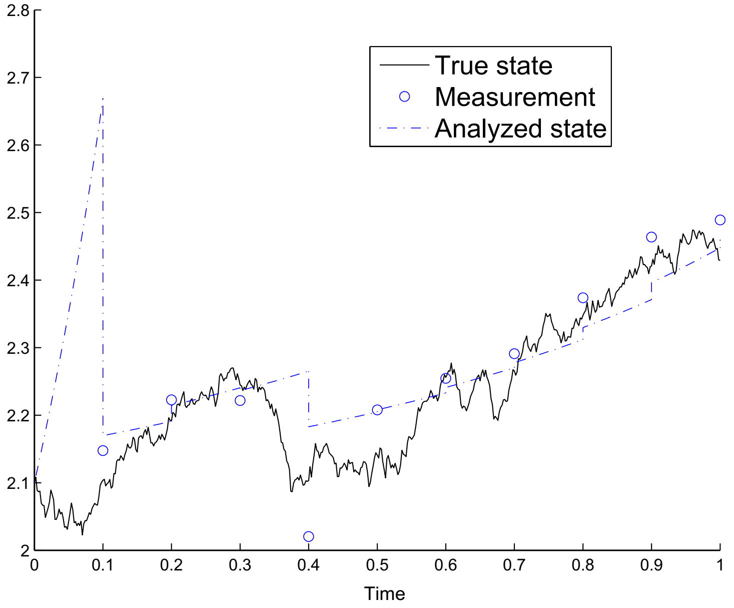

Fig. 1. Dynamic performance of gPC–EnSRF.

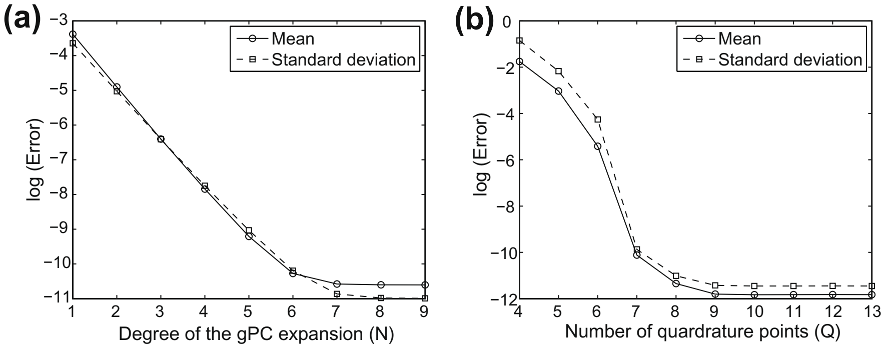

Fig. 2. Error convergence of gPC–EnSRF (a) with respect to the gPC expansion order (N); (b) with respect to the number (Q) of quadrature points in gPC collocation.

vergence. From the results it is clear that the problem can be fully resolved with $N = 8$ and $Q = 10$ (lower than the resolution used for our well-resolved exact solution).

The performance comparison between the gPC–EnSRF and the traditional EnSRF is in Fig. 3, where the ensemble size of the traditional EnSRF is varied from 10^2 to 10^6. While the gPC–EnSRF employs $M = 10^6$ ensemble in the analysis step, the number of simulations for the model equation – the effective ensemble size for simulations – is the number of quadrature points Q. The computational gain, both in accuracy and efficiency, can be clearly seen from Fig. 3 – with about $Q = 10$ simulations the gPC–EnSRF is more accurate (by about six orders in accuracy) than the traditional EnSRF with 10^6 simulations. We emphasize again that the accuracy improvement is made by reducing the sampling errors. The new methods does not improve the error caused by the inherent linear Gaussian assumption made by the Kalman filter for the nonlinear systems.

4.2. Advection equation

Here we consider the model problem used in [9,10], a one-dimensional linear advection model

$$\frac{\partial u^f}{\partial t} + c\frac{\partial u^f}{\partial x} = 0, \quad x \in [0, L], \quad t > 0, \tag{50}$$

where the length of the domain is $L = 100$ with periodic boundary condition and the advection speed is $c = 1$. The grid spacing is $\Delta x = 1$. A true state u^t is sampled from a Gaussian distribution, $\mathcal{N}$, with zero mean, unit variance, and a spatial de-correlation length of 10. This results in 10 i.i.d. Gaussian random variables and a random space of 10 dimension, i.e., $z \in \mathbb{R}^n$ with

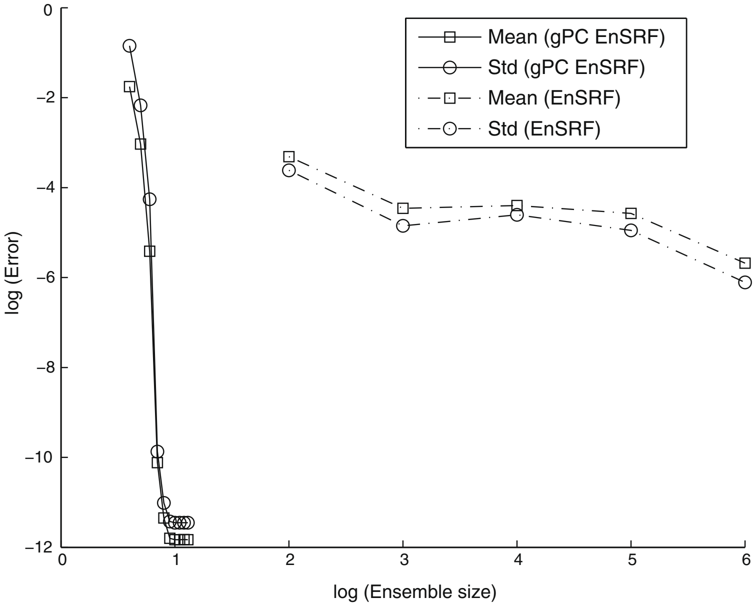

Fig. 3. Comparison of error convergence of gPC–EnSRF and standard EnSRF.

$n = 10$. Compared to [9,10], the length of the domain and dimensionality of the random space are smaller, in order to facilitate our simulations for convergence study.

The first guess solution is generated by drawing another sample from $\mathcal{N}$ and adding this to the true state. The initial ensemble is then generated by adding samples drawn from $\mathcal{N}$ to the first guess solution. Thus, the initial state has an error variance of one. Four measurements of the true solution, distributed evenly in the spatial domain, are assimilated every one time unit, i.e., $\Delta T = 1$, with observation errors of zero mean and standard deviation of 0.1.

According to Theorem 1, for this linear problem with Gaussian noise, the first-order gPC KF method is exact. Therefore, we fix the gPC order at $N = 1$ and use a set of sparse grid Hermite cubature points with second degree accuracy from [15] for the gPC coefficients evaluations. The number of cubature points is $Q = 21$. The qualitative behavior of the gPC–EnSRF is shown in Fig. 4, where the ensemble size at the analysis step is $M = 10^5$. As expected, the mean of the gPC–EnSRF estimates converge

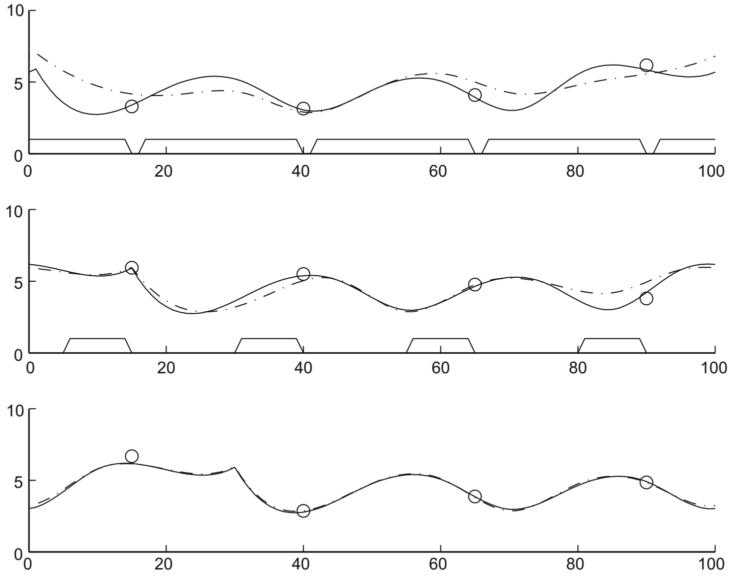

Fig. 4. Results of the gPC–EnSRF to the model problem (50) at three different times $t = 1$ (top figure), $t = 15$ (middle figure), and $t = 30$ (bottom figure). Solid lines are the true state, circles are the measurements, and dashed lines are the mean of the gPC–EnSRF estimates. Another set of solid lines near the bottom of each figure are the standard deviations of the gPC–EnSRF estimate.

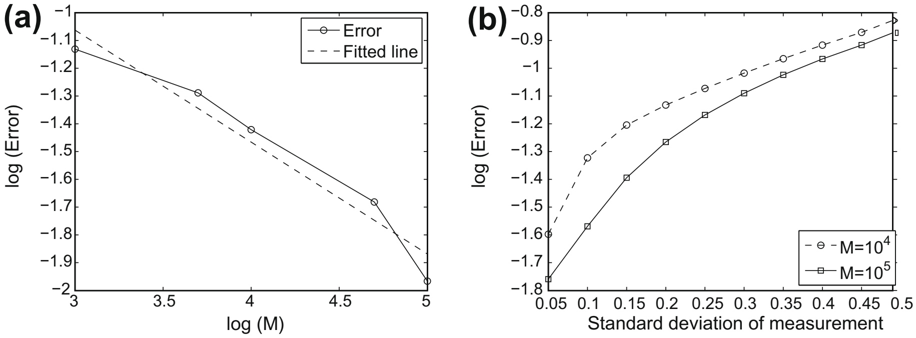

Fig. 5. Error convergence of gPC–EnSRF (a) with respect to ensemble size M; (b) with respect to the standard deviation of the measurement.

to the true state as time evolves, and the standard deviation of the estimates converges to the standard deviation of the measurements, which is 0.1 and visually indistinguishable in the bottom figure.

Since there is no need to refine the gPC order and cubature accuracy for this linear problem, we examine the error behavior of the gPC–EnSRF with respect to the ensemble size (M) at the analysis step and the level of measurement noise. Here error is defined as the difference between the exact KF estimates (available for this linear problem) and the numerical estimates obtained by the gPC–EnSRF. In Fig. 5(a) the error convergence at $t = 30$ with respect to the ensemble size M can be seen clearly. The slope of convergence is approximately -0.4 which is consistent with the rate of convergence of the traditional Monte Carlo sampling (-0.5). Again it is worth noting that the increase of the ensemble size M is achieved in the step of evaluating the gPC polynomial expression (40) and does not involve more simulations of the state equations. Hence increasing the ensemble size does not increase the computational effort of the gPC–EnSRF in a noticeable way. In Fig. 5(b), we observe that the error increases as the standard deviation of the measurement noise increases, and the dependency is almost linear. This is consistent with the error analysis of the classical EnKF ([23]).

4.3. Lorenz equations

A well-known example of a strongly nonlinear system is the Lorenz model, which has been intensively studied in the data assimilation community. See, for example, [7,24,25,27]. For certain values of parameters, this system exhibits chaotic behavior in the sense that very small perturbation in the initial values will lead to completely different trajectories. The system of Lorenz equations are

$$\frac{dx}{dt} = \sigma(y - x), \tag{51}$$

$$\frac{dy}{dt} = \rho x - y - xz, \tag{52}$$

$$\frac{dz}{dt} = xy - \beta z, \tag{53}$$

with the coefficients chosen as $\sigma = 10, \rho = 28$, and $\beta = 8/3$, and the initial condition

$$(x_0, y_0, z_0) = (1.508870, -1.531271, 25.46091). \tag{54}$$

These values have been employed extensively in the literature. The trajectories of the solution are shown in Fig. 6, along with another set of trajectories obtained by perturbing the initial condition of x by 0.001. The two sets of trajectories become completely different as the time evolves.

Here we use the following setting in our gPC assimilation.

- The system of equations are integrated for $t \in [0, 20]$ by the fourth-order Runge–Kutta method with a time step $\Delta t = 0.005$.
- A set of true states are constructed by perturbing the solutions of the system with the initial condition (54) by three independent Wiener processes with a distribution $0.1 \times \mathcal{N}(0, t)$. Measurements are made on all three components of the true states at intervals of $\Delta T = 0.05$ (every 10 integration steps) with independent measurement errors following a distribution $\mathcal{N}(0, 0.1^2)$.
- The gPC forecast model is the system with the random initial condition

  $$(x_0^f, y_0^f, z_0^f) = (x_0, y_0, z_0) + (Z_1, Z_2, Z_3),$$

  where $Z \sim N(0, \mathbf{I}_3)$ are i.i.d. Gaussian.

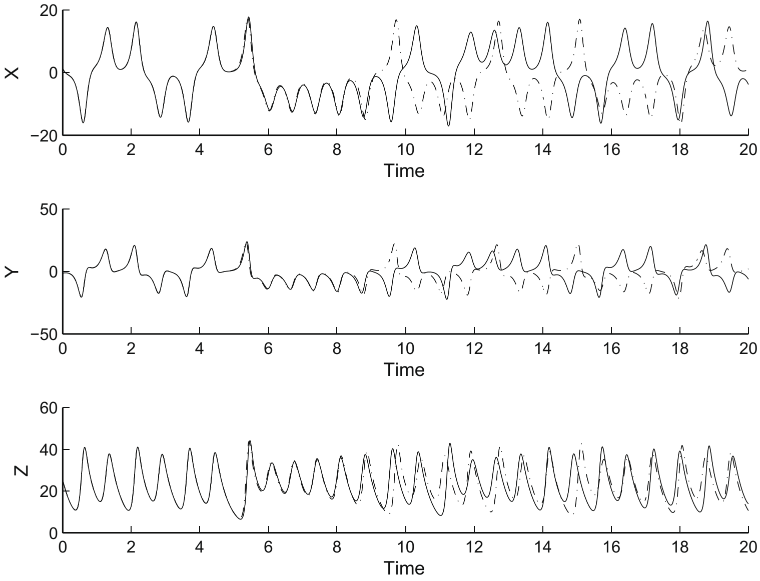

Fig. 6. Two different sets of trajectories of the Lorenz system with small deviation in the initial condition – a difference of 0.001 in the initial condition of x.

- The gPC–EnSRF employs the third-order Hermite polynomials $(N = 3)$, $Q = 5^3$ tensor product of the one-dimensional Hermite quadrature nodes, and $M = 10^4$ ensemble realizations in the analysis step. Therefore, the computational cost is $Q = 125$ number of simulations of the corresponding deterministic system.

The general behavior of gPC–EnSRF is illustrated in Fig. 7, where two set of curves are present. One is the true state and the other is the numerical estimate, and the two almost coincide with each other.

With a lack of the exact solution of the Kalman filter to the Lorenz system, we examine the errors in term of the difference between the assimilation results and the true states, in a qualitative manner, by following the existing studies on data assimilation of the Lorenz system. Let $\Delta X = x_{est} - x_{true}$ be the difference in the x variable between the numerical estimate x_{est} and the true state x_{true}. Similarly we define ΔY and ΔZ as the differences in y and z variables, respectively. The time evolution of the L^2 norm of the differences $(\Delta X, \Delta Y, \Delta Z)$ is shown in Fig. 8, with the dotted line obtained by the second-order gPC filter

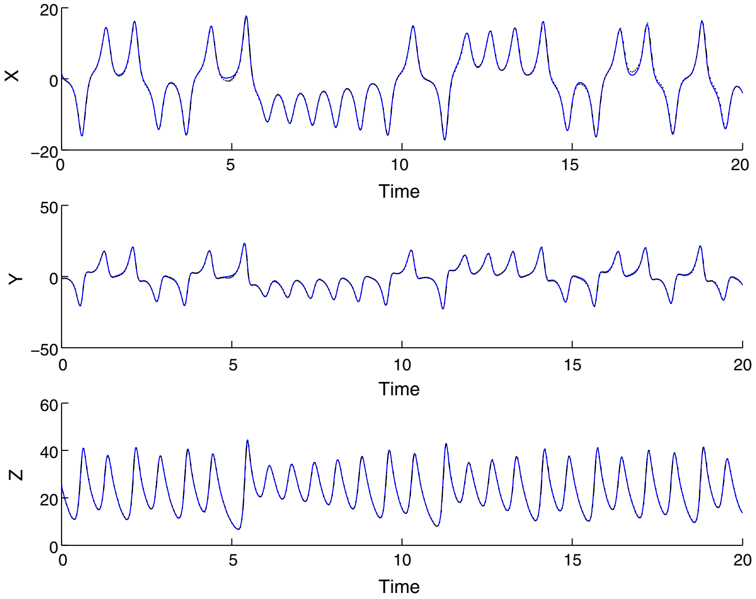

Fig. 7. Time evolution of the gPC–EnSRF estimates for the Lorenz system.

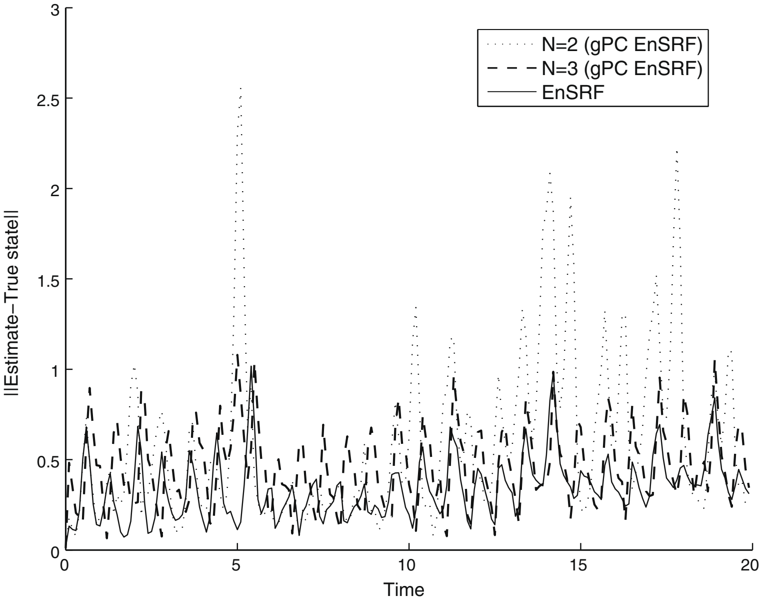

Fig. 8. L^2 norm of the difference between the estimate of gPC–EnSRF and the true states.

($N = 2$), dashed line by the third-order gPC filter ($N = 3$), and the solid line by the traditional EnSRF with 10^4 realizations. The convergence from the second-order gPC filter to the third-order is obvious. While the third-order gPC filter produces a very similar result to that of the EnSRF, it is much more efficient than the traditional EnSRF, as the simulation cost ratio is roughly 125 versus 10^4. At this stage, all results have converged, though not to the true states due to the errors in observation and the errors induced by the linear Gaussian assumption made in the Kalman filter, with the latter likely to be dominant for this nonlinear system.

5. Conclusion

In this paper we proposed a set of efficient ensemble Kalman filter algorithms based on generalized polynomial chaos (gPC) expansion. The algorithms employ gPC-based numerical methods, either a stochastic Galerkin or a stochastic collocation method, to solve the forecast problem with high accuracy and efficiency, then utilize the gPC expansion to generate arbitrarily large ensemble realizations, without incurring notable computational cost, to obtain the analyzed state estimates in the subsequent ensemble Kalman filter step. This naturally leads to significantly reduced sampling errors which is the main source of numerical errors in traditional ensemble Kalman filter methods. When combined with the ensemble square-root filter (EnSRF), the gPC–EnSRF algorithms can also eliminate the sampling errors associated with perturbing the measurement. The detailed algorithms were presented, and numerical examples were provided to demonstrate the efficiency of the algorithms. Also, the collocation based gPC–EnSRF can be extended to highly nonlinear and complex systems in a straightforward manner (at least on a conceptual level). Rigorous accuracy analysis, e.g. convergence rate, of the gPC-based algorithms and their applications to more complex systems are being pursued and will be reported in future work.

Appendix A

Proof of Theorem 1

Theorem 1. *Assume the forecast system of equations is linear*

$$\frac{d\mathbf{u}^f}{dt} = \mathbf{A}(t)\mathbf{u}^f + \mathbf{g}(t), \tag{55}$$

$$\mathbf{u}^f(t_0) = \mathbf{u}_0(Z), \tag{56}$$

where the initial condition $\mathbf{u}_0(Z)$ has a Gaussian distribution, and measurement (3) also has a Gaussian distribution. Let $\mathbf{u}_1^f(t)$ be the first-order gPC Galerkin solution, using Hermite polynomials, to (55) and (56), and $\mathbf{K}_1$ be the corresponding gPC Kalman gain matrix defined in (36). Then the analyzed state obtained by the first-order gPC Kalman filter

$$\mathbf{u}_1^a = \mathbf{u}_1^f + \mathbf{K}_1(\mathbf{d} - \mathbf{H}\mathbf{u}_1^f) \tag{57}$$

is equivalent to the analyzed state $\mathbf{u}^a$ obtained by the exact Kalman filter (4).

Proof. The general solution of (55) is

$$\mathbf{u}^f(t) = \mathbf{B}(t, t_0; \mathbf{A})\mathbf{u}_0 + \mathbf{C}(t, t_0; \mathbf{A}, \mathbf{g}), \tag{58}$$

where

$$\mathbf{B}(t, t_0; \mathbf{A}) = \mathbf{S}(t)\mathbf{S}^{-1}(t_0), \tag{59}$$

$$\mathbf{C}(t, t_0; \mathbf{A}, \mathbf{g}) = \mathbf{S}(t)\int_{t_0}^{t} \mathbf{S}^{-1}(s)\mathbf{g}(s)ds, \tag{60}$$

where $\mathbf{S}(t)$ is the fundamental matrix of the corresponding homogeneous equation of (55). It is obvious the forecast solution $\mathbf{u}^f(t)$ remains Gaussian, so does the analyzed solution $\mathbf{u}^a(t)$ following Gaussian assumption on the measurement noise.

Again, let $t_1 < t_2 < \cdots$ be the time instances when data arrive and assimilation is made. It suffices to prove the theorem for any interval from t_{n-1} to $t_n, n > 1$. Let $\mathbf{u}^a(t_{n-1})$ be the analyzed solution at t_{n-1} with mean $\bar{\mathbf{u}}^a(t_{n-1})$ and covariance function $\mathbf{P}^a(t_{n-1})$.

In Kalman filter, (55) is first solved from t_{n-1} to t_n with initial condition $\mathbf{u}^a(t_{n-1})$, and the forecast state is

$$\mathbf{u}^f(t_n) = \mathbf{B}(t_n, t_{n-1}; \mathbf{A})\mathbf{u}^a(t_{n-1}) + \mathbf{C}(t_n, t_{n-1}; \mathbf{A}, \mathbf{g}). \tag{61}$$

Therefore, $\mathbf{u}^f(t_n)$ follows Gaussian distribution with mean $\mathbf{B}\bar{\mathbf{u}}^a(t_{n-1}) + \mathbf{C}$ and covariance function

$$\mathbf{P}^f(t_n) = \mathbf{B}\mathbf{P}^a(t_{n-1})\mathbf{B}^T. \tag{62}$$

In the stochastic Galerkin based gPC–EnSRF, to solve (55) from t_{n-1} to t_n we first need to project the initial condition $\mathbf{u}^a(t_{n-1})$ by a set of gPC basis. Under the Gaussian assumption, the initial value can be represented by a first-order gPC expansion as follows:

$$\mathbf{u}^a(t_{n-1}) = \bar{\mathbf{u}}^a(t_{n-1}) + \mathbf{Q}\mathbf{z} \tag{63}$$

where $\mathbf{Q} \in \mathbb{R}^{m\times m}$ is the Cholesky decomposition of $\mathbf{P}^a(t_{n-1})$ satisfying $\mathbf{P}^a(t_{n-1}) = \mathbf{Q}\mathbf{Q}^T$, and $\mathbf{z} = (Z_1, \ldots, Z_m) \sim N(0, \mathbf{I}_m)$ is a Gaussian vector of length m whose components have zero mean and unit variance and are mutually independent.

The obvious basis polynomials in this case are the Hermite polynomials [13,?]. A straightforward application of the stochastic Galerkin procedure reveals the first-order expansion is sufficient, i.e., the coefficients of higher order terms are zero.

$$\mathbf{u}_1^f(t, \mathbf{z}) = \sum_{|\mathbf{i}|\leqslant 1} \hat{\mathbf{v}}_{\mathbf{i}}^f(t)\Phi_{\mathbf{i}}(\mathbf{z}) = \hat{\mathbf{v}}_0^f(t) + \sum_{k=1}^{m} \hat{\mathbf{v}}_k^f(t)Z_k, \tag{64}$$

where the expansion coefficients satisfy

$$\begin{aligned} &\frac{d\hat{\mathbf{v}}_0^f}{dt} = \mathbf{A}\hat{\mathbf{v}}_0^f + \mathbf{g}(t), \quad \hat{\mathbf{v}}_0^f(t_{n-1}) = \bar{\mathbf{u}}^a(t_{n-1}) \\ &\frac{d\hat{\mathbf{v}}_k^f}{dt} = \mathbf{A}\hat{\mathbf{v}}_k^f, \quad \hat{\mathbf{v}}_k^f(t_{n-1}) = \mathbf{q}_k, \quad 1 \leqslant k \leqslant m, \end{aligned} \tag{65}$$

where $\mathbf{q}_k$ is the kth column of matrix $\mathbf{Q}$. Following (58), the solutions to the above system are

$$\begin{aligned} &\hat{\mathbf{v}}_0^f(t_n) = \mathbf{B}\bar{\mathbf{u}}^a(t_{n-1}) + \mathbf{C} \\ &\hat{\mathbf{v}}_k^f(t_n) = \mathbf{B}\mathbf{q}_k, \quad 1 \leqslant k \leqslant m. \end{aligned}$$

By substituting the solution back into the first-order Hermite expansion (64), we obtain

$$\mathbf{u}_1^f(t_n) = (\mathbf{B}\bar{\mathbf{u}}^a(t_{n-1}) + \mathbf{C}) + \sum_{k=1}^{m} \mathbf{B}\mathbf{q}_k Z_k = (\mathbf{B}\bar{\mathbf{u}}^a(t_{n-1}) + \mathbf{C}) + \mathbf{B}\mathbf{Q}\mathbf{z}. \tag{66}$$

Therefore, the first-order gPC Galerkin solution $\mathbf{u}_1^f(t_n)$ follows Gaussian distribution with mean $(\mathbf{B}\bar{\mathbf{u}}^a(t_{n-1}) + \mathbf{C})$ and covariance function

$$\mathbf{P}_1^f = (\mathbf{B}\mathbf{Q})\mathbf{I}_m(\mathbf{B}\mathbf{Q})^T = \mathbf{B}\mathbf{P}^a(t_{n-1})\mathbf{B}^T,$$

which are the same as those of $\mathbf{u}^f(t_n)$. Since both $\mathbf{u}^f(t_n)$ and $\mathbf{u}_1^f(t_n)$ are Gaussian, we have $\mathbf{u}_1^f(t_n) = \mathbf{u}^f(t_n)$. Subsequently, the first-order gPC Galerkin method will produce the gPC Kalman gain matrix $\mathbf{K}_1$ from (36) that is the same as the exact Kalman gain matrix (5), and the analyzed solution $\mathbf{u}_1^a(t_n)$ from (57) will be the same as $\mathbf{u}^a(t_n)$ obtained by the exact Kalman filter (4), with both following the same Gaussian distribution. This completes the proof. □

References

[1] J.L. Anderson, An ensemble adjustment Kalman filter for data assimilation, Mon. Weather Rev. 129 (2001) 2884–2903.

[2] C.H. Bishop, B.J. Etherto, S.J. Majumdar, Adaptive sampling with the ensemble transform Kalman filter, part i: theoretical aspects, Mon. Weather Rev. 129 (2001) 420–436.

[3] G. Burgers, P.V. Leeuwen, G. Evensen, Analysis scheme in the ensemble Kalman filter, Mon. Weather Rev. 126 (1998) 1719–1724.

[4] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods in Fluid Dynamics, Springer-Verlag, Berlin/Heidelberg, 1988.

[5] S.E. Cohn, An introduction to estimation theory, J. Meteorol. Soc. Jpn. 75 (1997) 257–288.

[6] G. Evensen, Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, J. Geophys. Res. 99 (1994) 10143–10162.

[7] G. Evensen, An ensemble Kalman smoother for nonlinear dynamics, Mon. Weather Rev. 128 (2000) 1852–1867.

[8] G. Evensen, The ensemble Kalman filter: theoretical formulation and practical implementation, Ocean Dyn. 53 (2003) 343–367.

[9] G. Evensen, Sampling strategies and square root analysis schemes for the EnKF, Ocean Dyn. 54 (2004) 539–560.

[10] G. Evensen, Data Assimilation, The Ensemble Kalman Filter, Springer-Verlag, Berlin, 2007.

[11] A. Gelb, Applied Optimal Estimation, MIT Press, Cambridge, 1974.

[12] R.G. Ghanem, Ingredients for a general purpose stochastic finite element formulation, Comput. Meth. Appl. Mech. Eng. 168 (1999) 19–34.

[13] R.G. Ghanem, P. Spanos, Stochastic Finite Elements: A Spectral Approach, Springer-Verlag, 1991.

[14] D. Gottlieb, S.A. Orszag, Numerical Analysis of Spectral Methods: Theory and Applications, CBMS-NSF, SIAM, Philadelphia, PA, 1977.

[15] F. Heiss, V. Winschel, Estimation with numerical integration on sparse grids. Discussion Papers in Economics 916, University of Munich, Department of Economics, 2006.

[16] J.S. Hesthaven, S. Gottlieb, D. Gottlieb, Spectral Methods for Time-Dependent Problems, Cambridge University Press, 2007.

[17] P.L. Houtekamer, H.L. Mitchell, Data assimilation using an ensemble Kalman filter technique, Mon. Weather Rev. 126 (1998) 796–811.

[18] K. Ito, K. Xiong, Gaussian filters for nonlinear filtering problems, IEEE Trans. Automatic Control 45 (5) (2000) 910–927.

[19] A.H. Jazwinski, Stochastic Processes and Filtering Theory, Academic Press, San Diego, CA, 1970.

[20] S.J. Julier, J.K. Uhlmann, A new extension of the Kalman filter to nonlinear systems, in: The Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Orlando, FL, USA, 1997.

[21] S.J. Julier, J.K. Uhlmann, Unscented filter and nonlinear estimation, Proc. IEEE 92 (3) (2004) 401–422.

[22] R. Kalman, R. Bucy, New results in linear prediction and filter theory, Trans. AMSE J. Basic Eng. 83D (1961) 85–108.

[23] J. Li, D. Xiu, On numerical properties of the ensemble Kalman filter for data assimilation, Comput. Meth. Appl. Math. Eng. 197 (2008) 3574–3583.

[24] D.T. Pham, Stochastic methods for sequential data assimilation in strongly nonlinear systems, Mon. Weather Rev. 129 (2001) 1194–1207.

[25] M. Verlaan, A. Heemink, Non-linearity in data assimilation applications: a practical method for analysis, Mon. Weather Rev. 129 (2001) 1578–1589.

[26] J.S. Whitaker, T.M. Hamill, Ensemble data assimilation without perturbed observations, Mon. Weather Rev. 130 (2002) 1913–1924.

[27] X. Xiong, I.M. Navon, B. Uzunoglu, A note on the particle filter with posterior Gaussian resampling, Tellus 58 (2006) 456–460.

[28] D. Xiu, Efficient collocational approach for parametric uncertainty analysis, Commun. Comput. Phys. 2 (2) (2007) 293–309.

[29] D. Xiu, Fast numerical methods for stochastic computations: a review, Commun. Comput. Phys. 5 (2008) 242–272.

[30] D. Xiu, J.S. Hesthaven, High-order collocation methods for differential equations with random inputs, SIAM J. Sci. Comput. 27 (3) (2005) 1118–1139.

[31] D. Xiu, G.E. Karniadakis, The Wiener–Askey polynomial chaos for stochastic differential equations, SIAM J. Sci. Comput. 24 (2) (2002) 619–644.