



Long-time uncertainty propagation using generalized polynomial chaos and flow map composition



Dirk M. Luchtenburg ^{a,*}, Steven L. Brunton ^c, Clarence W. Rowley ^b

^a Department of Mechanical Engineering, The Cooper Union for the Advancement of Science and Art, NY 10003, United States

^b Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ 08544, United States

^c Department of Applied Mathematics, University of Washington, Seattle, WA 98195, United States

ARTICLE INFO

Article history:

Received 10 June 2013

Received in revised form 3 May 2014

Accepted 16 June 2014

Available online 25 June 2014

Keywords:

Uncertainty quantification

Long-time integration

Generalized polynomial chaos

Flow map composition

ABSTRACT

We present an efficient and accurate method for long-time uncertainty propagation in dynamical systems. Uncertain initial conditions and parameters are both addressed. The method approximates the intermediate short-time flow maps by spectral polynomial bases, as in the generalized polynomial chaos (gPC) method, and uses flow map composition to construct the long-time flow map. In contrast to the gPC method, this approach has spectral error convergence for both short and long integration times.

The short-time flow map is characterized by small stretching and folding of the associated trajectories and hence can be well represented by a relatively low-degree basis. The composition of these low-degree polynomial bases then accurately describes the uncertainty behavior for long integration times. The key to the method is that the degree of the resulting polynomial approximation increases exponentially in the number of time intervals, while the number of polynomial coefficients either remains constant (for an autonomous system) or increases linearly in the number of time intervals (for a non-autonomous system). The findings are illustrated on several numerical examples including a nonlinear ordinary differential equation (ODE) with an uncertain initial condition, a linear ODE with an uncertain model parameter, and a two-dimensional, non-autonomous double gyre flow.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

Uncertainty is an inherent feature in many complex systems ranging from simulations and experiments to natural systems such as the climate. There are a number of ways in which uncertainty enters a physical problem, including initial conditions, boundary conditions, model structure, and parameters of the model. We often describe these uncertainties statistically (e.g., mean, variance, etc.), and the goal is to quantify the statistical effects of these uncertainties on the evolution of the system at some time in the future.

In this work, we concentrate on uncertainty in initial conditions and parameters of a dynamical system, which may describe a physical process or the advection of particles (e.g., contaminants, chemicals, etc.) through the induced velocity field of a PDE. In particular, we focus on systems that can be represented by stochastic ordinary differential equations (ODEs) of the form:

* Corresponding author.

E-mail address: dluchten@cooper.edu (D.M. Luchtenburg).

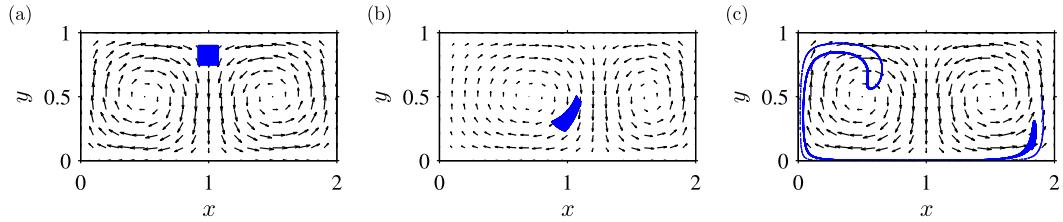


Fig. 1. Propagation of initially uniformly distributed particles through the double gyre flow: (a) initial distribution at $t = 0$, (b) distribution at $t = 2$, and (c) distribution at $t = 20$. See Section 5.3 for governing equations ($A = 0.1$, $\omega = 2\pi/10$, $\epsilon = 0.25$).

$$\frac{dx(t; Z)}{dt} = F(t, x; Z), \quad (1)$$

where $x \in \mathbb{R}^n$ is the state, $Z \in \mathbb{R}^d$ is a random vector representing uncertain parameters, and $F : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^d \rightarrow \mathbb{R}^n$ is a smooth function. F need only be Lipschitz continuous in x for existence and uniqueness of solutions; however, F is often twice continuously differentiable, as is the case when F is a velocity field that is a solution of the Navier–Stokes equations. Given an initial condition x_0 (possibly uncertain as well), we seek to obtain the solution $x(t; Z)$ as a function of time and the uncertain variables Z .

Understanding how uncertainty propagates through a dynamical system is an important engineering problem, and it is at the focus of significant current research efforts. However, propagating a probability density function (PDF) through a long-time, nonlinear dynamical system is a challenging task due to the stretching and folding of associated trajectories. Even simple fluid velocity fields may give rise to extremely complicated, chaotic particle trajectories in the induced velocity field [1]. Fig. 1 illustrates the propagation of a uniform distribution of particles through the (deterministic) dynamical system of a so-called double gyre. For short-time integration, the distribution is still well approximated by a uniform distribution; however, for longer integration time, the distribution becomes stretched and folded by the nonlinear dynamics. We observe that the mean and variance are no longer sufficient to describe the long-time density. The main idea explored in this paper is a simple one: we represent highly nonlinear deformations (as in Fig. 1(c)) using the composition of short-time flow maps, for which the deformation is not too great, and relatively easy to approximate (as in Fig. 1(b)). A popular approach to obtain statistical information is Monte Carlo simulation: sample known distributions of the uncertain parameters to obtain statistics or density estimates. However, accurate estimates often imply large sample sizes and each sample requires a solution of the dynamical system. This makes Monte Carlo methods computationally expensive. An alternative method is to employ spectral expansions to describe the dependence on the random variables. The method of polynomial chaos (PC) was introduced by Wiener [2] and used Hermite polynomials to expand the solution involving Gaussian random processes. The spectral expansion in stochastic random variables was pioneered by Ghanem and Spanos [3] in the context of finite elements for solid mechanics. Xiu and Karniadakis [4] later pointed out that the PC approach can be generalized to include non-Gaussian processes, resulting in generalized polynomial chaos (gPC).

Generalized polynomial chaos (gPC) is a method for solving stochastic differential equations, such as (1), based on Galerkin projections with an appropriately selected basis [4]. The so-called chaos basis is selected from the Askey scheme of orthogonal polynomials such that the weighting function is the same as the PDF of the associated random variable. This choice results in exponential convergence of the solution for relatively short integration times and also in cases where the statistics are constrained [4–7]. However, in general, gPC has nonuniform convergence when there is a strong nonlinear dependence on random parameters. Hence gPC tends to break down for long-time integration since nonlinearities prevail.

A possible solution to address the nonuniform convergence of the gPC method is to adjust the degree of the chaos expansion as the PDF changes in time. In [8], a multi-element polynomial chaos method (ME-gPC) was developed, which adaptively decomposes the space of random inputs into multiple elements and subsequently employs polynomial chaos expansions at element level. Similar techniques have also been employed by [9,10]. Another solution, coined time-dependent gPC (TD-gPC), keeps a global expansion which is adapted in time [11]. The main idea of TD-gPC is to determine new, optimal polynomials for the chaos expansion at a number of discrete instants in time. These new polynomials are based on the stochastic properties of the solution at the particular time level. In [12], a hybrid approach is proposed which combines ME-gPC and TD-gPC.

1.1. Our approach – generalized polynomial chaos and flow map composition

We propose an alternative method to deal with the long-time integration of probability densities, which are poorly represented by gPC. The method approximates the short-time flow map by a spectral basis as in gPC and uses flow map composition to construct the long-time flow map.

Distributions undergo relatively modest deformation during short-time advection through the dynamical system, and so a short-time flow map may be efficiently represented with a low-degree polynomial basis. These short-time maps are then composed via spectral interpolation to obtain the long-time flow map. Flow map composition has been investigated in the autonomous [13] and non-autonomous cases [14]. It was demonstrated that the flow map composition has a number of

favorable properties, which have been leveraged to accurately and efficiently compute time-series of finite-time Lyapunov exponent (FTLE) fields [14].

As time increases, so do the number of flow map compositions, which results in an exponentially increasing polynomial degree, but with relatively low-degree polynomials for each of the local short-time flow maps. The short-time flow maps are approximated over a large domain of interest so that after one up-front computation of each flow map (a small number of collocation points are advected, similar to Monte Carlo) the propagation of PDFs through the system becomes a simple and inexpensive operation. In a sense, the low-degree polynomials comprise a reduced order model of the individual short-time flow maps. These maps may then be leveraged to efficiently propagate *any* PDF through the system with minimal additional cost.

This method has a number of advantages. First, accurate long-time integration of PDFs is achieved because the degree of the resulting polynomial is exponential in the number of compositions (time steps). However, the number of polynomial coefficients needed to represent the composition increases only linearly in the number of compositions (for a non-autonomous system), or remains constant (for an autonomous system). The resulting expansion is thus quite efficient. After the one-time up-front cost of computing these approximate short-time flow maps, *any* PDF may be propagated through the system. There is an additional benefit of having the approximate flow map for use with other dynamical systems methods, such as finite-time Lyapunov exponents (FTLEs) and the Perron–Frobenius operator [15]. Finally, this method does not require significant overhaul of a computational code that integrates the original ODE. In contrast, these flow-map methods are built from individual trajectories obtained through a standard simulation. As in the case of computing FTLEs, pre-computing short-time flow maps allows us to propagate uncertainty at neighboring times without significant overlap in the computation.

Although there are many positive features of the proposed method, there are certain drawbacks that should be clearly stated. The computational cost for a non-autonomous system may be significant, especially if we are only interested in propagating a single PDF through the system. However, in the case when significant stretching occurs, gPC and Monte Carlo will in general be more expensive, even for a single PDF. We refer to Section 6.3 for a comparison of the computational costs of these methods. In addition, the method relies on the existence of slow manifolds or attractors on which the dynamics are restricted, or a large enough computational domain so that composition may be performed. Unlike gPC, the mean and the variance do not appear as polynomial coefficients in our framework, but must be approximated by a subsequent (significantly cheaper) Monte Carlo sampling. However, for long-time integration higher order statistics are generally necessary anyway [7].

In general, this method and gPC both suffer from the so-called “curse of dimensionality,” whereby the number of polynomial basis functions and collocation points becomes prohibitively large for even a moderately high-dimensional uncertain variables. Therefore, we view this method as best applying to physical vector fields of two and three dimensions, such as those induced by fluid velocity fields or gravitational potentials in astrophysics. In addition, these methods assume a global basis so that we are limited to simply connected global domains.

2. Problem formulation

2.1. Governing equation and flow map

Let $\mathcal{D} \subseteq \mathbb{R}^n$ be compact and $F : \mathbb{R} \times \mathcal{D} \times \mathbb{R}^d \rightarrow \mathbb{R}^n$ be a smooth vector field. We consider the ordinary differential equation (ODE)

$$\frac{dx(t; Z)}{dt} = F(t, x; Z), \quad (2)$$

where $Z = (z_1, \dots, z_d)$ are parameters of interest. In the case where $F(t, x; Z) = F(x; Z)$, the ODE is called *autonomous*, otherwise it is called *time-dependent* or *non-autonomous*. A specific solution or trajectory satisfies the ODE and a given initial condition

$$x(t_0; Z) = x_0(Z). \quad (3)$$

We emphasize that $F(t, x; Z)$ and $x_0(Z)$ are functions of different subsets of Z : the first subset contains uncertain parameters in the ODE, and the second one uncertain initial conditions.

Let $\varphi_{t_0}^{t_f} : \mathcal{D} \times \mathbb{R}^d \rightarrow \mathcal{D}$ be the *flow map*, so that $\varphi_{t_0}^{t_f}(x_0; Z)$ is a solution to the ODE (2) with initial condition $x(t_0; Z) = x_0(Z)$ from initial time t_0 to final time t_f . The flow map satisfies

$$\left. \frac{d\varphi_{t_0}^t(x_0; Z)}{dt} \right|_{t=t_f} = F(t_f, x_0; Z), \quad \forall x_0 \in \mathcal{D}, t_f \in \mathbb{R}. \quad (4)$$

Equivalently, we can write the flow map (particle position) according to the integral equation:

$$\varphi_{t_0}^{t_f}(x_0; Z) = x(t_0; Z) + \int_{t_0}^{t_f} F(t, x(t; Z); Z) dt. \quad (5)$$

We note that $\varphi_{t_0}^{t_f}$ satisfies the following properties:

$$\varphi_{t_0}^{t_0} = \text{id}, \quad (6a)$$

$$\varphi_{t_0}^{t_2} = \varphi_{t_1}^{t_2} \circ \varphi_{t_0}^{t_1}. \quad (6b)$$

The latter property shows that a “long-time” flow map can be composed using “short-time” flow maps.

In the special case of an autonomous dynamical system, we use the notation $\varphi^{\Delta t}$ to represent the flow map of duration Δt (equivalent to $\varphi_{t_0}^{t_0+\Delta t}$, $\forall t_0 \in \mathbb{R}$).

2.2. Probabilistic framework

As in [16], we will adopt a probabilistic framework and model the parameters of interest $Z = (z_1, \dots, z_d)$ as a vector of d independent random variables in a properly defined probability space $(\Omega, \mathcal{F}, \mathcal{P})$. Here, Ω is the sample space, which is equipped with a σ -algebra \mathcal{F} and probability measure \mathcal{P} . We will only consider continuous random variables.

Let $\rho_i : \Gamma_i \rightarrow \mathbb{R}^+$ be the probability density function (PDF) of the random variable $z_i(\omega)$, $\omega \in \Omega$, whose image is $\Gamma_i = z_i(\Omega) \subseteq \mathbb{R}$ for $i = 1, \dots, d$. Then

$$\rho(Z) = \prod_{i=1}^d \rho_i(z_i), \quad (7)$$

is the joint probability density of the random vector $Z = (z_1, \dots, z_d)$ with the support

$$\Gamma = \prod_{i=1}^d \Gamma_i \subseteq \mathbb{R}^d. \quad (8)$$

The random space Γ is d -dimensional, and the governing equation (2) should be valid for all $Z \in \Gamma$.

3. Generalized polynomial chaos

3.1. Univariate expansion

Let us consider a solution $x(t; Z)$ of the initial value problem (2), (3) which depends on a *single* random variable Z . The PDF of Z is denoted $\rho(Z)$ and has support $\Gamma \subseteq \mathbb{R}$. The generalized polynomial chaos (gPC) expansion approximates the solution x via orthogonal polynomials of the random variable Z . The key observation in gPC is that the optimal expansion polynomials are orthogonal with respect to the PDF as weighting function. More precisely, we select a set of orthogonal polynomials $\{\psi_i(Z)\}$, satisfying

$$\langle \psi_i, \psi_j \rangle = \int_{\Gamma} \psi_i(Z) \psi_j(Z) \rho(Z) dZ = h_i \delta_{ij}, \quad (9)$$

where δ_{ij} is the Kronecker delta function and h_i is a normalization factor:

$$h_i = \int_{\Gamma} \rho(Z) \psi_i^2(Z) dZ.$$

(In the following we will assume that the basis functions are normalized, i.e. $h_i = 1$.) Thus, if Z is a uniformly distributed variable in $[-1, 1]$, its PDF is a constant and (9) defines the Legendre polynomials. For a normally distributed random variable, the Gaussian PDF defines the Hermite polynomials. For other cases and extensions we refer to [4,11].

The degree- P gPC expansion of x in Z is obtained by projection onto the selected basis

$$\mathbb{P}x(t; Z) = \sum_{i=0}^P \hat{x}_i^e(t) \psi_i(Z), \quad (10)$$

where \mathbb{P} is the projector onto $\text{span}\{\psi_i(Z)\}_{i=0}^P$, and \hat{x}_i^e are the *exact* expansion coefficients defined by

$$\hat{x}_i^e(t) = \langle x, \psi_i \rangle = \int_{\Gamma} x(t; Z) \psi_i(Z) \rho(Z) dZ, \quad i = 0, \dots, P. \quad (11)$$

Note that the exact expansion coefficients \hat{x}_i^e rely on the exact solution $x(t; Z)$.

The convergence of gPC expansions like (10) has been recently studied by [17]. In particular, conditions are presented for mean-square convergence.

3.2. Multivariate expansion

Here, we are interested in the more general situation of a *multidimensional* probability space, i.e., $Z = (z_1, \dots, z_d)$, $d > 1$ and $\rho(Z) = \prod_{i=1}^d \rho_i(z_i)$, where $z_i \in \Gamma_i$. Therefore, we define multivariate basis functions for the gPC expansion of the solution $x(t; Z)$. The multivariate basis functions are constructed as products of (univariate) orthogonal polynomials. In particular, given d families $\{\psi_{i_k}^{(k)}(z_k)\}_{k=1}^d$ of univariate polynomials,¹ then the d -variate polynomial basis functions are defined as

$$\Psi_i(Z) = \prod_{k=1}^d \psi_{i_k}^{(k)}(z_k), \quad i_k \leq P_k; Z \in \Gamma, \quad (12)$$

where $i = (i_1, \dots, i_d)$ is a multi-index with $|i| = i_1 + \dots + i_d$. Note that the maximum polynomial degree of a basis function of family k is P_k , and consequently the maximum degree of all multivariate polynomials is $P_{\max} = \sum_{k=1}^d P_k$. In gPC, usually, all expansion polynomials up to a maximum degree P are included, i.e., $|i| \leq P$. The number of basis functions Ψ_i is thus

$$N = \binom{d+P}{d}. \quad (13)$$

The multivariate basis polynomials satisfy the orthogonality relation

$$\langle \Psi_i, \Psi_j \rangle = \int_{\Gamma} \Psi_i(Z) \Psi_j(Z) \rho(Z) dZ = \Delta_{ij}, \quad (14)$$

where $\Delta_{ij} = \prod_{k=1}^d \delta_{i_k j_k}$. As in (10), the degree- P multivariate gPC expansion of x in Z is obtained by projection onto the appropriate basis

$$\mathbb{P}x(t; Z) = \sum_{|i|=0}^P \hat{x}_i^e(t) \Psi_i(Z), \quad (15)$$

but now i is a multi-index, \mathbb{P} is the projector onto $\text{span}\{\Psi_i(Z)\}_{|i|=0}^P$, and \hat{x}_i^e are the exact expansion coefficients defined by

$$\hat{x}_i^e(t) = \langle x, \Psi_i \rangle = \int_{\Gamma} x(t; Z) \Psi_i(Z) \rho(Z) dZ, \quad \forall |i| = 0, \dots, P. \quad (16)$$

Note that when $d = 1$, we obtain a univariate gPC expansion as described in Section 3.1.

3.3. Non-intrusive techniques

In practice, we cannot employ (11) or (16) to find the expansion coefficients since it requires knowledge of the *exact* solution. However, we can compute approximations of those integrals using numerical quadrature methods. This approach is commonly referred to as non-intrusive since it does not require modification of the numerical solver; it requires only sampling of the deterministic solution at the quadrature nodes. When the number of random variables d is small, a multidimensional quadrature can be applied, for instance using tensor products of one-dimensional quadrature rules. Sparse grids become advantageous as the number of random variables increases. In high dimensions, Monte Carlo sampling is preferred since the convergence rate is independent of the number of random variables.

Let the *approximate* gPC expansion be given by

$$\tilde{x}(t; Z) = \sum_{|i|=0}^P \hat{x}_i(t) \Psi_i(Z). \quad (17)$$

Then the coefficients are determined as

$$\hat{x}_i(t) = \sum_{k=1}^K x(t; Z^{(k)}) \Psi_i(Z^{(k)}) \alpha^{(k)}, \quad \forall |i| = 0, \dots, P, \quad (18)$$

where $\{Z^{(k)}, \alpha^{(k)}\}$ are a set of quadrature nodes and weights, and $x(t; Z^{(k)})$ is the *deterministic* solution of (2) with fixed $Z^{(k)}$. A Gaussian quadrature rule yields an exact result for polynomials of degree $2K - 1$ or less by a suitable choice of the nodes and weights. This quadrature rule is chosen such that the approximate gPC expansion interpolates the solution at the nodes, and hence is a (pseudo-spectral) collocation method. For a Monte Carlo method the nodes over the domain Γ are randomly drawn from the probability distribution ρ , and the weights are $1/K$.

¹ Superscript (k) indicates the type of polynomial basis, and subscript i_k the degree of the corresponding basis function.

3.4. Statistical information

Statistical information can be readily obtained when a gPC approximation (17) is available. The approximate mean is obtained as

$$\mathbb{E}[x] \approx \tilde{\mu}(t) = \mathbb{E}[\tilde{x}] = \int_{\Gamma} \sum_{|\mathbf{i}|=0}^P \hat{x}_i(t) \Psi_i(Z) \rho(Z) dZ = \langle \tilde{x}, \Psi_0 \rangle = \hat{x}_0(t), \quad (19)$$

where we used orthogonality of the basis functions. The variance is approximated by

$$\mathbb{E}[(x - \mu)^2] \approx \tilde{\sigma}(t) = \mathbb{E}[(\tilde{x} - \tilde{\mu})^2] = \sum_{|\mathbf{i}|=1}^P \hat{x}_i^2(t). \quad (20)$$

4. Short-time flow map approximation and composition

Consider the flow map (5). We are interested in approximating the flow map for relatively short integration times, i.e. $\Delta t = t_f - t_0$ is small. This is motivated by the fact that short-time flow maps are characterized by small stretching and folding of the associated trajectories, and hence can be represented by relatively low-order bases. Using the property (6b), we compose these low-dimensional bases to obtain the flow map for long integration times.

4.1. Short-time flow map approximation

Similarly to the approximate gPC expansion (17), we approximate the short-time flow map $\varphi_{t_0}^{t_0+\Delta t}(x; Z)$ by a spectral expansion. The dependencies on both x and Z are described by orthogonal polynomials, i.e.

$$\varphi_{t_0}^{t_0+\Delta t}(x; Z) \approx \varphi_{t_0}^{t_0+\Delta t}(x; Z) = \sum_{|\mathbf{i}|=0}^Q \sum_{|\mathbf{j}|=0}^P \hat{\varphi}_{t_0,ij}^{t_0+\Delta t} \Theta_i(x) \Psi_j(Z), \quad (21)$$

where $\Psi_j(Z)$ is defined by (12) and likewise

$$\Theta_i(x) = \prod_{l=1}^n \theta_{i_l}^{(l)}(x_l), \quad i_l \leq Q_l; x = (x_1, \dots, x_n) \in \mathcal{D} \subseteq \mathbb{R}^n, \quad (22)$$

where $\theta_{i_l}^{(l)}$ are univariate orthonormal polynomials, with $Q = \sum_{k=1}^n Q_k$. We note that polynomial order of the multivariate basis function may differ in each coordinate direction x_l as denoted by the use of Q_l . The expansion coefficients are computed using a collocation method, as discussed in Section 3.3. Thus, the flow map approximation is obtained in two steps:

1. The deterministic solution is computed at the quadrature nodes, i.e., at particular values of Z and x ;
2. The expansion coefficients are determined by numerical quadrature as in (18).

The approximate expansion is then given by (21).

4.2. Flow map composition

We use flow map composition to compute long-time flow maps $\varphi_{t_0}^{t_0+T}$ with duration T by composition of a number of intermediate short-time flow maps $\varphi_{t_0}^{t_0+\Delta t}$. For $t_0 = 0$ and $T = M\Delta t$, we may use equally spaced intermediate short-time maps with duration Δt :

$$\varphi_0^{M\Delta t} = \varphi_{(M-1)\Delta t}^{M\Delta t} \circ \dots \circ \varphi_{\Delta t}^{2\Delta t} \circ \varphi_0^{\Delta t}. \quad (23)$$

In practice, we know the flow map only on some discrete set of points, $X_0 \subset \mathcal{D}$, so we must use an interpolation scheme to go between the various flow maps. In [14] linear interpolation was used on a high-resolution grid of particles. This work differs from linear interpolation by introducing a spectral approximation of the short-time flow map, given by (21), which requires propagation of a comparatively small number of collocation points. Then, neighboring flow maps are composed by spectral interpolation. This procedure is straightforward since the short-time flow maps are polynomial vector functions.

Since memory and speed are both important computational issues, one wishes to propagate as few particles as possible. The method of flow map composition was originally used to speed up the computation of a time-series of flow maps by re-using intermediate flow maps to eliminate redundant particle integrations. For example, $\varphi_{t_0}^{t_0+M\Delta t}$ and $\varphi_{t_0+\Delta t}^{t_0+(M+1)\Delta t}$ have a significant overlap. It is observed that the short-time flow map, does not stretch the flow significantly and may

be accurately represented by a relatively small number of basis polynomials, as above. Moreover, favorable error properties allow us to compose many such maps and obtain an accurate description of highly nonlinear, stretched flows over the entire domain. Therefore, it is possible to use many fewer points when representing each $\varphi_{k\Delta t}^{(k+1)\Delta t}$, $k = 0, \dots, M - 1$, compared with $\varphi_0^{M\Delta t}$. In some sense, the spectral approximations are reduced-order models for each short-time flow map. An immediate advantage is that we are able to rapidly pass any distribution through the flow map, and it is possible to efficiently compute the long-time flow map at neighboring times without redundant computation.

4.3. Composition on various computational domains

All of the example dynamical systems in this work are defined on compact domains \mathcal{D} , so that $\varphi_0^t(\mathcal{D}) \subseteq \mathcal{D}$ for all t . There are several computational issues that arise when applying flow map composition on non-compact domains. In particular, it may be difficult or impossible to define a domain of interest \mathcal{D}_i so that $\varphi_0^t(\mathcal{D}_i) \subseteq \mathcal{D}_i$. Instead, we define a larger domain \mathcal{D}' so that $\mathcal{D}_i \subseteq \mathcal{D}'$ and, $\varphi_0^t(\mathcal{D}_i) \subseteq \mathcal{D}'$ for time t of interest.

This generally involves defining a domain of interest and a larger containing domain where short-time flow maps are computed. Then, the smaller domain of interest is passed through each short-time map using interpolation, without the need for extrapolation.

This procedure has been successfully applied to two-dimensional convective flows, such as the fluid flow past a pitching airfoil [14]. In this case, a larger computational domain is used to ensure that particles starting on a smaller domain of interest remain in the larger domain for the entire long-time flow map. As is often the case in free convective flows, the fluid flow may be uninteresting or unquantifiable sufficiently far upstream or downstream of the domain of interest. In this case, it is possible to apply an analytic outflow condition, significantly reducing the necessary size of the outer domain.

4.4. Stability and error analysis

Stability and error analyses are notoriously challenging for the long-time behavior of trajectories in a chaotic dynamical system. Long-time stretching and folding of trajectories through a chaotic dynamical system is often confounded by the exponential growth of errors. Traditional local error analyses, such as those used to assess the accuracy scaling of integration schemes, may yield misleading or counterintuitive results. The exponential growth of even the smallest errors, on the order of machine precision, will eventually grow to an order determined by the largest scale of the chaotic attractor.

Fortunately, the finite-time exponential separation observed in chaotic systems is mediated by regions of large finite-time Lyapunov exponent (FTLE), which often occur as ridges. The FTLE field represents a finite-time sensitive of trajectories to initial disturbances or errors. It was shown in [14] that flow map composition benefits from the favorable property that particles flee from ridges of the FTLE field in forward time (which is in fact a property of the FTLE field), and therefore, they suffer less from the dominant exponential separation and error sensitivity.

It is extremely difficult to rigorously quantify the benefit of the aforementioned property in an error analysis of a time-varying system, since the unsteady nature of the chaotic attractor remains a challenging open problem in dynamical systems. It is our understanding that current and ongoing research on Lagrangian coherent structures and the FTLE field are at the frontier in answering these challenging questions. However, the phase flow method of Ying and Candès [13] (autonomous version of flow map composition) includes a rigorous error analysis for the autonomous flow map composition. This work provides an upper bound for the error, but it does not consider the chaotic structure of stretching within the flow field, which may result in significantly tighter error bounds. In the autonomous case, additional computational savings may be achieved by composing short-time flow maps in powers of two.

A detailed error analysis of the non-autonomous flow map composition in chaotically mixing vector fields is the subject of ongoing work. In particular, building on the error results in [13] to include the particle shedding properties of FTLE ridges would be an important theoretical step in the development of computational methods for chaotic advection. However, this is beyond the scope of this work.

5. Results and discussion

We present three examples that illustrate the benefits of using spectral approximations of short-time flow maps and composing them for long-time uncertainty propagation. The first example is a nonlinear autonomous ordinary differential equation (ODE), which highlights the connection between the flow map and the gPC framework in case of an uncertain initial condition. The flow map composition is shown to be accurate and efficient although the solution evolves into the discontinuous Heaviside function and is not well captured by traditional polynomial expansions. Our second example is a simple linear autonomous ODE with an uncertain model parameter. It highlights the challenges that arise when using gPC for long-time integration, and it elucidates the inner-workings of our approach. The last example is the so-called double gyre flow [18,19], which is defined by a nonlinear time-dependent ODE system. This example is more involved and we illustrate how to deal with time dependency. In addition, we exploit the fact that we have low-dimensional expansions of the short-time flow maps to efficiently propagate multiple probability density functions through the flow for long times. Computational cost is addressed at the end of this section.

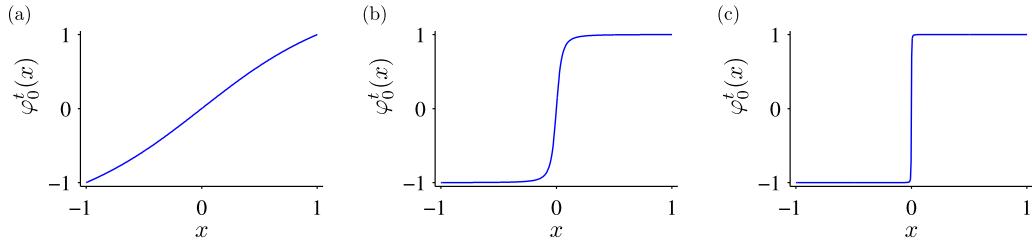


Fig. 2. Nonlinear ODE – Flow map (25) for $t = 0.2$ (a), $t = 3.0$ (b), and $t = 6.0$ (c).

5.1. Nonlinear autonomous ordinary differential equation (nonlinear ODE)

In this first example, we consider a nonlinear autonomous ODE with an uncertain initial condition. We start out with the deterministic equation and consider the flow map over a range of initial conditions. We then model this range of initial conditions with an uncertain variable Z and illustrate how polynomial chaos is related to the flow map.

Let us consider the ODE

$$\frac{dx(t)}{dt} = x(t)(1 - x^2(t)). \quad (24)$$

This ODE has three fixed points: two stable ones at $x = \pm 1$, and an unstable one at $x = 0$. We are interested in all solution curves that start on the interval $[-1, 1]$; i.e., we wish to solve (24) for all initial conditions $x(0) = x_0$, $x_0 \in [-1, 1]$. This family of curves is given by the flow map $\varphi_0^t : [-1, 1] \rightarrow [-1, 1]$, and satisfies

$$\frac{d\varphi^t(x)}{dt} = \varphi^t(x)(1 - (\varphi_0^t(x))^2), \quad (25)$$

for all $x \in [-1, 1]$ and $t \geq 0$. The flow map is shown in Fig. 2 for $t = 0.2, 3.0, 6.0$. Note that, for long integration times, the flow map converges to a step function.

5.1.1. Generalized polynomial chaos interpretation

Alternatively, we can interpret the flow map of (24) in the polynomial chaos framework. As before, consider the initial value problem

$$\frac{dx(t)}{dt} = x(t)(1 - x^2(t)), \quad x(0) = Z, \quad (26)$$

but now the initial condition is a uniformly distributed random variable between -1 and 1 , which we denote $Z \sim \mathcal{U}(-1, 1)$. More precisely, we have the following formulation:

$$\frac{dx(t, Z)}{dt} = x(t, Z)(1 - x^2(t, Z)), \quad x(0) = Z \sim \mathcal{U}(-1, 1). \quad (27)$$

Thus, argument Z of the solution $x(t, Z)$ plays the role of argument x of the flow map $\varphi_0^t(x)$ (compare with (25)). The appropriate gPC basis is given by the Legendre polynomials, as described in Section 3.1. Hence, we express the degree- P approximation of the solution as

$$\tilde{x}(t, Z) = \sum_{i=0}^P \hat{x}_i(t) L_i(Z), \quad (28)$$

where L_i are Legendre polynomials and \hat{x}_i the corresponding coefficients. The coefficients are computed using a collocation method, as described in Section 3.3. For propagation of the collocation points, in this and the following subsections, we use Matlab's fourth-order Runge–Kutta integrator with a relative tolerance of 10^{-13} and absolute tolerance of 10^{-15} . In Figs. 3(a, b), gPC approximations with $P = 20$ are shown for $t = 3.0$ and $t = 6.0$. As time progresses, the exact solution approaches a step function and becomes increasingly difficult to approximate. In Fig. 3(c), we increase the polynomial degree up to $P = 140$. This reduces the error, but the approximation still exhibits an undesired Gibbs phenomenon. The convergence of the relative error at $t = 6.00$ is shown in Fig. 5(a); it can be observed that exponential convergence is not achieved.

5.1.2. Generalized polynomial chaos and flow map composition

Analogous to the gPC expansion (28), we approximate the flow map (25) for a short time Δt by a Legendre expansion. Thus, we write the approximate short-time flow map as

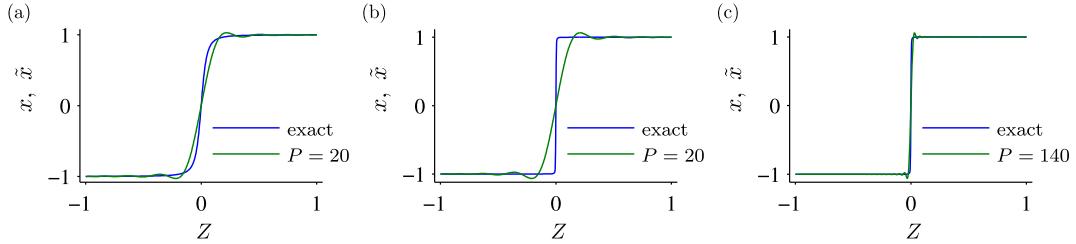


Fig. 3. Nonlinear ODE – Exact solutions and P -th degree gPC approximations of (27) for $t = 3.0$ (a), and $t = 6.0$ (b, c). Note that in (c) the 140th degree approximation exhibits ringing (Gibbs phenomena).

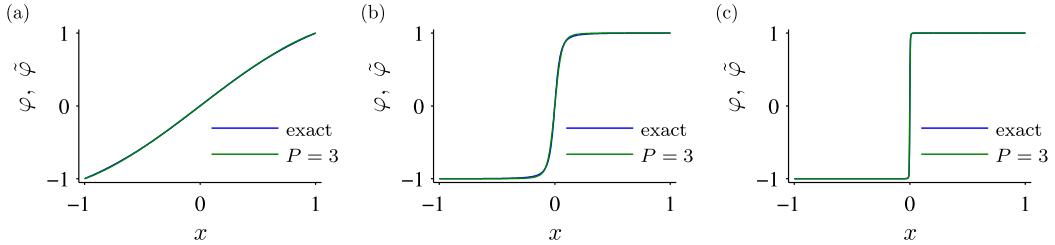


Fig. 4. Nonlinear ODE – Exact flow maps and their approximations. In (a) the short-time flow map for $\Delta t = 0.2$, see (29), is shown. In (b, c) the approximate flow maps are obtained by composition, as in (30), for $T = 3$ and $T = 6.0$ respectively.

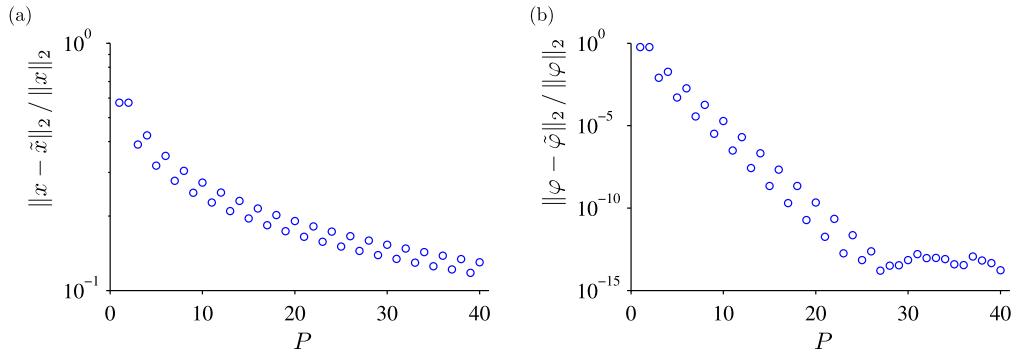


Fig. 5. Nonlinear ODE – L_2 norm of the relative error for increasing order P at $t = 6.0$ for: (a) the gPC expansion (28), and (b) the flow map obtained by composition of the short-time flow map (30). Note the spectral convergence of flow map composition in (b).

$$\tilde{\varphi}^{\Delta t}(x) = \sum_{i=0}^P (\hat{\varphi}^{\Delta t})_i L_i(x). \quad (29)$$

The approximate long-time flow map at $T = M\Delta t$ can now be obtained by composing this short-time flow map with itself M times:

$$\tilde{\varphi}^T(x) = \underbrace{(\tilde{\varphi}^{\Delta t} \circ \tilde{\varphi}^{\Delta t} \circ \cdots \circ \tilde{\varphi}^{\Delta t})(x)}_{M \text{ times}}. \quad (30)$$

The system is autonomous, so we use the notation $\varphi^{\Delta t}$ and drop the dependence on initial time t_0 . Note that the degree of this long-time flow map is P^M ; at each time step Δt the degree increases by a factor P . Hence, *an increase of the integration time will automatically result in an increasing polynomial degree*. In Fig. 4(a), the degree-3 short-time flow map approximation for $\Delta t = 0.2$ is shown. In Figs. 4(b, c), this short-time flow map is composed to get the flow map at $t = 15\Delta t = 3.0$ and $t = 30\Delta t = 6.0$ respectively. We observe that the composition approach outperforms the gPC method (compare with Fig. 3). Furthermore, we obtain exponential convergence of the error at all times, as shown in Fig. 5. In particular, in Fig. 5(b) it can be seen that the relative error at $t = 6.0$ decays exponentially for increasing order P of the short-time flow map expansion, whereas the gPC expansion in Fig. 5(a) does not.

5.2. Linear autonomous ordinary differential equation with an uncertain model coefficient (uncertain linear ODE)

In our second example, we consider a linear ODE with an uncertain parameter; this example has been used to illustrate the gPC method and its convergence breakdown for long term integration [4,11]. The model is given by a linear autonomous ODE

$$\frac{dx}{dt} = -kx, \quad x(0) = 1, \quad (31)$$

where the coefficient k is uniformly distributed between 0 and 1, i.e., $k \sim \mathcal{U}(0, 1)$. Defining $k = \frac{1}{2}(1 + Z)$, where $Z \sim \mathcal{U}(-1, 1)$, we write

$$\frac{dx(t, Z)}{dt} = -k(Z)x(t, Z), \quad x(0) = 1. \quad (32)$$

The stochastic solution to this initial value problem is

$$x(t, Z) = e^{-k(Z)t}. \quad (33)$$

The flow map is given by

$$\varphi_0^t(x; Z) = e^{-k(Z)t}x. \quad (34)$$

Note that the flow map is linear in x (linear ODE) and nonlinear in the random coefficient k . The mean and the variance of the stochastic solution are known exactly:

$$\mu(t) = \frac{1 - e^{-t}}{t}, \quad (35a)$$

$$\sigma(t) = \frac{1 - e^{-2t}}{2t} - \left(\frac{1 - e^t}{t}\right)^2. \quad (35b)$$

The time-dependent PDF of the stochastic solution is

$$f_x(t, x) = \frac{1}{xt}, \quad e^{-t}x_0 \leq x \leq x_0. \quad (36)$$

For future reference we define the following relative errors for the mean (expected value) and variance:

$$\epsilon_{\text{mean}}(t) = \left| \frac{\tilde{\mu}(t) - \mu(t)}{\mu(t)} \right|, \quad \epsilon_{\text{var}}(t) = \left| \frac{\tilde{\sigma}(t) - \sigma(t)}{\sigma(t)} \right|, \quad (37)$$

where μ and σ denote exact values, and $\tilde{\mu}$, $\tilde{\sigma}$ denote approximate values of the mean and variance respectively. As in the previous example, for all numerical integration we use Matlab's fourth-order Runge–Kutta integrator with a relative tolerance of 10^{-13} and absolute tolerance of 10^{-15} .

5.2.1. Generalized polynomial chaos

Since the coefficient k is uniformly distributed, we approximate the solution of (32) as an expansion in Legendre polynomials:

$$\tilde{x}(t, Z) = \sum_{i=0}^P \hat{x}_i(t) L_i(Z), \quad (38)$$

where the coefficients \hat{x}_i are again computed using collocation.

The approximate mean and variance are easily obtained by exploiting orthogonality of the gPC basis functions (see Section 3.4):

$$\tilde{\mu}(t) = \hat{x}_0(t), \quad \tilde{\sigma}(t) = \sum_{i=1}^P \hat{x}_i^2(t). \quad (39)$$

In Fig. 6 the mean and variance of the 4th-degree gPC expansion are compared with the exact values given by (35). The gPC solution agrees only for short integration times. In particular, the variance of the gPC solution diverges after about 15 time units. This observation is highlighted in Fig. 7, which shows the relative mean and variance errors given by (37). For long integration times the error quickly grows to order $O(1)$. This effect was summarized nicely in [11]: "This poor behavior can be somewhat alleviated by increasing the expansion order. However, for a fixed polynomial degree P , the error levels will become definitely unacceptable after some time. Hence, continuing to increase the end-time will require an ever-increasing polynomial degree, which is not feasible in practice." In the next section, we will employ flow map composition to effectively increase the polynomial degree.

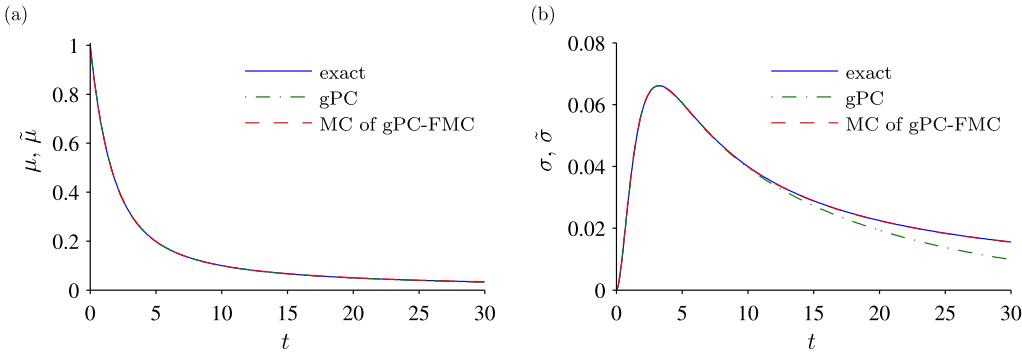


Fig. 6. Uncertain linear ODE – Evolution of the mean (a) and variance (b). In each subfigure the exact values are compared with the results of the 4th degree gPC expansion, and Monte Carlo sampling of the 4th degree gPC-FMC expansion. The exact mean and the approximations are indistinguishable. The exact variance and MC sampling of gPC-FMC are also in good agreement, whereas the variance obtained by gPC is incorrect (compare with Fig. 7).

5.2.2. Generalized polynomial chaos and flow map composition

In this section, we focus on an approximation of the flow map (34). First, we approximate the parametrized flow map (32) for a short integration time Δt by the expansion:

$$\tilde{\varphi}^{\Delta t}(x; Z) = \sum_{i=0}^1 \sum_{j=0}^3 \hat{\varphi}_{ij} L_i(x) L_j(Z), \quad (40)$$

where the basis functions are Legendre polynomials. Note that the flow map is linear in x and cubic in Z , resulting in a total degree $P = 4$. The coefficients $\hat{\varphi}_{ij}$ are computed using stochastic collocation as detailed in Section 3.3. Thus, computation of $\hat{\varphi}_{ij}$ requires the propagation of $2 \times 4 = 8$ collocation points. We also remark that only one short-time flow map needs to be computed since the flow map is autonomous. Starting with the deterministic initial condition $x_0 = 1$, the approximate solution of (32) at discrete times $t_i = i\Delta t$, $i = 0, 1, 2, 3, \dots$, is composed as

$$x_0 \rightarrow \underbrace{\tilde{\varphi}^{\Delta t}(x_0; Z)}_{=x_1(Z)} \rightarrow \underbrace{\tilde{\varphi}^{\Delta t}(x_1(Z); Z)}_{=x_2(Z)} \rightarrow \underbrace{\tilde{\varphi}^{\Delta t}(x_2(Z); Z)}_{=x_3(Z)} \rightarrow \dots \quad (41)$$

Note that, as before, the polynomial degree is increased by a factor P at each time step. This sequence also illustrates that, after one time step, the uncertain model parameter introduces an *uncertain initial condition*.

5.2.3. Monte Carlo sampling

A drawback of the expansion (41) is that the mean and variance cannot be computed as in gPC, since we cannot exploit the orthogonality of basis functions as in the derivation of (39). In this section, we employ Monte Carlo sampling to compute the mean and variance of (41). While using Monte Carlo sampling may at first seem contrary to the very nature of gPC, it has been used in the context of gPC-based ensemble Kalman filters to reduce sampling errors at reduced computational cost [20]. Sampling of the spectral expansion is also employed to estimate probability density functions [21], and to determine the statistics of the solution for inputs having different statistics [7]. For the sequence x_i defined by (41), the mean and variance are approximated as

$$\tilde{\mu}(t_i) = \frac{1}{K} \sum_{k=1}^K x_i(Z^{(k)}), \quad (42a)$$

$$\tilde{\sigma}(t_i) = \frac{1}{K} \sum_{k=1}^K x_i^2(Z^{(k)}) - \tilde{\mu}^2(t_i), \quad (42b)$$

where $\{Z^{(k)}\}$ are N numbers drawn from a uniform distribution over $[-1, 1]$.

For our numerical experiments, we approximate the short-time flow map (40) for a time interval $\Delta t = 0.1$ by a expansion of degree $P = 4$. The long-time flow map is constructed using (41), and the mean and variance are computed as in (42) with $K = 10^6$. We will refer to this approach as Monte Carlo sampling of the polynomial chaos/flow map composition (MC of gPC-FMC). In Fig. 6, this approach is compared with the exact and gPC results. In particular, we observe that the variance obtained by MC sampling of the gPC-FMC method stays close to the exact values, even for long-time integration. In Fig. 7 the relative mean and variance errors are shown. We see that initially, the gPC-FMC approach has larger error than gPC (due to the Monte Carlo sampling), but in the long run the error increases slowly and remains small, while errors for gPC grow to order 1 for longer times. Numerical simulations confirm that $\epsilon_{\text{var},\text{mean}} = O(10^{-4})$ up to $t = 120$. For comparison, the mean and variance were also computed using Monte Carlo simulations (MC) of the exact solution. This also confirms the accuracy

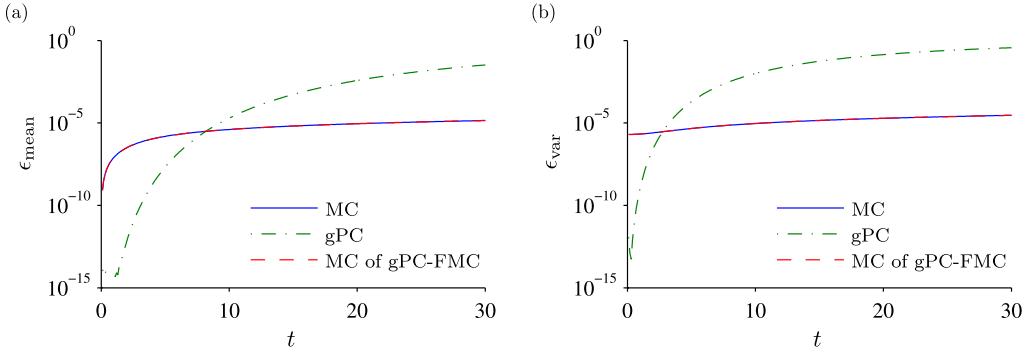


Fig. 7. Uncertain linear ODE – Evolution of the relative mean (a) and variance error (b). In each subfigure the errors for (1) Monte Carlo sampling of the exact solution using 10^6 points (MC), (2) gPC expansion of 4th degree (gPC), and (3) Monte Carlo sampling of the 4th degree gPC-FMC expansion using 10^6 points (MC of gPC-FMC) are shown.

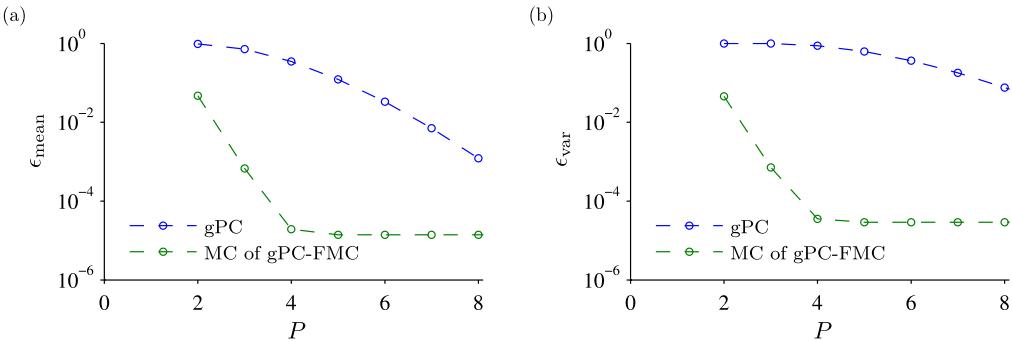


Fig. 8. Uncertain linear ODE – Error convergence of the relative mean (a) and variance errors (b) at $t = 30$, for the gPC expansion (39), and Monte Carlo sampling of (41).

of the gPC-FMC approach since the results are nearly indistinguishable. Fig. 8 shows the error convergence of the mean and the variance at $t = 30$, as the polynomial degree P increases. The gPC method converges very slowly, while both the mean and variance obtained by Monte Carlo sampling of the gPC-FMC expansion converge exponentially. The saturation at $P = 4$ is due to the number of Monte Carlo sampling points ($K = 10^6$); more sampling points result in a lower saturation level. This observation is consistent with the results in Fig. 7 since the MC and MC of gPC-FMC curves are nearly identical.

5.2.4. Numerical quadrature

In this section, we employ a quadrature rule to compute the mean and variance of (41), in place of the Monte Carlo sampling used in the previous section. In particular, we compute

$$\tilde{\mu}(t_i) = \sum_{k=1}^K x_i(Z^{(k)}) \alpha^{(k)}, \quad (43a)$$

$$\tilde{\sigma}(t_i) = \sum_{k=1}^K x_i^2(Z^{(k)}) \alpha^{(k)} - \tilde{\mu}^2(t_i), \quad (43b)$$

where $\{Z^{(k)}, \alpha^{(k)}\}$ are a set of nodes and corresponding weights. In the following we will use Gauss-Legendre quadrature.

We approximate the short-time flow map (40) as in the previous section, with $\Delta t = 0.1$ and $P = 4$. The mean and variance of the long-time flow map are computed using (43) with $K = 10^2$ nodes. This implies that we can exactly compute integrands with polynomial degree up to $P = 2 \cdot 10^2 - 1 = 199$. At $t = 30$ we have composed 300 flow maps, which results in a polynomial of degree 4^{300} . However, such a high degree is not necessary for computation of the mean and variance and the quadrature approximation suffices. In Fig. 9 the relative mean and variance errors are shown. We see that gPC only performs better than gPC-FMC for short integration times ($t < 4$). Comparing with Fig. 9, we also see that numerical quadrature with 10^2 nodes, outperforms MC with 10^6 points. Numerical simulations confirm that $\epsilon_{\text{var},\text{mean}} = O(10^{-7})$ up to $t = 120$. In Fig. 10, the error convergence of the mean and the variance at $t = 1$ and $t = 30$ are shown. The gPC-FMC achieves minimal error for an expansion order $P = 7$ of the short-time flow map for both $t = 1$ and $t = 30$. Note that gPC-FMC converges faster to the exact variance than gPC for both short and long integration times.

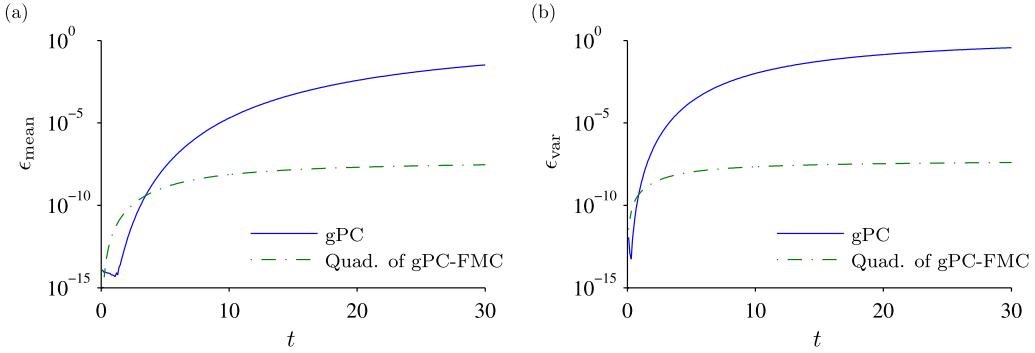


Fig. 9. Uncertain linear ODE – Evolution of the relative mean (a) and variance error (b). In each subfigure the errors for a gPC expansion of 4th degree (gPC), and a the 4th degree gPC-FMC expansion using 10^2 points for numerical quadrature (Quad.) are shown.

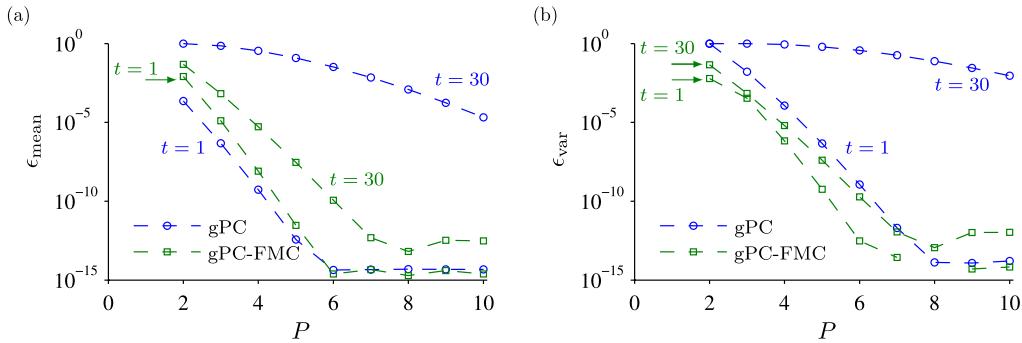


Fig. 10. Uncertain linear ODE – Error convergence of the relative mean (a) and variance errors (b) at $t = 1$ and $t = 30$. Results for the gPC expansion, see (39), and numerical quadrature of (41) using 10^2 nodes, see (43), are shown.

5.3. Nonlinear time-dependent double gyre flow

In this section, we consider a nonlinear time-dependent double gyre flow, which was used in Section 1 (see Fig. 1) to illustrate the significant effect of nonlinearities for long-time integration. The double gyre flow is analytically defined in [18,19] by the stream function

$$\psi(x_1, x_2, t) = A \sin(\pi f(x_1, t)) \sin(\pi x_2), \quad (44)$$

where

$$f(x_1, t) = a(t)x_1^2 + b(t)x_1, \quad (45)$$

$$a(t) = \epsilon \sin(\omega t), \quad (46)$$

$$b(t) = 1 - 2\epsilon \sin(\omega t), \quad (47)$$

on the domain $\mathcal{D} = [0, 2] \times [0, 1]$. The velocity field is obtained by differentiation

$$u = -\frac{\partial \psi}{\partial x_2}, \quad v = \frac{\partial \psi}{\partial x_1}. \quad (48)$$

Thus, using $u = d/dt(x_1)$, $v = d/dt(x_2)$, the governing equations are given by the system of nonlinear ODEs

$$\frac{dx_1}{dt} = -\pi A \sin(\pi f(x_1, t)) \cos(\pi x_2), \quad (49a)$$

$$\frac{dx_2}{dt} = \pi A \cos(\pi f(x_1, t)) \sin(\pi x_2) \frac{df(x_1, t)}{dx_1}. \quad (49b)$$

For $\epsilon = 0$ the ODE is autonomous, and for $\epsilon \neq 0$ it is *time-dependent* and the gyres conversely expand and contract periodically in the x_1 -direction such that the rectangle enclosing the gyres remains invariant.

In the next sections, we will consider the propagation of uniformly distributed particles, and also the propagation of density functions. For all numerical integration we use a fixed-step fourth-order Runge-Kutta method with time-step $h = 0.01$.

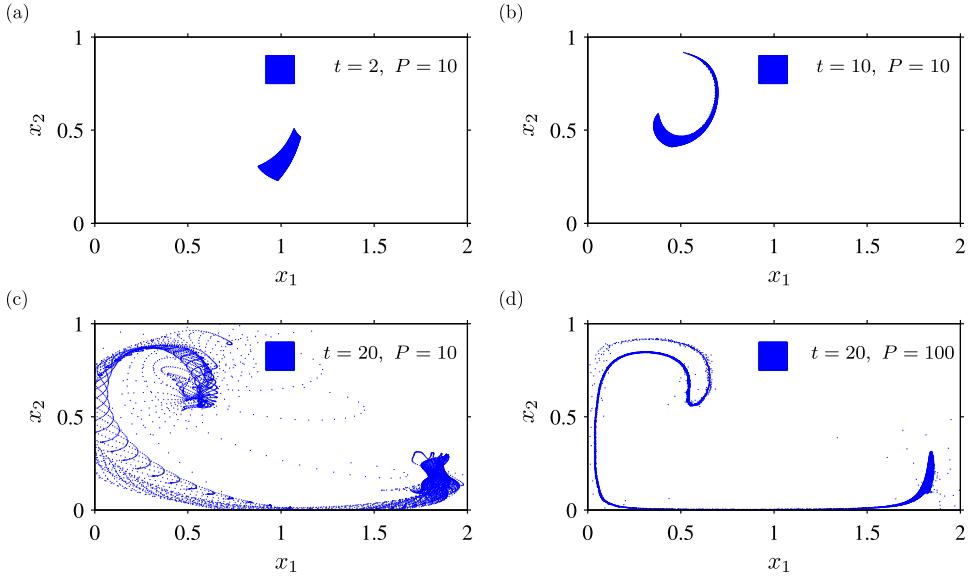


Fig. 11. Double gyre – Propagation of particles using gPC. Each subfigure shows the initial distribution (box) and a deformed distribution at a later time. More specifically, (a), (b), and (c) show the particle distributions at $t = 2$, $t = 10$, and $t = 20$ respectively, that are obtained using a 10th degree gPC expansion, (d) also shows the distribution at $t = 20$, however here a 100th gPC expansion is employed. See Section 5.3 for governing equations ($A = 0.1$, $\omega = 2\pi/10$, $\epsilon = 0.25$).

5.3.1. Generalized polynomial chaos

In this section, we use the gPC method to describe the dependence of the solution of (49) on a uniformly distributed uncertain initial condition. Here, the uncertain initial condition $x(t_0; Z)$ depends on two independent random variables ($Z = (z_1, z_2)$). Thus, the gPC approximation is given by

$$\tilde{x}(t; Z) = \sum_{|i|=1}^P \hat{x}_i(t) L_i(Z), \quad (50)$$

where $i = (i_1, i_2)$ is a multi-index, \hat{x}_i denote expansion coefficients (see Section 3.2), and

$$L_i = L_{i_1}^{(1)}(z_1) L_{i_2}^{(2)}(z_2), \quad i_1 + i_2 \leq P, \quad (51)$$

is a product of Legendre polynomials. The expansion coefficients are computed using a collocation method as described in Section 3.3. The initial condition is specified as

$$x(t_0; Z) = (x_1(t_0; Z), x_2(t_0; Z)) = (a + bz_1, c + dz_2), \quad (52)$$

where a, b are constants such that $0 \leq a \pm b \leq 2$, c, d are constants such that $0 \leq c \pm d \leq 1$, and z_1, z_2 are uniformly distributed on $[-1, 1]$.

For our numerical experiments we set $a = 0.995$, $b = 0.075$, $c = 0.825$, and $d = 0.075$. Thus, the uncertain initial condition is uniformly distributed on a small square as depicted in Fig. 11. We employ a 10th degree gPC expansion ($P = 10$) to propagate particles from this distribution. The gPC expansion approximates the particle distribution at $t = 2$ and $t = 10$ accurately as shown in Figs. 11(a) and (b). However, at later times the gPC approach breaks down, and we obtain incorrect non-smooth particle distributions. This is illustrated in Fig. 11(c), where the non-smooth distribution of particles at $t = 20$ is shown. Increasing the polynomial degree does not alleviate the situation: even for a 100th degree gPC approximation we obtain an inaccurate particle distribution as shown in Fig. 11(d).

5.3.2. Generalized polynomial chaos and flow map composition

In this section, we employ approximations of the short-time flow maps (21), and composition, see (23), to propagate uncertainty. As in Section 5.3.1 we consider uncertain initial conditions. The approximate sequential flow maps for a short integration time Δt are given by (21) and (22), where $t_0 = 0, \dots, (M-1)\Delta t$, $\Psi_j(Z) = 0$ (no uncertain model parameters), $n = 2$, and $\theta_{i_1}, \theta_{i_2}$ are Legendre polynomials, which are defined over $[0, 2]$ and $[0, 1]$ respectively. For our numerical example we set $\Delta t = 0.1$, $Q_1 = 10$ and $Q_2 = 5$. As in the second gPC example, we consider an uncertain initial condition (52), which is uniformly distributed on a square as shown in Fig. 11. Composing 100 short-time flow map approximations, we obtain the distribution of uniform particles at $t = 10$ as shown in Fig. 12(a). In contrast to the gPC result in Figs. 11(c) and (d), we observe that the composition of approximate short-time flow maps is able to describe the stretching and folding of the trajectories accurately. The results highlight the fact that composition of spectral approximations of short-time flow maps lead to numerically stable and accurate solutions even in the case of strong nonlinearities.

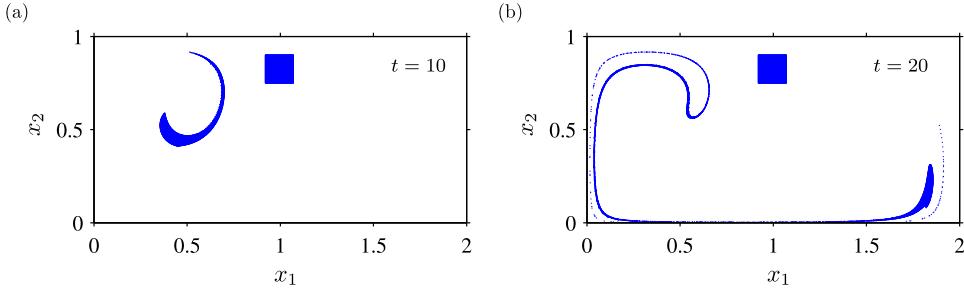


Fig. 12. Double gyre – Propagation of particles using gPC/FMC. Each subfigure shows the initial distribution (box) and a deformed distribution at a later time. More specifically, (a) and (b) shows the particle distribution at $t = 10$, and $t = 20$ respectively, that are obtained using flow map composition; the short-time flow map is computed for $\Delta t = 0.1$ and is of order 10 in the x_1 -direction and 5 in x_2 -direction. See Section 5.3 for governing equations ($A = 0.1$, $\omega = 2\pi/10$, $\epsilon = 0.25$).

5.3.3. Propagation of a PDF

In this section, we use short-time flow map approximations to propagate two different probability density functions (PDFs). Once these approximations are computed, any PDF can be propagated at little extra cost since only composition of the short-time flow maps is required. To simplify our calculations, we use the Perron–Frobenius operator to propagate the PDFs. Let the PDF at some initial time t_0 be given by $\rho_0(x)$. Then the PDF at some later time t_f is

$$\rho(x, t_f) = \rho_0(\varphi_{t_f}^{t_0}(x)) |\det(D\varphi_{t_f}^{t_0}(x))|, \quad (53)$$

where $D\varphi_{t_f}^{t_0}(x)$ is the Jacobian matrix corresponding to the spatial derivatives of the flow map. Since the double gyre flow is incompressible,² we have $|\det(D\varphi_{t_f}^{t_0}(x))| = 1$, and the density is simply

$$\rho(x, t_f) = \rho_0(\varphi_{t_f}^{t_0}(x)). \quad (54)$$

As above, we approximate the short-time flow maps by Legendre polynomials. However, here we need approximations of the reversed short-time flow maps, i.e. $\tilde{\varphi}_{(k+1)\Delta t}^{k\Delta t}(x)$, $k = 0, \dots, M - 1$. Once these approximations are calculated, the PDF at time $t = M\Delta t$ is obtained on a discrete grid of points, $X_0 \in \mathcal{D}$, as follows:

$$X_0 \rightarrow \underbrace{\tilde{\varphi}_{M\Delta t}^{(M-1)\Delta t}(X_0)}_{=X_1} \rightarrow \underbrace{\tilde{\varphi}_{(M-1)\Delta t}^{(M-2)\Delta t}(X_1)}_{=X_2} \rightarrow \dots \rightarrow \underbrace{\tilde{\varphi}_{\Delta t}^0(X_{M-1})}_{=X_M} \quad (55a)$$

$$\rho(X_0, M\Delta t) = \rho_0(X_M). \quad (55b)$$

For our two numerical examples we use the same values as above for the short-time flow map approximations, i.e. $\Delta t = 0.1$, $Q_1 = 10$ and $Q_2 = 5$. In our first example, we consider a Gaussian-like PDF with local support as shown in Fig. 13(a). This PDF is propagated through the double gyre flow, and the time-varying PDF at times $t = 2$, $t = 10$, and $t = 20$ is shown in Figs. 13(b, c). Using the same flow map approximations, we propagate another Gaussian-like PDF with support over the whole domain, see Fig. 14. The PDF at later times $t = 2$, $t = 10$, and $t = 20$ is shown in Figs. 14(b, c).

The evolution of any other PDF can be computed efficiently since it only requires evaluation of the initial PDF at points that are propagated using composition of pre-computed short-time flow maps. In addition, this representation can be employed for fast dynamical systems calculations, such as computation of finite-time Lyapunov exponent (FTLE) fields and almost-invariant sets [14,15,22].

6. Comparison with other methods

It has been demonstrated that gPC-FMC is accurate for long-time uncertainty propagation through a dynamical system, compared to traditional gPC. As mentioned in the introduction, multi-element gPC (ME-gPC) is a way to maintain the accuracy of gPC for long integration times [8]. In this section we compare, for the first two examples in Section 5, the gPC-FMC method with fourth-order Runge–Kutta (RK4) integration and ME-gPC. The gPC method is a spectral method, and ME-gPC is literally a refinement of gPC in the sense that it employs spectral elements (piecewise polynomial basis functions). In the ME-gPC method, the space of random inputs is decomposed into spectral elements when the relative error in variance becomes greater than a threshold value θ . In each random element a gPC expansion of fixed order is employed. For details and definition of the threshold criterion we refer to [8]. In comparing these methods, we focus on a computational efficiency in terms of walltime and error. All algorithms were implemented carefully in Matlab; however our implementation might

² The approximate flow map does not necessarily preserve area. However, we verified numerically that the flow map is nearly incompressible (on average about 0.1% of area is lost).

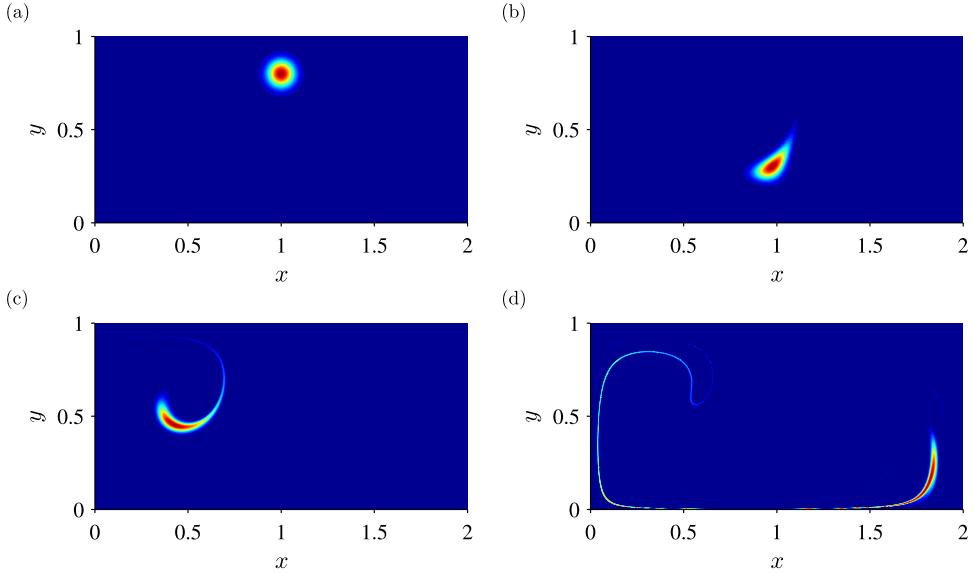


Fig. 13. Double gyre – Propagation of a ‘local’ PDF through the time-dependent double gyre flow using flow map composition; the approximate short-time flow map is computed for $\Delta t = 0.1$ and is of order 10 in the x -direction and 5 in y -direction: (a) initial PDF at $t = 0$, (b) PDF at $t = 2$, (c) PDF at $t = 10$, and (d) PDF at $t = 20$. The colormap varies from 0 (blue) to maximum (red). See Section 5.3 for governing equations ($A = 0.1$, $\omega = 2\pi/10$, $\epsilon = 0.25$).

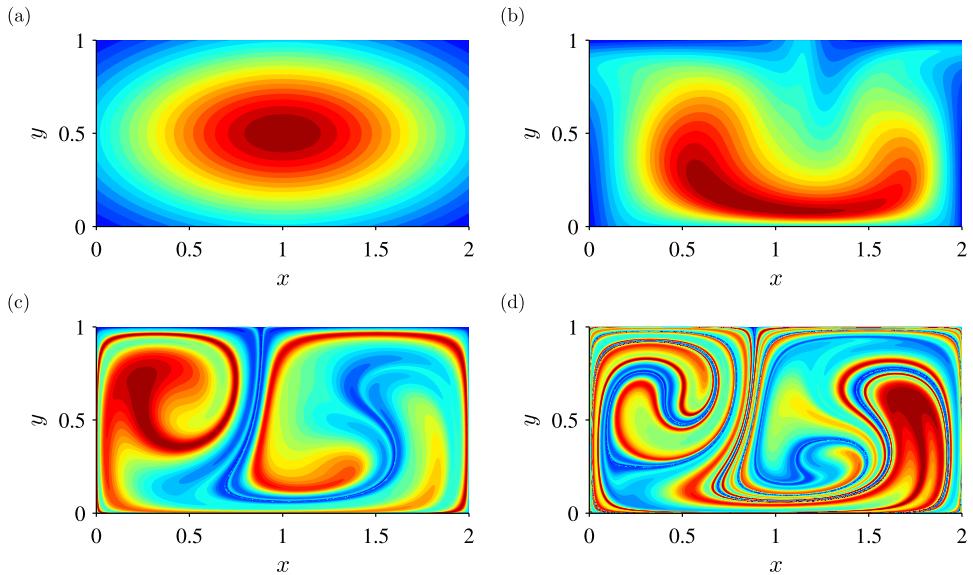


Fig. 14. Double gyre – Propagation of a ‘global’ PDF through the time-dependent double gyre flow using flow map composition; the approximate short-time flow map is computed for $\Delta t = 0.1$ and is of order 10 in the x -direction and 5 in y -direction: (a) initial PDF at $t = 0$, (b) PDF at $t = 2$, (c) PDF at $t = 10$, and (d) PDF at $t = 20$. The colormap varies from 0 (blue) to maximum (red). See Section 5.3 for governing equations ($A = 0.1$, $\omega = 2\pi/10$, $\epsilon = 0.25$).

not be optimal. In that sense, walltimes should be used for qualitative rather than quantitative comparison, although results indicate that gPC-FMC performs well.

6.1. Nonlinear ODE

We consider the example of Section 5.1. We first integrate the nonlinear ODE (24) using a vectorized RK4 integrator with constant time step h . We take the RK4 solution with $h = 10^{-5}$ as the ‘exact’ solution. The error converges for a time step $h_s = 10^{-3}$, and we use this time step for ME-gPC and gPC-FMC. For computation of the L_2 error and walltime, we propagate 500 uniformly distributed points over the interval $[-1, 1]$. The L_2 error versus walltime for RK4 integration, ME-gPC, and gPC-FMC is shown in Fig. 15 for integration times $T = 3, 6$. First, we focus on the RK4 branch (Monte Carlo simulation). The time step h is decreased from 0.5 to 10^{-4} , and consequently the L_2 error decreases while the walltime increases. As

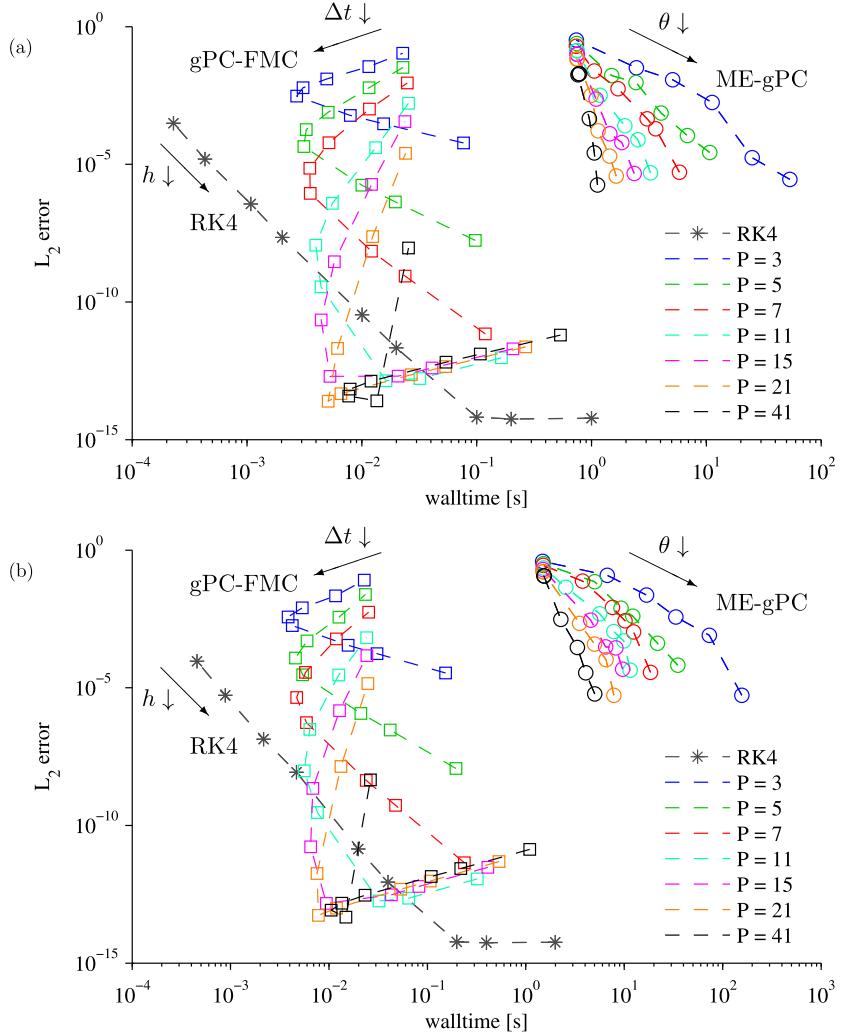


Fig. 15. Nonlinear ODE – L_2 error versus waltime. Comparison of RK4, ME-gPC, and gPC-FMC for (a) $T = 3$, and (b) $T = 6$. Along the branch denoted by RK4, the time step h is decreased from 0.5 to 10^{-4} . For all other branches the RK4 time step is fixed at $h_s = 10^{-3}$, and each branch is associated with a particular polynomial expansion order P . Along a gPC-FMC branch (\square), the time step for the short-time flow map Δt is decreased from 1 to 0.001. Along a ME-gPC branch (\circ), the threshold value θ is varied from 1 to 10^{-5} ($\theta \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0\}$).

mentioned above, the error levels off at $h_s = 10^{-3}$. Second, we consider the gPC-FMC branches. The RK4 time step is fixed for all branches at $h_s = 10^{-3}$. Each branch represents a different polynomial expansion order (P ranges from 3 to 41). Along each branch, the time step for the short-time flow map Δt is decreased from 1 to 0.001. The curves are ‘C-shaped’ showing that the L_2 error decreases, whereas the waltime initially decreases, but later increases. The waltime reflects a trade off between the cost of integration of the collocation points for a time Δt and the number of compositions $M = T/\Delta t$ that are required for a total integration time T . There is a small region where gPC-FMC outperforms RK4 integration. Third, we consider the ME-gPC method. The RK4 time step is fixed at $h_s = 10^{-3}$, and each branch also represents a particular polynomial expansion order. Along each branch, the threshold value θ is varied from 1 to 10^{-6} . Thus, more and more spectral elements are introduced when θ is lowered, and when it is high enough no spectral elements are introduced, and ME-gPC is equivalent to gPC. In the graphs, the latter are the ME-gPC points with smallest waltime and largest L_2 error. The maximum number of spectral elements that corresponds to each θ value is shown in Fig. 16.

The flow map for this example is autonomous. This implies that we need to compute one short-time flow map for the gPC-FMC method. The propagation of points follows from composition evaluation of this polynomial approximation (flow map composition). In the gPC method, the collocation points need to be propagated for each RK4 time step, the polynomial expansion needs to be built and evaluated for all points that are propagated. This procedure is more costly and consequently we have a larger waltime compared to gPC-FMC. Furthermore, for the ME-gPC method, the threshold value θ is checked at each time step, and if needed more elements are created to keep the error bounded. It is clear that for this example gPC-FMC outperforms ME-gPC.

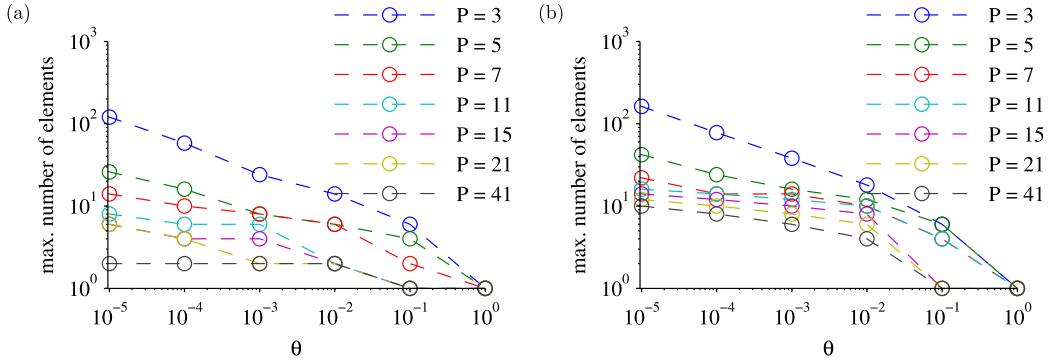


Fig. 16. Nonlinear ODE – ME-gPC: Maximum number of spectral elements as the threshold θ in Fig. 15 is varied. Subfigures (a) and (b) correspond to Figs. 15(a) and (b) respectively.

6.2. Uncertain linear ODE

Here, we consider the example of Section 5.2. As above, we first integrate the ODE (31) using a vectorized RK4 integrator with constant time step h . We take the RK4 solution with $h = 10^{-5}$ as the ‘exact’ solution. The error converges for a time step $h_s = 10^{-3}$, and we use this time step for ME-gPC and gPC-FMC. We evaluate the error for 1000 uniformly distributed points over the interval $[-1, 1]$. The L_2 error versus walltime for RK4 integration, ME-gPC, and gPC-FMC is shown in Fig. 17 for integration times $T = 10, 30$. The description of this figure is completely analogous to Fig. 15, and we refer the reader to Section 6.1 for more details. For both the gPC-FMC and ME-gPC method we consider polynomial expansions ranging from $P = 3$ to 21. For the gPC-FMC method, the time step of the short-term flow map varies over the range $\Delta t = 0.001\text{--}1$. For the ME-gPC method, we vary the threshold θ from 1 to 10^{-3} . We observe that for relatively high-degree polynomial expansions ($P = 15, 21$), lowering the threshold value to 10^{-3} does not have an effect. The maximum number of spectral elements that corresponds to each θ value is shown in Fig. 18. For this example we also observe that gPC-FMC outperforms ME-gPC.

6.3. Computational resources

Computationally, gPC-FMC should be compared against Monte Carlo sampling, since in both algorithms a large cloud of initial conditions are passed through the dynamical system to obtain statistics of the long-time distribution. However, in gPC-FMC the computational burden is shifted from integration to polynomial evaluation required for interpolation.

In order to compute the expansion coefficients of the short-time flow expansion (21) we need to integrate the governing equations (2) for all quadrature points, as described in Sections 4.1 and 3.3. However, an accurate approximation of the short-time solution requires significantly less points than required by Monte Carlo sampling of the long-time solution.

Typically, it is assumed that the cost of polynomial evaluation is negligible compared to the cost of integrating a trajectory. For example, compared to large-scale numerical simulations it is clear that simple algebraic evaluations do not incur any notable cost. In [20], where a gPC-based ENKF is considered, it is also assumed that simulations are expensive compared to the cost of gPC. Thus, for problems with a large state-space, the only relevant number for the computational cost is the number of quadrature points that needs to be integrated.

6.3.1. Other considerations

In many engineering applications integrating particle trajectories through measured velocity fields will already involve expensive interpolation between data snapshots [23,24]. Many of these methods may be extended to data collected from physical collocation points, realized by ocean gliders, UAVs, etc. In this case, the number of physical collocation points should be minimized.

Finally, there is a possibility for significant reduction in memory requirements with the proposed method. Extracting short-time flow maps using a small number of collocation points effectively compresses the relevant information in velocity field snapshots. This makes it possible to minimize data transfer and to potentially store all of the short-time flow maps in RAM, even for problems with large datasets.

7. Conclusions

We have presented a flexible approach for long-time uncertainty propagation in dynamical systems. The method approximates short-time flow maps by low-order spectral polynomial expansions and uses flow map composition to construct the flow map for longer integration times. In contrast to the gPC method, which performs poorly for long integration times, this approach has exponential error convergence for both short and long integration times. The composition of short-time flow maps, each represented in a low-order polynomial basis, leads to an overall flow map whose degree is exponential in the number of compositions, while the number of polynomial coefficients required to represent the overall flow map is

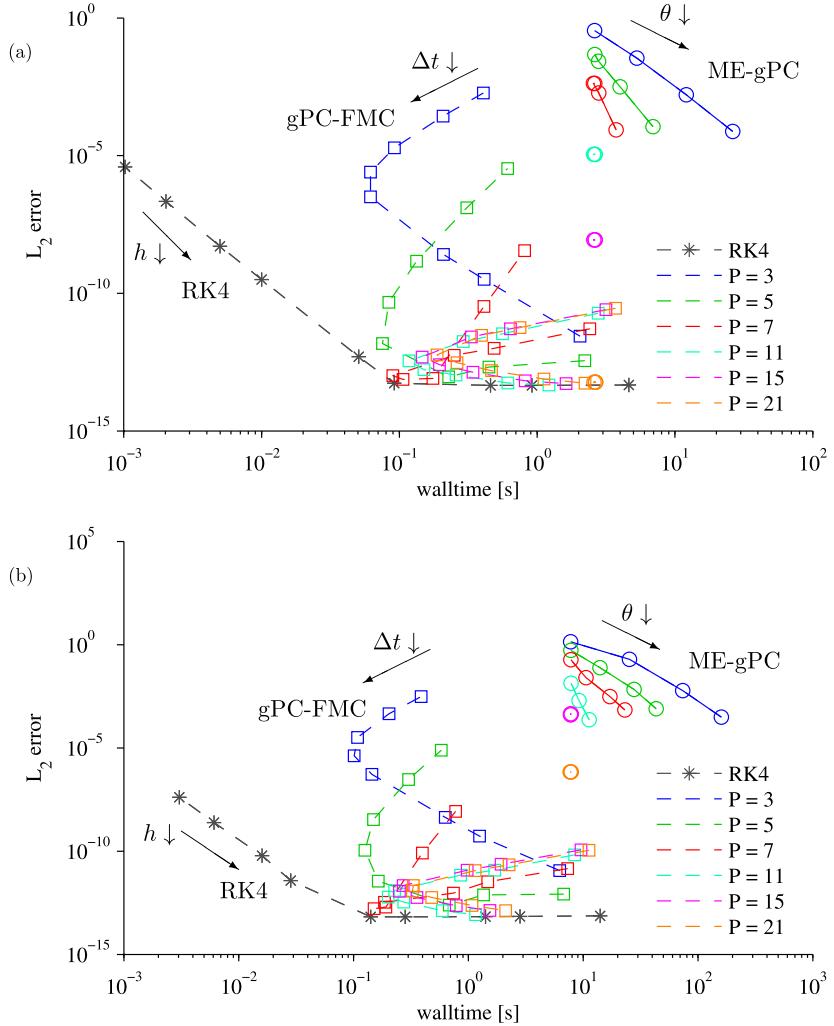


Fig. 17. Uncertain linear ODE – L_2 error versus walltime. Comparison of RK4, ME-gPC, and gPC-FMC for (a) $T = 10$, and (b) $T = 30$. Along the branch denoted by RK4, the time step h is decreased from 0.5 to 10^{-4} . For all other branches the RK4 time step is fixed at $h_s = 10^{-3}$, and each branch is associated with a particular polynomial expansion order P . Along a gPC-FMC branch (\square), the time step for the short-time flow map Δt is decreased from 1 to 0.001. Along a ME-gPC branch (\circ), the threshold value θ is varied from 1 to 10^{-3} ($\theta \in \{10^{-3}, 10^{-2}, 10^{-1}, 10^0\}$).

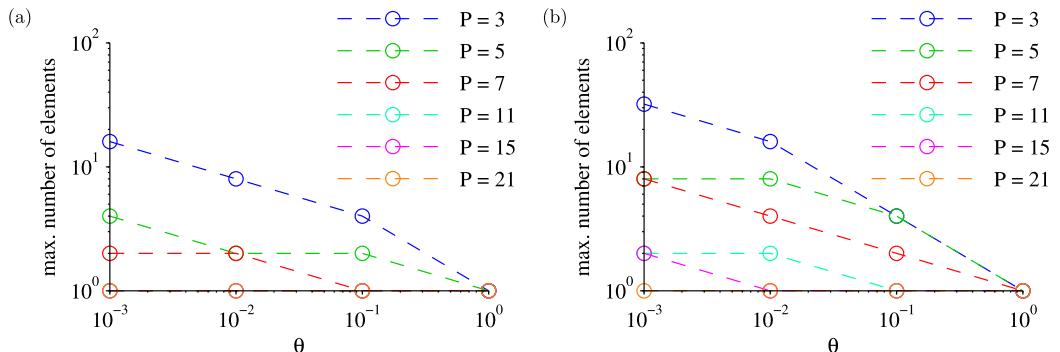


Fig. 18. Uncertain linear ODE – ME-gPC: Maximum number of spectral elements as the threshold θ in Fig. 17 is varied. Subfigures (a) and (b) correspond to Figs. 17(a) and (b) respectively.

only linear in the number of compositions (or constant, for an autonomous system). The method is particularly efficient for propagating multiple PDFs through the same dynamical system, since the short-time flow maps need to be computed only once.

The above approach was applied to three examples. In the first example, flow map composition is shown to be accurate and efficient although the uniformly distributed initial condition evolves into the discontinuous Heaviside function and is not well captured by traditional polynomial expansions, such as gPC. Our second example highlights the problem of long-time integration and the non-uniform convergence of the gPC method when a nonlinearity enters through an uncertain model parameter. The last example is the time-dependent double gyre flow, and illustrates the effectiveness of the approach even in the case of strong nonlinearities.

A number of future directions arise from this work. The efficiency and low memory requirements of this method make it directly applicable to large-scale contaminant release problems, such as tracking oil spills [25]. In particular, one may view the spectral flow map composition as a reduced-order model for the long-time flow map, whereby it is possible to rapidly advect any distribution after the up-front cost of integrating collocation points. It would also be interesting to apply this method to other dynamical systems such as mechanical linkages and multi-body problems in astrophysics, where long-time integration and uncertainty management are issues. It is important that the state dimension of these systems is not too large.

In addition to demonstrating this approach on large-scale physical problems, there are further aspects of the method which should be investigated. The favorable error properties of flow map composition have previously been investigated in the context of finite-time Lyapunov exponents. However, the structure of the resulting high-degree polynomial obtained through composition is tailored to represent long-time mixing structures in the flow, and understanding the nature of these polynomials is a topic of current research. Finally, it may be possible to determine high-order statistics directly from the leading terms of the long-time polynomial to reduce the burden of Monte Carlo sampling.

Acknowledgements

This work was supported by the Air Force Office of Scientific Research, grant FA9550-10-1-0143, and by the Federal Aviation Administration's Joint University Program. We are grateful for the comments and suggestions of the anonymous reviewers.

References

- [1] H. Aref, Stirring by chaotic advection, *J. Fluid Mech.* 143 (1984) 1–21.
- [2] N. Wiener, The homogeneous chaos, *Am. J. Math.* 60 (1938) 897–936.
- [3] R.G. Ghanem, P.D. Spanos, *Stochastic Finite Elements: A Spectral Approach*, Dover, 1991.
- [4] D. Xiu, G.E. Karniadakis, The Wiener–Askey polynomial chaos for stochastic differential equations, *SIAM J. Sci. Comput.* 24 (2002) 619–644.
- [5] H.N. Najm, Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics, *Annu. Rev. Fluid Mech.* 41 (2009) 35–52.
- [6] B.J. Debusschere, H.N. Najm, A. Matta, O.M. Knio, R.G. Ghanem, O.P. Le Maître, Protein labeling reactions in electrochemical microchannel flow: numerical simulation and uncertainty propagation, *Phys. Fluids* 15 (2003) 2238.
- [7] O.M. Knio, O.P. Le Maître, Uncertainty propagation in CFD using polynomial chaos decomposition, *Fluid Dyn. Res.* 38 (2006) 616–640.
- [8] X. Wan, G.E. Karniadakis, An adaptive multi-element generalized polynomial chaos method for stochastic differential equations, *J. Comput. Phys.* 209 (2005) 617–642.
- [9] G. Kewlani, K. Iagnemma, A multi-element generalized polynomial chaos approach to analysis of mobile robot dynamics under uncertainty, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1177–1182.
- [10] P. Prempraneeeracha, F. Hover, M. Triantafyllou, G. Karniadakis, Uncertainty quantification in simulations of power systems: multi-element polynomial chaos methods, *Reliab. Eng. Syst. Saf.* 95 (2010) 632–646.
- [11] M. Gerritsma, J.-B. van der Steen, P.E.J. Vos, G.E. Karniadakis, Time-dependent generalized polynomial chaos, *J. Comput. Phys.* 229 (2010) 8333–8363.
- [12] V. Heuveline, M. Schick, Towards a hybrid numerical method using generalized polynomial chaos for stochastic differential equations, Technical Report, Karlsruhe Institute of Technology, 2011.
- [13] L. Ying, E.J. Candès, The phase flow method, *J. Comput. Phys.* 220 (2006) 184–215.
- [14] S.L. Brunton, C.W. Rowley, Fast computation of finite-time Lyapunov exponent fields for unsteady flows, *Chaos* 20 (2010) 017503.
- [15] G. Froyland, K. Padberg, Almost-invariant sets and invariant manifolds – connecting probabilistic and geometric descriptions of coherent structures in flows, *Physica D* 238 (2009) 1507–1523.
- [16] D. Xiu, Fast numerical methods for stochastic computations: a review, *Commun. Comput. Phys.* 5 (2009) 242–272.
- [17] O.G. Ernst, A. Mugler, H.-J. Starkloff, E. Ullmann, On the convergence of generalized polynomial chaos expansions, *Modél. Math. Anal. Numér.* 46 (2012) 317–339.
- [18] T.H. Solomon, J.P. Gollub, Chaotic particle transport in time-dependent Rayleigh–Bénard convection, *Phys. Rev. A* 38 (1988) 6280–6286.
- [19] S.C. Shadden, F. Lekien, J.E. Marsden, Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows, *Physica D* 212 (2005) 271–304.
- [20] J. Li, D. Xiu, A generalized polynomial chaos based ensemble Kalman filter with high accuracy, *J. Comput. Phys.* 228 (2009) 5454–5469.
- [21] O.P. Le Maître, O.M. Knio, *Spectral Methods for Uncertainty Quantification*, Springer, 2010.
- [22] P. Tallapragada, S.D. Ross, A set oriented definition of finite-time Lyapunov exponents and coherent sets, *Commun. Nonlinear Sci. Numer. Simul.* 18 (2013) 1106–1126.
- [23] F. Lekien, J. Marsden, Tricubic interpolation in three dimensions, *Int. J. Numer. Methods Biomed. Eng.* 63 (2005) 455–471.
- [24] F. Lekien, C. Couillette, A.J. Mariano, E.H. Ryan, L.K. Shay, G. Haller, J.E. Marsden, Pollution release tied to invariant manifolds: a case study for the coast of Florida, *Physica D* 210 (2005) 1–20.
- [25] I. Mezić, S. Loire, V.A. Fonoferov, P. Hogan, A new mixing diagnostic and Gulf oil spill movement, *Science* 330 (2010) 486–489.