



A least-squares approximation of partial differential equations with high-dimensional random inputs

Alireza Doostan^{*}, Gianluca Iaccarino

Mechanical Engineering Department, Stanford University, Stanford, CA 94305, USA

ARTICLE INFO

Article history:

Received 22 October 2008

Accepted 2 March 2009

Available online 19 March 2009

Keywords:

Uncertainty quantification

Separated representation

Alternating least-squares

Curse of dimensionality

Stochastic partial differential equations

ABSTRACT

Uncertainty quantification schemes based on stochastic Galerkin projections, with global or local basis functions, and also stochastic collocation methods in their conventional form, suffer from the so called *curse of dimensionality*: the associated computational cost grows exponentially as a function of the number of random variables defining the underlying probability space of the problem. In this paper, to overcome the curse of dimensionality, a low-rank separated approximation of the solution of a stochastic partial differential (SPDE) with high-dimensional random input data is obtained using an *alternating least-squares* (ALS) scheme. It will be shown that, in theory, the computational cost of the proposed algorithm grows linearly with respect to the dimension of the underlying probability space of the system. For the case of an elliptic SPDE, an *a priori* error analysis of the algorithm is derived. Finally, different aspects of the proposed methodology are explored through its application to some numerical experiments.

© 2009 Elsevier Inc. All rights reserved.

1. Introduction

The behavior and evolution of complex systems are known only partially due to lack of knowledge about the governing physical laws or limited information regarding their operating conditions and input parameters (e.g. material properties). Uncertainty quantification (UQ) plays a crucial role in the construction of credible mathematical/computational models for such systems. In this work problems governed by a set of partial differential equations (PDEs) are considered. The uncertain parameters are introduced using available observations and are described as random variables/processes in a probabilistic framework. A computational model is then defined to approximate the statistics of the solution of these PDEs, thus propagating the uncertainty from the input parameters to the response of the system.

Monte Carlo sampling has been utilized for a long time as a general purpose scheme for uncertainty propagation. Recently there has been an increasing interest in developing more efficient computational models for the analysis of uncertain systems as Monte Carlo techniques are known to have slow rate of convergence. In particular, *perturbation-based* techniques, [14], are shown to be effective for situations where input parameters exhibit small variabilities. *Stochastic Galerkin schemes*, [11,6,28,17,2,26], have been successfully applied to problems arising from different areas of engineering and are extremely useful for situations in which the number of uncertain parameters is not large. More recently, there has been a great amount of attention to *collocation-based* techniques that take advantage of existing deterministic solvers for moderate $\mathcal{O}(10)$ number of uncertain variables [24,18,27,1,20,19].

In many applications one has to deal with a large number of uncertain input parameters. In such situations, with the exception of the Monte Carlo technique, the methods mentioned above suffer from the so called *curse of dimensionality*: their

^{*} Corresponding author. Tel.: +1 650 723 2330; fax: +1 650 725 3525.

E-mail addresses: doostan@stanford.edu (A. Doostan), jops@stanford.edu (G. Iaccarino).

computational cost grows exponentially as a function of the number of random variables defining the underlying probability space of the problem. More specifically, the computational cost of the stochastic Galerkin schemes with global polynomials, e.g. Wiener Hermite chaos, [11], depends on the number of terms in the solution expansion. For the case of order p approximation in d independent random variables, the cardinality of the associated basis is $P = (p + d)!/p!d!$ which increases exponentially with respect to p and d . Model reduction techniques have been developed in [7,10,22,21] to reduce this drawback for the case of stochastic Galerkin schemes.

Stochastic collocation schemes, [27,1,20,19] based on isotropic and anisotropic sparse grids reduce the problem of curse of dimensionality normally associated with conventional tensor-product schemes. The computational cost of the stochastic collocation scheme using a tensor-product grid and isotropic sparse-grid constructed from M points in each direction of a d -dimensional space is $\mathcal{O}(M^d)$ and $\mathcal{O}(C^d M(\log M)^{d-1})$, respectively. Note that both estimates grow exponentially with respect to d .

In this work a novel, alternative approach is investigated; the present approach extends the *alternating least-squares* (ALS) approximation technique of [3,4] to obtain a *separated representation* of the solution to partial differential equations with high-dimensional random inputs. In theory, the computational cost of this algorithm grows linearly with respect to the dimension of the probability space of the system. The fundamental feature of the proposed technique that addresses the curse of dimensionality is the adoption of a low-rank separated representation of functions in the high-dimensional probability space. The proposed approach is fundamentally different from stochastic Galerkin and collocation schemes. More specifically, it deviates from stochastic Galerkin schemes by not assuming any pre-determined basis along the stochastic dimension for the approximation. It is also different from stochastic collocation schemes as it does not solve the problem on the quadrature grid *exactly*.

The paper is organized as follows. Section 2 summarizes the separated representation of a d -dimensional function as well as the ALS algorithm of [3,4] to construct such representation. For the sake of consistency throughout this section, a similar notation to that of [3,4] is adopted. Following that, in Section 3, it will be shown how one can cast the discrete analog of the governing SPDE in a form that can be readily incorporated in the separated approximation framework. Finally, in Section 4, numerical experiments are performed on a one-dimensional (1D) elliptic and a two-dimensional (2D) hyperbolic SPDE, each with 30-dimensional probability space, to illustrate the performance of the proposed procedures.

2. Separated representations and the ALS algorithm

Separation-of-variable techniques have been widely used to approximate high-dimensional functions using one-dimensional operations, thus virtually eliminating the curse of dimensionality, see [23,16,15,3,25,13,4] and references therein. Let u be a d -dimensional function. It can be approximated as

$$u(y_1, \dots, y_d) \approx \phi_1(y_1) \cdots \phi_d(y_d). \tag{1}$$

The above approximation can be improved by introducing a series of such representations,

$$u(y_1, \dots, y_d) = \sum_{l=1}^r s_l \phi_1^l(y_1) \cdots \phi_d^l(y_d) + \mathcal{O}(\epsilon), \tag{2}$$

which is called a *separated representation* with *separation rank* r . Given a target accuracy ϵ , the approximation (2) can be achieved by tailoring unknown quantities $\{\phi_i^l(y_i)\}$, $\{s_l\}$, and an optimal separation rank r , for instance, through a nonlinear optimization scheme. For general functions, the separated representation (2) is not unique. However, any separated approximation satisfying the accuracy ϵ is acceptable. The main advantage of adopting the separated representation (2) is that many algebraic operations in d dimensions can be performed using a series of one-dimensional operations. Therefore, in theory, the computational efforts increase only linearly with respect to d . The goal is to present the ALS algorithm of [3,4] for the separated representation of the solution of a linear system of equations arising from spatial/temporal discretization of SPDEs using only one-dimensional operations. To achieve this objective some preliminary notations and definitions are introduced.

Notation 1. A scalar function $u(y_1, \dots, y_d)$ in d dimensions is a mapping from the Euclidean space \mathbb{R}^d to the real line \mathbb{R} , $u : \mathbb{R}^d \rightarrow \mathbb{R}$. A vector $\mathbf{u} \equiv u(j_1, \dots, j_d)$ with $j_k = 1, \dots, M_k$ is a discrete representation of function u on a d -dimensional tensor-product grid \mathcal{G} with $\prod_{k=1}^d M_k$ nodes. Without loss of generality it is assumed that the grid is isotropic, i.e. $M_k = M$ for $k = 1, \dots, d$.

Definition 1 ([3,4] (*Separated representation of a vector*)). Let \otimes denote the Kronecker product. For a given accuracy, ϵ , a vector $\mathbf{u} = u(j_1, \dots, j_d)$ in d dimension is approximated by

$$\sum_{l=1}^r s_l u_1^l(j_1) u_2^l(j_2) \cdots u_d^l(j_d) \equiv \sum_{l=1}^r s_l \mathbf{u}_1^l \otimes \mathbf{u}_2^l \otimes \cdots \otimes \mathbf{u}_d^l, \tag{3}$$

with $s_l \in \mathbb{R}$ being a scalar and $\mathbf{u}_k^l \in \mathbb{R}^M$ a one-dimensional vector with entries $u_k^l(j_k)$ and unit Euclidean norm. The approximation error is required to be less than ϵ , i.e.

$$\|\mathbf{u} - \sum_{l=1}^r s_l \mathbf{u}_1^l \otimes \mathbf{u}_2^l \otimes \cdots \otimes \mathbf{u}_d^l\| \leq \epsilon, \tag{4}$$

where $\|\cdot\|$ denotes the Frobenius norm over \mathcal{G} .

2.1. Reducing the separation rank r using ALS

Let v be a function whose discrete form \mathbf{v} has a large separation rank r_v . The ALS algorithm aims at reducing the separation rank of \mathbf{v} while maintaining an accurate representation. More specifically, let

$$\mathbf{v} = \sum_{l=1}^{r_v} s_l^v \mathbf{v}_1^l \otimes \mathbf{v}_2^l \otimes \cdots \otimes \mathbf{v}_d^l \tag{5}$$

with large r_v . Given an accuracy ϵ , the goal is to find

$$\mathbf{u}_\epsilon^M = \sum_{l=1}^{r_u} s_l^u \mathbf{u}_1^l \otimes \mathbf{u}_2^l \otimes \cdots \otimes \mathbf{u}_d^l, \tag{6}$$

with $r_u \ll r_v$ such that

$$\|\mathbf{v} - \mathbf{u}_\epsilon^M\| \leq \epsilon. \tag{7}$$

It is assumed that such lower separation rank approximation of \mathbf{v} exists, otherwise the ALS reduction algorithm is not useful.

For a fixed r_u , one can minimize the distance between \mathbf{v} and \mathbf{u} , in the Frobenius sense over \mathcal{G} , by adapting $\{\mathbf{u}_i^l\}$ and $\{s_l^u\}$. Due to the already separated form of \mathbf{v} and \mathbf{u}_ϵ^M such minimization can be performed, for instance, in the form of a sequence of linear one-dimensional least-squares problems which is referred to as alternating least-squares (ALS) algorithm.

Alternating least-squares (ALS) algorithm. For a fixed r_u , an initial guess for \mathbf{u} is constructed by randomly initializing $\{\mathbf{u}_i^l\}$ and thus s_l^u . The optimization steps are then as follows [3,4].

- Loop over dimensions $k = 1, \dots, d$
 - Loop over grid points $j_k = 1, \dots, M$ in direction k
 - Fix $\{\mathbf{u}_i^l\}_{i \neq k}$ and solve the following normal equation associated with a linear least-squares problem to update $\{\mathbf{u}_k^l\}$ and thus s_l^u :

$$\mathbb{B} \mathbf{c}_{j_k} = \mathbf{b}_{j_k}, \tag{8}$$

with

$$\mathbb{B}(\tilde{l}, \tilde{l}) = \prod_{i \neq k} \langle \mathbf{u}_i^1, \mathbf{u}_i^1 \rangle$$

and

$$\mathbf{b}_{j_k}(\tilde{l}) = \sum_{l=1}^{r_v} s_l^v v_k^l(j_k) \prod_{i \neq k} \langle \mathbf{v}_i^l, \mathbf{u}_i^l \rangle.$$

- Update $s_l^u = \left(\sum_{j_k} c_{j_k}^2(\tilde{l}) \right)^{1/2}$ and $\mathbf{u}_k^l(j_k) = c_{j_k}(\tilde{l}) / s_l^u$, where $\tilde{l} = 1, \dots, r_u$ and $j_k = 1, \dots, M$.

Here $\langle \cdot, \cdot \rangle$ denotes the usual inner-product of two vectors. The above algorithm monotonically reduces the difference between \mathbf{v} and \mathbf{u} until the rate of the reduction is small. If the desired accuracy has not been achieved, the rank r_u must be increased to reduce the error further. It is in general not possible to determine the optimal rank r_u a priori. However, in order to achieve a near-optimal rank r_u , it is proposed to start from a low separation rank, e.g. $r_u = 1$, and reduce the representation error using the ALS algorithm and, if necessary, increase r_u until $\|\mathbf{v} - \mathbf{u}_\epsilon^M\| \leq \epsilon$ is attained. The overall procedure is then summarized as:

Separation rank reduction algorithm:

- Set $r_u = 1$; (randomly) initialize $\{\mathbf{u}_i^1\}$ and s_l^u .
- Loop while $\|\mathbf{v} - \mathbf{u}_\epsilon^M\| > \epsilon$.
 - Perform the ALS algorithm until $\|\mathbf{v} - \mathbf{u}_\epsilon^M\|$ does not decrease much.
 - Set $r_u = r_u + 1$; (randomly) initialize $\{\mathbf{u}_i^{r_u}\}$ and $s_l^{r_u}$.

It is shown in [3,4] that one full ALS iteration, as described above, requires $\mathcal{O}(d \cdot r_u(r_u^2 + r_v \cdot M))$ operations. In theory, as far as r_u is finite and does not depend on d , the complexity of ALS algorithm scales linearly with d . However, in practice, r_u may depend mildly on d ; therefore the computational cost of the above ALS scheme is nearly linear with respect to d .

2.2. Regularization

In general, the problem (7) is ill-conditioned due to loss of precision and thus requires regularization. A simple *Tikhonov* regularization of (7) has been proposed in [4] which aims at controlling the *condition number*,

$$\kappa \equiv \frac{\left(\sum_{l=1}^{r_u} (s_l^u)^2\right)^{1/2}}{\|\mathbf{u}_\epsilon^M\|}, \tag{9}$$

of the separated representation \mathbf{u}_ϵ^M . More specifically, to prevent the loss of significant digits and to achieve the approximation (7), one has to satisfy $\kappa\mu\|\mathbf{u}_\epsilon^M\| \leq \epsilon$, where μ is the machine precision. Practically, such regularization can be implemented by simply replacing \mathbb{B} with $\mathbb{B} + \lambda\mathbb{I}$ in Eq. (8) and choosing the scalar λ slightly larger than μ [4]. Here \mathbb{I} denotes the identity matrix of the same size of \mathbf{B} .

In the following, the ALS scheme will be applied to the solution of the linear system of equations arising from spatial-temporal discretization of an SPDE.

3. Solution of an SPDE

3.1. Problem definition and numerical method

Consider a linear partial differential equation with stochastic operator and forcing, and let $u(x, t, \omega)$ be the solution in $\mathcal{D} \times [0, T] \times \Omega \rightarrow \mathbb{R}$, such that the following equation holds almost surely in Ω ,

$$\begin{aligned} \mathcal{L}(x, t, \omega; u) &= f(x, t, \omega) \quad (x, t) \in \mathcal{D} \times [0, T] \\ u(x, t, \omega; u) &= g(x, t) \quad (x, t) \in \partial\mathcal{D} \times [0, T] \\ u(x, 0, \omega) &= h(x, \omega) \quad x \in \mathcal{D}, \end{aligned} \tag{10}$$

where Ω is the set of elementary events and $\omega \in \Omega, \mathcal{D}$ and T denote the spatial extent and the time interval of the problem, respectively. The randomness in the problem is induced by the uncertainty in the underlying parameters of the corresponding physical system, e.g. heat conductivity, viscosity, initial conditions, etc., and is assumed to be a function of a finite, possibly very large, number of random variables. Such representation can be obtained through, for instance, spectral decomposition of the covariance kernel of the underlying (second-order) random fields/processes which is referred to as Karhunen–Loeve expansion. Therefore, the random differential operator of (10) can be represented as

$$\mathcal{L}(x, t, \omega; u) = \mathcal{L}(x, t, y_1(\omega), \dots, y_{d_o}(\omega); u), \tag{11}$$

where $\{y_k\}_{k=1}^{d_o}$ are random variables, e.g. Karhunen–Loeve expansion random variables, that define the finite-dimensional noise in \mathcal{L} . Similarly, one can rewrite $f(x, t, \omega) = f(x, t, y_1(\omega), \dots, y_{d_f}(\omega))$ and $h(x, \omega) = h(x, y_1(\omega), \dots, y_{d_h}(\omega))$.

In the present work, it is assumed that random variables $\{y_k\}_{k=1}^d$ with $d \equiv d_o + d_f + d_h$ are independent random variables with probability density functions $\{\rho_k\}_{k=1}^d$, respectively. Let $\Gamma_k \equiv y_k(\Omega)$ be the image of the random variable $y_k(\omega)$; the underlying probability space Γ is the product of images of random variables $y_k(\omega)$, i.e. $\Gamma \equiv \prod_{k=1}^d \Gamma_k$.

The solution of (10) is a mapping of $\{x, t, y_1(\omega), \dots, y_d(\omega)\}$, i.e.

$$u(x, t, \omega) := u(x, t, y_1(\omega), \dots, y_d(\omega)), \tag{12}$$

which is in general nonlinear.

Considering a spatial-temporal discretization scheme, the semi-discrete equivalent form of (10) typically simplifies to solution of a random linear system of equations of the form

$$\mathbb{A}_n(\omega)\mathbf{u}_n(\omega) = \mathbf{f}_n(\omega) \quad \forall n, \tag{13}$$

where n denotes the index associated with the time integration scheme and is henceforth omitted for the sake of a simpler notation. In the present study, it is assumed that the random matrix $\mathbb{A}(\omega)$ and the random vector $\mathbf{f}(\omega)$ in (13) admit a separated representation with respect to their coordinates $\{y_1(\omega), \dots, y_d(\omega)\}$. For the case of linear problem (10), such setting is readily available once the corresponding uncertain parameters are characterized in a separated representation, for instance, through a procedure similar to Karhunen–Loeve expansion (here with independent random variables) or a polynomial chaos expansion (possibly with large separation rank that can be further reduced using the ALS algorithm). When large samples of stochastic processes representing the uncertainty in the system are available, a more consistent approach is to use procedures of [5]. On the other hand the formalism of [9] can be employed when the uncertain parameters are assimilated based on the available (possibly limited) data. Consequently, the discrete representation of (13) on a tensor-product grid \mathcal{G} consisting of M nodes along each direction y_k (total of M^d nodes) reads,

$$\left(\sum_{l=1}^{r_A} \mathbb{A}_0^l \otimes \mathbb{A}_1^l \otimes \dots \otimes \mathbb{A}_d^l\right)\mathbf{u} = \sum_{l=1}^{r_f} \mathbf{f}_0^l \otimes \mathbf{f}_1^l \otimes \dots \otimes \mathbf{f}_d^l. \tag{14}$$

$\mathbb{A}_0^l \in \mathbb{R}^{N,N}$ and $\mathbf{f}_0^l \in \mathbb{R}^N$ are deterministic (sparse) matrices and vectors, respectively, whose size, N , is determined by the spatial discretization scheme and are obtained from the discretization of deterministic modes in the representation of underlying stochastic fields. For $k = 1, \dots, d$, the diagonal matrices $\mathbb{A}_k^l \in \mathbb{R}^{M,M}$ and vectors $\mathbf{f}_k^l \in \mathbb{R}^M$ hold quadrature values in the finite-dimensional representation of corresponding stochastic fields. Finally, \mathbf{u} is the tensor-product representation of the solution to be calculated. In general, the construction of separated representations of the coefficient tensor \mathbb{A} and the right-hand-side tensor \mathbf{f} in (14) along with their separation ranks $r_{\mathbb{A}}$ and $r_{\mathbf{f}}$, respectively, depend on the SPDE, the spatial-temporal discretizations, and how the finite-dimensional noise representations are obtained. One example of such construction is provided in Section 4.1.

Remark 1. Notice that the representation (14) is in fact a compact notation for the tensor-product stochastic collocation approximation of (13) on \mathcal{G} . The solution \mathbf{u} in (14) is therefore the solution of the tensor-product stochastic collocation applied to (13). The procedure proposed here, however, departs from the collocation approach by approximating the solution \mathbf{u} on \mathcal{G} , within the class of separated representations, and not evaluating it exactly at the quadrature points. As is described in more details in what follows, such approximation allows one to break the issue of curse of dimensionality associated with the tensor-product collocation approach.

3.2. Separated approximation of (14) using ALS

A separated representation of \mathbf{u} with low separation rank is used to approximate the solution of (14). This can be achieved as in Section 2.1 using the ALS algorithm. In particular,

$$\mathbf{u}_\epsilon^M \equiv \sum_{l=1}^{r_u} s_l^u \mathbf{u}_0^l \otimes \mathbf{u}_1^l \otimes \dots \otimes \mathbf{u}_d^l, \tag{15}$$

is sought such that

$$\left\| \sum_{l=1}^{r_{\mathbb{A}}} \mathbb{A}_0^l \otimes \mathbb{A}_1^l \otimes \dots \otimes \mathbb{A}_d^l \left(\sum_{l=1}^{r_u} s_l^u \mathbf{u}_0^l \otimes \mathbf{u}_1^l \otimes \mathbf{u}_2^l \otimes \dots \otimes \mathbf{u}_d^l \right) - \sum_{l=1}^{r_{\mathbf{f}}} \mathbf{f}_0^l \otimes \mathbf{f}_1^l \otimes \dots \otimes \mathbf{f}_d^l \right\| \leq \epsilon, \tag{16}$$

for a desired accuracy ϵ . To this end, the separated rank reduction algorithm of Section 2.1 is modified as follows.

Separation rank reduction algorithm:

- Set $r_u = 1$; (randomly) initialize $\{\mathbf{u}_i^1\}_{i=0}^d$ and thus s_1^u .
- Loop while $\|\mathbf{f} - \mathbb{A}\mathbf{u}_\epsilon^M\| > \epsilon$.
 - Perform a modified ALS algorithm, as described below, until $\|\mathbf{f} - \mathbb{A}\mathbf{u}_\epsilon^M\|$ does not decrease much.
 - Set $r_u = r_u + 1$; (randomly) initialize $\{\mathbf{u}_i^{r_u}\}_{i=0}^d$ and thus $s_{r_u}^u$.

Due to spatial-temporal discretization of (10), one has to solve different normal equations in the alternating least-squares algorithm of Section 2.1. The modified normal equations for updates along the spatial direction, $k = 0$, and each random direction $y_k, k = 1, \dots, d$, are as follows.

Modified normal equations. Let $J = N$ when $k = 0$ and $J = M$ when $k = 1, \dots, d$. For each $k = 0, 1, \dots, d$, the quantities $\{\mathbf{u}_k^{\bar{l}}\}_{l=1}^{r_u}$ are simultaneously updated by solving the following normal equation,

$$\mathbb{B}\mathbf{c} = \mathbf{b}, \tag{17}$$

where \mathbb{B} contains $r_u \times r_u$ blocks with size $J \times J$. The (\bar{l}, \bar{l}) th block is obtained from

$$\mathbb{B}_{(\bar{l}, \bar{l})}[i', j'] = \sum_{l=1}^{r_{\mathbb{A}}} \sum_{l=1}^{r_{\mathbb{A}}} \left(\sum_{j=1}^J \mathbb{A}_k^l[j, i'] \mathbb{A}_k^l[j, j'] \right) \prod_{i \neq k} \langle \mathbb{A}_i^l \mathbf{u}_i^{\bar{l}}, \mathbb{A}_i^l \mathbf{u}_i^{\bar{l}} \rangle. \tag{18}$$

Moreover \mathbf{b} contains r_u vectors each with size J where the \bar{l} th component is computed as

$$\mathbf{b}_{\bar{l}}(i') = \sum_{l=1}^{r_{\mathbf{f}}} \sum_{l=1}^{r_{\mathbb{A}}} \left(\sum_{j=1}^J \mathbb{A}_k^l[j, i'] \mathbf{f}_k^l(j) \right) \prod_{i \neq k} \langle \mathbf{f}_i^l, \mathbb{A}_i^l \mathbf{u}_i^{\bar{l}} \rangle. \tag{19}$$

Finally, $s_l^u = (\sum_{j_k} \mathbf{c}_l^2(j_k))^{1/2}$ and $\mathbf{u}_k^{\bar{l}}(j_k) = \mathbf{c}_l(j_k) / s_l^u$ for all $\bar{l} = 1, \dots, r_u$.

Note that for updates along random directions, $k = 1, \dots, d$, the linear system (17) decouples into M smaller linear systems each with size $r_u \times r_u$ for each nodal point j_k . However, when $k = 0$ the linear system (17) is coupled according to the spatial discretization and thus has the size $(N \cdot r_u) \times (N \cdot r_u)$.

In practice, the linear system (17) is not formed explicitly, particularly when updates along spatial direction x , i.e. $k = 0$, are performed. Depending on the choice of the solver, the linear system (17) can be solved by only matrix–vector multiplications and vector–vector dot-products in which the major memory requirement is due to having sparse matrices $\{\mathbb{A}_0^l\}_{l=1}^{r_{\mathbb{A}}}$ and vectors $\{\mathbf{f}_0^l\}_{l=1}^{r_{\mathbf{f}}}$ stored. In the present study, a preconditioned conjugate gradient solver is employed; an incomplete LU decomposition of the mean of the random matrix $\mathbb{A}(\omega)$, i.e. \mathbb{A}_0^1 , is used as the preconditioner.

A thorough discussion on the order of computations required to cast and solve (17) with an explicit solver, which requires more computations compared to iterative ones, is presented in [4]. There, it is shown that, in theory, when ranks r_A , r_f , and r_u are small and independent of d , the computational cost of a full ALS iteration from $k = 0$ to $k = d$, remains linear with respect to d . For cases where iterative solvers with matrix–vector and vector–vector operations are employed, as in the present study, the computational cost of a full ALS iteration with updates along $k = 0$ to $k = d$ is much less than those of explicit solvers with the order remaining linear in d . However, in some practical situations, r_A , r_f , and r_u might mildly depend on d which results in an increase in the computational cost to more than linear growth.

Remark 2. Similar to tensor-product stochastic collocation schemes, the choice of grid points j_k affects the convergence and the accuracy of the approximation. In the present work, based on the distribution of the random variables y_k , quadrature rules are used to distribute these abscissas along each direction y_k .

Remark 3. In practice, it is more natural to monitor the convergence of the ALS algorithm based on the *normalized* norm of the residual rather than the norm of the residual itself. Throughout the rest of this study, the condition $\|\mathbf{f} - \mathbf{A}\mathbf{u}_\epsilon^M\| \leq \epsilon\|\mathbf{f}\|$ is used as the stopping criterion.

3.3. Response statistics

The computation of response statistics for the proposed technique follows that of a tensor-product stochastic collocation technique [27,1] except that the computational cost grows linearly with respect to dimension d . Given the discrete solution \mathbf{u}_ϵ^M in Eq. 15, one can estimate the desirable statistics, e.g. moments, of u based on: (1) sampling from interpolating surface of \mathbf{u}_ϵ^M ; or (2) numerical integration using quadrature rules for its moments, e.g. mean and variance. In the following, these are described in more details.

Statistics based on interpolation. A multi-dimensional interpolation technique with, for instance, Lagrange polynomials can be employed to construct the solution response surface based on the nodal values of u along the random dimensions. Given the separated solution \mathbf{u}_ϵ^M , such interpolation is cast as a sequence of one-dimensional interpolations, hence, resulting in a total computational cost that grows linearly with respect to d . More specifically, let $\mathcal{I}(\mathbf{u}_\epsilon^M)(y_1, \dots, y_d) : \mathbb{R}^N \times \mathcal{G} \rightarrow \mathbb{R}^N \times \Gamma$ denote the interpolation of the random solution \mathbf{u}_ϵ^M in Γ , then

$$\mathcal{I}(\mathbf{u}_\epsilon^M)(y_1, \dots, y_d) = \sum_{l=1}^{r_u} s_l^u \mathbf{u}_0^l \prod_{k=1}^d \mathcal{I}(u_k^l)(y_k), \tag{20}$$

where

$$\mathcal{I}(u_k^l)(y_k) = \sum_{j_k=1}^M u_k^l(j_k) L_{j_k}(y_k), \quad \forall k, l, \tag{21}$$

is the one-dimensional interpolation along direction y_k and $L_{j_k}(y_k)$ is the Lagrange polynomial corresponding to the node j_k . Desirable statistics of \mathbf{u}_ϵ^M are then computed by sampling from $\mathcal{I}(\mathbf{u}_\epsilon^M)(y_1, \dots, y_d)$.

Statistics based on quadrature integration. The integral-form statistics, e.g. moments, of the response quantities of interest can be obtained as a sequence of one-dimensional quadrature integrations once their separated representation is available. For instance, the estimate of the mean and second moment of the nodal solution vector \mathbf{u} is obtained as

$$\mathbb{E}[\mathbf{u}_\epsilon^M] = \int_{\Gamma} \mathbf{u}_\epsilon^M(y_1, \dots, y_d) \prod_{k=1}^d (\rho_k dy_k) = \sum_{l=1}^{r_u} s_l^u \mathbf{u}_0^l \prod_{k=1}^d \left(\int_{\Gamma_k} u_k^l(y_k) \rho_k dy_k \right) = \sum_{l=1}^{r_u} s_l^u \mathbf{u}_0^l \prod_{k=1}^d \left(\sum_{j_k=1}^M u_k^l(j_k) w_{j_k} \right) \tag{22}$$

and similarly

$$\mathbb{E}[(\mathbf{u}_\epsilon^M)^2] \equiv \int_{\Gamma} (\mathbf{u}_\epsilon^M \circ \mathbf{u}_\epsilon^M)(y_1, \dots, y_d) \prod_{k=1}^d (\rho_k dy_k) = \sum_{l=1}^{r_u} \sum_{l'=1}^{r_u} s_l^u s_{l'}^u (\mathbf{u}_0^l \circ \mathbf{u}_0^{l'}) \prod_{k=1}^d \left(\sum_{j_k=1}^M u_k^l(j_k) u_k^{l'}(j_k) w_{j_k} \right), \tag{23}$$

respectively. In Eqs. (22) and (23), \mathbb{E} is the mathematical expectation operator, \circ denotes the Hadamard product of two vectors, and w_{j_k} is the weight associated with the quadrature point j_k .

3.4. Error analysis

In this Section, an *a priori* error analysis of the proposed scheme is provided. It will be demonstrated how the error decays as the approximation is refined by simultaneously increasing grid points M along random directions and decreasing the target accuracy ϵ in the separated approximation of the solution. In particular, the present error analysis is obtained for the random linear system of Eq. (13) arising from a symmetric spatial discretization of a linear elliptic SPDE of the form,

$$\begin{aligned} -\nabla \cdot (a(x, \omega) \nabla u(x, \omega)) &= f(x, \omega), \quad x \in \mathcal{D}, \\ u(x, \omega) &= 0, \quad x \in \partial\mathcal{D}, \end{aligned} \tag{24}$$

that holds almost surely in Ω and $\omega \in \Omega$. Under some standard regularity conditions on the stochastic functions a and $f : \mathcal{D} \times \Omega \rightarrow \mathbb{R}$, the problem (24) is well-posed and its solution u is analytic with respect to $\{y_k\}_{k=1}^d$ that characterize the finite-dimensional noise representation of a and $f : \mathcal{D} \times \Gamma \rightarrow \mathbb{R}$, [1]. In particular, it is assumed that $0 < a(x, y_1, \dots, y_d) < \infty$ almost surely in Γ and $\forall x \in \mathcal{D}$.

Proposition 1. Let \mathbf{u} be the exact solution of (13) associated with problem (24). Let $\mathcal{I}(\mathbf{u}_\epsilon^M)$, as in (20), be the interpolation of separated approximation of \mathbf{u} based on M Gaussian quadrature nodes corresponding to probability density functions ρ_k and with a target accuracy ϵ for the ALS algorithm of Section 3. Then for some constants C_1 and C_2 independent of M and ϵ ,

$$\|\mathbf{u} - \mathcal{I}(\mathbf{u}_\epsilon^M)\|_{L_2(\mathbb{R}^N) \otimes L_2(\Gamma)} \leq C_1(\sigma, d)e^{-\sigma M} + C_2 \epsilon \mathcal{K}(\mathbb{A}) \|\mathbf{u}\|_{L_2(\mathbb{R}^N) \otimes L_\infty(\Gamma)}, \tag{25}$$

where d is the dimension of the random space, σ depends on the smoothness of u with regard to $\{y_k\}_{k=1}^d$ [1], and $\mathcal{K}(\mathbb{A}) \equiv \|\mathbb{A}\| \cdot \|\mathbb{A}^{-1}\|$ is the condition number of the discrete representation of \mathbb{A} on the tensor-product grid \mathcal{G} .

Proof. Let $\mathcal{I}(\mathbf{u}) : \mathbb{R}^N \times \mathcal{G} \rightarrow \mathbb{R}^N \times \Gamma$ be the Lagrange interpolation of the exact solution \mathbf{u} on \mathcal{G} . Then,

$$\|\mathbf{u} - \mathcal{I}(\mathbf{u}_\epsilon^M)\|_{L_2(\mathbb{R}^N) \otimes L_2(\Gamma)} \leq \|\mathbf{u} - \mathcal{I}(\mathbf{u})\|_{L_2(\mathbb{R}^N) \otimes L_2(\Gamma)} + \|\mathcal{I}(\mathbf{u}) - \mathcal{I}(\mathbf{u}_\epsilon^M)\|_{L_2(\mathbb{R}^N) \otimes L_2(\Gamma)}, \tag{26}$$

where $\|\mathbf{u} - \mathcal{I}(\mathbf{u})\|_{L_2(\mathbb{R}^N) \otimes L_2(\Gamma)}$ is identical to the error arising from a tensor-product stochastic collocation (using \mathcal{G}) applied to problem (13). From [1],

$$\|\mathbf{u} - \mathcal{I}(\mathbf{u})\|_{L_2(\mathbb{R}^N) \otimes L_2(\Gamma)} \leq C_1(\sigma, d)e^{-\sigma M}, \tag{27}$$

in which σ is related to the smoothness of \mathbf{u} in Γ . The second term on the right-hand-side of inequality (29), $\|\mathcal{I}(\mathbf{u}) - \mathcal{I}(\mathbf{u}_\epsilon^M)\|_{L_2(\mathbb{R}^N) \otimes L_2(\Gamma)}$, is due to the separated approximation of the tensor-product stochastic collocation solution and decays by decreasing ϵ . Let w_k be the Gaussian quadrature weights corresponding to abscissas $j_k = 1, \dots, M$. Under some mild constraints (c.f. [8]) on the probability density functions ρ_k associated with random variables y_k with bounded ranges, the maximum Gaussian quadrature weight satisfies, [8],

$$\max_{j_k} w_{j_k} \leq \frac{C_q}{M}, \tag{28}$$

where C_q is a constant that depends on the type of quadrature rule (e.g. Legendre–Gaussian quadrature vs. Chebyshev–Gaussian quadrature) and is independent of M . As a result of (28) and the linearity of the interpolation operator together with other properties of Gaussian quadrature integration, the proof proceeds as

$$\begin{aligned} \|\mathcal{I}(\mathbf{u}) - \mathcal{I}(\mathbf{u}_\epsilon^M)\|_{L_2(\mathbb{R}^N) \otimes L_2(\Gamma)}^2 &= \|\mathcal{I}(\mathbf{u} - \mathbf{u}_\epsilon^M)\|_{L_2(\mathbb{R}^N) \otimes L_2(\Gamma)}^2 = \sum_{j_0} \sum_{j_1} \dots \sum_{j_d} (u(j_0, j_1, \dots, j_d) - u_\epsilon^M(j_0, j_1, \dots, j_d))^2 \prod_{k=1}^d w_{j_k} \\ &\leq \left(\max_{j_k} w_{j_k}\right)^d \|\mathbf{u} - \mathbf{u}_\epsilon^M\|_{L_2(\mathbb{R}^N) \otimes L_2(\mathbb{R}^{M^d})}^2 \leq \left(\frac{C_q}{M}\right)^d \epsilon^2 \mathcal{K}^2(\mathbb{A}) \|\mathbf{u}\|_{L_2(\mathbb{R}^N) \otimes L_2(\mathbb{R}^{M^d})}^2 \\ &\leq C_q^d \epsilon^2 \mathcal{K}^2(\mathbb{A}) \|\mathbf{u}\|_{L_2(\mathbb{R}^N) \otimes L_\infty(\mathbb{R}^{M^d})}^2 \leq C_q^d \epsilon^2 \mathcal{K}^2(\mathbb{A}) \|\mathbf{u}\|_{L_2(\mathbb{R}^N) \otimes L_\infty(\Gamma)}^2, \end{aligned}$$

where,

$$\|\mathbf{u} - \mathbf{u}_\epsilon^M\|_{L_2(\mathbb{R}^N) \otimes L_2(\mathcal{G})} \leq \epsilon \mathcal{K}(\mathbb{A}) \|\mathbf{u}\|_{L_2(\mathbb{R}^N) \otimes L_2(\mathcal{G})}$$

is obtained from standard matrix analysis techniques [12]. \square

4. Numerical examples

In this section, two numerical examples are considered to verify the proposed algorithm for the solution of SPDEs with high-dimensional random inputs. In the first example, the *method of manufactured solution* (MMS) is employed as a verification means for the separated approximation of an elliptic SPDE in one physical dimension (1D). The second example illustrates a 2D unsteady scalar transport equation with random diffusion and inflow velocities.

4.1. Example I: 1D elliptic SPDE

Consider the following one-dimensional stochastic elliptic equation that holds a.s. in Ω :

$$\begin{aligned} -\frac{\partial}{\partial x} \left(a(x, \omega) \frac{\partial u(x, \omega)}{\partial x} \right) &= f(x, \omega) \quad x \in \mathcal{D} = (0, 1) \\ u(0, \omega) &= u(1, \omega) = 0. \end{aligned} \tag{29}$$

The coefficient $a(x, \omega)$ is represented by

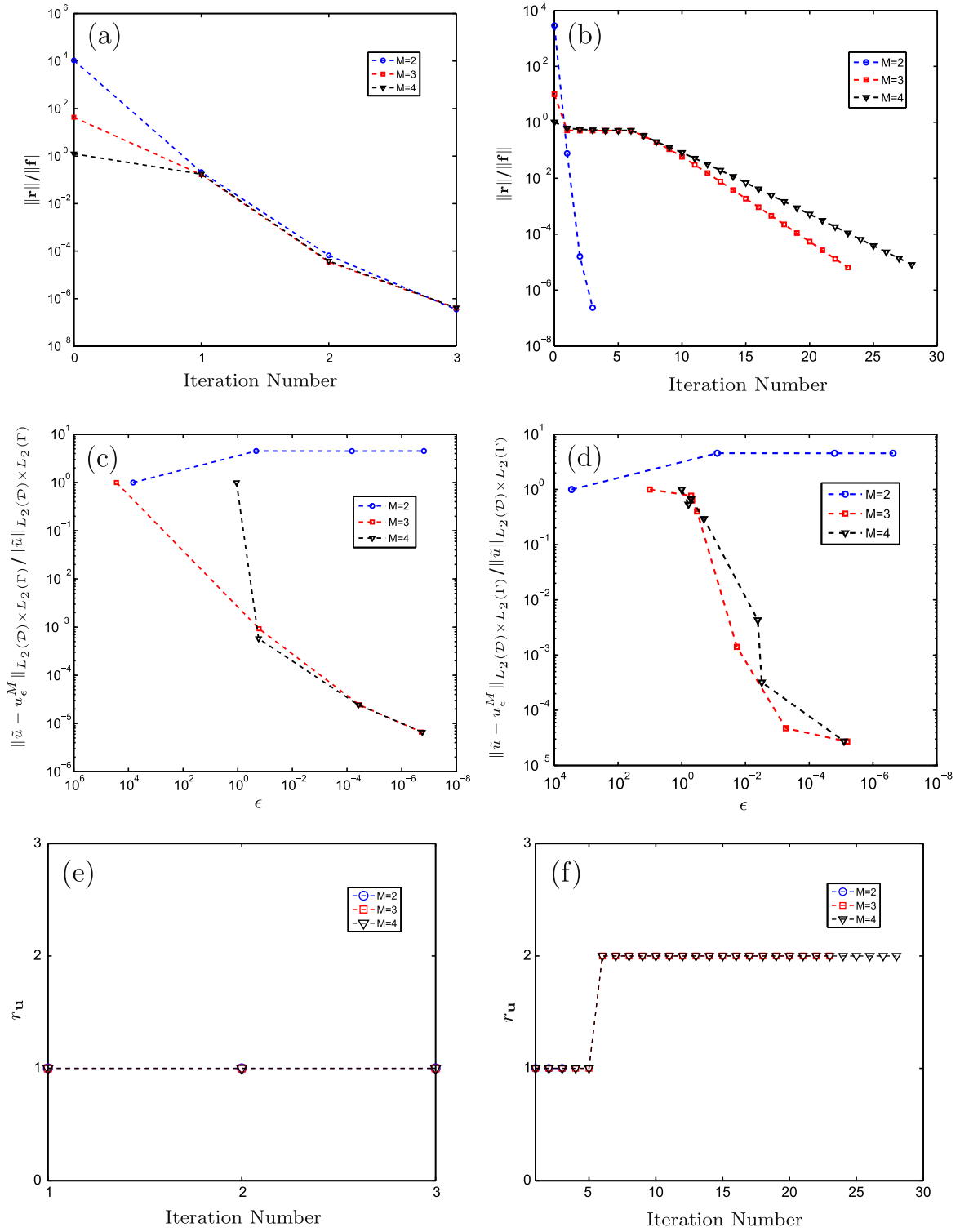


Fig. 1. (a) and (b): convergence of the residual $\mathbf{r} = \mathbf{f} - \mathbb{A}\mathbf{u}_\epsilon^M$ for $r_{\bar{u}_1} = 1$ and $r_{\bar{u}_2} = 2$, respectively. (c) and (d): $L_2(\mathcal{D}) \times L_2(\Gamma)$ convergence of the separated approximation u_ϵ^M for $r_{\bar{u}_1} = 1$ and $r_{\bar{u}_2} = 2$, respectively, (e) and (f): rank of the approximate solution, $r_{\mathbf{u}}$, in each ALS iteration for $r_{\bar{u}_1} = 1$ and $r_{\bar{u}_2} = 2$, respectively. For all cases smallest target accuracy $\epsilon = 1 \times 10^{-5}$ is considered.

$$a(x, \omega) = \bar{a} + \sigma \sum_{k=1}^{d_a} \cos(2\pi kx) y_k(\omega), \tag{30}$$

with $\{y_k\}_{k=1}^{d_a}$ i.i.d. uniform random variables $U(-1, 1)$, $\bar{a} = 50$, $\sigma = 1$ and $d_a = 30$. Notice that given the above setting, $a(x, \omega)$ is strictly positive on $\mathcal{D} \times \Omega$. The method of manufactured solution (MMS) is utilized to verify the solution of the proposed scheme with a finite element discretization of (29). Two manufactured solutions

$$\begin{aligned} \tilde{u}_1(x, \omega) &= \sin(\pi x)y_1^2(\omega)y_2^2(\omega) \quad \text{and} \\ \tilde{u}_2(x, \omega) &= \sin(\pi x)y_1^2(\omega)y_2^2(\omega) + 0.1 \sin(10\pi x)y_4^2(\omega)y_5^2(\omega) \end{aligned} \tag{31}$$

with ranks $r_{\tilde{u}_1} = 1$ and $r_{\tilde{u}_2} = 2$, respectively, are considered. The corresponding manufactured inputs \tilde{f}_1 and \tilde{f}_2 are then obtained from $\tilde{f}_j \equiv -\frac{\partial}{\partial x}(a(x, \omega)\frac{\partial \tilde{u}_j(x, \omega)}{\partial x})$ for $j = 1, 2$. Consequently, \tilde{u}_1 and \tilde{u}_2 are the exact solutions of (29) with $f = \tilde{f}_1$ and $f = \tilde{f}_2$, respectively. A linear finite element discretization of (29) with uniform mesh size $h = 1 \times 10^{-5}$ is considered. Such resolution leads to a spatial discretization error that is not dominant in the approximation. The convergence of the proposed scheme to the exact (manufactured) solution, with respect to different number of quadrature points M and smallest target accuracy $\epsilon = 1 \times 10^{-5}$, is explored. The tensor-product grid \mathcal{G} is constructed by distributing M points along each direction y_k based on the Clenshaw–Curtis rule, i.e.

$$y_k(j_k) = -\cos\left(\frac{\pi(j_k - 1)}{M - 1}\right), \quad j_k = 1, \dots, M. \tag{32}$$

Notice that one may consider other types of quadrature, e.g Legendre–Gaussian quadrature, as well considering the fact that the quality of the approximation of output statistics based on one rule is, in general, different from those of others. More investigation of such discrepancy is outside the scope of this study.

For the case of $r_{\tilde{u}_1} = 1$, the semi-discretization of problem (29) leads to the linear system of equations

$$\left(\mathbb{A}_0^1 + \sum_{k=1}^{d_a} \mathbb{A}_0^{k+1} y_k(\omega)\right) \mathbf{u}(\omega) = \left(\mathbf{f}_0^1 + \sum_{k=1}^{d_a} \mathbf{f}_0^{k+1} y_k(\omega)y_1^2(\omega)y_2^2(\omega)\right), \tag{33}$$

where sparse matrices $\mathbb{A}_0^1, \mathbb{A}_0^2, \dots, \mathbb{A}_0^{d_a+1}$ are obtained from the standard linear finite element discretization of the differential operator of (29) when the coefficient $a(x, \omega)$ is replaced by $\bar{a}, \sigma \cos(2\pi x), \dots, \sigma \cos(2\pi d_a x)$, respectively. Similarly, vectors $\mathbf{f}_0^1, \mathbf{f}_0^2, \dots, \mathbf{f}_0^{d_a+1}$ are computed by, first, sorting all terms in the right-hand-side of (29) with respect to contributions from random variables y_k and then calculating the finite element discretization of corresponding deterministic coefficients.

Notice that, in (33), the random matrix $\mathbb{A}(\omega)$ and random vector $\mathbf{f}(\omega)$ are already in separated form with respect to $\{y_k\}_{k=1}^{d_a}$ and with separation ranks $r_{\mathbb{A}} = r_{\mathbf{f}} = d_a + 1$. Therefore, the discrete representation (14) of system (33) on \mathcal{G} is readily available by picking values of each univariate function, along $\{y_k\}_{k=1}^{d_a}$, at quadrature points. For instance, let δ_{ij} denote the Kronecker's delta, then $\mathbb{A}_k^1[i, j] = \delta_{ij}$ for $k = 1, \dots, d_a$, $\mathbb{A}_k^{l>1}[i, j] = \delta_{ij}y_k(j)$ if $l = k + 1$ and $\mathbb{A}_k^{l>1}[i, j] = \delta_{ij}$ otherwise, $r_{\mathbf{f}} = d_a + 1, \mathbf{f}_k^1(i) = 1, \mathbf{f}_2^2(i) = (y_2(i))^2, \mathbf{f}_1^3(i) = y_1(i), \mathbf{f}_2^3(i) = (y_2(i))^3$, and so on. Similar procedures are carried out for the case of $r_{\tilde{u}_2} = 2$.

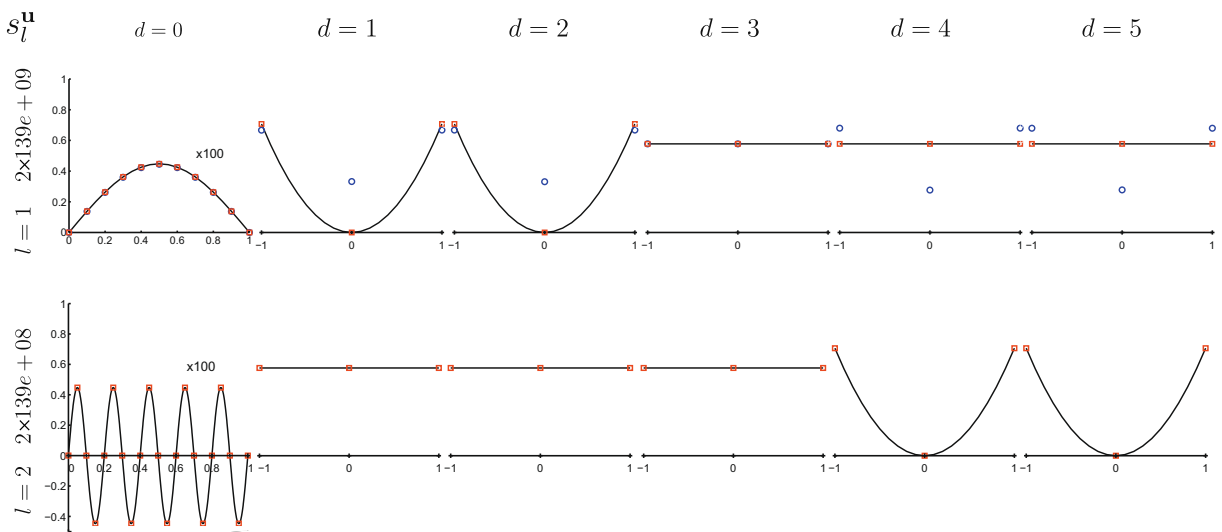


Fig. 2. Comparison of components of the exact solution \tilde{u} with their corresponding approximations along the physical space as well as the first five random directions (at the collocation nodes). The circles and squares correspond to the solution of the ALS algorithm at the end of rank one, $r_{\mathbf{u}} = 1$, and rank two, $r_{\mathbf{u}} = 2$, iterations, respectively. Solid lines correspond to the exact quantities from \tilde{u} .

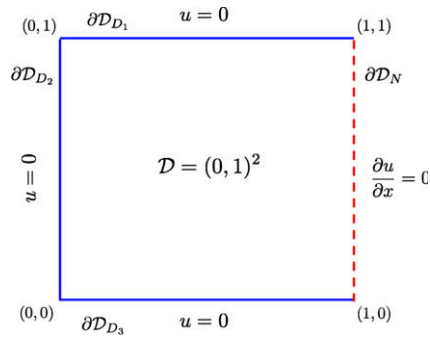


Fig. 3. Schematic of the Example II geometry along with its Dirichlet ($\partial\mathcal{D}_D \equiv \partial\mathcal{D}_{D_1} \cup \partial\mathcal{D}_{D_2} \cup \partial\mathcal{D}_{D_3}$) and Neumann ($\partial\mathcal{D}_N$) boundary conditions.

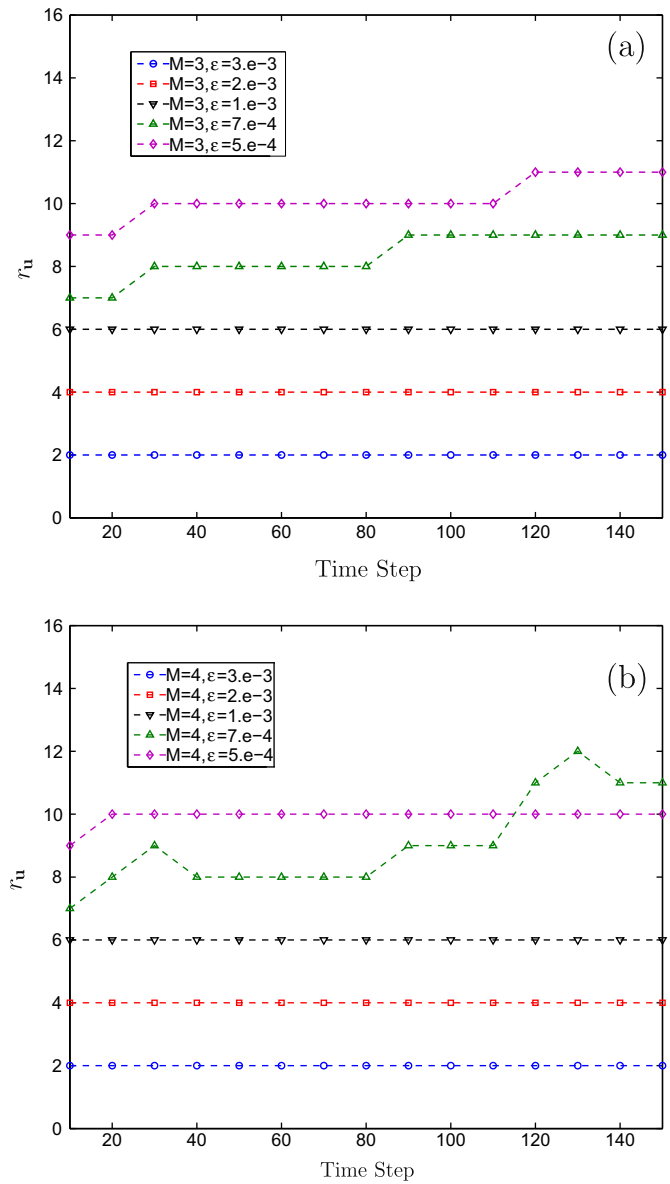


Fig. 4. Evolution of the rank of \mathbf{u}_c^M , r_u , corresponding to different analysis time steps and different accuracies ϵ . (a) $M = 3$, and (b) $M = 4$.

Remark 4. In this example, the ranks of \mathbb{A} and \mathbf{f} are directly related to the problem dimension d_a , i.e. $r_{\mathbb{A}} = r_{\mathbf{f}} = d_a + 1$, which is simply a consequence of using a Karhunen–Loeve-type expansion for finite-dimensional noise representation of $a(x, \omega)$. This will increase the computational cost of the algorithm to more than linear with respect to d_a , e.g. quadratic in d_a when an iterative solver is used and $N \gg M$. Alternatively, one can obtain the finite-dimensional noise representation of $a(x, \omega)$ by applying the separated rank reduction technique of this study on the stochastic differential operator itself to possibly achieve $r_{\mathbb{A}} < d$. The interested reader is referred to [3–5] for more details.

Fig. 1(a) and (b) shows the decay of the residual, $\mathbf{r} \equiv \mathbf{f} - \mathbb{A}\mathbf{u}_\epsilon^M$, of the corresponding linear system for $r_{\mathbf{u}_1} = 1$ and $r_{\mathbf{u}_2} = 2$, respectively. The root mean-square error $\|\tilde{u} - u_\epsilon^M\|_{L_2(\mathcal{D}) \times L_2(\Gamma)}$ is calculated based on the Lagrange interpolation of \mathbf{u}_ϵ^M , as described in Section 3.3, and is plotted in Fig. 1(c) and (d) for different values of ϵ . Finally, the rank of the approximate solution \mathbf{u}_ϵ^M at any iteration of the ALS algorithm is shown in Fig. 1(e) and (f). For the case of $r_{\mathbf{u}_1} = 1$ and when $M = 3, 4$, the ALS algorithm converges with rank $r_{\mathbf{u}_1} = 1$ and therefore identifies the rank of the exact solution \tilde{u}_1 . However, the solution fails to converge with $M = 2$ simply due to the fact that the second order terms in \tilde{u}_1 are not captured when $M = 2$. This is, clearly, an indication of the role of number of quadrature points M in the overall convergence of the solution as described in Section 3.4. For the case of $r_{\mathbf{u}_2} = 2$ and $M = 3, 4$ the algorithm performs five iterations with rank $r_{\mathbf{u}_1} = 1$ until it increases the approximation rank ($r_{\mathbf{u}_2} = 2$) in order to achieve the target accuracy $\epsilon = 1 \times 10^{-5}$. When $r_{\mathbf{u}_2} = 2$ and $M = 2$, the residual of the ALS approximation decays enough and the algorithm stops with $r_{\mathbf{u}_2} = 1$ thus failing to detect the correct rank of the solution.

More interestingly, the components of \mathbf{u}_ϵ^M along each direction x, y_1, \dots, y_{d_a} converge to the corresponding quantities of \tilde{u} as ϵ decreases and with sufficiently large M . This is depicted in Fig. 2 only for the case of $r_{\mathbf{u}_2} = 2$ and when $M = 3$.

4.2. Example II: 2D unsteady convection diffusion equation

A two-dimensional (2D) transient convection diffusion equation with stochastic velocity and diffusion fields are considered to model the transport of a scalar quantity (e.g. heat, concentration of a pollutant) in a random medium where the convection effects are not known completely. The mathematical model for the random scalar field $u(\mathbf{x}, t, \omega)$ reads

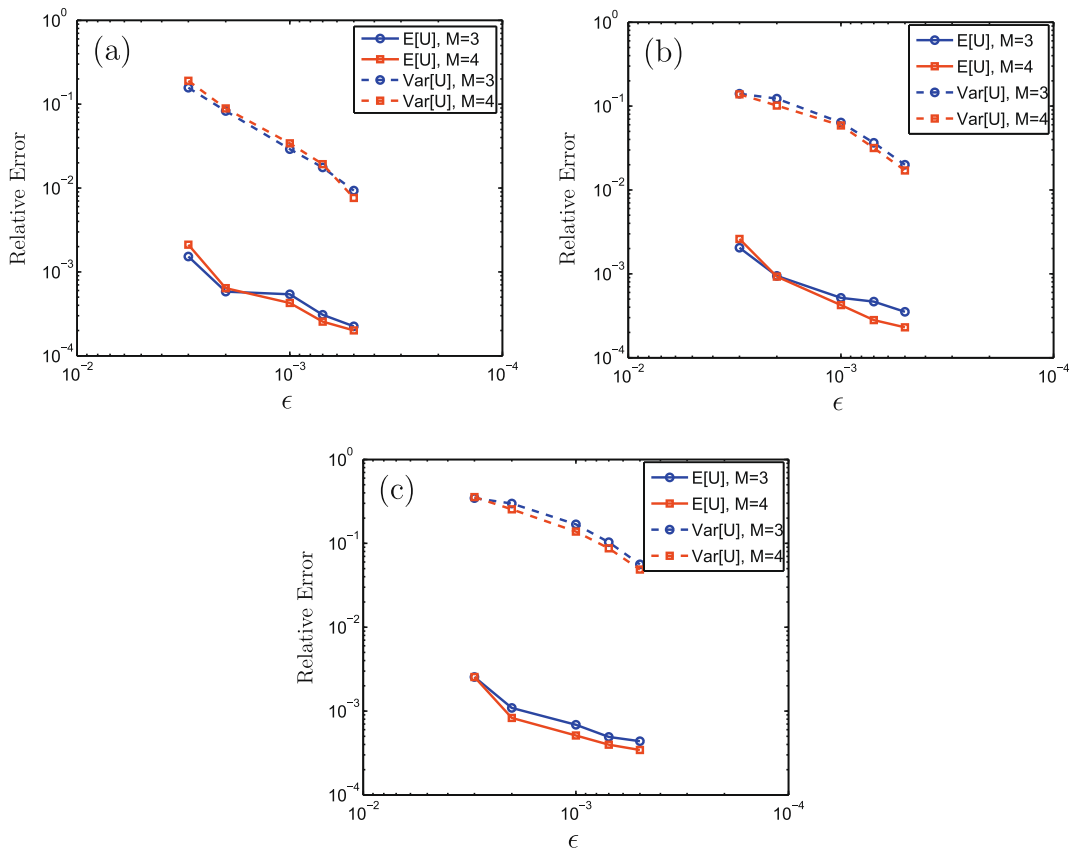


Fig. 5. Mean-square convergence of the solution mean and variance at three different time steps $t = 20\Delta t, 70\Delta t, 150\Delta t$. The reference quantities are obtained from a Monte Carlo simulation with a sample size of 10^6 . The solid and dashed lines correspond to the solution mean and variance respectively. (a) $t = 20\Delta t$, (b) $t = 70\Delta t$, and (c) $t = 150\Delta t$.

$$\begin{aligned}
 \frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u &= \nabla \cdot (a \nabla u) \quad \text{in } \mathcal{D} \times [0, T] \times \Omega, \\
 u &= 0 \quad \text{on } \partial \mathcal{D}_D \times [0, T] \times \Omega, \\
 \nabla u \cdot \mathbf{n} &= 0 \quad \text{on } \partial \mathcal{D}_N \times [0, T] \times \Omega, \\
 u(\mathbf{x}, 0, \omega) &= g(\mathbf{x}) \quad \text{on } \mathcal{D} \times \Omega,
 \end{aligned}
 \tag{34}$$

and holds almost surely in Ω . Here, \mathbf{n} is the unit vector normal to the boundary. The Dirichlet and Neumann boundary conditions are denoted by $\partial \mathcal{D}_D$ and $\partial \mathcal{D}_N$, respectively, and are schematically depicted in Fig. 3. The initial scalar field is modeled by a Gaussian function $g(\mathbf{x}) = \exp(-(x - 0.5)^2/0.1^2 - (y - 0.5)^2/0.1^2)$ defined on $\mathcal{D} = (0, 1)^2$. The diffusion coefficient a and velocity field $\mathbf{v} \equiv (v_x, v_y)$ are stochastic fields represented by a total of 30 independent random variables. More specifically,

$$a(\mathbf{x}, \omega) = \bar{a} + \sigma_a \sum_{k=1}^{d_a} \frac{1}{k^{1/2}} \cos(k\pi x) \cos(k\pi y) y_k(\omega),
 \tag{35}$$

where $\bar{a} = 0.01$, $d_a = 10$, $\sigma_a = 0.0018$, and $\{y_k\}_{k=1}^{d_a}$ are *i.i.d.* uniform random variables on $U(-1, +1)$. Notice that the construction (35) along with its particular choice of parameters lead to a strictly positive coefficient a on $\mathcal{D} \times \Omega$. Finally, the divergence-free random velocity field \mathbf{v} is assumed to have the form,

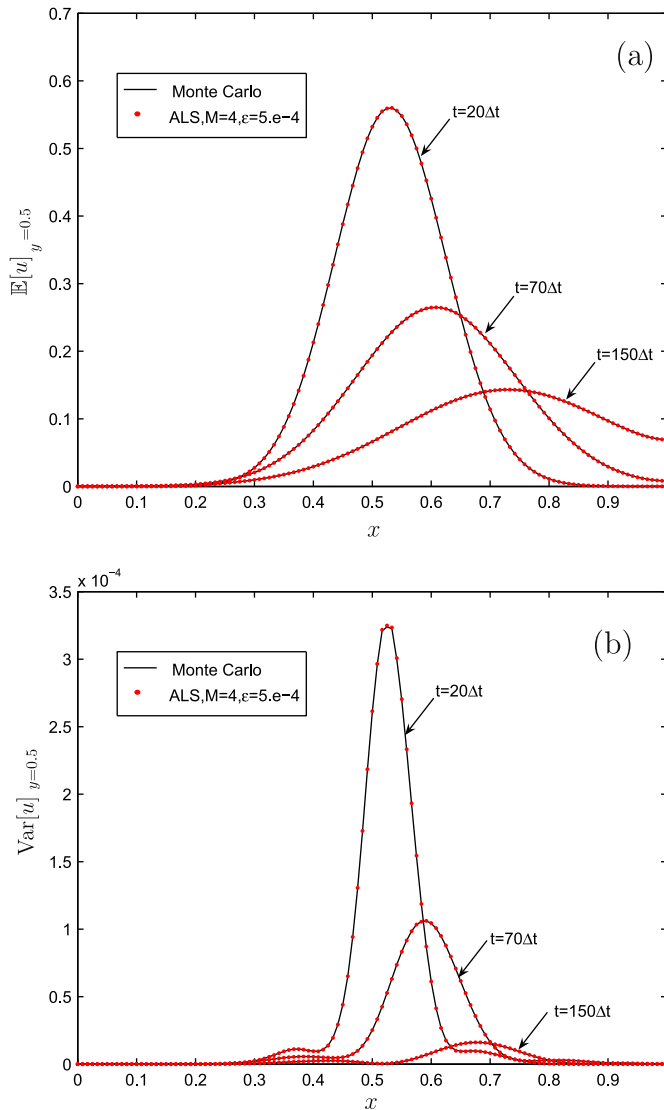


Fig. 6. Evolution of solution mean and variance along the centerline of the domain, $y = 0.5$, and the corresponding ALS approximation with $M = 4$ and $\epsilon = 5 \times 10^{-4}$. (a) Evolution of the solution mean for analysis times $t = 20\Delta t, 70\Delta t, 150\Delta t$, and (b) evolution of the solution variance for analysis times $t = 20\Delta t, 70\Delta t, 150\Delta t$.

$$\begin{aligned}
 v_x(\mathbf{x}, \omega) &= v_0 y(1-y) + \sigma_v \sum_{k=1}^{d_v} \frac{1}{k^{1/2}} \cos(2k\pi x) \sin(2k\pi y) \zeta_k(\omega) \\
 v_y(\mathbf{x}, \omega) &= \sigma_v \sum_{k=1}^{d_v} \frac{1}{k^{1/2}} \sin(2k\pi x) (1 - \cos(2k\pi y)) \zeta_k(\omega).
 \end{aligned} \tag{36}$$

The subsequent analysis are done with $v_0 = 0.64$, $\sigma_v = 0.0032$, $d_v = 20$. Random variables $\{\zeta_k\}_{k=1}^{d_v}$ are also *i.i.d.* uniformly distributed on $[-1, +1]$ and are independent of $\{y_k\}_{k=1}^{d_a}$ in Eq. (35).

4.2.1. Discretization schemes

4.2.1.1. Spatial-temporal discretization. The proposed method has the flexibility to accommodate any desirable spatial-temporal discretization of a given SPDE as far as it satisfies the usual discretization requirements such as stability and well-posedness. In the present study an *implicit-central difference* scheme with a uniform cartesian grid of size $\Delta x = \Delta y = 1/120$ and $\Delta t = 0.01$ is used for spatial-temporal discretization of Eq. (34). The analysis is performed till $T = 1.5$ to let the scalar u decay sufficiently from its initial value.

4.2.1.2. Random discretization. A tensor-product grid with Legendre–Gaussian quadratures of size $M = 3, 4$ along the range of random variables $\{y_k\}_{k=1}^{d_a}$ and $\{\zeta_k\}_{k=1}^{d_v}$ is considered to construct the approximation.

4.2.2. Results

The performance of the ALS algorithm in estimating the temporal mean and variance of u is investigated. Analyses are carried out with $M = 3, 4$ in combination with $\epsilon = 3 \times 10^{-3}, 2 \times 10^{-3}, 1 \times 10^{-3}, 7 \times 10^{-4}, 5 \times 10^{-4}$. For the first analysis time step, the initial approximation rank $r_u = 1$ is selected and the algorithm is executed to achieve the target accuracy. For the subsequent time steps, the solution is initialized based on the solution of the previous time step with a rank that is minimum of 1 or $r_u - 3$. This will significantly reduce the overall computational cost as opposed to resetting the initial rank to $r_u = 1$ for all time steps.

Fig. 4 shows the evolution of the solution rank with respect to different analysis time steps and accuracies ϵ . Notice that while u is 30-dimensional, the rank r_u never exceeds 12 during the entire analysis. In general, depending on the nature of the problem, the solution rank may decrease or increase in time. With sufficiently small time steps and by choosing the subsequent solution rank slightly smaller than that of the previous time step, one can achieve a *rather* minimal computation for the current time step.

The convergence of the mean and variance of the solution corresponding to three different analysis time steps is illustrated in Fig. 5. Similar to other uncertainty propagation schemes, e.g. stochastic Galerkin and stochastic collocation schemes, the proposed approach in its present form suffers from the *long time integration*: the accuracy of approximation deteriorates with respect to time. This, for instance, can be observed by comparing the magnitude and decay of relative variance error as a function of time in Fig. 5. A brute force approach to eliminate this issue is to simultaneously increase M and decrease ϵ in time.

Finally, the evolution of solution mean and variance and the corresponding finest conducted approximation, $M = 4$ and $\epsilon = 5 \times 10^{-4}$, along the centerline of the physical domain, $y = 0.5$, is depicted in Fig. 6.

5. Conclusion

The present study tackles the issue of curse of dimensionality associated with the approximation of high-dimensional stochastic partial differential equations. The proposed method is based on the low-rank separated representation of multivariate functions and an alternating least-squares (ALS) algorithm that minimizes the difference between the quantity to be estimated and the corresponding separated representation. This is considered to be a generalization of the singular value decomposition technique without being optimal. The number of required operations, hence the computational cost, and also the memory requirements for the proposed scheme is formally linear with respect to the dimension of the underlying probability space in which the solution exists. This is, in general, a significant advantage over the widely used stochastic Galerkin and collocation schemes when solving problems with large number of random variables.

The proposed ALS algorithm is verified by an *a priori* error analysis of the separated solution and through its application to a simple 1D elliptic SPDE using the method of manufactured solution. Components of the arising error have been identified and have been shown to decay when the analysis is refined. The proposed algorithm is furthermore applied to estimate the first two statistics of the solution of a 2D scalar transport equation with uncertain diffusion and inflow velocities. The existence of a low-rank separated solution to an example problem with a high-dimensional probability space has been observed. Furthermore, the ability of the ALS scheme to capture the low-rank separated approximate solution is verified in both numerical experiments.

The effect of long time integration has been investigated briefly in a numerical experiment. The target accuracy, ϵ , and the number of collocation points, M , along the range of each random variable in the ALS algorithm can be adjusted a priori to achieve a certain accuracy at the final analysis time. However, this might require a relatively small ϵ (large separation rank)

to begin with, thus making the proposed procedures not as efficient. Similar to the case of stochastic Galerkin and collocation schemes, further studies are needed to improve the efficiency of the proposed approach for long time integrations of unsteady problems.

Acknowledgment

This material is based upon work supported by the Department of Energy, National Nuclear Security Administration under Award Number NA28614.

References

- [1] I. Babuška, F. Nobile, R. Tempone, A stochastic collocation method for elliptic partial differential equations with random input data, *SIAM J. Numer. Anal.* 45 (3) (2007) 1005–1034.
- [2] I. Babuška, R. Tempone, G. Zouraris, Galerkin finite element approximations of stochastic elliptic partial differential equations, *SIAM J. Numer. Anal.* 42 (2) (2004) 800–825.
- [3] G. Beylkin, M.J. Mohlenkamp, Numerical operator calculus in higher dimensions, *Proceedings of the National Academy of Science* 99 (2002) 10246–10251.
- [4] G. Beylkin, M.J. Mohlenkamp, Algorithms for numerical analysis in high dimensions, *SIAM Journal on Scientific Computing* 26 (6) (2005) 2133–2159.
- [5] G. Beylkin, M.J. Mohlenkamp, Multivariate regression and machine learning with sums of separable functions, *SIAM J. Sci. Comput.* 31 (3) (2009) 1840–1857.
- [6] M.K. Deb, I. Babuska, J.T. Oden, Solution of stochastic partial differential equations using Galerkin finite element techniques, *Comput. Methods Appl. Mech. Eng.* 190 (2001) 6359–6372.
- [7] A. Doostan, R. Ghanem, J. Red-Horse, Stochastic model reduction for chaos representations, *Comput. Methods Appl. Mech. Eng.* 196 (37–40) (2007) 3951–3966.
- [8] W. Gautschi, The circle theorem and related theorems for Gauss-type quadrature rules, *ETNA Electron. Trans. Numer. Anal.* 25 (2006) 129–137.
- [9] R. Ghanem, A. Doostan, On the construction and analysis of stochastic models: characterization and propagation of the errors associated with limited data, *J. Comput. Phys.* (217) (2006) 63–81.
- [10] R. Ghanem, G. Saad, A. Doostan, Efficient solution of stochastic systems: application to the embankment dam problem, *Struct. Saf.* 29 (3) (2007) 238–251.
- [11] R. Ghanem, P. Spanos, *Stochastic Finite Elements: A Spectral Approach*, Dover, 2002.
- [12] G.H. Golub, C.F. Van Loan, *Matrix Computations*, second ed., Johns Hopkins Press, Baltimore, MD, 1989.
- [13] W. Hackbusch, B.N. Khoromskij, Kronecker Tensor-Product Approximation to Certain Matrix-Valued Functions in Higher Dimensions. Technical Report Preprint 16, Max-Planck-Institut für Mathematik in den Naturwissenschaften, 2004.
- [14] M. Kleiber, T.D. Hien, D.H. Tran, *The Stochastic Finite Element Method: Basic Perturbation Technique and Computer Implementation*, John Wiley & Sons, Chichester, 1992.
- [15] T.G. Kolda, Orthogonal tensor decompositions, *SIAM J. Matrix Anal. Appl.* 23 (1) (2001) 243–255. July.
- [16] Pieter Kroonenberg, Jan Leeuw, Principal component analysis of three-mode data by means of alternating least squares algorithms, *Psychometrika* 45 (1) (1980) 69–97. March.
- [17] O.P. Le Maitre, H. Najm, R. Ghanem, O. Knio, Multi-resolution analysis of Wiener-type uncertainty propagation schemes, *J. Comput. Phys.* 197 (2) (2004) 502–531.
- [18] L. Mathelin, M.Y. Hussaini. A Stochastic Collocation Algorithm for Uncertainty Analysis. Technical Report NAS 1.26:212153; NASA/CR-2003-212153, NASA Langley Research Center, 2003.
- [19] F. Nobile, R. Tempone, C.G. Webster, An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data, *SIAM J. Numer. Anal.* 46 (5) (2008) 2411–2442.
- [20] F. Nobile, R. Tempone, C.G. Webster, A sparse grid stochastic collocation method for partial differential equations with random input data, *SIAM J. Numer. Anal.* 46 (5) (2008) 2309–2345.
- [21] A. Nouy, Generalized spectral decomposition method for solving stochastic finite element equations: invariant subspace problem and dedicated algorithms, *Comput. Methods Appl. Mech. Eng.*, Available online 3 July, 2008.
- [22] A. Nouy, A generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations, *Comput. Methods Appl. Mech. Eng.* 196 (37–40) (2007) 4521–4537.
- [23] V. Pereyra, G. Scherer, Efficient computer manipulation of tensor products with applications to multidimensional approximation, *Math. Comput.* 27 (1973) 595–605.
- [24] M.A. Tatang, *Direct Incorporation of Uncertainty in Chemical and Environmental Engineering Systems*. Ph.D. Thesis, Massachusetts Institute of Technology, 1995.
- [25] E. Tyrtshnikov, Kronecker-product approximations for some function-related matrices, *Linear Algebra and its Applications* 379 (2004) 423–437.
- [26] X. Wan, G. Karniadakis, An adaptive multi-element generalized polynomial chaos method for stochastic differential equations, *J. Comput. Phys.* 209 (2005) 617–642.
- [27] D. Xiu, J.S. Hesthaven, High-order collocation methods for differential equations with random inputs, *SIAM J. Sci. Comput.* 27 (3) (2005) 1118–1139.
- [28] D. Xiu, G.M. Karniadakis, The Wiener–Askey polynomial chaos for stochastic differential equations, *SIAM J. Sci. Comput.* 24 (2) (2002) 619–644.