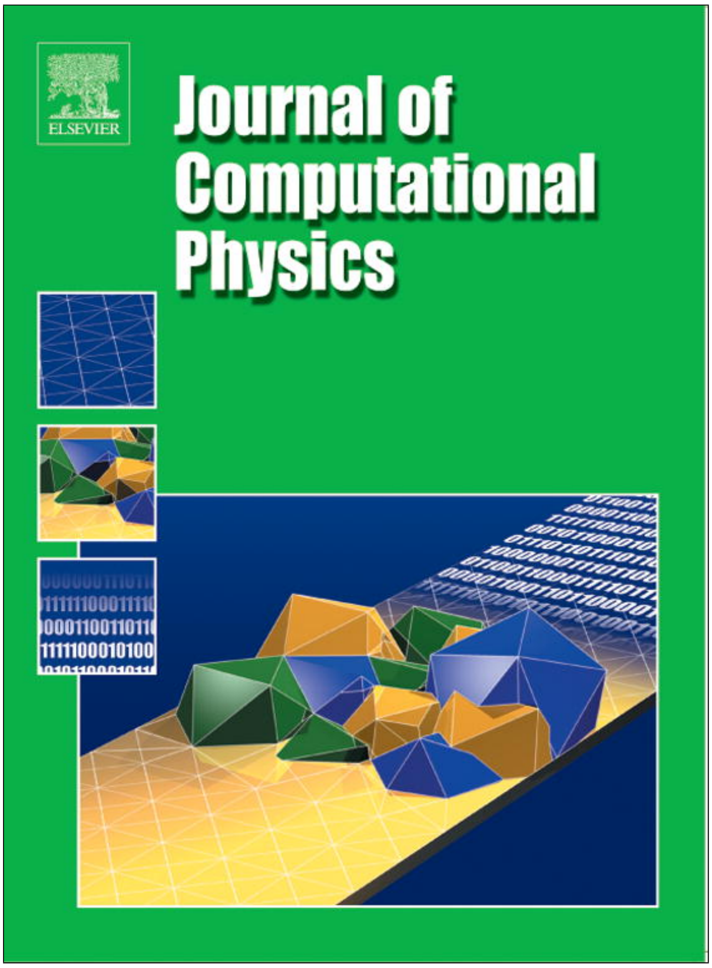

ELSEVIER
Journal of Computational Physics

Journal of Computational Physics 242 (2013) 753–776

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

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

A dynamically bi-orthogonal method for time-dependent stochastic partial differential equations II: Adaptivity and generalizations

Mulin Cheng, Thomas Y. Hou *, Zhiwen Zhang

Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125, United States

ARTICLE INFO

Article history:
Received 31 October 2012
Received in revised form 12 January 2013
Accepted 8 February 2013
Available online 28 February 2013

Keywords:
Stochastic partial differential equations
Karhunen–Loeve expansion
Low-dimensional structure
Adaptivity algorithm
Sparsity
Stochastic flow

ABSTRACT

This is part II of our paper in which we propose and develop a dynamically bi-orthogonal method (DyBO) to study a class of time-dependent stochastic partial differential equations (SPDEs) whose solutions enjoy a low-dimensional structure. In part I of our paper [9], we derived the DyBO formulation and proposed numerical algorithms based on this formulation. Some important theoretical results regarding consistency and bi-orthogonality preservation were also established in the first part along with a range of numerical examples to illustrate the effectiveness of the DyBO method. In this paper, we focus on the computational complexity analysis and develop an effective adaptivity strategy to add or remove modes dynamically. Our complexity analysis shows that the ratio of computational complexities between the DyBO method and a generalized polynomial chaos method (gPC) is roughly of order $O((m/N_p)^3)$ for a quadratic nonlinear SPDE, where m is the number of mode pairs used in the DyBO method and N_p is the number of elements in the polynomial basis in gPC. The effective dimensions of the stochastic solutions have been found to be small in many applications, so we can expect m is much smaller than N_p and computational savings of our DyBO method against gPC are dramatic. The adaptive strategy plays an essential role for the DyBO method to be effective in solving some challenging problems. Another important contribution of this paper is the generalization of the DyBO formulation for a system of time-dependent SPDEs. Several numerical examples are provided to demonstrate the effectiveness of our method, including the Navier–Stokes equations and the Boussinesq approximation with Brownian forcing.

1. Introduction

This is the second part of the paper in developing a dynamically bi-orthogonal (DyBO) method for solving time-dependent stochastic partial differential equations (SPDEs). It is well known that solving SPDEs is very challenging due to the introduction of random variables and/or stochastic processes. The computational cost increases exponentially fast as the number of random variables increases, which is also known as the curse of dimensionality. In the past two decades, there has been tremendous progress in numerical simulations of the SPDEs. To our knowledge, these methods can be classified into two major groups, Monte Carlo methods [25,26,12,2] and polynomial chaos methods [31,5,11,32,33,20,14]. Monte Carlo methods are very robust and have the advantage of being independent of the dimensionality of random variables, but they suffer from slow convergence due to their sampling nature. Polynomial chaos methods provide more accurate approximations because

* Corresponding author. Tel.: +1 626 395 4546.
E-mail addresses: mulinch@caltech.edu (M. Cheng), hou@cms.caltech.edu (T.Y. Hou), zhangzw@caltech.edu (Z. Zhang).

0021-9991/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2013.02.020

of their spectral representation property. However, they suffer from the curse of dimensionality when the number of independent random variables is high.

Many of physical and engineering simulation problems that appear to be high-dimensional have some hidden low-dimensional or sparse structures. In recent years, we have witnessed a surge of interests in exploring sparse structures prevailing in many physical and engineering problems. These methods include compressed sensing in signal reconstruction [6,10], hierarchical matrix in discretization of integral operators [13], adaptive data analysis in signal processing [17,18], signal processing for speech and music via l^1 minimization [23,34], proper orthogonal decomposition (POD) methods [3,30], reduced basis (RB) methods [4,24,27] in solving parameterized PDEs, and the dynamically Orthogonal (DO) method in solving SPDEs [28,29]. Most of these methods emphasize the use of spatial basis, but ignore stochastic basis. Thus they do not preserve the bi-orthogonality of the spatial and the stochastic basis in their expansions.

The dynamically bi-orthogonal method (DyBO) that we proposed and developed in [9] and this paper (see also [8]) aims at preserving the dynamic bi-orthogonality, thus essentially tracking the Karhunen–Loeve expansions [19,21] of stochastic solutions. The Karhunen–Loeve expansion provides the optimal spatial and stochastic basis in the sense that it minimizes the total mean squared error and gives the sparsest representation of stochastic solutions. One important advantage of DyBO over other reduced basis methods is that we construct our reduced basis on the fly without the need to compute the reduced basis offline by sampling the stochastic solution. Another advantage of our method is that we do not need to compute the covariance matrix, which could be very computationally expensive especially for high-dimensional problems. By solving an equivalent system that governs the evolution of the spatial and stochastic basis, our method explores the low-dimensional structure intrinsically hidden in a wide range of time-dependent SPDEs.

In part I of our paper [9], we introduced the derivation of dynamically bi-orthogonal formulation for time-dependent SPDEs, and proved several theoretical properties, such as the dynamically bi-orthogonality preservation and the consistency between the DyBO formulation and the original SPDE. We also gave some details on the numerical implementation of the DyBO methods, including the representation of stochastic basis and how to deal with eigenvalue crossing. One of the purposes of this paper is to study several important issues concerning the numerical performance of the DyBO method. These include the computational complexity analysis and an adaptive strategy for adding or removing spatial and stochastic basis on the fly. We also generalize the dynamically bi-orthogonal formulation for a system of SPDEs and propose an effective parallel algorithm for DyBO. The parallel implementation is important for industrial-scale applications.

Our complexity analysis gives a detailed comparison between the complexity of our DyBO method and that of gPC. Our analysis shows that the ratio between the complexity of DyBO and that of gPC is of order $O(m/N_h^d + (m/N_p)^3)$ for a quadratic nonlinear SPDE. Here m is the number of modes used in DyBO, N_p is the number of polynomial basis functions used in gPC, and N_h^d is the total number of spatial grid points in a d-dimensional problem. Typically, we expect $m \ll N_p$ and $m/N_h^d \ll (m/N_p)^3$. Thus the ratio of complexities between DyBO and gPC is roughly of order $O((m/N_p)^3)$. This has been confirmed by our numerical experiments. Our complexity analysis also indicates that DyBO consumes less memory compared with gPC. The ratio of memory consumptions between DyBO and gPC is of order $O(m/N_p)$.

The ability to add or remove modes dynamically is crucial for the successful applications of our DyBO method to more challenging SPDEs. The adaptive strategy that we develop in this paper is based on solving both DyBO and gPC solutions with the same initial condition for a short time. We then extract the dominating spatial and stochastic modes by performing KL expansion on the difference of the two solutions. By adding these dominant modes back to the DyBO formulation, we recapture previously unresolved dynamics and maintain the accuracy of our method as these unresolved components become important later in time. We have applied this adaptive strategy to solve the 1D stochastic Burgers equation, the 2D incompressible Navier–Stokes equation and the Boussinesq approximation with Brownian motion forcing. Our numerical results indicate that the adaptive strategy indeed works quite effectively. The adaptive method gives the results that are almost indistinguishable from those obtained by using a large m from the beginning. Further, we demonstrate the convergence of our method as the number of modes increases.

This paper is organized as follows. In Section 2, we provide a brief overview of the DyBO formulation. We perform the computational complexity analysis of our DyBO method in gPC version in Section 3, and compare the complexity of DyBO with that of gPC. A parallel strategy is also proposed. In Section 4 a local error analysis between DyBO method and gPC method is conducted and an adaptivity strategy in changing the number of the spatial and stochastic basis is proposed. We generalize the DyBO formulation for a system of time-dependent SPDEs in Section 5. Several numerical examples are provided in Section 6 to demonstrate these ideas. Finally, some conclusion remarks will be made in Section 7.

2. Overview of the DyBO formulation for SPDEs

In order to set up notations for discussion and ease readers for further reading, we give a brief overview of the DyBO formulation in this section. Further details can be found in part I of this paper [9]. Consider the following time-dependent SPDE:

$$\frac{\partial u}{\partial t}(x,t,\omega) = \mathcal{L}u(x,t,\omega), \quad x \in \mathcal{D} \subset \mathbb{R}^d, \quad t \in [0,T], \quad \omega \in \Omega, \tag{1a}$$

$$u(x,0,\omega) = u_0(x,\omega), \quad x \in \mathcal{D}, \quad \omega \in \Omega, \tag{1b}$$

$$\mathcal{B}(u(x,t,\omega)) = h(x,t,\omega), \quad x \in \partial\mathcal{D}, \quad \omega \in \Omega, \tag{1c}$$

where $\mathcal{L}$ is a differential operator that may contain random coefficients and/or stochastic forces and $\mathcal{B}$ is a linear differential operator. The randomnesses may also enter the system through initial u_0 and/or boundary conditions $\mathcal{B}$.

We assume the stochastic solution $u(x,t,\omega)$ of the system (1) is a second-order stochastic process at each fixed time $t > 0$, i.e., $u(\cdot,t,\cdot) \in \mathbb{L}^2(\mathcal{D} \times \Omega)$. We consider the following truncated KL expansion,

$$\widetilde{u}(x,t,\omega) = \bar{u}(x,t) + \sum_{i=1}^{m} u_i(x,t)Y_i(\omega,t) = \bar{u}(x,t) + \mathbf{U}(x,t)\mathbf{Y}^T(\omega,t) \approx u(x,t,\omega), \tag{2}$$

where $\mathbf{U} = (u_1, u_2, \cdots, u_m)$ and $\mathbf{Y} = (Y_1, Y_2, \cdots, Y_m)$. Define an anti-symmetrization operator $\mathcal{Q} : \mathbb{R}^{k\times k} \to \mathbb{R}^{k\times k}$ and a partial anti-symmetrization operator $\tilde{\mathcal{Q}} : \mathbb{R}^{k\times k} \to \mathbb{R}^{k\times k}$ as follows:

$$\mathcal{Q}(\mathbf{A}) = \frac{1}{2}(\mathbf{A} - \mathbf{A}^T), \quad \tilde{\mathcal{Q}}(\mathbf{A}) = \frac{1}{2}(\mathbf{A} - \mathbf{A}^T) + \mathrm{diag}(\mathbf{A}),$$

where $\mathbf{A} \in \mathbb{R}^{k\times k}$ and $\mathrm{diag}(\mathbf{A})$ is a diagonal matrix whose diagonal entries are equal to those of matrix $\mathbf{A}$.

By enforcing the bi-orthogonal condition via $\mathcal{Q}$ and $\tilde{\mathcal{Q}}$ and a compatibility condition, we obtain the DyBO formulation for the SPDE (1)

$$\frac{\partial \bar{u}}{\partial t} = \mathbb{E}[\mathcal{L}\widetilde{u}], \tag{3a}$$

$$\frac{\partial \mathbf{U}}{\partial t} = -\mathbf{U}\mathbf{D}^T + \mathbb{E}[\tilde{\mathcal{L}}\widetilde{u}\mathbf{Y}], \tag{3b}$$

$$\frac{\mathrm{d}\mathbf{Y}}{\mathrm{d}t} = -\mathbf{Y}\mathbf{C}^T + \langle \tilde{\mathcal{L}}\widetilde{u}, \mathbf{U}\rangle \mathbf{\Lambda}_{\mathbf{U}}^{-1}, \tag{3c}$$

where $\mathbf{\Lambda}_{\mathbf{U}} = \mathrm{diag}\left(\left\langle \mathbf{U}^T, \mathbf{U}\right\rangle\right) = (\langle u_i, u_j\rangle \delta_{ij}) \in \mathbb{R}^{m\times m}$ and $\tilde{\mathcal{L}}\widetilde{u} = \mathcal{L}\widetilde{u} - \mathbb{E}[\mathcal{L}\widetilde{u}]$ and the m-by-m matrices $\mathbf{C}$ and $\mathbf{D}$ can be solved uniquely from the following linear system,

$$\mathbf{C} - \mathbf{\Lambda}_{\mathbf{U}}^{-1}\tilde{\mathcal{Q}}(\mathbf{\Lambda}_{\mathbf{U}}\mathbf{C}) = 0, \tag{4a}$$

$$\mathbf{D} - \mathcal{Q}(\mathbf{D}) = 0, \tag{4b}$$

$$\mathbf{D}^T + \mathbf{C} = G_*(\bar{u}, \mathbf{U}, \mathbf{Y}), \tag{4c}$$

where the matrix $G_*(\bar{u}, \mathbf{U}, \mathbf{Y}) = \mathbf{\Lambda}_{\mathbf{U}}^{-1}\left\langle \mathbf{U}^T, \mathbb{E}[\tilde{\mathcal{L}}\widetilde{u}\mathbf{Y}]\right\rangle \in \mathbb{R}^{m\times m}$.

The first two equations in the DyBO formulation (3) are time-dependent *deterministic* PDEs for the mean solution $\bar{u}$ and the spatial basis function $\mathbf{U}$ and they are coupled to the third equation, a system of stochastic ODEs for the stochastic basis function $\mathbf{Y}$. Various spatial discretization schemes, such as finite difference schemes or spectral methods, along with ODE solvers, such as the fourth-order Runge–Kutta method can be used to solve the first two deterministic PDEs. For the numerical simulations of the stochastic ODEs (3c), three representations of the stochastic modes $\mathbf{Y}$ have been proposed in the first part of the paper [9], leading to three variants of DyBO method, i.e., DyBO-MC, DyBO-gSC and DyBO-gPC. In this paper, we primarily focus on DyBO-gPC methods, although similar arguments can also be applied to DyBO-MC and DyBO-gSC.

The Cameron–Martin theorem [5] implies the stochastic modes $Y_i(\omega,t)$'s in the KL expansion (2) can be approximated by the linear combination of polynomial chaos, i.e.,

$$Y_i(\omega,t) = \sum_{\boldsymbol{\alpha}\in\mathfrak{J}} \mathbf{H}_{\boldsymbol{\alpha}}(\xi(\omega))A_{\boldsymbol{\alpha}i}(t), \quad i = 1,2,\cdots,m, \tag{5}$$

or in a matrix form, if we write $\mathbf{H}(\xi) = (\mathbf{H}_{\boldsymbol{\alpha}}(\xi))_{\boldsymbol{\alpha}\in\mathfrak{J}}$ as a row vector,

$$\mathbf{Y}(\omega,t) = \mathbf{H}(\xi(\omega))\mathbf{A}, \tag{6}$$

where $\mathbf{A} \in \mathbb{R}^{N_p\times m}$ and N_p is the number of polynomial basis functions. The expansion (2) now reads

$$\widetilde{u} = \bar{u} + \mathbf{U}\mathbf{A}^T\mathbf{H}^T.$$

We can derive equations for $\bar{u}, \mathbf{U}$ and $\mathbf{A}$, instead of $\bar{u}, \mathbf{U}$ and $\mathbf{Y}$. In other words, the stochastic modes $\mathbf{Y}$ are identified with a matrix $\mathbf{A} \in \mathbb{R}^{N_p\times m}$, which leads to the DyBO-gPC formulation of SPDE (1),

$$\frac{\partial \bar{u}}{\partial t} = \mathbb{E}[\mathcal{L}\widetilde{u}], \tag{7a}$$

$$\frac{\partial \mathbf{U}}{\partial t} = -\mathbf{U}\mathbf{D}^T + \mathbb{E}[\tilde{\mathcal{L}}\widetilde{u}\mathbf{H}]\mathbf{A}, \tag{7b}$$

$$\frac{\mathrm{d}\mathbf{A}}{\mathrm{d}t} = -\mathbf{A}\mathbf{C}^T + \left\langle \mathbb{E}[\mathbf{H}^T\tilde{\mathcal{L}}\widetilde{u}], \mathbf{U}\right\rangle \mathbf{\Lambda}_{\mathbf{U}}^{-1}, \tag{7c}$$

where $\mathbf{C}(t)$ and $\mathbf{D}(t)$ can be solved from the linear system (4) with

$$G_*(\bar{u},\mathbf{U},\mathbf{Y}) = \mathbf{\Lambda}_{\mathbf{U}}^{-1}\left\langle \mathbf{U}^T, \mathbb{E}[\tilde{\mathcal{L}}\tilde{u}\mathbf{Y}]\right\rangle = \mathbf{\Lambda}_{\mathbf{U}}^{-1}\left\langle \mathbf{U}^T, \mathbb{E}[\tilde{\mathcal{L}}\tilde{u}\mathbf{H}]\right\rangle \mathbf{A}. \tag{8}$$

By solving the system (7), we have an approximate solution to SPDE (1)

$$u^{\text{DyBO-gPC}} = \bar{u} + \mathbf{U}\mathbf{A}^T\mathbf{H}^T.$$

The orthonormal property of $\mathbf{Y}$ implies that the columns of $\mathbf{A}$ are orthonormal, i.e., $\mathbf{A}^T\mathbf{A} = \mathbf{I} \in \mathbb{R}^{m\times m}$. We would like to point out that $\mathbf{A}\mathbf{A}^T \in \mathbb{R}^{N_p\times N_p}$ in general is *not* an identity matrix as $m \ll N_p$ if the SPDE solution has a low-dimensional or sparse structure.

3. Computational complexity analysis

As we discussed in the previous sections, the DyBO method explores the low-dimensional structure of the stochastic solutions of time-dependent SPDE and represents the solution in the most compact form in the $\mathbb{L}^2$ sense. The DyBO method not only offers savings in memory consumption, but also reduces the computational cost since we have much fewer entries to update in each step time compared to gPC methods. In this section, the storage complexity and the computational cost between the DyBO-gPC method and the gPC method will be analysed and compared. We provide the analysis for a typical scenario, i.e., the quadratic nonlinear PDE driven by stochastic forces. Examples of this type of SPDEs include the stochastic Burgers equation and the stochastic Naiver–Stokes equation. In Section 6, numerical examples will be provided to confirm the complexity analysis.

To make the discussion concrete, we assume throughout this section that the randomness is given in terms of r independent random variables $\xi_i(\omega)$ of the same distribution $\rho(\cdot)$, and the set of polynomial chaos basis $\mathbf{H}$ has N_p elements, i.e., the cardinality of multi-index set $|\mathfrak{J}| = N_p$. Furthermore, N_h grid nodes are used along each direction of the hyper-cube $\mathcal{D} \in \mathbb{R}^d$, which results in a spatial grid of total N_h^d nodes. Such discretizations generally lead to large systems for both gSC and gPC. As a reminder, we have assumed throughout this paper that the solutions of SPDEs under consideration enjoy low-dimensional structures, i.e., $m \ll N_p$.

3.1. Storage complexity

Consider the gPC expansion of the stochastic solution, i.e.,

$$u^{\text{gPC}}(x,t,\xi) = \bar{v}(x,t) + \sum_{\boldsymbol{\alpha}\in\mathfrak{J}} v_{\boldsymbol{\alpha}}(x,t)\mathbf{H}_{\boldsymbol{\alpha}}(\xi) = \bar{v}(x,t) + \mathbf{V}(x,t)\mathbf{H}^T(\xi), \tag{9}$$

where $\mathbf{V}(x,t) = (v_{\boldsymbol{\alpha}}(x,t))_{\boldsymbol{\alpha}\in\mathfrak{J}}$ is a row vector of length N_p. It is easy to derive the gPC formulation of the SPDE (1),

$$\frac{\partial\bar{v}}{\partial t} = \mathbb{E}[\mathcal{L}v], \tag{10a}$$

$$\frac{\partial\mathbf{V}}{\partial t} = \mathbb{E}[\tilde{\mathcal{L}}v\mathbf{H}]. \tag{10b}$$

From the above gPC formulation (10), it is clear that $\bar{v}$ and $\mathbf{V}$ have to be updated in each time iteration. Thus, the storage cost of the gPC solution is proportional to $O(N_h^d) + O(N_pN_h^d) = O(N_pN_h^d)$.

On the other hand, the mean $\bar{u}(x,t)$, the spatial modes $\mathbf{U}(x,t) = (u_1(x,t), u_2(x,t), \cdots, u_m(x,t))$ and the stochastic modes $\mathbf{A}(t) \in \mathbb{R}^{N_p\times m}$ in the DyBO-gPC formulation (7) are updated every time iteration, which implies the memory consumption is proportional to $O(N_h^d) + O(mN_h^d) + (O(mN_p) = O(mN_h^d + mN_p)$. Here we have ignored the storage cost of axillary matrices in the DyBO formulation, i.e., matrices $\mathbf{C}, \mathbf{D}$ and $G_* \in \mathbb{R}^{m\times m}$, which are just $O(m^2)$.

The above discussion regarding the storage complexity is summarized in Table 1. Typically, the number of spatial grid nodes is much larger than the number of polynomial basis, i.e., $N_h^d \gg N_p$. Thus, the reduction of DyBO-gPC compared to gPC in terms of memory consumption is $O\left(\frac{m}{N_p}\right)$.

Table 1
Storage complexity comparison between gPC and DyBO-gPC methods.

Method	Variables to update	Storage complexity	Reduction
gPC (10)	$\bar{v}, \mathbf{V}$	$O(N_pN_h^d)$	NA
DyBO-gPC (7)	$\bar{u}, \mathbf{U}, \mathbf{A}$	$O(mN_h^d + mN_p)$	$O\left(\frac{m}{N_p}\right) + O\left(\frac{m}{N_h^d}\right) \approx O\left(\frac{m}{N_p}\right)$

3.2. Computational costs for the quadratic nonlinear PDE driven by stochastic forces

We next consider the computational complexity in terms of computational time. Unlike the analysis of the storage complexity in the previous section, the analysis of computational time requires knowing the specific form of the stochastic differential operator $\mathcal{L}$. Here, we consider a typical scenario, i.e., the quadratic nonlinear PDE driven by stochastic forces, where quadratic nonlinear PDEs are defined as second-order polynomials of the solution u and its partial derivatives of any orders.

As has been shown in the first part of our paper [9], the DyBO-gPC formulation is a good approximation to the gPC formulation if m is chosen properly. This in turn implies the computational time step sizes used to numerically integrate both systems should be comparable. Thus, to compare the total computational time, we only need to compare the time required to evaluate the right-hand sides if the same explicit ODE solver is adopted for both gPC and DyBO-gPC formulations. Before we consider each case, we make the following two assumptions regarding the computational complexity of the spatial derivatives and the stochastic forces.

Assumption 1. In our complexity analysis, we assume that the spatial derivative can be computed in linear time and the gPC expansion of stochastic force **F** can be evaluated in linear time.

Under Assumption 1, any quadratic nonlinear PDE driven by stochastic forces is equivalent to the following SPDE in regard to computational cost

$$\frac{\partial u}{\partial t} = \mathcal{L}u = (\mathring{\mathcal{L}}\,u)^2 + f = (\mathring{\mathcal{L}}\,u)^2 + \mathbf{F}\mathbf{H}^T, \tag{11}$$

where $\mathring{\mathcal{L}}$ is a deterministic linear differential operator.

We first consider the computational complexity of the gPC formulation (10). With the gPC expansion of the solution $v = \bar{v} + \mathbf{V}\mathbf{H}^T$, simple calculations give the gPC formulation for SPDE (11)

$$\frac{\partial \bar{v}}{\partial t} = (\mathring{\mathcal{L}}\,\bar{v})^2 + \mathring{\mathcal{L}}\,\mathbf{V}\,\mathring{\mathcal{L}}\,\mathbf{V}^T, \tag{12a}$$

$$\frac{\partial \mathbf{V}}{\partial t} = 2\,\mathring{\mathcal{L}}\,\bar{v}\,\mathring{\mathcal{L}}\,\mathbf{V} + \left(\mathring{\mathcal{L}}\,v_{\alpha}\,\mathring{\mathcal{L}}\,v_{\beta}\mathfrak{T}^{(\mathbf{H})}_{\alpha\beta\gamma}\right)_{1\times\gamma} + \mathbf{F}, \tag{12b}$$

where the third-order tensor $\mathfrak{T}^{(\mathbf{H})} = (\mathbb{E}[\mathbf{H}_{\alpha}\mathbf{H}_{\beta}\mathbf{H}_{\gamma}])_{\alpha\beta\gamma}$. The computational cost of some typical terms on the right hand sides of (12) is listed in Table 2. Note that the third-order tensor $\mathfrak{T}^{(\mathbf{H})}$ only depends on the polynomial basis **H** and can be pre-computed, so its computational cost is ignored in this analysis. To evaluate a single entry of row vector $\left(\mathring{\mathcal{L}}\,v_{\alpha}\,\mathring{\mathcal{L}}\,v_{\beta}\mathfrak{T}^{(\mathbf{H})}_{\alpha\beta\gamma}\right)_{1\times\gamma}$, we have to compute the summation $\sum_{\alpha,\beta\in\mathfrak{J}}\mathring{\mathcal{L}}\,v_{\alpha}\,\mathring{\mathcal{L}}\,v_{\beta}\mathfrak{T}^{(\mathbf{H})}_{\alpha\beta\gamma}$, which costs $O(N_p^2N_h^d)$, because a single evaluation of $\mathring{\mathcal{L}}\,v_{\alpha}$ costs $O(N_h^d)$ and a total of N_p^2 terms are summed up. Therefore, the total cost of the whole row vector is $O(N_p^3N_h^d)$.

Next we consider the computational cost of DyBO-gPC. With the truncated KL expansion of the solution $\widetilde{u} = \bar{u} + \mathbf{U}\mathbf{A}^T\mathbf{H}^T$, simple calculations give the DyBO-gPC formulation for SPDE (11),

$$\frac{\partial \bar{u}}{\partial t} = (\mathring{\mathcal{L}}\,\bar{u})^2 + \mathring{\mathcal{L}}\,\mathbf{U}\,\mathring{\mathcal{L}}\,\mathbf{U}^T, \tag{13a}$$

$$\frac{\partial \mathbf{U}}{\partial t} = -\mathbf{U}\mathbf{D}^T + 2\,\mathring{\mathcal{L}}\,\bar{u}\,\mathring{\mathcal{L}}\,\mathbf{U} + \underbrace{\left(\mathcal{L}^{\circ}u_i\mathcal{L}^{\circ}u_jA_{\alpha i}A_{\beta j}A_{\gamma k}\mathfrak{T}^{(\mathbf{H})}_{\alpha\beta\gamma}\right)_{1\times k}}_{\text{Term }\mathfrak{A}} + \mathbf{F}\mathbf{A}, \tag{13b}$$

$$\frac{\partial \mathbf{A}}{\partial t} = -\mathbf{A}\mathbf{C}^T + 2\mathbf{A}\left\langle \mathring{\mathcal{L}}\,\bar{u}\,\mathring{\mathcal{L}}\,\mathbf{U}^T, \mathbf{U}\right\rangle \Lambda_{\mathbf{U}}^{-1} + \underbrace{\left(\mathfrak{T}^{(\mathbf{U})}_{ijk}A_{\alpha i}A_{\beta j}\mathfrak{T}^{(\mathbf{H})}_{\alpha\beta\gamma}\right)_{\gamma\times k}}_{\text{Term }\mathfrak{B}} + \left\langle \mathbf{F}^T, \mathbf{U}\right\rangle \Lambda_{\mathbf{U}}^{-1}, \tag{13c}$$

where the third-order m-by-m-by-m tensor $\mathfrak{T}^{(\mathbf{U})} = \left(\left\langle \mathring{\mathcal{L}}\,u_i\,\mathring{\mathcal{L}}\,u_j, u_k\right\rangle\right)_{ijk}$, and matrices **C** and **D** can be solved from the linear system (4) with

Table 2
The computational cost of the gPC formulation.

Term	$(\mathring{\mathcal{L}}\,\bar{v})^2$	$\mathring{\mathcal{L}}\,\mathbf{V}\,\mathring{\mathcal{L}}\,\mathbf{V}^T$	$\left(\mathring{\mathcal{L}}\,v_{\alpha}\,\mathring{\mathcal{L}}\,v_{\beta}\mathfrak{T}^{(\mathbf{H})}_{\alpha\beta\gamma}\right)_{1\times\gamma}$	**F**	Total
Time	$O(N_h^d)$	$O(N_pN_h^d)$	$O(N_p^3N_h^d)$	$O(mN_h^d)$	$O(N_p^3N_h^d)$

Table 3
The computational costs of the DyBO-gPC formulation.

Term	$\overset{\circ}{\mathcal{L}}\mathbf{U}\overset{\circ}{\mathcal{L}}\mathbf{U}^T$	$\left\langle \mathbf{U}^T, \overset{\circ}{\mathcal{L}}\,\bar{u}\,\overset{\circ}{\mathcal{L}}\,\mathbf{U} \right\rangle$	$\mathfrak{T}^{(\mathbf{U})}$	Term $\mathfrak{A}, \mathfrak{B}$ or $\mathfrak{C}$	Total
Time	mN_h^d	$m^2N_h^d$	$m^3N_h^d$	$mN_p^3 + m^3N_h^d$	$mN_p^3 + m^3N_h^d$

$$\Lambda_{\mathbf{U}}G_* = 2\left\langle \mathbf{U}^T, \overset{\circ}{\mathcal{L}}\,\bar{u}\,\overset{\circ}{\mathcal{L}}\,\mathbf{U} \right\rangle + \underbrace{\left(\mathfrak{T}_{ijk}^{(\mathbf{U})} A_{\alpha i} A_{\beta j} A_{\gamma l} \mathfrak{T}_{\alpha\beta\gamma}^{(\mathbf{H})} \right)_{k\times l}}_{\text{Term } \mathfrak{C}} + \left\langle \mathbf{U}^T, \mathbf{FA} \right\rangle. \tag{14}$$

Please note that the Einstein summation convention is implicitly assumed and the matrix-tensor product should be computed in a recursive way, i.e., $A_{\alpha i}A_{\beta j}A_{\gamma k}\mathfrak{T}_{\alpha\beta\gamma}^{(\mathbf{H})} = A_{\alpha i}\left(A_{\beta j}\left(A_{\gamma k}\mathfrak{T}_{\alpha\beta\gamma}^{(\mathbf{H})}\right)\right)$. It is not difficult to find out such products can be computed in order of $O(mN_p^3)$.

The computational costs of some typical terms on the right hand side of the DyBO-gPC formulation (13) are given in Table 3, where the estimate of term $\mathfrak{B}$ goes as follows. The computation of $\mathfrak{T}^{(\mathbf{U})}$ in term $\mathfrak{B}$ costs $O(m^3N_h^d)$, while the computation of the matrix-tensor product $A_{\alpha i}A_{\beta j}\mathfrak{T}_{\alpha\beta\gamma}^{(\mathbf{H})}$ costs $O(mN_p^3)$. The last step of computing tensor–tensor product costs $O(N_pm^3)$. Thus, the total computational cost of term $\mathfrak{B}$ in Eq. (13c) is $O(m^3N_h^d) + O(mN_p^3) + O(N_pm^3) \leqslant O(m^3N_h^d) + O(mN_p^3)$ since $m \ll N_p$.

In light of the above discussions, the ratio of the computational costs between DyBO-gPC and gPC for the quadratic nonlinear PDE driven by stochastic force is

$$O\left(\frac{m}{N_h^d}\right) + O\left(\left(\frac{m}{N_p}\right)^3\right) \approx O\left(\left(\frac{m}{N_p}\right)^3\right) = O\left(m^\alpha N_p^{-\beta}\right), \tag{15}$$

where the exponents $\alpha = 3$ and $\beta = 3$. In Section 6.2, we will numerically verify these two exponents for the Navier–Stokes equation driven by stochastic forces.

Remark 1. If the distribution of ξ_i's is Gaussian, the tensor $\mathfrak{T}^{(\mathbf{H})}$ can be quite sparse, i.e., a few non-zero entries out of total N_p^3 entries. However, this may not be the case for general distributions, so we do not explore this sparsity in the above analysis. Later in numerical results, we will show that even if such sparsity is explored in numerical implementations of the gPC method, our DyBO-gPC method is still superior. We should also emphasize that gPC is a forward-model independent procedure while DyBO is derived from the forward model. In this sense, DyBO uses more information about the forward model than gPC.

3.3. Parallel computation strategy

Nowadays, parallelization almost becomes an indispensable tool for successful numerical simulations of PDEs in industrial applications. Although the proposed DyBO methods have explored the inherent sparsity within SPDEs themselves, further computational reductions by parallelization are still necessary, especially for spatially three-dimensional SPDEs with multiple physical components. Based on the computational complexity analysis in the previous section, we propose a parallelization strategy based on domain decomposition for the quadratic nonlinear PDE driven by a stochastic force. Specifically, the computation costs of the third-order tensor $\mathfrak{T}^{\mathbf{U}}$, term $\mathfrak{A}, \mathfrak{B}$ and $\mathfrak{C}$, dominate others and bear prohibitive costs of $O(m^3N_h^d)$. Without resorting to other fancy parallelization techniques, the definitions of these terms actually suggest a simple strategy. We explain this in details for the computation of the third-order tensor $\mathfrak{T}^{(\mathbf{U})}$ while the same strategy applies to other three terms similarly.

Suppose the whole spatial domain $\mathcal{D}$ is partitioned to Q disjoint subdomains $\mathcal{D}_q$'s, i.e., $\cup_{q=1}^{Q}\mathcal{D}_q = \mathcal{D}$ and $\mathcal{D}_{q_1} \cap \mathcal{D}_{q_2} = \emptyset$ for $q_1 \neq q_2$. From the definition of $\mathfrak{T}^{(\mathbf{U})}$, each entry

$$\mathfrak{T}_{ijk}^{(\mathbf{U})} = \int_{\mathcal{D}} \overset{\circ}{\mathcal{L}}u_i\overset{\circ}{\mathcal{L}}u_ju_k\mathrm{d}x = \sum_{q=1}^{Q}\int_{\mathcal{D}_q}\overset{\circ}{\mathcal{L}}u_i\overset{\circ}{\mathcal{L}}u_ju_k\mathrm{d}x = \sum_{q=1}^{Q}\mathfrak{T}_{ijk}^{(\mathbf{U},q)}, \tag{16}$$

where $\mathfrak{T}^{(\mathbf{U},q)}$ is the portion of $\mathfrak{T}^{(\mathbf{U})}$ on the q'th subdomain.

Assume Q processors or computational nodes are available and the qth processor is assigned to the subdomain $\mathcal{D}_q$. On the qth processor, only the solutions constrained to the subdomain $\mathcal{D}_q$, i.e., $\bar{u}|_{\mathcal{D}_i}$ and $\mathbf{U}|_{\mathcal{D}_i}$, are stored in in-core memory. Thus, each process can compute its own portion of the third-order tensor $\mathfrak{T}^{(\mathbf{U},q)}$ via Eq. (16). The result on each subdomain will be combined at the end to get $\mathfrak{T}^{(\mathbf{U})}$. The partition of domain may be problem-dependent. Numerical examples that confirm the speedup of the parallel computation strategy will be given in Section 6.

4. Adaptive DyBO algorithms

So far, we assume that the number of the spatial and stochastic basis pairs, $\{u_i, Y_i\}$'s, in the DyBO formulation is fixed to some integer m, which determines the number of functions $\mathbf{U}$ and the size of matrix $\mathbf{A}$. Fixing m for all times is not a good strategy for practical applications. For example, some spatial and stochastic basis pairs may become negligibly small as the system evolves. Keeping such pairs in computation not only wastes computational resource and increases computational time, but may also bring in unexpected numerical instability since extremely small spatial modes may lead to ill-condition of the evolution system. On the other hand, some previously neglected mode pairs may become important later on. Ignoring them may introduce $O(1)$ numerical errors. Therefore, developing an adaptive strategy to add or remove mode pairs dynamically is important for the success of DyBO method. In this section, we propose an adaptive strategy to remove and add basis pairs on the fly for the DyBO-gPC formulation (7).

4.1. Type-KL error analysis

Our adaptive strategy is based on the analysis of a special type of error, which we call *Type-KL* error and is defined as follows

$$\bar{\epsilon} = \bar{u} - \bar{v}, \tag{17a}$$

$$\boldsymbol{\epsilon} = \mathbf{U} - \mathbf{V}\mathbf{A}, \tag{17b}$$

where $v = \bar{v} + \mathbf{V}\mathbf{H}^T$ is the gPC solution defined in (10). Simple calculations give

$$\frac{\partial \bar{\epsilon}}{\partial t} = \mathbb{E}[\mathcal{L}\widetilde{u} - \mathcal{L}v], \tag{18a}$$

$$\frac{\partial \boldsymbol{\epsilon}}{\partial t} = \mathbb{E}\left[(\tilde{\mathcal{L}}\widetilde{u} - \tilde{\mathcal{L}}v)\mathbf{H}\right]\mathbf{A} + \boldsymbol{\epsilon}\mathbf{D} + \mathbf{V}(\mathbf{A}\mathbf{A}^T - \mathbf{I})\left\langle \mathbb{E}\left[\tilde{\mathcal{L}}\widetilde{u}\mathbf{H}^T\right], \mathbf{U}\right\rangle \Lambda_{\mathbf{U}}^{-1}, \tag{18b}$$

where Eq. (7b) (7c) and Eq. (4c) are used.

4.2. The adaptive algorithm

The strategy to remove modes is simple. Since the stochastic basis $\mathbf{Y}$ is orthonormal, i.e., $\mathbb{E}[Y_i^2] = 1$, we only need to check the norm of $u_i(x,t)$ to evaluate the importance of the mode pair (u_i, Y_i). At the end of each time step, we compute $\lambda_i = \|u_i\|^2$ and drop the ith pair $\{u_i, Y_i\}$ if $\lambda_i < \eta\lambda_{\max}$, where $\eta \in (0,1)$ is a pre-selected threshold and $\lambda_{\max} = \max_{i=1,2,\ldots,m}\lambda_i$.

The situation to add mode pairs is more involved. Essentially, we want an algorithm to know *when* and *what* to add without sacrificing too much computational efficiency. A naive approach would be adding some spatial and stochastic mode pair if the smallest eigenvalue rises above some threshold, i.e., $\lambda_{\min} > \eta\lambda_{\max}$. An immediate question is what spatial function and random variable should be used as the initial conditions for the new spatial mode $u_{m+1}(x,t)$ and the stochastic mode $\mathbf{H}a_{m+1}(t)$ with $a_{m+1}(t) \in \mathbb{R}^{N_p\times 1}$ at some time $t = s$. What's more, the newly added mode pair may remain small and be removed later, which may happen repeatedly and should be avoided. In other words, we should estimate the growth rate of the largest unresolved eigenvalue, i.e., $\frac{\mathrm{d}\lambda_{m+1}}{\mathrm{d}t}$ or $\frac{\mathrm{d}\sqrt{\lambda_{m+1}}}{\mathrm{d}t}$ and check if it may potentially grow above the threshold, i.e., $\frac{\mathrm{d}\sqrt{\lambda_{m+1}}}{\mathrm{d}t}\Delta T \geqslant \sqrt{\eta\lambda_{\max}}$ after some finite time interval ΔT. It turns out that these two questions are related.

The basis idea for adding mode pairs is to start from the same initial condition, evolve the SPDE system by gPC and DyBO-gPC methods for a short time Δs, respectively. We use the solution discrepancy at final time to estimate the growth rate of unresolved eigenvalues. If the growth rate is above certain threshold, we will use the dominant mode pair as the initial conditions for the new spatial and stochastic modes. This heuristic conjecture can be made more rigorous by looking at the type-KL errors that we discussed in the previous subsection.

Suppose at time $t = s$, the DyBO-gPC solution

$$\widetilde{u}(x,s,\xi) = \bar{u}(x,s) + \mathbf{U}(x,s)\mathbf{A}(s)^T\mathbf{H}(\xi)^T, \tag{19}$$

remains a good approximation to the gPC solution, i.e.,

$$v(x,s,\xi) \approx \widetilde{u}(x,s,\xi). \tag{20}$$

However, as the system continues to evolve for a short time, discrepancy between these two solutions arises and cannot be ignored any more. We should enrich the DyBO-gPC basis to capture this discrepancy, otherwise the type-KL error will accumulate significantly and affect the accuracy of the DyBO-gPC solution. Specifically, we can add one pair of spatial mode $\sqrt{\lambda_{m+1}(t)}\check{u}_{m+1}(x,t)$ and stochastic modes $\mathbf{H}(\xi(\omega))a_{m+1}(t)$ to compensate such discrepancy, i.e., at time $t = s$,

$$\begin{aligned}\widetilde{u}(x,s,\xi) &= \bar{u}(x,s) + (\mathbf{U}(x,s), u_{m+1}(x,s))(\mathbf{A}(s), a_{m+1}(s))^T\mathbf{H}(\xi)^T \\ &= \bar{u}(x,s) + (\mathbf{U}(x,s), \sqrt{\lambda_{m+1}(s)}\check{u}_{m+1}(x,s))(\mathbf{A}(s), a_{m+1}(s))^T\mathbf{H}(\xi)^T,\end{aligned} \tag{21}$$

where $\lambda_{m+1} \approx 0$, $\langle \mathbf{U}(x,s), \check{u}_{m+1}(x,s)\rangle = \mathbf{0}$, $\|\check{u}(x,s)\|_{\mathbb{L}^2(\mathcal{D})} = 1$, $\mathbf{A}(s)^T a_{m+1}(s) = \mathbf{0}$. Both $\sqrt{\lambda_{m+1}(t)}\check{u}_{m+1}(x,t)$ and $\mathbf{H}(\xi(\omega))a_{m+1}(t)$ are unknown at this moment $t = s$ and will be derived later. After including the unresolved $(m+1)$th basis pair, the type-KL error is given by Eq. (18b). Now let's estimate both sides at time $t = s$. From Eq. (20), we know that $\mathbf{V}(x,s) = \mathbf{U}(x,s)\mathbf{A}(s)^T$, so

$$\epsilon(x,s) = \left(\mathbf{U}(x,s), \sqrt{\lambda_{m+1}(s)}\check{u}_{m+1}(x,s)\right) - \mathbf{U}(x,s)\mathbf{A}(s)^T(\mathbf{A}(s), a_{m+1}(s)) = \left(\mathbf{0}, \sqrt{\lambda_{m+1}(s)}\check{u}_{m+1}(x,s)\right) = \mathbf{0} \quad \text{as } \lambda_{m+1} \ \to \ 0,$$

where we have used orthogonality of $\mathbf{A}(s)$ and $a_{m+1}(s)$ in the first equality. This simply implies that the second term $\boldsymbol{\epsilon}\mathbf{D}$ on the right hand side of Eq. (18b) is zero at time $t = s$. Similar calculations reveal the third term on the right hand side is also zero as $\lambda_{m+1} \ \to \ 0$, i.e.,

$$\mathbf{V}(x,s)\left((\mathbf{A}(s), a_{m+1}(s))(\mathbf{A}(s), a_{m+1}(s))^T - \mathbf{I}\right) = \mathbf{U}(x,s)\mathbf{A}(s)^T\left((\mathbf{A}(s), a_{m+1}(s))(\mathbf{A}(s), a_{m+1}(s))^T - \mathbf{I}\right) = \mathbf{0}.$$

Therefore, only the first term on the right hand side of Eq. (18b) really contributes, which can be approximated to the first-order accuracy $O(\Delta s)$ as follows,

$$\mathbb{E}[\tilde{\mathcal{L}}v(x,s,\xi)\mathbf{H}] = \mathbb{E}\left[\frac{v(x,s+\Delta s,\xi) - v(x,s,\xi)}{\Delta s}\mathbf{H}\right],$$

$$\mathbb{E}[\tilde{\mathcal{L}}\tilde{u}(x,s,\xi)\mathbf{H}] = \mathbb{E}\left[\frac{\tilde{u}(x,s+\Delta s,\xi) - \tilde{u}(x,s,\xi)}{\Delta s}\mathbf{H}\right].$$

Because $\tilde{u}|_{t=s} \to v|_{t=s}$ as $\lambda_{m+1} \to 0$,

$$\mathbb{E}[(\tilde{\mathcal{L}}\tilde{u} - \tilde{\mathcal{L}}v)\mathbf{H}](\mathbf{A}, a_{m+1})|_{t=s} \approx \mathbb{E}\left[\frac{\tilde{u}(x,s+\Delta s,\xi) - v(x,s+\Delta s,\xi)}{\Delta s}\mathbf{H}\right](\mathbf{A}(s), a_{m+1}(s)).$$

The last component of the above equality is

$$\mathbb{E}\left[(\tilde{\mathcal{L}}\tilde{u} - \tilde{\mathcal{L}}v)\mathbf{H}\right]a_{m+1}|_{t=s} = \mathbb{E}\left[\frac{\tilde{u}(x,s+\Delta s,\xi) - v(x,s+\Delta s,\xi)}{\Delta s}\mathbf{H}\right]a_{m+1}(s) + O(\Delta s). \tag{22}$$

Now we calculate the last component on the left hand side of Eq. (18b), i.e., $\frac{\partial \epsilon_{m+1}}{\partial t}(x,s,\xi)$. As $\lambda_{m+1} \ \to \ 0$, we have

$$\left.\frac{\partial \epsilon_{m+1}}{\partial t}\right|_{t=s} = \left.\frac{\partial\left(\sqrt{\lambda_{m+1}}\check{u}_{m+1}\right)}{\partial t}\right|_{t=s} = \left.\frac{d\sqrt{\lambda_{m+1}}}{dt}\check{u}_{m+1}\right|_{t=s} + \left.\sqrt{\lambda_{m+1}}\frac{\partial \check{u}_{m+1}}{\partial t}\right|_{t=s} = \frac{d\sqrt{\lambda_{m+1}}}{dt}(s)\check{u}_{m+1}(x,s). \tag{23}$$

Combining the above discussion, we have the following equality from Eq. (17),

$$\frac{d\sqrt{\lambda_{m+1}}}{dt}(s)\check{u}_{m+1}(x,s) \approx \mathbb{E}\left[\frac{\tilde{u}(x,s+\Delta s,\xi) - v(x,s+\Delta s,\xi)}{\Delta s}\mathbf{H}\right]a_{m+1}(s). \tag{24}$$

Now consider the KL expansion of the solution discrepancy at time $t = s + \Delta s$, i.e.,

$$\Delta\tilde{u}(x,s+\Delta s,\xi) = \tilde{u}(x,s+\Delta s,\xi) - v(x,s+\Delta s,\xi) = \sqrt{\theta_1}w_1(x)b_1^T\mathbf{H}^T + \cdots, \tag{25}$$

where $\theta_1, w_1(x)$ and $\mathbf{H}b_1$ are the largest eigenvalue, normalized spatial and stochastic basis, respectively. The above equality implies that the growth rate of the largest unresolved eigenvalue λ_{m+1} can be estimated from the largest eigenvalue of $\Delta\tilde{u}$. What's more, a sensible choice of initial conditions for the newly added basis pair $u_{m+1}(x,s)$ and $a_{m+1}(s)$ would be the largest spatial and stochastic mode of $\Delta\tilde{u}(x,s+\Delta s,\xi)$, i.e., $\sqrt{\theta_1}w_1(x)$ and $\mathbf{H}b_1$. This strategy involves computation of gPC solutions for a short time Δs, which can be expensive. Instead of invoking such strategy every time step, we invoke such procedure every duration $\Delta T, \Delta T \gg \Delta s$. See Fig. 1 for illustrations.

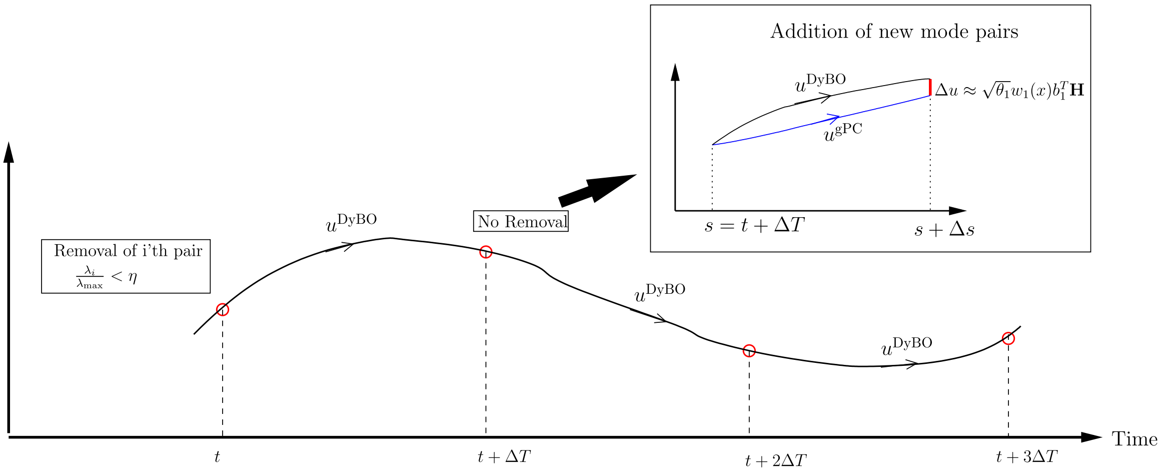


Fig. 1. Illustration of strategies of adding and removing basis pairs.

We remark that a similar strategy of removing and adding spatial and stochastic basis pairs may be developed for DyBO-gSC and DyBO-MC. The adaptive algorithm can be generalized to add more than one pair of spatial and stochastic basis at a time. The corresponding spatial and stochastic basis pairs can be obtained from the KL expansion of $\Delta\widetilde{u}(x,t,\xi)$ in (25). Moreover, we note that the new spatial basis u_{m+1} and new stochastic basis $\mathbf{H}a_{m+11}(t)$ may not be perfectly orthogonal to other basis $\mathbf{U}(x,t)$ and $\mathbf{HA}$ at time $t=s$. Due to the bi-orthogonality-preserving property of the DyBO method (see Theorem 3.1 in [9]), such deviation from the bi-orthogonality will not be amplified.

5. Generalization of the DyBO method to a SPDE system

In addition to computational complexity analysis and adaptivity in changing basis number, the generalizations of the DyBO method will be another focus of this paper. In this section, we will discuss the generalization of the DyBO formulation for SPDE systems.

Many applications involve multiple physical fields, or physical components, for instance, the standard three-dimensional incompressible Naiver-Stokes equations involve four physical components, three velocity components along x-, y-, z-axis and pressure. When compressibility cannot be ignored, e.g., in aerodynamic [1], two additional components, typically density and temperature fields, get involved. Therefore, generalizing the DyBO method for a system of time-dependent SPDEs is important and necessary. More precisely, we consider a system of time-dependent SPDEs as follows:

$$\frac{\partial u_l}{\partial t}(x,t,\omega)=\mathcal{L}_l\{u_1,u_2,\cdots,u_N\},\quad l=1,2,\cdots,N,\quad x\in\mathcal{D}\subset\mathbb{R}^d,\quad t\in[0,T], \tag{26}$$

where each $\mathcal{L}_l$ is a stochastic differential operator acting on the physical components $u_1,u_2,\cdots,u_N$ and N is the total number of physical components. To simplify the notation, we omit the boundary condition and initial condition for each component. When no ambiguity arises, we simply use shorthand notation

$$u=\{u_1,u_2,\cdots,u_N\}\quad\text{and}\quad\mathcal{L}_l u=\mathcal{L}_l\{u_1,u_2,\cdots,u_N\}.$$

Unlike a single SPDE, randomness in (26) introduced through initial conditions, boundary conditions, stochastic forcing terms propagates not only in space and time, but also among different physical components. Randomness introduced by one physical component may affect other components. In general, different physical components may have different stochastic properties. Thus, using a common basis, such as the orthonormal polynomial basis, may not be the most efficient way to represent the solution of a stochastic system. The most compact representations in $\mathbb{L}^2$ sense are the KL expansions of each physical component, which is our starting point to derive the DyBO formulation for the SPDE system (26).

Consider the m_l-term truncated KL expansion of the lth physical component $u_l(x,t,\omega)$,

$$\widetilde{u}_l=\bar{u}_l+\sum_{i=1}^{m_l}u_{li}Y_{li}=\bar{u}_l+\mathbf{U}_l\mathbf{Y}_l^T\approx u_l, \tag{27}$$

where $\mathbf{U}_l$ is a row vector of functions of spatial coordinate x and temporal coordinate t,

$$\mathbf{U}_l(x,t)=(u_{l1}(x,t),u_{l2}(x,t),\cdots,u_{lm_l}(x,t))\in\mathbb{R}^{1\times m_l}$$

and $\mathbf{Y}_l$ is a row vector of random variables,

$$\mathbf{Y}_l(\omega,t)=(Y_{l1}(\omega,t),Y_{l2}(\omega,t),\cdots,Y_{lm_l}(\omega,t))\in\mathbb{R}^{1\times m_l}.$$

With these preparations, we are now ready to derive our DyBO method for a system of SPDEs. We follow the steps in the derivation of DyBO method for a single SPDE (see Section 2 of [9]) by substituting the expansion (27) into the system (26) and using anti-symmetrization operators $\mathcal{Q}$ and $\tilde{\mathcal{Q}}$ to enforce the bi-orthogonality of the spatial and stochastic modes $\mathbf{U}_l$ and $\mathbf{Y}_l$ of each physical component u_l. After projecting the growth rate of the spatial and stochastic modes $\frac{\partial\mathbf{U}_l}{\partial t}$ and $\frac{\mathrm{d}\mathbf{Y}_l}{\mathrm{d}t}$ onto themselves, we arrive at the generalized DyBO formulation for SPDE system (26) (see Cheng's thesis [8] for details).

$$\frac{\partial\bar{u}_l}{\partial t}=\mathbb{E}[\mathcal{L}_l\widetilde{u}], \tag{28a}$$

$$\frac{\partial\mathbf{U}_l}{\partial t}=-\mathbf{U}_l\mathbf{D}_l^T+\mathbb{E}[\tilde{\mathcal{L}}_l\widetilde{u}\mathbf{Y}_l], \tag{28b}$$

$$\frac{\mathrm{d}\mathbf{Y}_l}{\mathrm{d}t}=-\mathbf{Y}_l\mathbf{C}_l^T+\langle\tilde{\mathcal{L}}_l\widetilde{u},\mathbf{U}_l\rangle\mathbf{\Lambda}_{\mathbf{U}_l}^{-1}, \tag{28c}$$

where $l=1,2,\cdots,N$ and $\widetilde{u}=\{\widetilde{u}_1,\ldots,\widetilde{u}_N\}$. The matrices $\mathbf{C}_l$'s and $\mathbf{D}_l$'s can be solved from linear systems

$$\mathbf{C}_l-\mathbf{\Lambda}_{\mathbf{U}_l}^{-1}\tilde{\mathcal{Q}}(\mathbf{\Lambda}_{\mathbf{U}_l}\mathbf{C}_l)=0, \tag{29a}$$

$$\mathbf{D}_l-\mathcal{Q}(\mathbf{D}_l)=0, \tag{29b}$$

$$\mathbf{D}_l^T+\mathbf{C}_l=G_{*l}(\bar{u}_l,\mathbf{U}_l,\mathbf{Y}_l), \tag{29c}$$

with $G_{*l}(\bar{u}_l, \mathbf{U}_l, \mathbf{Y}_l) = \mathbf{\Lambda}_{\mathbf{U}_l}^{-1}\left\langle \mathbf{U}_l^T, \mathbb{E}[\tilde{\mathcal{L}}_l \widetilde{u} \mathbf{Y}_l]\right\rangle \in \mathbb{R}^{m_l \times m_l}$. The boundary conditions and initial conditions for each physical components can be obtained correspondingly. We assume the randomnesses of the SPDE system follows the same distribution, then we can derive the DyBO-gPC formulation for the SPDE system. For DyBO-gPC, the stochastic modes $\mathbf{Y}_l$ are presented in the form of gPC expansions, i.e., $\mathbf{Y}_l(\omega, t) = \mathbf{H}(\boldsymbol{\xi}(\omega))\mathbf{A}_l$, where $\mathbf{A}_l \in \mathbb{R}^{N_p \times m_l}$. The DyBO-gPC formulation for each component is given by

$$\frac{\partial \bar{u}_l}{\partial t} = \mathbb{E}[\mathcal{L}_l \widetilde{u}], \tag{30a}$$

$$\frac{\partial \mathbf{U}_l}{\partial t} = -\mathbf{U}_l \mathbf{D}_l^T + \mathbb{E}[\tilde{\mathcal{L}}_l \widetilde{u} \mathbf{H}]\mathbf{A}_l, \tag{30b}$$

$$\frac{\mathrm{d}\mathbf{A}_l}{\mathrm{d}t} = -\mathbf{A}_l \mathbf{C}_l^T + \left\langle \mathbb{E}[\mathbf{H}^T \tilde{\mathcal{L}}_l \widetilde{u}], \mathbf{U}_l \right\rangle \mathbf{\Lambda}_{\mathbf{U}_l}^{-1}, \tag{30c}$$

where $\mathbf{C}_l(t)$ and $\mathbf{D}_l(t)$ can be solved from

$$G_{*l}(\bar{u}_l, \mathbf{U}_l, \mathbf{Y}_l) = \mathbf{\Lambda}_{\mathbf{U}_l}^{-1}\left\langle \mathbf{U}_l^T, \mathbb{E}[\tilde{\mathcal{L}}_l \widetilde{u} \mathbf{Y}_l]\right\rangle = \mathbf{\Lambda}_{\mathbf{U}_l}^{-1}\left\langle \mathbf{U}_l^T, \mathbb{E}[\tilde{\mathcal{L}}_l \widetilde{u} \mathbf{H}]\right\rangle \mathbf{A}_l. \tag{31}$$

Various theoretical results of the DyBO formulation for single SPDE, such as the preservation of bi-orthogonality and error analysis can be generalized to the DyBO formulation for a SPDE system. The strategies proposed for a single SPDE such as eigenvalues crossings and adding or removing basis pairs can also be generalized to a system of SPDE. Similar results can be obtained for the DyBO-gSC version and DyBO-MC version. More details about the numerical implementation will be given in the next section.

6. Numerical examples

Previous sections highlight the analytical aspects of the DyBO formulation and algorithm, this section demonstrates its success by several numerical examples, each of which emphasizes and verifies some of analytical results in the previous sections. In the first example, a SPDE driven purely by stochastic forces is considered, which shows the necessity of adaptivity in the DyBO method. More involved numerical examples, such as spatially two-dimensional SPDE and/or a system of SPDEs, which require adaptivity, parallelization strategy and other numerical techniques, will also be reported in this section.

6.1. SPDE purely driven by stochastic forces

In the first numerical example, we consider the SPDE driven purely by a stochastic force f, i.e.,

$$\frac{\partial u}{\partial t} = \mathcal{L}u = f(x, t, \xi(\omega)), \quad x \in \mathcal{D} = [0, 1], \quad t \in [0, T], \tag{32}$$

where $\boldsymbol{\xi} = (\xi_1, \xi_2, \cdots, \xi_r)$ are independent standard Gaussian random variables, i.e., $\xi_i \sim \mathcal{N}(0, 1)$. A similar example has been used in the first part of the paper [9] for eigenvalue crossing. Here we consider a different stochastic force f to test the adaptive strategy proposed in Section 4. To construct such force, we consider an exact solution given in the following form,

$$u(x, t, \boldsymbol{\xi}) = \bar{v}(x, t) + \mathbf{V}(x, t)\mathbf{Z}^T(\xi, t), \tag{33}$$

where $\mathbf{V}(x, t) = \mathring{\mathbf{V}}(x)\mathbf{W}_{\mathbf{V}}(t)\mathbf{\Lambda}_{\mathbf{V}}^{\frac{1}{2}}(t), \mathbf{Z}(\xi, t) = \mathring{\mathbf{Z}}(\xi)\mathbf{W}_{\mathbf{Z}}(t), \mathring{\mathbf{V}}(x) = (\mathring{v}_1(x), \cdots, \mathring{v}_m(x))$ with $\left\langle \mathring{v}_i(x), \mathring{v}_j(x)\right\rangle = \delta_{ij}$ and $\mathring{\mathbf{Z}}(\xi) = (\mathring{Z}_1(\xi), \cdots, \mathring{\mathbf{Z}}_m(\xi))$ with $\mathbb{E}[\mathring{Z}_i \mathring{Z}_j] = \delta_{ij}$ for $i, j = 1, 2, \cdots, m$. $\mathbf{W}_{\mathbf{V}}(t)$ and $\mathbf{W}_{\mathbf{Z}}(t)$ are m-by-m orthonormal matrices, and $\mathbf{\Lambda}_{\mathbf{V}}^{\frac{1}{2}}(t)$ is a diagonal matrix. By differentiation, we can get the corresponding stochastic force

$$f = \frac{\partial \bar{v}}{\partial t} + \frac{\partial \mathbf{V}}{\partial t}\mathbf{Z}^T + \mathbf{V}\frac{\mathrm{d}\mathbf{Z}^T}{\mathrm{d}t}. \tag{34}$$

By substituting the above equalities into DyBO-gPC system (7), we arrive at the DyBO-gPC formulation for SPDE (32)

$$\frac{\partial \bar{u}}{\partial t} = \frac{\partial \bar{v}}{\partial t}, \tag{35a}$$

$$\frac{\partial \mathbf{U}}{\partial t} = -\mathbf{U}\mathbf{D}^T + \left(\frac{\partial \mathbf{V}}{\partial t}\mathbf{W}_{\mathbf{Z}}^T + \mathbf{V}\frac{\mathrm{d}\mathbf{W}_{\mathbf{Z}}^T}{\mathrm{d}t}\right)\mathbb{E}[\mathring{\mathbf{Z}}^T \mathbf{H}]\mathbf{A}, \tag{35b}$$

$$\frac{\mathrm{d}\mathbf{A}}{\mathrm{d}t} = -\mathbf{A}\mathbf{C}^T + \mathbb{E}[\mathbf{H}^T \mathring{\mathbf{Z}}]\left\langle \mathbf{W}_{\mathbf{Z}}\frac{\partial \mathbf{V}^T}{\partial t} + \frac{\mathrm{d}\mathbf{W}_{\mathbf{Z}}}{\mathrm{d}t}\mathbf{V}^T, \mathbf{U}\right\rangle \mathbf{\Lambda}_{\mathbf{U}}^{-1} \tag{35c}$$

and

$$G_*(u, \mathbf{U}, \mathbf{A}) = \mathbf{\Lambda}_{\mathbf{U}}^{-1}\left\langle \mathbf{U}, \frac{\partial \mathbf{V}}{\partial t}\mathbf{W}_{\mathbf{Z}}^T + \mathbf{V}\frac{\mathrm{d}\mathbf{W}_{\mathbf{Z}}^T}{\mathrm{d}t}\right\rangle \mathbb{E}[\mathring{\mathbf{Z}}^T \mathbf{H}]\mathbf{A}.$$

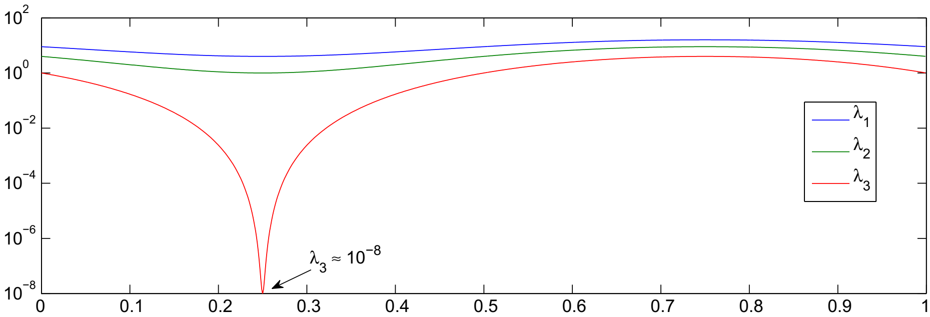


Fig. 2. Eigenvalues are plotted as function of time. λ_3 becomes small near $t = 0.25$.

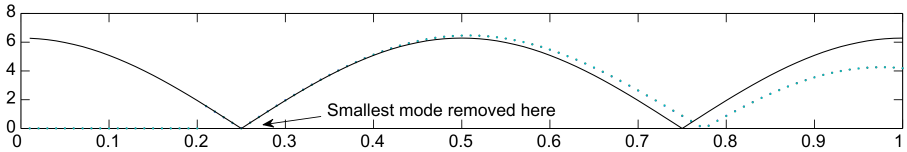


Fig. 3. Change rate of the largest unresolved eigenvalue $\frac{d\sqrt{\lambda_3}}{dt}$. Solid line is the exact solution, while the dotted line are computed the adaptive algorithm based on type-KL error analysis.

We consider a small system $m = 3$ and use the following settings,

$$\mathbf{V}(\overset{\circ}{x}) = (\sqrt{2}\sin(\pi x), \sqrt{2}\sin(5\pi x), \sqrt{2}\sin(9\pi x)), \quad \mathbf{Z}(\overset{\circ}{x}) = (\mathrm{H}_1(\xi_1), \mathrm{H}_2(\xi_1), \mathrm{H}_3(\xi_1)),$$

$$\mathbf{W_V}(t) = \mathbf{P_V}\begin{pmatrix} \cos b_{\mathbf{V}}t & -\sin b_{\mathbf{V}}t & 0 \\ \sin b_{\mathbf{V}}t & \cos b_{\mathbf{V}}t & 0 \\ 0 & 0 & 1 \end{pmatrix}\mathbf{P}_{\mathbf{V}}^T, \quad \mathbf{W_Z}(t) = \mathbf{P_Z}\begin{pmatrix} \cos b_{\mathbf{Z}}t & -\sin b_{\mathbf{Z}}t & 0 \\ \sin b_{\mathbf{Z}}t & \cos b_{\mathbf{Z}}t & 0 \\ 0 & 0 & 1 \end{pmatrix}\mathbf{P}_{\mathbf{Z}}^T,$$

where $b_{\mathbf{V}} = 2.0, b_{\mathbf{Z}} = 2.0, \mathbf{P_V}$ and $\mathbf{P_Z}$ are two orthonormal matrices generated randomly at the beginning, and $\mathbf{H}(\xi) = (\mathrm{H}_1(\xi_1), \mathrm{H}_2(\xi_1), \cdots, \mathrm{H}_5(\xi_1))$. To simulate the scenario where adding and removing mode pairs are necessary, we consider the following eigenvalues

$$\Lambda_{\mathbf{V}}^{\frac{1}{2}} = \mathrm{diag}(3.0001 + \sin(2\pi t), 2.0001 + \sin(2\pi t), 1.0001 + \sin(2\pi t)),$$

where λ_3 becomes very small $\sim 10^{-8}$ near $t = 0.25$. See Fig. 2. We use this example to test the effectiveness of our first adaptive method based on the type-KL error analysis. When the adaptive strategy for adding mode pairs is invoked, it is crucial to know the growth rate of the largest unresolved eigenvalue and avoid adding such mode pair if it continues to be small in the near future ΔT. This is accomplished by computing solutions by DyBO and gPC for a short time Δs and estimating the growth rate $\frac{d\sqrt{\lambda_{m+1}}}{dt}$ from the difference of the two solutions.

In Fig. 3, we verify the accuracy of such estimates, where the third mode pair is intentionally dropped at $t = 0.2$ when it becomes small ($\sim 10^{-3}$) and never put back in the remaining computation. The solid line is the exact growth rate of the largest unresolved eigenvalue, i.e., $\frac{d\sqrt{\lambda_3}}{dt}$, while the dotted line is the estimate. In computations, we actually use different short time duration $\Delta s = 8\delta t, 4\delta t, 2\delta t, \delta t$ to verify the convergence of such estimate. However, all of theses estimates cluster together and cannot be distinguished from the figure. As we can see from Fig. 3, such estimates are very accurate when the largest unresolved eigenvalue is indeed small and become less accurate when the largest unresolved eigenvalue is not so small compared to the resolved ones.

In Fig. 4, we consider the effect of the invoking frequency for adding mode pairs, i.e., $\frac{1}{\Delta T}$. If no mode pair is added, the relative error of STD at $t = 1.0$ is about 26%. When the adaptive algorithm is incorporated, the error can be brought down to $\leqslant 1.5\%$ depending on the invoking frequency. The threshold η in the adaptive algorithm is taken to be 10^{-4} and $\sqrt{\eta} = 10^{-2}$, so we see such difference is relatively marginal. We will continue to demonstrate the effectiveness of the adaptive algorithm in more involved numerical examples in the following subsections.

6.2. 2D stochastic flow

As a model to test numerically the proposed DyBO formulation for a spatially two-dimensional nonlinear SPDE, we consider the incompressible Navier–Stokes equations driven by stochastic forces. Specifically, we consider the stochastic flow in an unit square, i.e., $\mathcal{D} = [0, 1] \times [0, 1]$, with periodic boundary conditions on both spatial directions x and y (see Fig. 5(a)). The governing equation of this stochastic flow is the Stochastic Navier–Stokes equations (SNSEs). For spatially two-dimensional

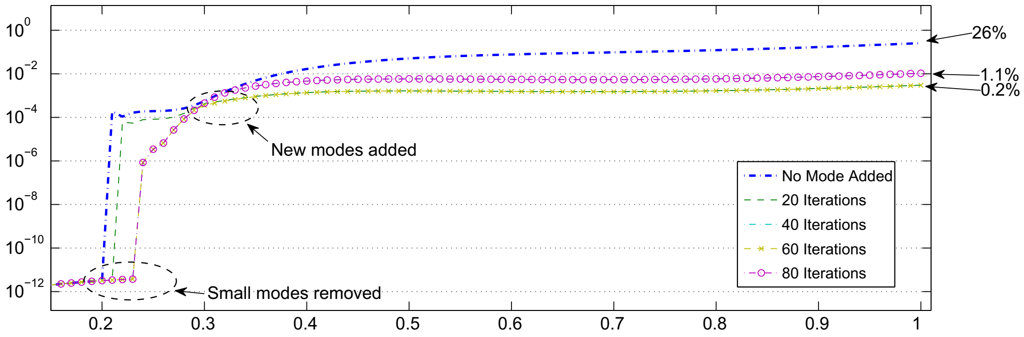
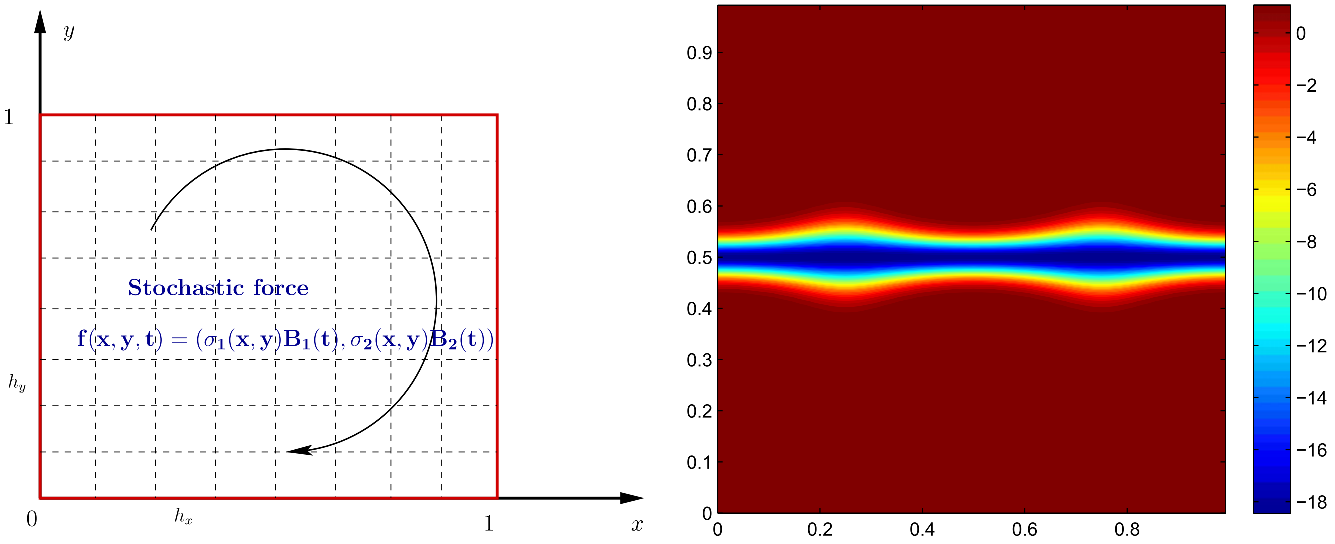


Fig. 4. $\mathbb{L}^2$ relative errors of STD given by DyBO with adding or removing basis pairs.



(a) Diagram of flow domain

(b) Initial vorticity field

Fig. 5. Stochastic flows driven by stochastic force f in 2D unit square.

incompressible flow problems, it is more convenient to use the vorticity-stream function formulation. The vorticity-stream function formulation gives $w = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$ and ψ, with velocity $u = \frac{\partial \psi}{\partial y}$ and $v = -\frac{\partial \psi}{\partial x}$. The vorticity-stream formulation is given by

$$\frac{\partial w}{\partial t} = \mathcal{L}w = -\left(u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y}\right)w + \nu\Delta w + \left(\frac{\partial f_2}{\partial x} - \frac{\partial f_1}{\partial y}\right), \tag{36}$$

$$-\Delta\psi = w, \tag{37}$$

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x}. \tag{38}$$

We assume the randomness is given in terms of r independent standard Gaussian random variables, $\boldsymbol{\xi} = (\xi_1, \xi_2, \cdots, \xi_r)$, and the initial vorticity is deterministic, i.e., $w(x, y, 0, \xi) = \mathring{w}(x, y)$. That is to say, the randomness is injected into the system only through the stochastic force $f = (f_1, f_2)$. In the following numerical example, we choose $\nu = 2.0 \times 10^{-4}$ and adopt the initial vorticity field used in [14,22],

$$\mathring{w}(x,y) = const - \frac{1}{2\delta_1}\exp\left(-\frac{I(x)(y-0.5)^2}{2\delta_1^2}\right),$$

where $I(x) = 1 + \delta_2(\cos(4\pi x) - 1)$ and the constant is taken such that $\int_{\mathcal{D}} \mathring{w} dxdy = 0$. $\delta_1 = 0.025$ and $\delta_2 = 0.3$, so the initial vorticity concentrates in a narrow band along $y = 0.5$ as shown in Fig. 5(b), which models a perturbed flat vortex sheet in the limit that $\delta_1 \to 0$. For the driving stochastic force f, we consider an approximated version of Brownian force $f = (\sigma_1(x, y)B_1(t), \sigma_2(x, y)B_2(t))$ (see [8,9,22,14] for the details of the construction)

$$\left(\frac{\partial f_2}{\partial x} - \frac{\partial f_1}{\partial y}\right) = -\frac{\partial \sigma_1}{\partial y}\sum_{i=1}^{4}\frac{1}{\sqrt{T}}M_i\left(\frac{t}{T}\right)\xi_i + \frac{\partial \sigma_2}{\partial x}\sum_{i=5}^{8}\frac{1}{\sqrt{T}}M_{i-4}\left(\frac{t}{T}\right)\xi_i = \mathbf{FH}^T. \tag{39}$$

The functions σ_1 and σ_2 are chosen such that

$$\frac{\partial\sigma_1}{\partial y} = 0.3\pi\cos(2\pi x)\cos(2\pi y), \quad \frac{\partial\sigma_2}{\partial x} = 0.3\pi\sin(2\pi x)\sin(2\pi y).$$

The derivations of gPC or DyBO-gPC formulations of SNSE is standard. By considering the gPC expansion $w = \bar{w} + \mathbf{W}\mathbf{H}^T$, we can derive the gPC formulation of SNSE (36),

$$\frac{\partial\bar{w}}{\partial t} = \nu\Delta\bar{w} - \mathfrak{D}_{(\bar{u},\bar{v})\bar{w}} - \mathfrak{D}_{(\mathbf{U},\mathbf{V})\mathbf{W}}, \tag{40a}$$

$$\frac{\partial\mathbf{W}}{\partial t} = \nu\Delta\mathbf{W} - \mathfrak{D}_{(\bar{u},\bar{v})\mathbf{W}^T} - \mathfrak{D}_{(\mathbf{U},\mathbf{V})\bar{w}} - \left(\mathfrak{D}_{(u_{\boldsymbol{\alpha}},v_{\boldsymbol{\alpha}})w_{\boldsymbol{\beta}}}\mathfrak{T}^{(\mathbf{H})}_{\boldsymbol{\alpha\beta\gamma}}\right)_{1\times\gamma} + \mathbf{F}, \tag{40b}$$

where $\mathfrak{D}_{(\cdot,\cdot)(\cdot)}$ is a generalized material derivative defined as follows. For a scalar or row-vector field θ under scalar or row-vector velocity field u and v,

$$\mathfrak{D}_{(u,v)\theta} = \begin{cases} u\frac{\partial\theta}{\partial x} + v\frac{\partial\theta}{\partial y}, & u, v, \theta \text{ are scalars},\\ u\frac{\partial\theta}{\partial x} + v\frac{\partial\theta}{\partial y}, & u, v \text{ are scalars and } \theta \text{ is a row vector},\\ \frac{\partial\theta}{\partial x}u + \frac{\partial\theta}{\partial y}v, & u, v \text{ are row vectors and } \theta \text{ is a scalar},\\ u\frac{\partial\theta^T}{\partial x} + v\frac{\partial\theta^T}{\partial y}, & u, v, \theta \text{ are row vectors}, \end{cases} \tag{41}$$

where u and v can be row vectors of the same length and the right hand side is understood in the usual sense of vector-vector multiplications or scalar-vector multiplications.

The DyBO-gPC formulation of SNSE (36) can be obtained by considering the m-term truncated KL expansion $w = \bar{w} + \mathbf{W}\mathbf{A}^T\mathbf{H}^T$, see A for more details about its derivation.

$$\frac{\partial\bar{w}}{\partial t} = \nu\Delta\bar{w} - \mathfrak{D}_{(\bar{u},\bar{v})\bar{w}} - \mathfrak{D}_{(\mathbf{U},\mathbf{V})\mathbf{W}}, \tag{42a}$$

$$\frac{\partial\mathbf{W}}{\partial t} = -\mathbf{W}\mathbf{D}_w^T + \left[\nu\Delta\mathbf{W} - \mathfrak{D}_{(\bar{u},\bar{v})\mathbf{W}} - \mathfrak{D}_{(\mathbf{U},\mathbf{V})\bar{w}}\right] - \left[\mathfrak{D}_{(u_i,v_i)w_j}A_{\boldsymbol{\alpha}i}A_{\boldsymbol{\beta}j}A_{\gamma k}\mathfrak{T}^{(\mathbf{H})}_{\boldsymbol{\alpha\beta\gamma}}\right]_{1\times k} + \mathbf{F}\mathbf{A}, \tag{42b}$$

$$\frac{\mathrm{d}\mathbf{A}}{\mathrm{d}t} = \mathbf{A}\left(-\mathbf{C}_w^T + \left\langle\nu\Delta\mathbf{W}^T - \left(\mathfrak{D}_{(\bar{u},\bar{v})\mathbf{W}}\right)^T - \left(\mathfrak{D}_{(\mathbf{U},\mathbf{V})\bar{w}}\right)^T, \mathbf{W}\right\rangle\Lambda_{\mathbf{W}}^{-1}\right) - \left[\mathfrak{T}^{(\mathbf{W})}_{ijk}A_{\boldsymbol{\alpha}i}A_{\boldsymbol{\beta}j}\mathfrak{T}^{(\mathbf{H})}_{\boldsymbol{\alpha\beta\gamma}}\right]_{\gamma\times k}\Lambda_{\mathbf{W}}^{-1} + \left\langle\mathbf{F}^T, \mathbf{W}\right\rangle\Lambda_{\mathbf{W}}^{-1}, \tag{42c}$$

where matrices $\mathbf{C}_w$ and $\mathbf{D}_w$ can be solved from the linear system (4) from G_{*w},

$$\Lambda_{\mathbf{W}}G_{*w} = \left\langle\mathbf{W}^T, \nu\Delta\mathbf{W} - \mathfrak{D}_{(\bar{u},\bar{v})\mathbf{W}} - \mathfrak{D}_{(\mathbf{U},\mathbf{V})\bar{w}}\right\rangle - [\mathfrak{T}^{(\mathbf{W})}_{ijk}A_{\boldsymbol{\alpha}i}A_{\boldsymbol{\beta}j}A_{\gamma l}\mathfrak{T}^{(\mathbf{H})}_{\boldsymbol{\alpha\beta\gamma}}]_{k\times l} + \left\langle\mathbf{W}^T, \mathbf{F}\right\rangle\mathbf{A}.$$

For notation compactness, the spatial basis $\mathbf{W}$ in DyBO-gPC formulation should not be confused with the notation in gPC formulation (39).

Both the gPC system and DyBO-gPC systems are numerically solved by the fourth-order RK method with time step $\Delta t = 10^{-3}$ on a 128×128 spatial grid. The pseudo-spectral method with the 36-dealiasing rule [15,16] is used to compute spatial derivatives. Various numerical results are presented in the following.

Verification of complexity analysis. Clearly, SNSE (36) is a quadratic nonlinear PDE driven by stochastic forces, so the computational complexity analysis in Section 3 is applicable. Before presenting computational results, we first verify the complexity analysis, i.e., Eq. (15). To this end, we record the wall time of a single time step when the gPC system (39) or the DyBO-gPC system (41) is numerically integrated by the fourth-order Runge–Kutta method. For $N_p = 80, 100, 120$ and $m = 4, 8, 12, 16$, the computational times are summarized in Table 4. To improve the accuracy of recorded wall times, we actually compute the average wall time of 10 time iterations.

In Table 4, the exponents α and β in Eq. (15) are estimated by linear regression. The last column uses wall times corresponding to $m = 8, 12, 16$, while the second to last column uses all four values of m. As we can see from the fourth column of the table, the computational time is relatively small when $m = 4$. In this case, the dominant terms in our previous analysis may not truly dominate other terms and some inevitable programming overheads, such as memory allocations and function

Table 4
Comparison of wall times of a single RK step of gPC and DyBO-gPC systems.

N_p	gPC (*sec*)		DyBO-gPC (*sec*)				α	
	Sparse	Non-Sparse	$m = 4$	$m = 8$	$m = 12$	$m = 16$	α_1	α_2
80	17.242	772.10	0.3946	1.6238	4.8850	10.5483	2.3670	2.7006
100	26.302	1482.7	0.4221	1.5666	4.9577	10.7119	2.3334	2.7779
120	36.440	2558.3	0.4246	1.6567	5.0451	10.8200	2.3342	2.7099
β	β-Sparse		1.6621	1.8056	1.7683	1.7844		
	β-Non-Sparse		2.7683	2.9117	2.8744	2.8905		

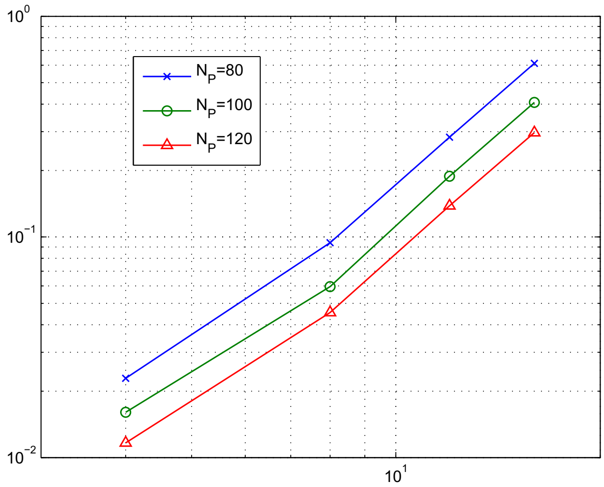
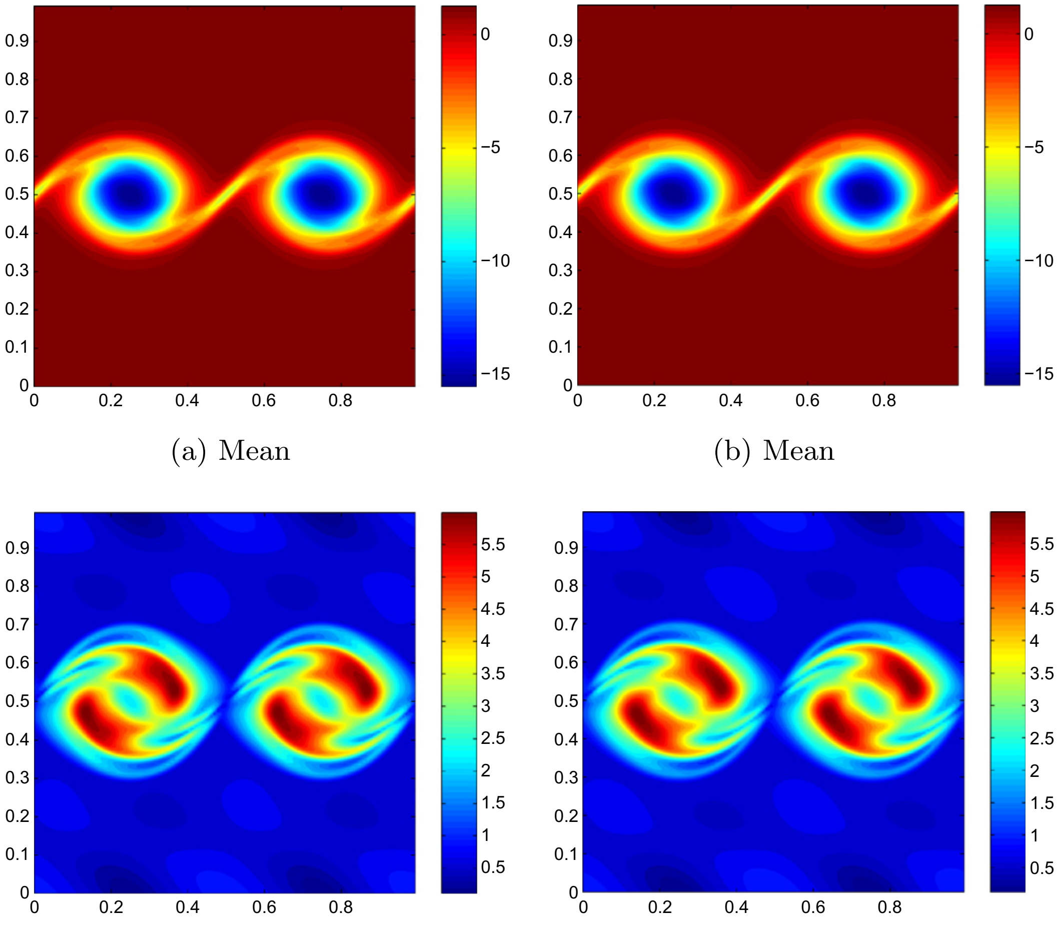


Fig. 6. Wall times of a single RK step of the DyBO-gPC system. The horizontal axis is the number of mode pairs, m.

calling overheads, may kick in. In Fig. 6 accompanying Table 4, the computational times corresponding to $m = 8, 12, 16$ align nicely into a straight line for each $N_p = 80, 100, 120$, respectively, but the computational times corresponding to $m = 4$ drift up. If we remove these points from our fitting, the linear regression estimate of the exponent α in Eq. (15) would be approximately equal to 2.73, close to the theoretically predicated value 3.

As we mentioned in Remark 1, $\mathfrak{T}^{(\mathbf{H})}$ is very sparse when Hermite polynomials are used for Gaussian random variables. In Table 4, we also report the wall times of gPC in the second and third columns, respectively, depending on whether such



(a) Mean

(b) Mean

(c) STD

(d) STD

Fig. 7. Mean and STD of the vorticity field at time $t = 1.0$. The left column is computed by DyBO-gPC, while the right column is computed by gPC.

sparsity is explored or not in the numerical implementation of gPC methods. Clearly, the computational cost is significantly smaller if such sparsity is considered. However, we may not have such luxury for arbitrary non-Gaussian random variables, i.e., general distributions. In the last two rows of Table 4, the exponent β is estimated by linear regression, respectively, when sparsity is explored or not. The last row gives ~ 2.9 for the exponent β confirming our analysis in Eq. (15).

Numerical errors of DyBO-gPC. The number of polynomial basis functions $\mathbf{H}_{\boldsymbol{\alpha}}$ grows exponentially fast as the number of random variables r and the total order p increase. The scheme of sparse truncation proposed in Luo's thesis [22] (see also [14]) proves to be a relatively effective method to alleviate the situation. In the following computation, we follow this scheme and choose a multi-index set $\mathfrak{J}$ obtained from a sparse truncation of the multi-index set $\mathfrak{J}_8^3$,

$$\mathfrak{J} = \{\boldsymbol{\alpha} \in \mathfrak{J}_8^3 \text{ and if } \mid \boldsymbol{\alpha} \mid = 3, \text{then } \alpha_2 \leqslant 2, \alpha_3 \leqslant 1, \alpha_4 \leqslant 1, \alpha_6 \leqslant 2, \alpha_7 \leqslant 1, \alpha_8 \leqslant 1\} \setminus \{\mathbf{0}\},$$

which still results in total 130 multi-indices!

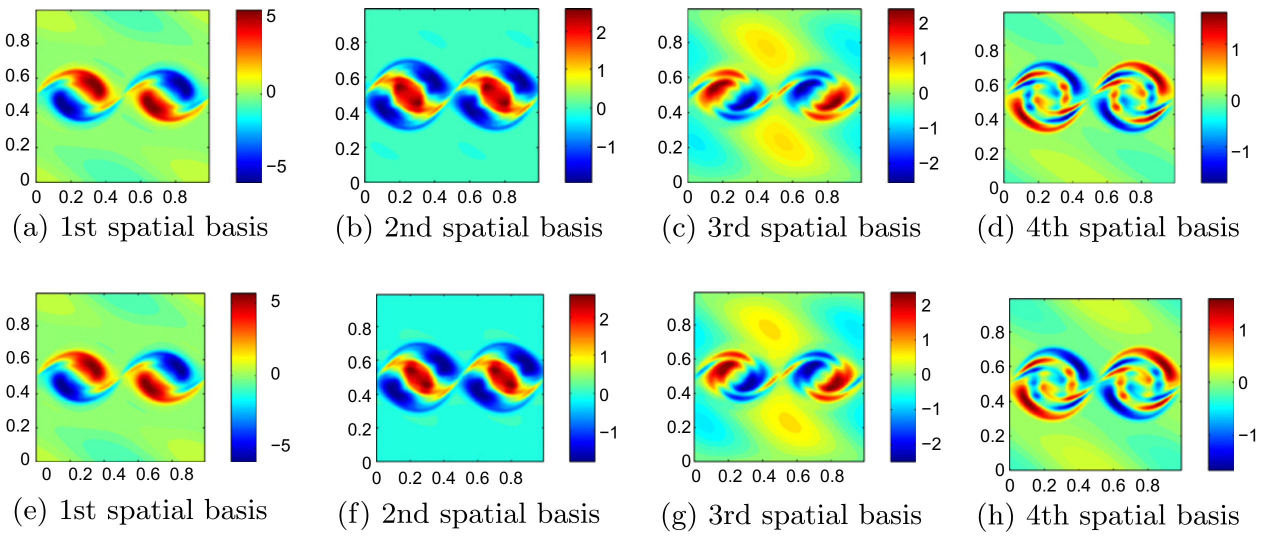
The mean and the STD of the vorticity field and the first four spatial modes in the KL expansion of the vorticity field at time $t = 1.0$ are given in Fig. 7 and Fig. 8, respectively. In both figures, the results by our DyBO-gPC method with $m = 8$ are given in the left column and compared to the results by the gPC method in the right column. The results are almost indistinguishable. We further confirm the convergence of DyBO-gPC to gPC by plotting the relative errors of both mean and STD of the vorticity field as functions of time in the top two subplots of Fig. 9.

In the same figure, we also report the relative errors as functions of time when the adaptive strategy of adding and/or removing basis pairs is enacted. Two numerical examples are provided: one starts with four basis pairs, i.e., $m_0 = 4$ and the other starts with six basis pairs, i.e., $m_0 = 6$. In Fig. 9(c), the number of basis pairs in the DyBO-gPC method is plotted against time t. Because of the special form of the stochastic force f considered in this numerical example, the randomness is only introduced through low-order gPC coefficients and then spread to the mean and other high-order gPC coefficients. At the beginning of the evolution of the stochastic flow, the randomness is not strong and the adaptive algorithm finds no need to add new basis pairs before time $t = 0.35$. As the system evolves, the randomnesses get strong through interactions among different basis pairs and the adaptive algorithm automatically adds more basis pairs when necessary.

Avoiding selection of multi-index Set $\mathfrak{J}$. The gPC method suffers greatly from the curse of dimensionality. In the above numerical example, we use low-order ($\leqslant 3$) polynomials and also the sparse truncation technique to further reduce the size of multi-index set $\mathfrak{J}$, which still results in a set of 130 polynomials. It takes more than 8 h of wall time to numerically integrate the gPC system from $t = 0.0$ to $t = 1.0$. Adaptive gPC methods try to include only the most important gPC coefficients $w_{\boldsymbol{\alpha}}$ in the computation, i.e., a selection of multi-index set $\mathfrak{J}$.

In Fig. 10(a), we plot the energy spectrum of the gPC solution at $t = 1.0$, i.e., the L^2 norm of the gPC coefficient $w_{\boldsymbol{\alpha}}$, which does not decay monotonically. Index $\mathfrak{J}$ is a multi-index set, so we do not have sufficient information and a good strategy, prior to the computation, to sort $\mathfrak{J}$ and select the most important ones.

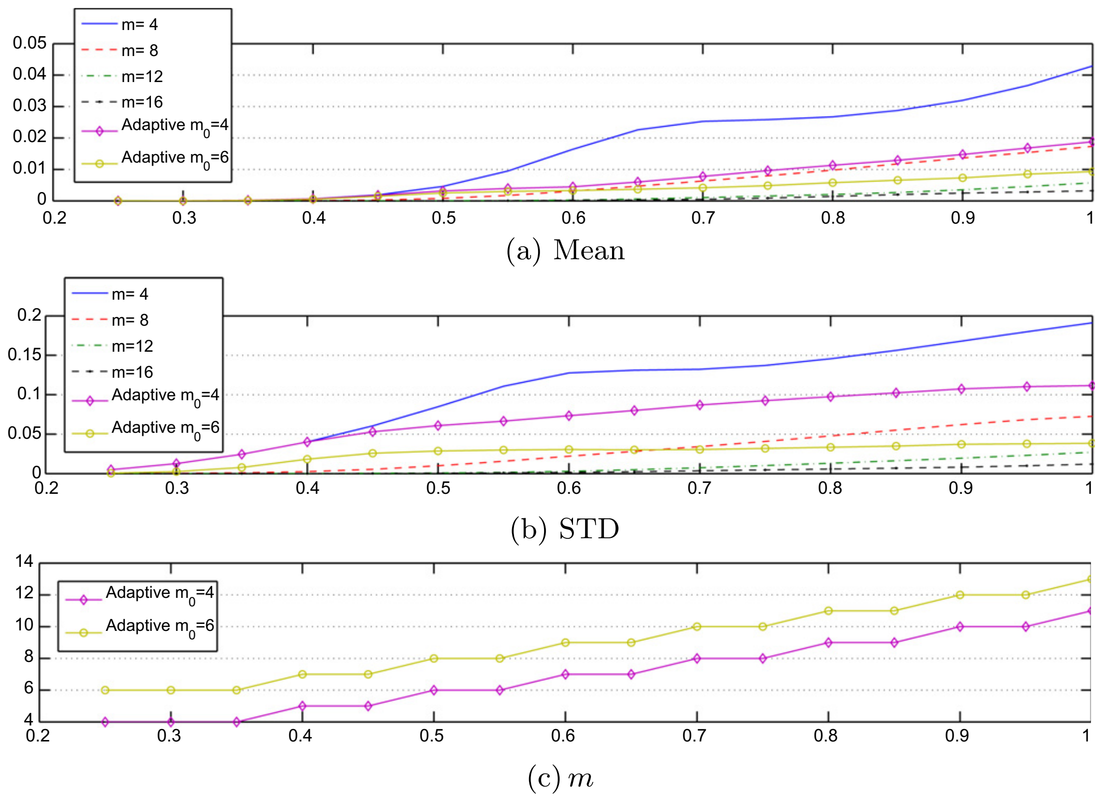
On the other hand, our DyBO method tracks the KL expansion of the true solution and *automatically* includes only the most important ones. Furthermore, the KL expansion is known to provide the most compact representation of a second-order stochastic process, so the energy spectrum of the DyBO solution, i.e., the L^2 norm of the DyBO-gPC coefficient $w_i, i = 1, 2, \ldots, m$, has a faster decay rate even compared to the sorted energy spectrum of gPC solution (see Fig. 10(b)). This difference in the decay rate implies that our method leads to a smaller system to solve, leading to less computational cost.



(a) 1st spatial basis (b) 2nd spatial basis (c) 3rd spatial basis (d) 4th spatial basis

(e) 1st spatial basis (f) 2nd spatial basis (g) 3rd spatial basis (h) 4th spatial basis

Fig. 8. The first four spatial basis at time $t = 1.0$. Top are computed by DyBO-gPC, while the bottom are computed by gPC.



(a) Mean

(b) STD

(c) m

Fig. 9. The $\mathbb{L}^2$ relative errors of mean and STD of vorticity field computed by DyBO. The errors are plotted as functions of time in the top two figures, while the numbers of basis pairs used in the adaptive strategy are given in the last figure.

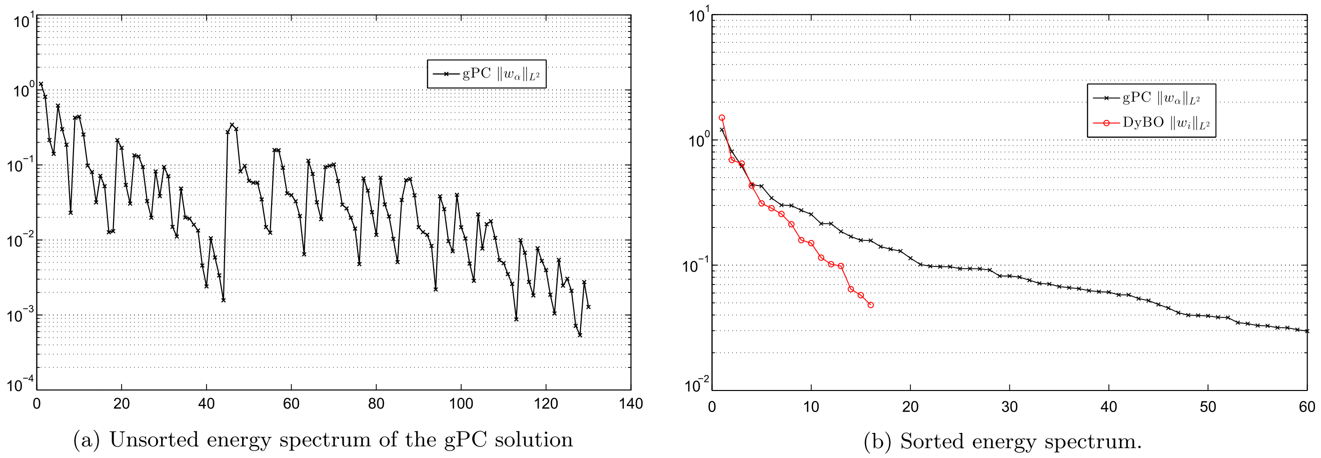
To further illustrate and understand the benefits of the DyBO method, we consider a little "stronger" stochastic force

$$\left(\frac{\partial f_2}{\partial x} - \frac{\partial f_1}{\partial y}\right) = -\frac{\partial \sigma_1}{\partial y}\sum_{i=1}^{4}\frac{it}{T}\xi_i + \frac{\partial \sigma_2}{\partial x}\sum_{i=5}^{8}\frac{(i-4)t}{T}\left(\frac{t}{T}\right)\xi_i. \tag{43}$$

With this stochastic force, the sorted energy spectrum of both gPC and DyBO solutions along with the square roots of eigenvalues are plotted in Fig. 11(a). Clearly, the energy spectrum of DyBO decays much faster than that of gPC.

Once the gPC coefficients w_α's are sorted in the descending order in L^2 norm, we can use the first several gPC coefficients, i.e., the most important ones, to compute a solution and compare with the exact one. The relative errors of STD computed by this procedure are plotted in Fig. 11(b) against the number of gPC coefficients.

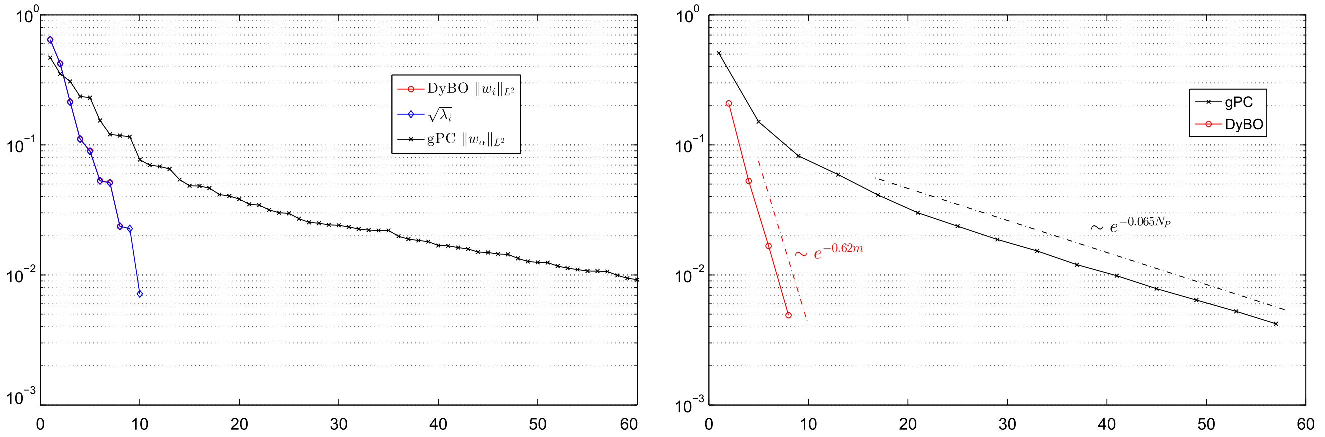
At the first glance, this procedure may seem effective. But we would like to point out that the multi-indices $\boldsymbol{\alpha}$'s corresponding to the most important gPC coefficients are in general not known prior to the beginning of computations. What's more, such set of multi-indices may change with respect to time t, making the selection of an effective multi-index set $\mathfrak{J}$ even harder. Moreover, the less important gPC coefficients excluded from the gPC system may induce additional errors when



(a) Unsorted energy spectrum of the gPC solution

(b) Sorted energy spectrum.

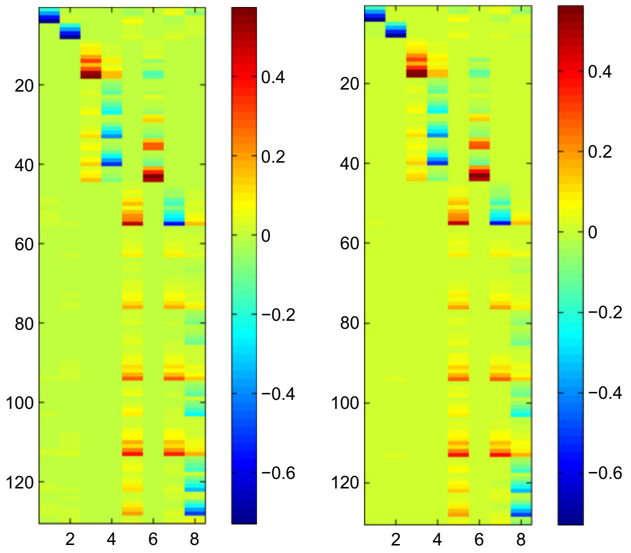
Fig. 10. Comparison of energy spectrum of gPC and DyBO solutions at time $t = 1.0$.



(a) (Sorted) Energy spectrum of the gPC and DyBO solutions.

(b) Relative errors of STD computed by gPC and DyBO methods

Fig. 11. Comparisons of gPC and DyBO methods.



(a) DyBO

(b) gPC

Fig. 12. Stochastic basis computed by DyBO and gPC.

we solve the system which only includes the most important ones. In fact, we observe that the solution obtained by this procedure is less accurate than that by our DyBO method with the same number of basis pairs, as shown in Fig. 11(b). With only 8 basis pairs, our DyBO method achieves the same accuracy ($\sim 0.5\%$) as that by gPC method with 60 gPC coefficients. By using Table 4, we can estimate speedup in this case. When the sparsity of tensor $\mathfrak{T}^{\mathbf{H}}$ is not exploited in the numerical implementation of gPC, the speedup is $\sim 200X$ (327.8 s v.s. 1.6567 s per time iteration). When the sparsity is exploited (see Remark 3.1 in Section 3.2), the speedup is $\sim 6X$ (10.628 s v.s. 1.6567 s per time iteration). Fig. 11(a) also confirms numerically that our DyBO method can accurately recover the eigenvalues in the KL expansion.

Looking into the stochastic basis $\mathbf{Y} = \mathbf{HA}$ reveals the origin of the fast error decay in our DyBO-gPC method. In Fig. 12, we plot the stochastic basis computed by DyBO in the second figure and ones recovered from the gPC solution in the third figure, respectively. Clearly, each stochastic basis Y_i is a linear combination of several, possibly many, gPC basis. Therefore, unlike gPC methods, where the stochastic basis $\mathbf{H}$ is fixed and does not change with time, the DyBO method aggregates the polynomial basis and forms a more efficient stochastic basis $\mathbf{Y}$. Moreover, this set of stochastic basis is automatically adapted in time.

6.3. 2D stochastic flow driven by buoyance force

As a model to test numerically the generalized DyBO formulation for a system of time-dependent SPDEs, we consider the Navier–Stokes equations whose velocity components are driven by *both* stochastic forces and buoyancy forces due to small

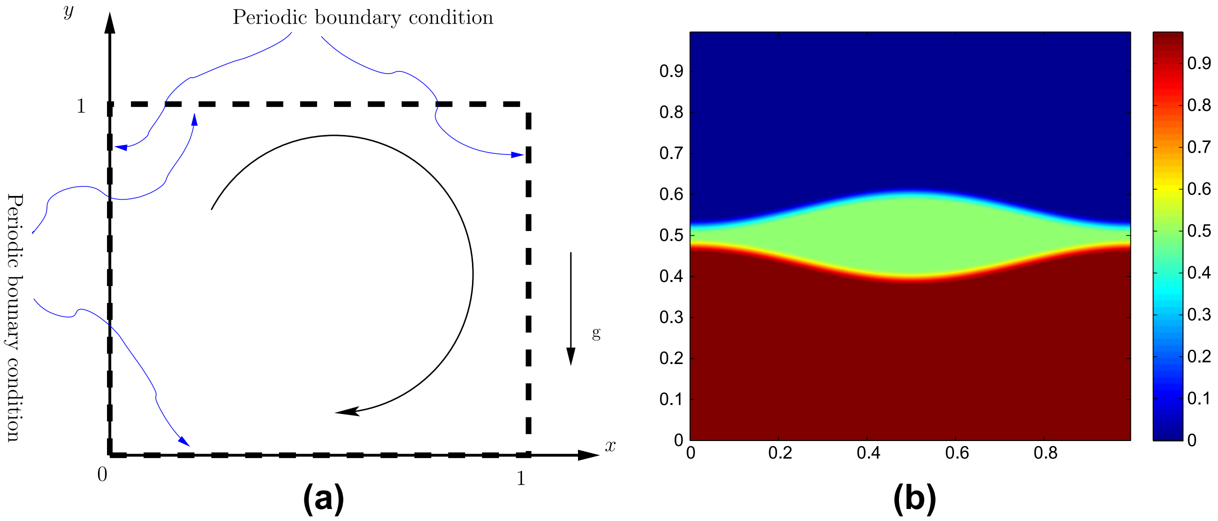


Fig. 13. Stochastic flow driven by stochastic force and buoyancy force due to Boussinesq approximation. On the left: diagram of the stochastic flow in an unit square. The gravity is downward parallel to *y*-axis and periodic boundary conditions are assumed on both *x* and *y* directions. On the right, the initial temperature field is plotted, while the initial vorticity is uniformly zero.

density difference induced by temperature variations. Specifically, we consider the stochastic flow in an unite square, i.e., $\mathcal{D} = [0,1] \times [0,1]$, with periodic boundary conditions on both spatial directions. See Fig. 13(a).

The temperature field is not spatially uniform and causes variations of the density field. Such variations are small because we assume the thermal expansion coefficient is very small. The induced buoyancy force may drive the flow motion in addition to external stochastic forces. Here, we adopt the Boussinesq approximation to model such buoyancy force. The governing SPDE of such stochastic flow in Fig. 13(a) is the Stochastic Navier–Stokes equations (SNSE)

$$\frac{\partial\theta}{\partial t} + u\frac{\partial\theta}{\partial x} + v\frac{\partial\theta}{\partial y} = \kappa\Delta\theta, \tag{44a}$$

$$\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u} = \nu\triangle\mathbf{u} - \nabla p + \mathbf{f} + \mathbf{F} \tag{44b}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{44c}$$

where the viscosity $\nu = 2.0 \times 10^{-4}$ and the thermal diffusivity $\kappa = 2.0 \times 10^{-4}$. θ is temperature field, $\mathbf{u} = (u, v)$ is the velocity field, p is the pressure, $\mathbf{f} = (f_1, f_2)$ is the zero-mean stochastic force vector and $\mathbf{F} = (0, \mu g\theta)$ is the buoyancy force, where μ is the thermal expansion coefficient, g is the gravity of Earth and scaled gravity $\mu g = 11.31$ (see [7] for more details).

At the first glance, this numerical example is similar to the one in Section 6.2 except introducing the additional temperature field and gravity effect. However, the stochastic flow is very different from the one considered previously for the following reason. We see that the buoyancy force depends on the gradient of temperature field and is actually a stochastic force. In other words, the stochastic flow considered in this section is driven by two kinds of stochastic forces: one *"external"* stochastic force injecting randomness from the ambient environment into the unit square and one *"internal"* stochastic force feeding randomness back to the system, from temperature component to vertical velocity component. Such stochastic flow provides a severe test model for our generalized DyBO methods.

Like the previous numerical example, it is convenient to consider the vorticity-stream function formulation in the standard form defined in the system of SPDEs (26),

$$\frac{\partial\theta}{\partial t} = \mathcal{L}_\theta\{\theta, w\} = -\left(u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y}\right)\theta + \kappa\Delta\theta, \tag{45a}$$

$$\frac{\partial w}{\partial t} = \mathcal{L}_w\{\theta, w\} = -\left(u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y}\right)w + \nu\Delta w + \left(\frac{\partial f_2}{\partial x} - \frac{\partial f_1}{\partial y}\right) + \mu g\frac{\partial\theta}{\partial x}. \tag{45b}$$

Consider the finite-term KL expansion of the solutions of SNSE (44),

$$\theta = \bar{\theta} + \boldsymbol{\theta}\mathbf{Y}^T = \bar{\theta} + \boldsymbol{\theta}\mathbf{A}^T\mathbf{H}^T, \tag{46a}$$

$$w = \bar{w} + \mathbf{W}\mathbf{Z}^T = \bar{w} + \mathbf{W}\mathbf{B}^T\mathbf{H}^T, \tag{46b}$$

where row vectors $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_{m_\theta})$ and $\mathbf{W} = (w_1, w_2, \cdots, w_{m_w})$, and matrices $\mathbf{A} \in \mathbb{R}^{N_p \times m_\theta}$ and $\mathbf{B} \in \mathbb{R}^{N_p \times m_w}$. We write basis number vector $\mathbf{m} = (m_\theta, m_\mathbf{W})$. By substituting the above expansion into Eq. (29), we obtain the DyBO-gPC formulation for SNSE (44) (see A for details). The system of equations for the temperature component are

$$\frac{\partial\bar{\theta}}{\partial t} = \kappa\Delta\bar{\theta} - \mathfrak{D}_{(\bar{u},\bar{v})\bar{\theta}} - \mathfrak{D}_{(\mathbf{U},\mathbf{V})\left(\boldsymbol{\theta}\mathbf{A}^T\mathbf{B}\right)}, \tag{47a}$$

$$\frac{\partial\boldsymbol{\theta}}{\partial t} = -\boldsymbol{\theta}\mathbf{D}_\theta^T + \kappa\Delta\boldsymbol{\theta} - \mathfrak{D}_{(\bar{u},\bar{v})\boldsymbol{\theta}} - \mathfrak{D}_{(\mathbf{U},\mathbf{V})\bar{\theta}}\mathbf{B}^T\mathbf{A} - \left[\mathfrak{D}_{(u_i,v_i)\theta_j}B_{\boldsymbol{\alpha}i}A_{\boldsymbol{\beta}j}A_{\boldsymbol{\gamma}k}\mathfrak{T}^{(\mathbf{H})}_{\boldsymbol{\alpha\beta\gamma}}\right]_{1\times k}, \tag{47b}$$

$$\frac{\mathrm{d}\mathbf{A}}{\mathrm{d}t} = \mathbf{A}\Big(-\mathbf{C}_\theta^T + \kappa\langle\Delta\boldsymbol{\theta}^T,\boldsymbol{\theta}\rangle\boldsymbol{\Lambda}_{\boldsymbol{\theta}}^{-1} - \Big\langle\big(\mathfrak{D}_{(\bar{u},\bar{v})\boldsymbol{\theta}}\big)^T,\boldsymbol{\theta}\Big\rangle\boldsymbol{\Lambda}_{\boldsymbol{\theta}}^{-1}\Big) - \mathbf{B}\Big\langle\big(\mathfrak{D}_{(\mathbf{U},\mathbf{V})\bar{\theta}}\big)^T,\boldsymbol{\theta}\Big\rangle\boldsymbol{\Lambda}_{\boldsymbol{\theta}}^{-1} - \left[\mathfrak{T}^{(\boldsymbol{\theta})}_{ijk}B_{\boldsymbol{\alpha}i}A_{\boldsymbol{\beta}j}\mathfrak{T}^{(\mathbf{H})}_{\boldsymbol{\alpha\beta\gamma}}\right]_{\gamma\times k}\boldsymbol{\Lambda}_{\boldsymbol{\theta}}^{-1}, \tag{47c}$$

where matrices $\mathbf{C}_\theta$ and $\mathbf{D}_\theta$ can be solved from the linear system (4) from $G_{*\theta}$,

$$\boldsymbol{\Lambda}_{\boldsymbol{\theta}}G_{*\theta} = \kappa\langle\boldsymbol{\theta}^T,\Delta\boldsymbol{\theta}\rangle - \langle\boldsymbol{\theta}^T,\mathfrak{D}_{(\bar{u},\bar{v})\boldsymbol{\theta}}\rangle - \langle\boldsymbol{\theta}^T,\mathfrak{D}_{(\mathbf{U},\mathbf{V})\bar{\theta}}\rangle\mathbf{B}^T\mathbf{A} - \left[\mathfrak{T}^{(\boldsymbol{\theta})}_{ijk}B_{\boldsymbol{\alpha}i}A_{\boldsymbol{\beta}j}A_{\boldsymbol{\gamma}l}\mathfrak{T}^{(\mathbf{H})}_{\boldsymbol{\alpha\beta\gamma}}\right]_{k\times l}. \tag{48}$$

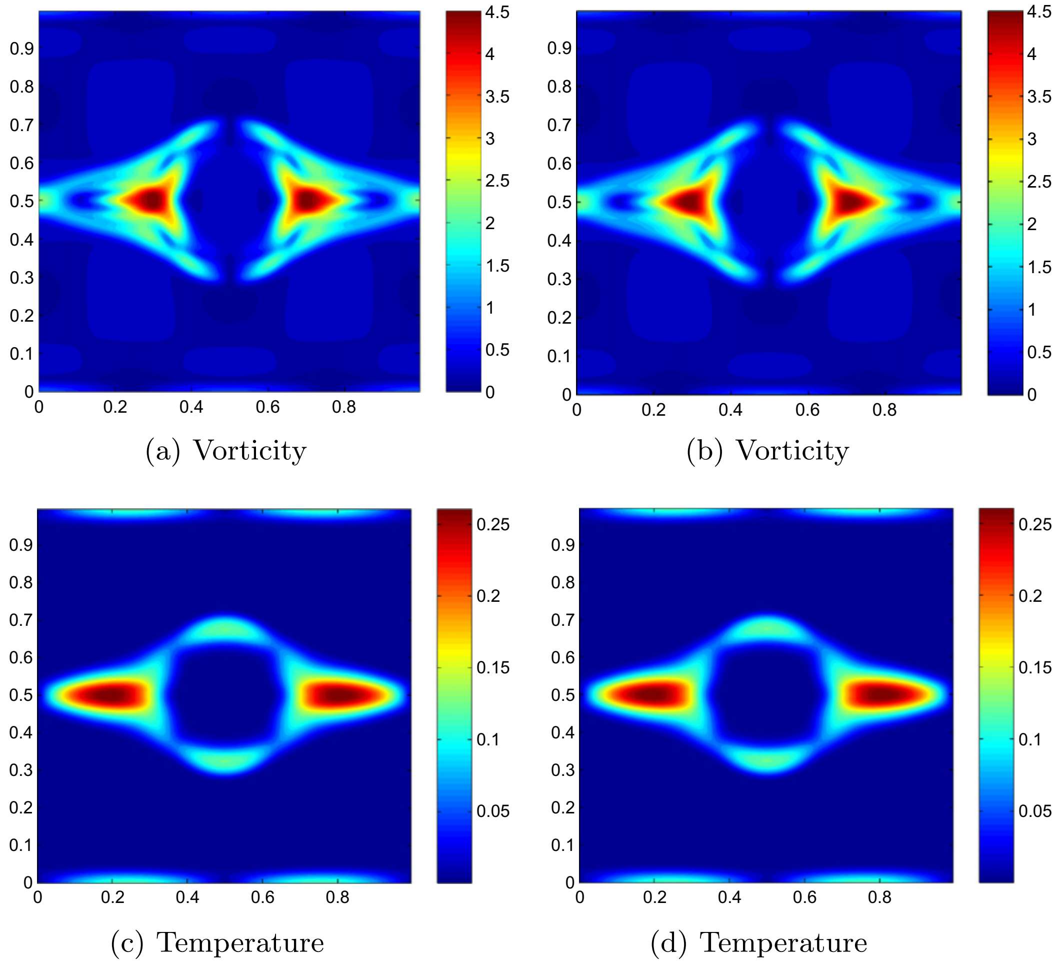
The system of equations for the vorticity component are

$$\frac{\partial\bar{w}}{\partial t} = \left(\nu\Delta\bar{w} - \mathfrak{D}_{(\bar{u},\bar{v})\bar{w}} + \mu g\frac{\partial\bar{\theta}}{\partial x}\right) - \mathfrak{D}_{(\mathbf{U},\mathbf{V})\mathbf{W}}, \tag{49a}$$

$$\frac{\partial\mathbf{W}}{\partial t} = -\mathbf{W}\mathbf{D}_w^T + \left[\nu\Delta\mathbf{W} - \mathfrak{D}_{(\bar{u},\bar{v})\mathbf{W}} - \mathfrak{D}_{(\mathbf{U},\mathbf{V})\bar{w}}\right] + \mu g\frac{\partial\boldsymbol{\theta}}{\partial x}\mathbf{A}^T\mathbf{B} - \left[\mathfrak{D}_{(u_i,v_i)w_j}B_{\boldsymbol{\alpha}i}B_{\boldsymbol{\beta}j}B_{\boldsymbol{\gamma}k}\mathfrak{T}^{(\mathbf{H})}_{\boldsymbol{\alpha\beta\gamma}}\right]_{1\times k} + \mathbf{FB}, \tag{49b}$$

$$\begin{aligned}\frac{\mathrm{d}\mathbf{B}}{\mathrm{d}t} = {}&\mathbf{B}\Big(-\mathbf{C}_w^T + \Big\langle\nu\Delta\mathbf{W}^T - \big(\mathfrak{D}_{(\bar{u},\bar{v})\mathbf{W}}\big)^T - \big(\mathfrak{D}_{(\mathbf{U},\mathbf{V})\bar{w}}\big)^T,\mathbf{W}\Big\rangle\boldsymbol{\Lambda}_{\mathbf{W}}^{-1}\Big) + \mu g\mathbf{A}\Big\langle\frac{\partial\boldsymbol{\theta}^T}{\partial x},\mathbf{W}\Big\rangle\boldsymbol{\Lambda}_{\mathbf{W}}^{-1} - \left[\mathfrak{T}^{(\mathbf{W})}_{ijk}B_{\boldsymbol{\alpha}i}B_{\boldsymbol{\beta}j}\mathfrak{T}^{(\mathbf{H})}_{\boldsymbol{\alpha\beta\gamma}}\right]_{\gamma\times k}\boldsymbol{\Lambda}_{\mathbf{W}}^{-1}\\ &+ \Big\langle\mathbf{F}^T,\mathbf{W}\Big\rangle\boldsymbol{\Lambda}_{\mathbf{W}}^{-1},\end{aligned} \tag{49c}$$

where matrices $\mathbf{C}_w$ and $\mathbf{D}_w$ can be solved from the linear system (4) from G_{*w},



(a) Vorticity

(b) Vorticity

(c) Temperature

(d) Temperature

Fig. 14. STD of vorticity and temperature fields at time $t = 1.0$. Left column by DyBO and right column by gPC.

$$\Lambda_{\mathbf{W}} G_{*w} = \left\langle \mathbf{W}^T, \nu\Delta \mathbf{W} - \mathfrak{D}_{(\bar{u},\bar{v})\mathbf{W}} - \mathfrak{D}_{(\mathbf{U},\mathbf{V})\bar{w}} \right\rangle + \left\langle \mathbf{W}^T, \mu g \frac{\partial \boldsymbol{\theta}}{\partial x} \right\rangle \mathbf{A}^T \mathbf{B} - \left[\mathfrak{T}^{(\mathbf{W})}_{ijk} B_{\boldsymbol{\alpha} i} B_{\boldsymbol{\beta} j} B_{\boldsymbol{\gamma} l} \mathfrak{T}^{(\mathbf{H})}_{\boldsymbol{\alpha\beta\gamma}} \right]_{k\times l} + \left\langle \mathbf{W}^T, \mathbf{F} \right\rangle \mathbf{B}. \tag{50}$$

Both the gPC system and the DyBO-gPC system are numerically integrated by the fourth-order RK method with a time step $\Delta t = 10^{-3}$. Unlike the stochastic flow only driven by the stochastic force in the previous numerical example, we found by numerical experiments a higher-resolution spatial grid is necessary to resolve some fine structures. Thus, we use 256×256 spatial grid in the numerical simulations. Computations on higher resolution grid, 512×512, are also performed for the gPC method to make sure that the solution is indeed well-resolved. The pseudo-spectral method with the 36-dealiasing rule [15,16] is used to compute spatial derivatives. For the DyBO method, the gPC solution at $\Delta T_0 = 0.2$ are used as initial conditions. Different values of ΔT_0, such as 0.1 and 0.15, have also been used and no significant differences have been found. Here, we use the sparse truncation technique and choose the multi-index set

$$\mathfrak{J} = \{\boldsymbol{\alpha} \in \mathfrak{J}^3_8 \text{ and if } \mid \boldsymbol{\alpha} \mid = 3, \text{then } \alpha_2 \leqslant 2, \alpha_3 \leqslant 1, \alpha_4 \leqslant 1, \alpha_6 \leqslant 2, \alpha_7 \leqslant 1, \alpha_8 \leqslant 1\} \setminus \{\mathbf{0}\}.$$

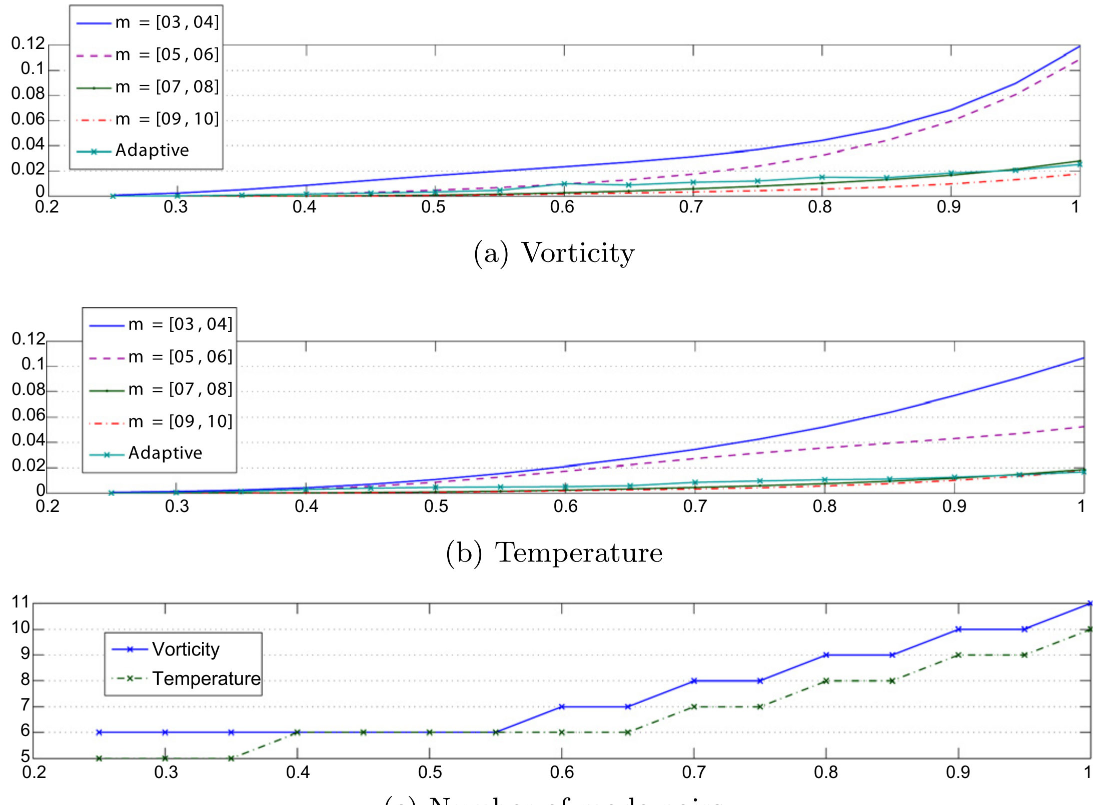
Both initial vorticity and temperature field are assumed to be deterministic. In this example, we are primarily interested in the combined effect of stochastic force f and the stochastic buoyancy force, so the vorticity is assumed to be zero initially. We adopt the initial temperature field from [7],

$$\theta(x, y, 0, \xi) = \overset{\circ}{\theta}(x, y) = \frac{1}{2}\mathbb{H}_{\delta_1}(y_{lb}(x) - y) + \frac{1}{2}\mathbb{H}_{\delta_1}(y_{ub}(x) - y), \tag{51}$$

where $y_{lb}(x) = \frac{1}{2} - \delta_2 - \delta_3 y_0(x), y_{ub}(x) = \frac{1}{2} + \delta_2 + \delta_3 y_0(x), y_0(x) = 1 + \sin(2\pi(x + \frac{3}{4}))$, and the modified Heaviside step function $\mathbb{H}_\epsilon(z) = \frac{x+\epsilon}{2\epsilon} + \frac{1}{2\pi}\sin\left(\frac{\pi x}{\epsilon}\right)$. In Fig. 13(b), the initial temperature field $\overset{\circ}{\theta}$ is plotted.

In the first numerical example, we choose basis number vector $\mathbf{m} = (7, 8)$. In Fig. 14, STD fields of vorticity and temperature are plotted at time $t = 1.0$ with the results by DyBO in the left column and ones by gPC in the right column. We also compare the spatial basis of vorticity and temperature, $\mathbf{W}$ and $\boldsymbol{\theta}$, given by DyBO and gPC at time $t = 1.0$, respectively (not shown here, see [8]). These results confirm that the solutions given by DyBO are not only a good approximation to the solution given by gPC, but also track directly the KL expansion of the SPDE solution.

To further study the numerical convergence of our DyBO method, we choose another two basis number vectors, $\mathbf{m} = (3, 4)$ and $\mathbf{m} = (9, 10)$, and repeat the DyBO computation. The relative errors of vorticity and temperature STD fields as functions of time are plotted in Fig. 15a) and (b), respectively. When the basis number vector is increased from



(a) Vorticity

(b) Temperature

(c) Number of mode pairs

Fig. 15. $\mathbb{L}^2$ relative errors of vorticity and temperature STDs as functions of time. The evolutions of the numbers of vorticity and temperature basis pairs are also given in the last figure.

Table 5
Speedups by proposed parallelization strategy for different spatial grids and mode number vectors. Wall times of a single RK time step for different parameters are given in seconds, fourth, sixth and eighth columns. All times are in seconds.

# proc	256×256				512×512			
	$\mathbf{m} = [7,8]$		$\mathbf{m} = [19,20]$		$\mathbf{m} = [7,8]$		$\mathbf{m} = [19,20]$	
	Time	Speedup	Time	Speedup	Time	Speedup	Time	Speedup
Serial	1.6930	NA	19.529	NA	10.658	NA	71.372	NA
2	1.2618	1.34	11.150	1.75	6.8218	1.56	34.471	2.07
4	1.0216	1.66	6.9873	2.79	4.4738	2.38	20.600	3.46
6	0.9787	1.73	4.4082	4.43	3.7902	2.81	16.940	4.21
8	0.9197	1.84	3.1262	6.25	3.5095	3.03	15.182	4.70

$\mathbf{m} = (3,4)$ to $\mathbf{m} = (9,10)$, the relative error of STD is brought down from 11.9% to 1.8% for vorticity and from 10.7% to 1.8% for temperature, respectively.

In the above numerical simulation, we have intentionally chosen $m_{\mathbf{W}} = m_\theta + 1$ due to the fact that the vorticity is more singular than the temperature in general and requires more modes to resolve the stochastic solution accurately. However, in general, the specific number of modes required in numerical simulations would be problem dependent. Furthermore, when the adaptive strategy developed in Section 4 is incorporated, such guess of mode numbers is eliminated. The number of modes for both components will adjust automatically on the fly as demonstrated in the next numerical example. Initially, the basis number vector is chosen as $\mathbf{m}_0 = (5,6)$. As we can see in the top two plots in Fig. 15, good accuracy is preserved as basis pairs are automatically added when necessary (see Fig. 15(c) for the evolution of numbers of vorticity and temperature basis pairs).

The parallelization strategy proposed in Section 3.3 for the quadratic nonlinear PDE driven by a stochastic force is implemented via POSIX multi-threaded programming in C++ and hooked to the main matlab code via matlab external APIs. Since we use FFT in the pseudo-spectral method, a simple domain partition scheme (see Cheng's thesis [8] for details) is adopted for the maximum performance of FFT. The simulations are conducted on the Shared Heterogeneous Cluster (SHC) at Caltech Center for Advanced Computing Research (CACR). Due to the limitations posted by available campus matlab licenses, our simulation is constrained onto a single computing node where total 8 computing cores are available from two AMD Opteron 2390 of 2.5 GHz.

For two spatial grids, 256×256 and 512×512, the wall times of a single RK iteration steps are recorded and reported in Table 5 for serial computation and parallel computation with $2,4,6$ and 8 cores, respectively. Speedups of parallel strategies are also reported in the same table. Two mode number vectors $\mathbf{m} = (7,8)$ and $\mathbf{m} = (19,20)$ are used in the computation. Confirming our complexity analysis, the speedup is more significant for larger spatial grid and more mode pairs. For example, the computational time on 8 cores is reduced to $1/6$ on 256×256 spatial grid compared to that on a single core.

7. Conclusions

The DyBO method proposed in the first part of our paper [9] exploits the intrinsic data-sparsity of SPDEs, tracks the KL expansion of the stochastic solutions, and provides an efficient numerical method for time-dependent SPDEs. In addition to providing applications for stochastic flow driven by stochastic forces, we also make the following important contributions to this method in this paper:

Computational complexity analysis. Detailed computational complexity analysis has been conducted, which shows a speedup factor of $O((m/N_p)^3)$ over the standard gPC method for a quadratic nonlinear PDE driven by a stochastic force. We have observed considerable speedups in the 2D stochastic flow problem. Furthermore, a simple, yet powerful, parallelization strategy based on domain decomposition of the spatial domain was proposed and its parallel efficiency was numerically verified.

Adaptive strategy for adding and removing mode pairs. Based on the analysis of the type-KL error, a sophisticated strategy has been proposed to adaptively and dynamically add or remove spatial and stochastic mode pairs by using the short-time "burst" of the DyBO-gPC and gPC method. The effectiveness of such strategies have been demonstrated in our numerical examples. We are currently exploring alternative adaptive strategy based on l^1 optimization to avoid the need of computing gPC solution for a short time..

Generalizations to a system of SPDEs. We have generalized the DyBO method for a system of SPDEs and applied it to solve a 2D Boussinesq approximation driven by both stochastic external forces and buoyancy forces. Such generalization potentially provides a way to tackle problems involving both multiscale phenomena and randomnesses. For example, the stochastic solution can be first decomposed into different spatial scales and then the generalized DyBO method can be applied to the system of SPDEs of stochastic solution components on different scales. The DyBO method can also be generalized to track KL expansion of SPDE solution in other Sobolev space, which will be reported in the subsequence paper.

Acknowledgement

This work was supported in part by AFOSR MURI Grant FA9550-09-1-0613, a DOE Grant DE-FG02-06ER25727 and a NSF Grant DMS-0908546.

Appendix A. Derivations of the DyBO formulation of SNSE

The derivations of the DyBO formulation of (36) and (44) are very similar. In this appendix, we provide the details of the derivations of the DyBO-gPC formulation of SNSE (44) as an example. Substituting the KL expansion into the Poisson equation for the steam function (37), we have

$$-\Delta\psi = \bar{w} + \mathbf{W}\mathbf{Z}^T, \tag{A.1}$$

which implies an expansion $\psi = \bar{\psi} + \boldsymbol{\psi}\mathbf{Z}^T$ with

$$\begin{aligned} &-\Delta\bar{\psi} = \bar{w},\\ &-\Delta\boldsymbol{\psi} = \mathbf{W}. \end{aligned}$$

From the above expansion of ψ, it is easy to get expansions of u and v,

$$\begin{aligned} u &= \frac{\partial\bar{\psi}}{\partial y} + \frac{\partial\boldsymbol{\psi}}{\partial y}\mathbf{Z}^T = \bar{u} + \mathbf{U}\mathbf{Z}^T, \quad \text{i.e., } \bar{u} = \frac{\partial\bar{\psi}}{\partial y} = -\frac{\partial}{\partial y}\Delta^{-1}\bar{w}, \mathbf{U} = \frac{\partial\boldsymbol{\psi}}{\partial y} = -\frac{\partial}{\partial y}\Delta^{-1}\mathbf{W},\\ v &= -\frac{\partial\bar{\psi}}{\partial x} - \frac{\partial\boldsymbol{\psi}}{\partial x}\mathbf{Z}^T = \bar{v} + \mathbf{V}\mathbf{Z}^T, \quad \text{i.e., } \bar{v} = -\frac{\partial\bar{\psi}}{\partial x} = \frac{\partial}{\partial x}\Delta^{-1}\bar{w}, \mathbf{V} = -\frac{\partial\boldsymbol{\psi}}{\partial x} = \frac{\partial}{\partial x}\Delta^{-1}\mathbf{W}. \end{aligned}$$

Note that all these expansions of ψ, u and v are not necessarily KL expansions.

First, we derive the equation for the temperature. Substituting the KL expansion into Eq. (45b), we get

$$\mathcal{L}_\theta\{\theta, w\} = -\left(\bar{u}\frac{\partial\bar{\theta}}{\partial x} + \bar{v}\frac{\partial\bar{\theta}}{\partial y}\right) + \kappa\Delta\bar{\theta} + [\kappa\Delta\boldsymbol{\theta}\mathbf{A}^T - \left(\bar{u}\frac{\partial\boldsymbol{\theta}}{\partial x} + \bar{v}\frac{\partial\boldsymbol{\theta}}{\partial y}\right)\mathbf{A}^T - \left(\frac{\partial\bar{\theta}}{\partial x}\mathbf{U} + \frac{\partial\bar{\theta}}{\partial y}\mathbf{V}\right)\mathbf{B}^T]\mathbf{H}^T - \mathbf{U}\mathbf{B}^T\mathbf{H}^T\mathbf{H}\mathbf{A}\frac{\partial\boldsymbol{\theta}^T}{\partial x} - \mathbf{V}\mathbf{B}^T\mathbf{H}^T\mathbf{H}\mathbf{A}\frac{\partial\boldsymbol{\theta}^T}{\partial y}.$$

Taking expectations on both sides yields

$$\begin{aligned} \mathbb{E}[\mathcal{L}_\theta\{\theta, w\}] &= -\left(\bar{u}\frac{\partial\bar{\theta}}{\partial x} + \bar{v}\frac{\partial\bar{\theta}}{\partial y}\right) + \kappa\Delta\bar{\theta} - \mathbf{U}\mathbf{B}^T\mathbf{A}\frac{\partial\boldsymbol{\theta}^T}{\partial x} - \mathbf{V}\mathbf{B}^T\mathbf{A}\frac{\partial\boldsymbol{\theta}^T}{\partial y},\\ \tilde{\mathcal{L}}_\theta\{\theta, w\} &= \left[\kappa\Delta\boldsymbol{\theta}\mathbf{A}^T - \left(\bar{u}\frac{\partial\boldsymbol{\theta}}{\partial x} + \bar{v}\frac{\partial\boldsymbol{\theta}}{\partial y}\right)\mathbf{A}^T - \left(\frac{\partial\bar{\theta}}{\partial x}\mathbf{U} + \frac{\partial\bar{\theta}}{\partial y}\mathbf{V}\right)\mathbf{B}^T\right]\mathbf{H}^T + \mathbf{U}\mathbf{B}^T\mathbf{A}\frac{\partial\boldsymbol{\theta}^T}{\partial x} + \mathbf{V}\mathbf{B}^T\mathbf{A}\frac{\partial\boldsymbol{\theta}^T}{\partial y} - \mathbf{U}\mathbf{B}^T\mathbf{H}^T\mathbf{H}\mathbf{A}\frac{\partial\boldsymbol{\theta}^T}{\partial x} - \mathbf{V}\mathbf{B}^T\mathbf{H}^T\mathbf{H}\mathbf{A}\frac{\partial\boldsymbol{\theta}^T}{\partial y}, \end{aligned}$$

and

$$\mathbb{E}\left[\tilde{\mathcal{L}}_\theta\{\theta, w\}\mathbf{H}\right] = \kappa\Delta\boldsymbol{\theta}\mathbf{A}^T - \left(\bar{u}\frac{\partial\boldsymbol{\theta}}{\partial x} + \bar{v}\frac{\partial\boldsymbol{\theta}}{\partial y}\right)\mathbf{A}^T - \left(\frac{\partial\bar{\theta}}{\partial x}\mathbf{U} + \frac{\partial\bar{\theta}}{\partial y}\mathbf{V}\right)\mathbf{B}^T - \left[\left(u_i\frac{\partial\theta_j}{\partial x} + v_i\frac{\partial\theta_j}{\partial y}\right)B_{\boldsymbol{\alpha}i}A_{\boldsymbol{\beta}j}\mathfrak{T}^{(\mathbf{H})}_{\boldsymbol{\alpha\beta\gamma}}\right]_{1\times\gamma}.$$

Further, we plug in the above equality into the inner product. We get

$$\langle\boldsymbol{\theta}^T, \mathbb{E}[\tilde{\mathcal{L}}_\theta\{\theta, w\}\mathbf{H}]\rangle = \kappa\langle\boldsymbol{\theta}^T, \Delta\boldsymbol{\theta}\rangle\mathbf{A}^T - \left\langle\boldsymbol{\theta}^T, \bar{u}\frac{\partial\boldsymbol{\theta}}{\partial x} + \bar{v}\frac{\partial\boldsymbol{\theta}}{\partial y}\right\rangle\mathbf{A}^T - \left\langle\boldsymbol{\theta}^T, \frac{\partial\bar{\theta}}{\partial x}\mathbf{U} + \frac{\partial\bar{\theta}}{\partial y}\mathbf{V}\right\rangle\mathbf{B}^T - [\mathfrak{T}^{(\boldsymbol{\theta})}_{ijk}B_{\boldsymbol{\alpha}i}A_{\boldsymbol{\beta}j}\mathfrak{T}^{(\mathbf{H})}_{\boldsymbol{\alpha\beta\gamma}}]_{k\times\gamma},$$

where the third-order m_w-by-m_θ-by-m_θ tensor

$$\mathfrak{T}^{(\boldsymbol{\theta})} = \left(\left\langle\theta_k, u_i\frac{\partial\theta_j}{\partial x} + v_i\frac{\partial\theta_j}{\partial y}\right\rangle\right)_{ijk}.$$

From Eq. (29), we have the DyBO-gPC formulation for the temperature component,

$$\begin{aligned} \frac{\partial\bar{\theta}}{\partial t} &= \kappa\Delta\bar{\theta} - \left(\bar{u}\frac{\partial\bar{\theta}}{\partial x} + \bar{v}\frac{\partial\bar{\theta}}{\partial y}\right) - \mathbf{U}\mathbf{B}^T\mathbf{A}\frac{\partial\boldsymbol{\theta}^T}{\partial x} - \mathbf{V}\mathbf{B}^T\mathbf{A}\frac{\partial\boldsymbol{\theta}^T}{\partial y},\\ \frac{\partial\boldsymbol{\theta}}{\partial t} &= -\boldsymbol{\theta}\mathbf{D}_\theta^T + \kappa\Delta\boldsymbol{\theta} - \left(\bar{u}\frac{\partial\boldsymbol{\theta}}{\partial x} + \bar{v}\frac{\partial\boldsymbol{\theta}}{\partial y}\right) - \left(\frac{\partial\bar{\theta}}{\partial x}\mathbf{U} + \frac{\partial\bar{\theta}}{\partial y}\mathbf{V}\right)\mathbf{B}^T\mathbf{A} - \left[\left(u_i\frac{\partial\theta_j}{\partial x} + v_i\frac{\partial\theta_j}{\partial y}\right)B_{\boldsymbol{\alpha}i}A_{\boldsymbol{\beta}j}A_{\gamma k}\mathfrak{T}^{(\mathbf{H})}_{\boldsymbol{\alpha\beta\gamma}}\right]_{1\times k},\\ \frac{\mathrm{d}\mathbf{A}}{\mathrm{d}t} &= \mathbf{A}\left(-\mathbf{C}_\theta^T + \kappa\langle\Delta\boldsymbol{\theta}^T, \boldsymbol{\theta}\rangle\boldsymbol{\Lambda}_{\boldsymbol{\theta}}^{-1} - \left\langle\bar{u}\frac{\partial\boldsymbol{\theta}^T}{\partial x} + \bar{v}\frac{\partial\boldsymbol{\theta}^T}{\partial y}, \boldsymbol{\theta}\right\rangle\boldsymbol{\Lambda}_{\boldsymbol{\theta}}^{-1}\right) - \mathbf{B}\left\langle\frac{\partial\bar{\theta}}{\partial x}\mathbf{U}^T + \frac{\partial\bar{\theta}}{\partial y}\mathbf{V}^T, \boldsymbol{\theta}\right\rangle\boldsymbol{\Lambda}_{\boldsymbol{\theta}}^{-1} - \left[\mathfrak{T}^{(\boldsymbol{\theta})}_{ijk}B_{\boldsymbol{\alpha}i}A_{\boldsymbol{\beta}j}\mathfrak{T}^{(\mathbf{H})}_{\boldsymbol{\alpha\beta\gamma}}\right]_{\gamma\times k}\boldsymbol{\Lambda}_{\boldsymbol{\theta}}^{-1}, \end{aligned}$$

where matrices $\mathbf{C}_\theta$ and $\mathbf{D}_\theta$ can be solved from the linear system (4) with $G_{*\theta}$,

$$\boldsymbol{\Lambda}_{\boldsymbol{\theta}}G_{*\theta} = \kappa\langle\boldsymbol{\theta}^T, \Delta\boldsymbol{\theta}\rangle - \left\langle\boldsymbol{\theta}^T, \bar{u}\frac{\partial\boldsymbol{\theta}}{\partial x} + \bar{v}\frac{\partial\boldsymbol{\theta}}{\partial y}\right\rangle - \left\langle\boldsymbol{\theta}^T, \frac{\partial\bar{\theta}}{\partial x}\mathbf{U} + \frac{\partial\bar{\theta}}{\partial y}\mathbf{V}\right\rangle\mathbf{B}^T\mathbf{A} - \left[\mathfrak{T}^{(\boldsymbol{\theta})}_{ijk}B_{\boldsymbol{\alpha}i}A_{\boldsymbol{\beta}j}A_{\gamma l}\mathfrak{T}^{(\mathbf{H})}_{\boldsymbol{\alpha\beta\gamma}}\right]_{k\times l}.$$

Next, we derive the DyBO equations for the vorticity component. Substituting the KL expansion into Eq. (45a), we have

$$\begin{aligned}\mathcal{L}_w\{\theta,w\} = {} & \left(\nu\Delta\bar{w} - \bar{u}\frac{\partial\bar{w}}{\partial x} - \bar{v}\frac{\partial\bar{w}}{\partial y} + \mu g\frac{\partial\bar{\theta}}{\partial x}\right) + \left[\nu\Delta\mathbf{W} - \left(\bar{u}\frac{\partial\mathbf{W}}{\partial x} + \bar{v}\frac{\partial\mathbf{W}}{\partial y}\right) - \left(\frac{\partial\bar{w}}{\partial x}\mathbf{U} + \frac{\partial\bar{w}}{\partial y}\mathbf{V}\right)\right]\mathbf{B}^T\mathbf{H}^T + \mu g\frac{\partial\boldsymbol{\theta}}{\partial x}\mathbf{A}^T\mathbf{H}^T \\ & - \mathbf{U}\mathbf{B}^T\mathbf{H}^T\mathbf{H}\mathbf{B}\frac{\partial\mathbf{W}^T}{\partial x} - \mathbf{V}\mathbf{B}^T\mathbf{H}^T\mathbf{H}\mathbf{B}\frac{\partial\mathbf{W}^T}{\partial y} + \mathbf{F}\mathbf{H}^T.\end{aligned}$$

Taking expectations on both sides, we have

$$\begin{aligned}\mathbb{E}[\mathcal{L}_w\{\theta,w\}] &= \left(\nu\Delta\bar{w} - \bar{u}\frac{\partial\bar{w}}{\partial x} - \bar{v}\frac{\partial\bar{w}}{\partial y} + \mu g\frac{\partial\bar{\theta}}{\partial x}\right) - \mathbf{U}\frac{\partial\mathbf{W}^T}{\partial x} - \mathbf{V}\frac{\partial\mathbf{W}^T}{\partial y}, \\ \tilde{\mathcal{L}}_w\{\theta,w\} &= \left[\nu\Delta\mathbf{W} - \left(\bar{u}\frac{\partial\mathbf{W}}{\partial x} + \bar{v}\frac{\partial\mathbf{W}}{\partial y}\right) - \left(\frac{\partial\bar{w}}{\partial x}\mathbf{U} + \frac{\partial\bar{w}}{\partial y}\mathbf{V}\right)\right]\mathbf{B}^T\mathbf{H}^T + \mu g\frac{\partial\boldsymbol{\theta}}{\partial x}\mathbf{A}^T\mathbf{H}^T\mathbf{U}\frac{\partial\mathbf{W}^T}{\partial x} + \mathbf{V}\frac{\partial\mathbf{W}^T}{\partial y} - \mathbf{U}\mathbf{B}^T\mathbf{H}^T\mathbf{H}\mathbf{B}\frac{\partial\mathbf{W}^T}{\partial x} - \mathbf{V}\mathbf{B}^T\mathbf{H}^T\mathbf{H}\mathbf{B}\frac{\partial\mathbf{W}^T}{\partial y} + \mathbf{F}\mathbf{H}^T\end{aligned}$$

and

$$\mathbb{E}[\tilde{\mathcal{L}}_w\{\theta,w\}\mathbf{H}] = \left[\nu\Delta\mathbf{W} - \left(\bar{u}\frac{\partial\mathbf{W}}{\partial x} + \bar{v}\frac{\partial\mathbf{W}}{\partial y}\right) - \left(\frac{\partial\bar{w}}{\partial x}\mathbf{U} + \frac{\partial\bar{w}}{\partial y}\mathbf{V}\right)\right]\mathbf{B}^T + \mu g\frac{\partial\boldsymbol{\theta}}{\partial x}\mathbf{A}^T - \left[\left(u_i\frac{\partial w_j}{\partial x} + v_i\frac{\partial w_j}{\partial y}\right)B_{\alpha i}B_{\beta j}\mathfrak{T}^{(\mathbf{H})}_{\alpha\beta\gamma}\right]_{1\times\gamma} + \mathbf{F}.$$

Substituting the above equality into the inner product gives

$$\begin{aligned}\left\langle\mathbf{W}^T, \mathbb{E}[\tilde{\mathcal{L}}_w\{\theta,w\}\mathbf{H}]\right\rangle = {} & \left\langle\mathbf{W}^T, \nu\Delta\mathbf{W} - \left(\bar{u}\frac{\partial\mathbf{W}}{\partial x} + \bar{v}\frac{\partial\mathbf{W}}{\partial y}\right) - \left(\frac{\partial\bar{w}}{\partial x}\mathbf{U} + \frac{\partial\bar{w}}{\partial y}\mathbf{V}\right)\right\rangle\mathbf{B}^T + \left\langle\mathbf{W}^T, \mu g\frac{\partial\boldsymbol{\theta}}{\partial x}\right\rangle\mathbf{A}^T - \left[\mathfrak{T}^{(\mathbf{W})}_{ijk}B_{\alpha i}B_{\beta j}\mathfrak{T}^{(\mathbf{H})}_{\alpha\beta\gamma}\right]_{k\times\gamma} \\ & + \left\langle\mathbf{W}^T, \mathbf{F}\right\rangle,\end{aligned}$$

where the third-order m_w-by-m_w-by-m_w tensor

$$\mathfrak{T}^{(\mathbf{W})} = \left(\left\langle w_k, u_i\frac{\partial w_j}{\partial x} + v_i\frac{\partial w_j}{\partial y}\right\rangle\right)_{ijk}. \tag{A.2}$$

From Eq. (29), we have the DyBO-gPC formulation for the vorticity component,

$$\begin{aligned}\frac{\partial\bar{w}}{\partial t} &= \left(\nu\Delta\bar{w} - \bar{u}\frac{\partial\bar{w}}{\partial x} - \bar{v}\frac{\partial\bar{w}}{\partial y} + \mu g\frac{\partial\bar{\theta}}{\partial x}\right) - \mathbf{U}\frac{\partial\mathbf{W}^T}{\partial x} - \mathbf{V}\frac{\partial\mathbf{W}^T}{\partial y}, \\ \frac{\partial\mathbf{W}}{\partial t} &= -\mathbf{W}\mathbf{D}_w^T + \left[\nu\Delta\mathbf{W} - \left(\bar{u}\frac{\partial\mathbf{W}}{\partial x} + \bar{v}\frac{\partial\mathbf{W}}{\partial y}\right) - \left(\frac{\partial\bar{w}}{\partial x}\mathbf{U} + \frac{\partial\bar{w}}{\partial y}\mathbf{V}\right)\right] + \mu g\frac{\partial\boldsymbol{\theta}}{\partial x}\mathbf{A}^T\mathbf{B}\left[\left(u_i\frac{\partial w_j}{\partial x} + v_i\frac{\partial w_j}{\partial y}\right)B_{\alpha i}B_{\beta j}B_{\gamma k}\mathfrak{T}^{(\mathbf{H})}_{\alpha\beta\gamma}\right]_{1\times k} + \mathbf{F}\mathbf{B}, \\ \frac{\mathrm{d}\mathbf{B}}{\mathrm{d}t} &= \mathbf{B}\left(-\mathbf{C}_w^T + \left\langle\nu\Delta\mathbf{W}^T - \left(\bar{u}\frac{\partial\mathbf{W}^T}{\partial x} + \bar{v}\frac{\partial\mathbf{W}^T}{\partial y}\right) - \left(\frac{\partial\bar{w}}{\partial x}\mathbf{U}^T + \frac{\partial\bar{w}}{\partial y}\mathbf{V}^T\right), \mathbf{W}\right\rangle\Lambda_{\mathbf{W}}^{-1}\right) + \mu g\mathbf{A}\left\langle\frac{\partial\boldsymbol{\theta}^T}{\partial x}, \mathbf{W}\right\rangle\Lambda_{\mathbf{W}}^{-1} - \left[\mathfrak{T}^{(\mathbf{W})}_{ijk}B_{\alpha i}B_{\beta j}\mathfrak{T}^{(\mathbf{H})}_{\alpha\beta\gamma}\right]_{\gamma\times k}\Lambda_{\mathbf{W}}^{-1} + \left\langle\mathbf{F}^T, \mathbf{W}\right\rangle\Lambda_{\mathbf{W}}^{-1},\end{aligned}$$

where matrices $\mathbf{C}_w$ and $\mathbf{D}_w$ can be solved from the linear system (4) with G_{*w},

$$\boldsymbol{\Lambda}_{\mathbf{W}}G_{*w} = \left\langle\mathbf{W}^T, \nu\Delta\mathbf{W} - \left(\bar{u}\frac{\partial\mathbf{W}}{\partial x} + \bar{v}\frac{\partial\mathbf{W}}{\partial y}\right) - \left(\frac{\partial\bar{w}}{\partial x}\mathbf{U} + \frac{\partial\bar{w}}{\partial y}\mathbf{V}\right)\right\rangle + \left\langle\mathbf{W}^T, \mu g\frac{\partial\boldsymbol{\theta}}{\partial x}\right\rangle\mathbf{A}^T\mathbf{B} - \left[\mathfrak{T}^{(\mathbf{W})}_{ijk}B_{\alpha i}B_{\beta j}B_{\gamma l}\mathfrak{T}^{(\mathbf{H})}_{\alpha\beta\gamma}\right]_{k\times l} + \left\langle\mathbf{W}^T, \mathbf{F}\right\rangle\mathbf{B}.$$

Combining the above discussion and using the generalized material derivative (41), we arrive at the DyBO-gPC formulation for SNSE (44).

References

[1] J. Anderson, Computational Fluid Dynamics, first ed., McGraw-Hill, 1995.

[2] I. Babuska, F. Nobile, R. Tempone, A stochastic collocation method for elliptic partial differential equations with random input data, SIAM J. Numer. Anal. 45 (3) (2007) 1005.

[3] G. Berkooz, P. Holmes, J.L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, Ann. Rev. Fluid Mech. 25 (1993) 539–575.

[4] S. Boyaval, C. LeBris, T. Lelièvre, Y. Maday, N. Nguyen, A. Patera, Reduced basis techniques for stochastic problems, Arch. Comput. Methods Eng. 17 (2012) 435–454.

[5] R.H. Cameron, W.T. Martin, The orthogonal development of non-linear functionals in series of Fourier–Hermite functionals, Ann. Math. 48 (2) (1947) 385–392.

[6] E. Candès, T. Tao, Near optimal signal recovery from random projections: universal encoding strategies?, IEEE Trans Inf. Theory 52 (12) (2006) 5406–5425.

[7] H.D. Ceniceros, T.Y. Hou, An efficient dynamically adaptive mesh for potentially singular solutions, J. Comput. Phys. 172 (2) (2001) 609–639.

[8] M.L. Cheng, Adaptive methods exploring intrinsic sparse structures of stochastic partial differential equations, Doctroal thesis, California Institute of Technology, 2012.

[9] M.L. Cheng, T.Y. Hou, Z.W. Zhang. A dynamically bi-orthogonal method for time-dependent stochastic partial differential equations I: derivation and algorithms. J. Comput. Phys. Accepted for publication, http://dx.doi.org/10.1016/j.jcp.2013.02.033.

[10] D.L. Donoho, Compressed sensing, IEEE Trans. Inf. Theory 52 (2006) 1289–1306.

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

[12] M.B. Giles, Multilevel Monte Carlo path simulation, Oper. Res. 56 (3) (2008) 607–617.

[13] W. Hackbusch, A sparse matrix arithmetic based on *H*-matrices Part I: introduction to *H*-matrices, Computing 62 (2) (1999) 89–108.

[14] T.Y. Hou, W. Luo, B. Rozovskii, H. Zhou, Wiener Chaos expansions and numerical solutions of randomly forced equations of fluid mechanics, J. Comput. Phys. 216 (2) (2006) 687–706.
[15] T.Y. Hou, R. Li, Dynamic depletion of vortex stretching and non-blowup of the 3-D incompressible Euler equations, J. Nonlinear Sci. 16 (6) (2006) 639–664.
[16] T.Y. Hou, R. Li, Computing nearly singular solutions using pseudo-spectral methods, J. Comput. Phys. 226 (2007) 379–397.
[17] T.Y. Hou, Z. Shi, Adaptive data analysis via sparse time-frequency representation, Adv. Adapt. Data Anal. 3 (2011) 1–28.
[18] T.Y. Hou and Z. Shi, Data-driven Time–frequency analysis, Appl. Comput. Harmonic Anal., in press. http://dx.doi.org/10.1016/j.acha.2012.10.001.
[19] K. Karhunen, Uber lineare methoden in der Wahrscheinlichkeitsrechnung, Ann. Acad. Sci. Fennicae. Ser. A.I. Math.-Phys. 37 (1947) 1–79.
[20] O. Le Maitre, Uncertainty propagation using Wiener–Haar expansions, J. Comput. Phys. 197 (1) (2004) 28–57.
[21] M. Loeve, Probability Theory, GTM. 46, fourth ed., vol. II, Springer-Verlag, 2006, ISBN 0-387-90262-7.
[22] W. Luo. Wiener chaos expansion and numerical solutions of stochastic partial differential equations, Ph.D. thesis, California Institute of Technology, 2006.
[23] W. Ma, M. Yu, J. Xin, S. Osher, A convex model and $L1$ minimization for musical noise reduction in blind source separation, Commun. Math. Sci. 10 (1) (2012) 223–238.
[24] Y. Maday, Reduced-basis method for the rapid and reliable solution of partial differential equations, Proceedings of International Conference on Mathematicians, Madrid, European Mathematical Society, Zurich, 2006.
[25] W.J. Morokoff, R.E. Caflisch, Quasi-sandom sequences and their discrepancies, SIAM J. Sci. Comput. 15 (6) (1994) 1251–1279.
[26] H. Niederreiter, Random Number Generation and Quasi-Monte Carlo Methods, SIAM Publ., 1992.
[27] G. Rozza, D.B. P Huynh, A.T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. Application to transport and continuum mechanics, Arch. Comput. Methods Eng. 15 (3) (2008) 229–275.
[28] T.P. Sapsis, P.F.J. Lermusiaux, Dynamically orthogonal field equations for continuous stochastic dynamical systems, Physica D 238 (2009) 2347–2360.
[29] T.P. Sapsis, P.F.J. Lermusiaux, Dynamical criteria for the evolution of the stochastic dimensionality in flows with uncertainty, Physica D 241 (1) (2012) 60–76.
[30] D. Venturi, On proper orthogonal decomposition of randomly perturbed fields with applications to flow past a cylinder and natural convection over a horizontal plate, J. Fluid Mech. 559 (2006) 215–254.
[31] N. Wierner, The homogeneous chaos, Amer. J. Math. 60 (1938) 897–936.
[32] D. Xiu, G.E. Karniadakis, The Wiener–Askey polynomial chaos for stochastic differential equations, SIAM J. Sci. Comput. 24 (2) (2002) 614–644.
[33] D. Xiu, G.E. Karniadakis, Modeling uncertainty in flow simulations via generalized polynomial chaos, J. Comput. Phys. 187 (1) (2003) 137–167.
[34] M. Yu, W. Ma, J. Xin, S. Osher, Multi-channel $L1$ regularized convex speech enhancement model and fast computation by the split Bregman method, IEEE Trans. Audio Speech Lang. Process. 20 (2) (2012) 661–675.