LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-644616

Module-based Hybrid Uncertainty Quantification for Multi-physics Applications: Theory and Software

C. Tong, X. Chen

October 8, 2013

Disclaimer

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Module-based Hybrid Uncertainty Quantification for Multi-physics Applications: Theory and Software LLNL-TR-644616

Charles Tong

Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
P.O. Box 808, L-557
Livermore, CA 94551-0808

tong10@llnl.gov

Co-investigators
Xiao Chen, Lawerence Livermore National Laboratory
Gianluca Iaccarino, Stanford University
Akshay Mittal, Stanford University

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, and was funded by the Department of Energy Office of Science Advanced Scientific Computing Research.

Contents

1 Summary of Accomplishments **2**

2 Introduction **3**

3 Uncertainty Quantification Methods **4**
3.1 Intrusive Methods . . . 4
3.2 Non-intrusive Methods . . . 5
3.3 Semi-intrusive Methods . . . 6

4 Mathematical Concepts for Hybrid and Modular UQ Methods **7**
4.1 Modular Polynomial-Chaos (PC)-based Uncertainty Representation . . . 8
4.2 Modular Operators and Uncertainty Propagation . . . 11
4.3 Non-intrusive Methods for Modular Uncertainty Propogation . . . 13
4.4 Intrusive Methods for Linear or Linearly Coupled Modules . . . 14
4.5 Intrusive Methods for Nonlinear or Non-linearly Coupled Modules . . . 19
4.6 Semi-intrusive Methods for Linear or Nonlinear Modules . . . 20
4.7 Stochastic Coupling . . . 20
4.8 Applicability to Modules with High Dimensions (With Correlated Inputs) . . . 21
4.9 Dimension Reduction in General Settings . . . 21
4.10 Bayesian Calibration . . . 21

5 A Computational Framework for Hybrid UQ Methods **22**
5.1 Basic Operators in Modules . . . 22
5.2 Hybrid UQ Software Infrastructure . . . 22

6 Numerical Study **24**
6.1 Two-dimensional Reactive Transport in Isotropic and Homogeneous Medium . . . 24
6.1.1 Stochastic Galerkin Method for Transport Module . . . 25
6.1.2 Sampling-based Methods for Reaction Module . . . 30
6.1.3 Propagation of Uncertainties through the Transport and Reaction Modules . . . 31
6.2 Two-dimensional Flow and Reactive Transport in Heterogeneous Porous Media . . . 36
6.2.1 Kahunen-Loeve Expansion (KLE) of Logarithmic Hydraulic Conductivity . . . 36
6.2.2 Non-intrusive PC-based Stochastic Solution of Flow Equation . . . 37
6.2.3 Generation of Stochastic Velocity Field . . . 38
6.2.4 PC-based Stochastic Transport . . . 42
6.2.5 Propagation of Uncertainties through the Transport and Reaction Modules . . . 42
6.3 Predator-prey Model . . . 47
6.4 Non-linearly Coupled Contaminant Flow . . . 48
6.5 Bayesian Inference of Flow and Reactive Transport in Heterogeneous Porous Media . . . 50

7 Summary and Future Work **52**

8 Appendix **54**
8.1 Element matrices in homogeneous media . . . 54
8.2 Stochastic Galerkin projection . . . 54
8.3 Element matrices in heterogeneous media . . . 56

1. Summary of Accomplishments

This project on uncertainty quantification (UQ) mathematics was funded by the Department of Energy Office of Science under the Advanced Scientific Computing Research Program from October 2010 to September 2013. In this project we proposed to develop an innovative uncertainty quantification methodology that captures the best of the two competing approaches in UQ, namely, intrusive and non-intrusive approaches. The idea is to develop the mathematics and the associated computational framework and algorithms to facilitate the use of intrusive or non-intrusive UQ methods in different modules of a multi-physics multi-module simulation model in a way that physics code developers for different modules are shielded (as much as possible) from the chores of accounting for the uncertainties introduced by the other modules.

As the result of our research and development, we have produced a number of publications, conference presentations, and a software product as listed below:

Journal Papers:

1. X. Chen, B. Ng, Y. Sun, and C. Tong, A Flexible Uncertainty Quantification Method for Linearly Coupled Multi-physics Systems, *J. Comput. Phys.*, http://doi:10.1016/j.jcp.2013.04.009, 2013.
2. X. Chen, B. Ng, Y. Sun, and C. Tong, A Computational Method for Simulating Subsurface Flow and Reactive Transport in Heterogeneous Porous Media Embedded with Flexible Uncertainty Quantification, *J. Water Resour. Res.*, http://doi: 10.1002/wrcr.20454, 2013.

Conference Presentations:

1. C. Tong, G. Iaccarino, and B. Lee, A High Performance Embedded Hybrid Methodology for Uncertainty Quantification with Applications, in *Proc. SciDAC*, Denver, CO, July 10–14, 2011.
2. X. Chen, B. Ng, Y. Sun, and C. Tong, Hybrid Uncertainty Quantification Techniques for Multi-species Reactive Transport Applications, in a session on *Uncertainty Assessment, Optimization, and Sensitivity Analysis in Integrated Hydrologic Modeling as Applications of Hydro-informatics, AGU Fall Meeting*, San Francisco, CA, December, 2011.
3. X. Chen, B. Ng, Y. Sun, and C. Tong, Modular Uncertainty Quantification based on Multi-physics Decomposition with Application in Reactive Transport, *SIAM Conference on Uncertainty Quantification, Raleigh*, NC, April 2, 2012.
4. G. Iaccarino, and B. Lee, C. Tong and X. Chen, A High Performance Embedded Hybrid Methodology for Uncertainty Quantification with Applications, in *Proc. Exascale Research Workshop*, Portland, OR, April 16, 2012.
5. X. Chen, B. Ng, Y. Sun, and C. Tong, Modular Uncertainty Quantification based on Multi-physics Decomposition with Application in Nonlinear Solute Reactive Transport Using Karhunen-Loeve Expansion, *Computational Methods in Water Resources Conference on Uncertainty Quantification*, Urbana-Champaign, IL, June 2012.
6. C. Tong, Some Thoughts on in-situ UQ and Exascale Computing, *Exascale Research Conference, Arlington*, VA, October 1, 2012.
7. X. Chen, Y. Sun, and C. Tong, Dimension Reduction and Bayesian Inference in Reactive Transport with Embedded Capability for Uncertainty Quantification, in a session on *Uncertainty Quantification and Parameter Estimation: Impact on Risk and Decision Making, AGU Fall Meeting*, San Francisco, CA, December, 2012.
8. X. Chen, J. Connors, and C. Tong, Modular Resolution of Uncertainty and Numerical Error Distributions for Operator-splitted Advection Diffusion Equation, *DOE Applied Mathematics Program Meeting*, Albuquerque, NM, August 2013.

Software Product:

1. MEDUSA - supports the propagation of uncertainty and global sensitivities through modules with different UQ methods (polynomial chaos, sampling, sampling with derivatives).

2. Introduction

The discipline of uncertainty quantification seeks to develop and applies rigorous methodologies and methods to quantify uncertainties associated with the modeling and simulation of physical processes, with the goals of estimating the variability of the output of interest from all relevant sources of uncertainty (uncertainty analysis) and quantifying the contribution of individual sources to the overall uncertainty (sensitivity analysis). Advances in mathematical/statistical techniques and the availability of high performance computers in recent years have provided an unprecedented opportunity to undertake the computationally intensive task of "model predictions with confidence."

UQ approaches are often categorized as either non-intrusive or intrusive. Non-intrusive methods generate model output probability distributions by first creating a sample drawn from the given uncertain parameter probability distributions, running deterministic simulations with the sample, and finally computing the statistical moments and/or sensitivities. There are many sampling designs such as Monte Carlo, quasi-Monte Carlo ([1], Latin Hypercube ([2], importance sampling [3, 4], etc. An advantage of non-intrusive methods is the simplicity of interfacing UQ software with deterministic simulation models. However, non-intrusive methods generally suffer from slow convergence rate of the statistical moments, although viable remedies using response surface techniques are available.

Intrusive methods, on the other hand, generally require re-formulation and re-implementation of determinstric models. A popular class of intrusive methods is the stochastic Galerkin method based on polynomial chaos expansion (PCE) [5, 6, 7]. PCE has been used successfully on many applications such as solid mechanics [5], transport in heterogeneous media [8], fluid mechanics [9, 10], combustion [11], etc. PCE-based methods are sometimes more computationally efficient than non-intrusive sampling-based methods [12, 13]. A review of spectral UQ methods in fluids has been discussed by Knio et al. [9] and Najm el al. [14].

Despite their successes, advances in intrusive methods, however, have not been keeping pace with the fast increases in the complexities of modern multi-physics models. A major reason is that implementation of intrusive methods requires major modifications to existing deterministic codes, a task that may be too cumbersome and time-consuming for most complex multi-physics multi-scale models. Moreover, the size of the coupled system arising from the discretization of the stochastic systems may become so large that renders large-scale and high-resolution model simulation intractable. Furthermore, there are still many challenges with intrusive methods for complex applications such as turbulent flow and highly nonlinear transient problems. To overcome some of the difficulties with intrusive PCE methods, sampling-based non-intrusive PCE methods have been proposed as viable alternatives (for example, [15]). However, non-intrusive PCE methods are largely similar to polynomial regression methods in existence for decades. Even there are generally some convergence benefits with non-intrusive PCE methods, the rigid requirement of sample point placement makes these methods unattractive in practice.

Recent developments in both intrusive and non-intrusive stochastic expansion methods (polynomial chaos [16], stochastic collocation [17, 18], low-rank approximations [19]) and response surface reconstruction (Kriging [20], radial basis functions [21], Pade-Legendre approaches [22]) have demonstrated impressive efficiency gains. These methods have demonstrated how the mathematical structure of the problems and the smoothness of the system responses can be exploited to achieve superlinear convergence [18]. The increasing attention towards development of new methodologies for uncertainty propagation begs the questions: (1) Is a monolithic approach to UQ the most effective one? (2) How can we accommodate new algorithmic enhancements in the solution of stochastic PDEs that are suited to a specific physical problem? (3) Can we tailor the UQ strategy to map optimally on evolving parallel architectures? (4) Is it possible to rigorously merge different methodologies into a UQ framework that offers flexibility and adaptivity? (5) What if we do not have the access to modify some commercial codes intrusively?

To overcome some of the difficulties with either intrusive or non-intrusive approaches, some hybrid approaches bridging the gaps between the two have been proposed recently, such as the multi-state procedure [23] and the mixed aleatory/epistemic uncertainties representations [24]. Another research effort on hybrid UQ methods is the domain hybridization method driven by the need to couple two different descriptions of turbulent flows [25]. The present article explores another alternative to propagate uncertainties using hybrid (or partially embedded) techniques. Instead of representing a stochastic multi-physics model purely in terms of polynomial chaos or through sampling the associated deterministic model, our hybrid UQ approach blends UQ methods best suited or only available for each individual physics module, and our hybrid UQ

framework seamlessly "glues" these modules together to facilitate uncertainty/sensitivity propagation for the full system.

In this report, we present mathematical concepts for hybrid UQ as well as a software infrastructure to capture the underlying numerical functions for propagating uncertainties and sensitivities across physics modules in a multi-physics environment. The outline of the paper is as follows: In Section 3 we provide a brief review of intrusive methods based on polynomial chaos, sampling-based UQ methods, and semi-intrusive methods such as methods that exploit additional information from simulation models such as derivatives. In Section 4, we give a detailed presentation of the underlying mathematics for propagation of uncertainties and sensitivities. In Section 5, we describe a computational framework for embedding the computational infrastructure of the underlying numerical algorithms for uncertainty propagation. In Section 6, we provide a few numerical examples to demonstrate how our hybrid UQ methodology works, followed by a short summary in Section 7.

3. Uncertainty Quantification Methods

In this section we give a brief review of the various UQ methods in use today. We begin with the intrusive polynomial chaos methods, followed by non-intrusive sampling methods, and ended with a derivative-based semi-intrusive method.

3.1. Intrusive Methods

One popular intrusive method is the stochastic Galerkin method using polynomial chaos expansion (PCE) [5, 6], which has been successfully applied to UQ analysis in many applications, including solid mechanics [5], transport in heterogeneous media [8], combustion [11], and fluid mechanics [9, 10]. The method represents the solution of stochastic equations using PCE. The stochastic equations are rewritten in terms of these solution expansions then projected onto an appropriate set of basis functions to derive a new, larger set of deterministic equations, from which the PCE coefficients are solved then used to reconstruct the output uncertainties.

Formally, polynomial chaos is a member of the set of homogeneous chaos, which was first defined by Wiener [26] as the span of Hermite polynomial functionals of a Gaussian random process. It was later pioneered by Ghanem and Spanos [5] for quantifying uncertainties in various applications. Subsequently, Xiu and Karniadakis [10] generalized Wiener's idea of chaos for various classes (based on Askey's classification [27]) of orthogonal polynomials that are coupled to their associated stochastic processes.

Consider a probability space (Θ, Σ, Pr), where Θ is the sample space, Σ is a σ-algebra on Θ (non-empty collection of subset of Ω that is closed under complementation and countable unions of its member), and Pr is a probability measure (mapping Σ to $[0, 1]$). Let's consider m-dimensional real-valued random variables $\boldsymbol{\xi} = \{\xi_1, \cdots, \xi_m\}$, with known distribution. Let Ξ denote the range of $\boldsymbol{\xi}$, and $(\Xi, \mathcal{B}_\Xi, Pr_\Xi)$ the associated probability space, and we have

$$X : \Theta \mapsto \Xi. \tag{1}$$

Let $L_2(\Theta, Pr)$ denote the set of second-order random variables such that for $\forall X \in L_2(\Theta, Pr)$, we have $E[|X|^2] < \infty$, where the mean E is defined as:

$$E[X] = \int_\Theta X(\boldsymbol{\xi}(\theta)) \, \mathrm{d}Pr(\theta). \tag{2}$$

The set of second-order random variables form a Hilbert space with respect to the inner product:

$$\langle X, Y \rangle = E[XY] = \int_\Theta X(\boldsymbol{\xi}(\theta)) Y(\boldsymbol{\xi}(\theta)) \, \mathrm{d}Pr(\theta) = \int_\Xi \int_\Xi X(\mathbf{x}) Y(\mathbf{y}) \, \mathrm{d}Pr_\Xi(\mathbf{x}, \mathbf{y}) \tag{3}$$

where $Pr_\Xi(\mathbf{x}, \mathbf{y})$ is the joint probability density function of $\mathbf{x}$ and $\mathbf{y}$. The associated norm of X is $\sqrt{E[|X|^2]}$.

An important concept in representing uncertainty using polynomial chaos is that one can express a second-order random variable X involving m random variables $\xi_1, \cdots, \xi_m$ as a sum of orthogonal polynomials:

$$X = X_0 + \sum_{j_1=1}^{m} X_{j_1} \Psi_1(\xi_{j_1}(\theta)) + \sum_{j_1=1}^{m} \sum_{j_2=1}^{j_1} X_{j_1, j_2} \Psi_2(\xi_{j_1}(\theta), \xi_{j_2}(\theta)) + \cdots \tag{4}$$

where $\Psi_k(\xi_{j_1}, \cdots, \xi_{j_k})$ are polynomial chaos of order k in the variables $(\xi_{j_1}, \cdots, \xi_{j_k})$ with its type and domain depending on the distributions imposed on the m-dimensional random variables $\boldsymbol{\xi} = \{\xi_1, \cdots, \xi_m\}$.

To simplify notation, this multi-dimensional expansion is usually mapped term-by-term to a single index form given by:

$$\hat{X} = \sum_{j=0}^{Q} X_j \Psi_j(\boldsymbol{\xi}) \tag{5}$$

where $\Psi_j(\boldsymbol{\xi})$'s are polynomial chaoses in single-index form and $(Q+1)$ is the total number of terms used for polynomial order $\leq p$ such that:

$$Q + 1 = \frac{(p+m)!}{p!m!}. \tag{6}$$

All the polynomial chaoses are mutually orthogonal with regards to the inner product associated to the space spanned by the random variables $\boldsymbol{\xi}$. In particular, when uniform distribution is assumed for the m random variables, the associated orthogonal polynomials are the Legendre polynomials which have the following property:

$$\int_{-1}^{1} L_j(x) L_i(x) \, \mathrm{d}x = \frac{2}{2j+1} \delta_{jl} \tag{7}$$

where δ_{jl} is the Kronecker delta. The Legendre polynomials satisfy the recurrence relation:

$$(l+1) L_{l+1}(x) - (2l+1) x L_l(x) + l L_{l-1}(x) = 0 \tag{8}$$

with $L_0(x) = 1$ and $L_1(x) = x$. Using PCE, it is straightforward to verify that the approximate mean of X (using $(Q+1)$ terms) is X_0 and the approximate variance is:

$$\int_{-1}^{1} \left[\sum_{j=0}^{Q} X_j \Psi_j(\boldsymbol{\xi}) - x_0 \Psi_0(\boldsymbol{\xi}) \right]^2 \mathrm{d}\boldsymbol{\xi} = \sum_{j=1}^{Q} \left[X_j^2 \left\langle \Psi_j^2 \right\rangle \right]. \tag{9}$$

Furthermore, let $\hat{\boldsymbol{\xi}}$ be a subset of $\boldsymbol{\xi}$ and $\hat{\Psi}$ be the set of all Legendre polynomials of order up to p involving only the random variables in $\hat{\xi}$. Then the partial variance (Sobol' index) for this subset can be calculated by:

$$S\left(\hat{\boldsymbol{\xi}}\right) \approx \frac{\sum_{\Psi \in \hat{\Psi}} \left[X_j^2 \left\langle \Psi_j^2 \right\rangle \right]}{\sum_{j=1}^{Q} \left[X_j^2 \left\langle \Psi_j^2 \right\rangle \right]}. \tag{10}$$

To model uncertainty in a given partial differential equation system with parametric uncertainties, the independent and the dependent random variables are both represented by the appropriate PCE. A Galerkin projection is then performed onto each of the $(Q+1)$ orthogonal polynomials giving rise to a coupled system of $(Q+1)$ equations, which are discretized and solved for the coefficients X_j's. Finally, the coefficients are substituted back into the expansion.

Intrusive methods generally, when the input-output relationship is sufficiently smooth, has the potential to be more computationally efficient than non-intrusive methods [12, 13]. However, this is at the expense of (1) higher development effort, since implementation require major modifications to existing deterministic codes, and (2) additional overhead in solving the larger set of equations, which might be structurally different enough to warrant new solvers. These two challenges might be too cumbersome and time-consuming to overcome for most complex multi-physics multi-scale models. As a result, intrusive methods have not been as widely adopted as non-intrusive methods because their advances have not been keeping pace with the fast increases in the complexities of modern multi-physics models.

3.2. Non-intrusive Methods

Non-intrusive methods consist of generating samples of random variables based on their distributions via a fixed sampling scheme. Different sampling schemes generate samples with different "space filling" properties. The samples are then propagated through the model by running the model repeatedly with different sample inputs. The outputs of interest are collected and the desired statistics such as mean and

standard deviation are computed. Again, let m be the number of random variables and let $P(\xi_1, \xi_2, ..., \xi_m)$ be the corresponding joint probability distribution function. Let $S \in \mathcal{R}^{N \times m}$ be the set of N samples drawn from this distribution where each row of S is a single sample of the input random variables. Let $Y \in \mathcal{R}^N$ be the (univariate) outputs corresponding to the N "input" samples. Then, the sample approximations of the mean and standard deviation are:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} Y_i \tag{11}$$

$$\sigma = \frac{1}{N-1} \sqrt{\sum_{i=1}^{N} (Y_i - \mu)^2}. \tag{12}$$

While straightforward to implement, the non-intrusive approach suffers from poor computational efficiency: the convergence rate of the computed mean is only $O(1/\sqrt{N})$ for the Monte Carlo sampling method. Thus, to increase the accuracy of the computed mean by one decimal would require increasing N by 100-fold. Despite the fact that other sampling strategies (e.g., Latin hypercube [2]), Quasi-Monte Carlo [1], importance sampling [3, 4]) have been proposed to improve convergence, existing sampling methods are still inadequate to handle large-scale and computationally expensive models.

To mitigate this problem of slow convergence, one might exploit the fact that, in many real-life models, the outputs of interest are smooth function of the input random variables. This motivates the use of response surface models (also known as surrogate models, meta-models,and emulators) that interpolate the values of the output variables. Interpolation is generally computationally much cheaper than the actual simulation runs, thus allowing for quicker and larger-scale generation of samples. The choice of interpolation scheme is heavily dependent on the specific application problem.

It should be noted that the non-intrusive form of the polynomial chaos method (sometimes called spectral collocation) is analogous to a special type of polynomial regression scheme. An advantage of this scheme is that if the output of interest is a polynomial function of the input random variables, the convergence rate of computing the basic statistics is exponential with respect to the sample size, and the basic statistical measures are readily available from the polynomial chaos coefficients associated with the output variables.

Once a good response surface model has been constructed, Sobol' indices can be computed in several ways such as Sobol' method [28], McKay's method [29], or direct numerical integration. For example, McKay's method makes use of the replicated Latin hypercube (r-LHS) sampling strategy. In the r-LHS design, each random variable $\xi_i, i = 1, ..., m$, takes on distinct values $x_{ij}, j = 1, ..., L$, where L is called the number of levels. These values are to be randomized and re-used R times to form the final design which has $N = LR$ sample points.

The estimator for the Sobol' index corresponding to ξ_i is given by:

$$S(\xi_i) = \frac{1}{V(Y)} \left\{ \frac{1}{L} \sum_{j=1}^{L} \left[\bar{Y}(\xi_i = x_{ij}) - \bar{Y} \right]^2 - \frac{1}{LR^2} \sum_{j=1}^{L} \sum_{r=1}^{R} \left[Y^{(r)}(\xi_i = x_{ij}) - \bar{Y}(\xi_i = x_{ij}) \right]^2 \right\} \tag{13}$$

where $\bar{Y}(\xi_i = x_{ij}) = \frac{1}{R} \sum_{r=1}^{R} Y^{(r)}(\xi_i = x_{ij})$ and $Y^{(r)}(\xi_i = x_{ij})$ is the output of the sample point corresponding to $\xi_i = x_{ij}$ in the r^{th} replication; $\bar{Y}$ and $V(Y)$ are the aggregate mean and variance of Y, respectively.

3.3. Semi-intrusive Methods

Semi-intrusive methods uses additional information (beyond typical model outputs of interest) to aid in the UQ task. For example, many simulation models are equipped with derivatives (deriative of outputs interest with respect to some uncertain parameters). These derivative information can greatly speed up UQ by reducing the sample size needed for, for example, building response surface models. In the following we show how derivative information is used in using sampling and polynomial regression to build response surface. Again, let $\{S_i, Y_i\}_1^N$ be a sample of size N where $S_i \in \Re^m$ is a sample point with m random variables and $Y_i \in \Re$ is the corresponding model output. In addition, let $(Y_i)_j$ be the derivative of Y with respect to the j-th random variable.

For example, a quadratic regression analysis assumes the following form:

$$Y = b_0 + \sum_{k=1}^{m} b_k \xi_k + \sum_{k=1}^{m} \sum_{j \leq k}^{m} b_{kj} \xi_k \xi_j + \epsilon \tag{14}$$

where b_k's and b_{kj}'s are the regression coefficients to be determined, and ϵ_i is the regression error. Subsituting the sample to this equation gives rise to N equations which can be expressed in matrix form as:

$$\boldsymbol{Y} = \boldsymbol{S}\boldsymbol{b} + \boldsymbol{\epsilon} \tag{15}$$

where $\boldsymbol{S}$ is a $N \times ((m+2)(m+1)/2)$ matrix where $\boldsymbol{b}$ consists of all b_k's and b_{kj}'s. Minimizing the 2-norm of the discrepancies $\boldsymbol{\epsilon}$ is equivalent to applying the least-squares approach so that

$$\boldsymbol{S}^T \boldsymbol{S}\boldsymbol{b} = \boldsymbol{S}^T \boldsymbol{Y} \tag{16}$$

giving

$$\boldsymbol{b} = (\boldsymbol{S}^T \boldsymbol{S})^{-1} \boldsymbol{S}^T \boldsymbol{Y}. \tag{17}$$

Here, unique solution ($\boldsymbol{b}$) is possible only when $\boldsymbol{S}^T \boldsymbol{S}$ is nonsingular, which is satisfied if the number of rows in the matrix is larger or equal to the number of columns, and the sample is well-distributed (no dependency between S_i and S_j for $i \neq j$).

When derivatives are also available at each sample points, we can use the same least-squares procedure. In this case, the derivatives of the quadratic equation with respect to each random variable is:

$$\frac{\partial Y}{\partial \xi_l} = b_l + \sum_{k=1}^{l-1} b_{lk} \xi_k + \sum_{k=l+1}^{m} b_{kl} \xi_k + 2 b_{ll} \xi_l. \tag{18}$$

Thus, since each sample point has $m+1$ output values (model output and m derivatives) and thus $m+1$ rows in the regression matrix, ideally the sample size can be reduced by a factor of $m+1$ and the uniqueness requirement can still be satisfied. In practice, the number of rows should be at least twice the number of columns for numerical stability. Once the regression coefficients have been computed, the polynomial function can be used to perform uncertainty and sensitivity analysis.

4. Mathematical Concepts for Hybrid and Modular UQ Methods

Hybrid methods seek to bridge the gap between the practicality of non-intrusive methods and the potential efficiency and robustness of intrusive methods. The main idea is to integrate both methods into a UQ framework of a complex multi-physics model.

The concept of hybridization for UQ has been realized in many forms ([30, 31, 32]). For example, Hosder et al. [15] presented a non-intrusive polynomial chaos method, which uses samples to construct the equations for solving the PCE coefficients. Surana and Banaszuk [33] proposed probabilistic waveform relaxation, applicable for networked systems with weakly coupled subsystems, which integrates generalized polynomial chaos and probabilistic collocation into waveform relaxation. Abgrall et al. [34] proposed a semi-intrusive method that solves the deterministic problem over an extended domain that encapsulates uncertainty, which requires less modification to the deterministic solver and can also handle more general types of input distributions. Constantine et al. [32, 35] combined the intrusive Galerkin method with non-intrusive stochastic allocation and demonstrated their method on a one-way decoupled system. Guadagnini et al. [36] employed random domain decomposition to divide physical domain for composite material according to a large-scale distribution of geological units and then quantify the local-level uncertainty within each disjoint, statistically homogeneous unit. Our work is most similar to [32], but allows uncertainty introduced in one module to influence the stochastic dependent variables in the second module, making it applicable to more general multi-physics models.

Most of the aforementioned methods are only applicable for specialized problems that satisfy the methods' assumptions of decoupled or weakly-coupled physics modules. In contrast, the hybrid UQ methodology we aspire to develp is more general. Our notion of "hybrid" refers to techniques that facilitate the use of different

intrusive and non-intrusive methods for strongly-coupled modules in a multi-physics simulation. This is a largely unexplored concept with many possible research directions. For example, one can use an intrusive method in one single-physics module and a non-intrusive method in another module, and then piece their uncertainties together (as done in this paper). Alternatively, a single-physics module can be made hybrid with some of its parameters handled intrusively and others handled non-intrusively to handle nonlinear dependencies.

The hybrid concept is ideally compatible with the current practice of multi-physics code development, namely, the use of "plug-and-play" to facilitate the natural division of physics expertise and to ensure that future model updates can be performed with minimal effort. Other potential benefits of hybrid UQ methods are: (1) higher computational efficiency; (2) higher degree of flexibility of selecting UQ schemes for each physics module; and (3) ability to naturally track evolution of uncertainties (and sensitivity) in detail (at mesh point level).

To successfully develop this methodology, several challenges must be addressed: (1) how to accurately propagate uncertainties between individual physics modules with minimal loss of uncertainty information; (2) how to propagate uncertainties when the overall number of parameters in the system is large; (3) how to perform parameter estimation and calibration in view of observational data; and (4) how to design a flexible software framework for supporting hybrid UQ for general multi-physics applications. We will address some of these issues in this report.

In this section, we introduce the general theory behind our *modularly hybrid* UQ framework for global uncertainty and sensitivity propagation. This UQ framework is different from other types of hybrid frameworks: while other hybrid frameworks combine properties of intrusive and non-intrusive methods into one monolithic algorithm for uncertainty propagation, our framework is hybrid in the modular sense, in that different algorithms (pure or combined versions of intrusive or non-intrusive) can be applied to different modules in the multi-physics system. Our objective is to develop the mathematics and algorithms so that each module is self-contained in its own uncertainty propagation yet together global uncertainty is propagated.

In the rest of this section, we first introduce a mathematical formalism for modular decomposition. We prove a few lemmas on the mathematics of transformations between the local (to each module) and the global stochastic space. Subsequently, we will provide details on the transformations for different types of modules (linearly- or nonlinearly-coupled in the stochastic space for intrusive modules, non-intrusive and semi-intrusive modules). The goal is to develop a computational framework to facilitate inter-module transformations in a manner that is transparent to physics module developers.

4.1. Modular Polynomial-Chaos (PC)-based Uncertainty Representation

Without loss of generality, for a component-wise formulation, we define $\Theta_1 \subseteq \mathbb{R}^{m_1}$ and $\Theta_2 \subseteq \mathbb{R}^{m_2}$ such that $\Theta_1 \cap \Theta_2 \equiv \emptyset$ and $\Theta_1 \times \Theta_2 \equiv \Theta$. We also assume that we have $Pr_1 : \Theta_1 \rightarrow [0,1]$ and $Pr_2 : \Theta_2 \rightarrow [0,1]$ such that $Pr\left(\theta_1, \theta_2\right) = Pr_1\left(\theta_1\right) Pr_2\left(\theta_2\right) \forall \theta_1 \in \Theta_1$ and $\theta_2 \in \Theta_2$. In this setting, we assume that each module has a local set of parameters that are different from and are uncorrelated to the set of external parameters.

We define a conditional polynomial basis in Θ_1 or Θ_2 using $\Psi_{\Theta_1,p}$ and $\Psi_{\Theta_2,p}$ with cardinality Q_1+1 and Q_2+1 respectively. i.e.

$$Q_1 + 1 = \frac{(p+m_1)!}{p!m_1!};\ Q_2 + 1 = \frac{(p+m_2)!}{p!m_2!}. \tag{19}$$

Next, we represent the coefficients corresponding to the two conditional polynomial bases in matrix form as:

$$\left[\hat{X}\right]_{\Theta_1,p} = \left[\ \hat{X}_{\Theta_1,0} \quad \cdots \quad \hat{X}_{\Theta_1,Q_1}\ \right] \tag{20}$$

and

$$\left[\hat{X}\right]_{\Theta_2,p} = \left[\ \hat{X}_{\Theta_2,0} \quad \cdots \quad \hat{X}_{\Theta_2,Q_2}\ \right] \tag{21}$$

such that

$$X(\theta_1,\theta_2) = \left[\hat{X}(\theta_2)\right]_{\Theta_1,p}[\Psi(\theta_1)]_{\Theta_1,p}, \tag{22}$$

$$= \left[\hat{X}(\theta_1)\right]_{\Theta_2,p}[\Psi(\theta_2)]_{\Theta_2,p}; \text{ and} \tag{23}$$

$$\left[\hat{X}(\theta_2)\right]_{\Theta_1,p} = \int_{\Theta_1} X(\theta_1,\theta_2)[\Psi(\theta_1)]^T_{\Theta_1,p}\, dPr_1(\theta_1), \tag{24}$$

$$\left[\hat{X}(\theta_1)\right]_{\Theta_2,p} = \int_{\Theta_2} X(\theta_1,\theta_2)[\Psi(\theta_2)]^T_{\Theta_2,p}\, dPr_2(\theta_2). \tag{25}$$

Based on the above definitions, now we can define conditional mean and variance as:

$$\mu_p(X;\theta_2) = \langle X\rangle_{\Theta_1} = \hat{X}_{\Theta_1,0}(\theta_2); \tag{26}$$

$$\Sigma_p(X;\theta_2) = \left\langle XX^T\right\rangle_{\Theta_1} - \langle X\rangle_{\Theta_1}\langle X\rangle^T_{\Theta_1} = \sum_{j=1}^{Q_1}\hat{X}_{\Theta_1,j}(\theta_2)\hat{X}^T_{\Theta_1,j}(\theta_2); \tag{27}$$

$$\mu_p(X;\theta_1) = \langle X\rangle_{\Theta_2} = \hat{X}_{\Theta_2,0}(\theta_1); \tag{28}$$

$$\Sigma_p(X;\theta_1) = \left\langle XX^T\right\rangle_{\Theta_2} - \langle X\rangle_{\Theta_2}\langle X\rangle^T_{\Theta_2} = \sum_{j=1}^{Q_2}\hat{X}_{\Theta_2,j}(\theta_1)\hat{X}^T_{\Theta_2,j}(\theta_1). \tag{29}$$

We will now prove two lemmas that are necessary in deriving the global mean and variance estimates using either of the conditional polynomial chaos expansions. The first lemma defines the restriction and prolongation transformations between the global and modular PCE.

Lemma 1: Let X be a second order random variable defined in the conditional probability space above, then the respective restriction and prolongation operations can be represented by:

$$\left[\hat{X}(\theta_2)\right]_{\Theta_1,p} = \left[\hat{X}\right]_{\Theta,p}[\Pi_{1,2}(\theta_2)]_p \quad \text{restriction and} \tag{30}$$

$$\left[\hat{X}\right]_{\Theta,p} = \left\langle\left[\hat{X}\right]_{\Theta_1,p}[\Pi_{1,2}]^T_p\right\rangle_{\Theta_2} \quad \text{prolongation,} \tag{31}$$

respectively, where

$$[\Pi_{1,2}(\theta_2)]_p = \begin{bmatrix} \pi_0(\theta_2) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \pi_{P_1}(\theta_2) \end{bmatrix} \tag{32}$$

such that

$$\forall 0 \le j \le Q_1,\ \pi_j(\theta_2) \equiv [\Psi(\theta_2)]^T_{\Theta_2,q(j)} : q(j) = \max_{0\le k\le p}\left\{p-k : \frac{(k+m_1)!}{k!m_1!} \le j+1 < \frac{(k+1+m_1)!}{(k+1)!m_1!}\right\} \tag{33}$$

is a transformation matrix with Q_1 columns and at-most one non-zero element per row.

Proof: By definition of X, we have

$$\int_{\Theta_1}\left[\hat{X}(\theta_2)\right]_{\Theta_1,p}[\Psi(\theta_1)]_{\Theta_1,p}[\Psi(\theta_1)]^T_{\Theta_1,p}\, dPr_1(\theta_1) = \int_{\Theta_1}\left[\hat{X}\right]_{\Theta,p}[\Psi(\theta;\theta_2)]_{\Theta,p}[\Psi(\theta_1)]^T_{\Theta_1,p}\, dPr_1(\theta_1);$$

$$\Rightarrow \left[\hat{X}(\theta_2)\right]_{\Theta_1,p}\left(\int_{\Theta_1}[\Psi(\theta_1)]_{\Theta_1,p}[\Psi(\theta_1)]^T_{\Theta_1,p}\, dPr_1(\theta_1)\right) = \left[\hat{X}\right]_{\Theta,p}\left(\int_{\Theta_1}[\Psi(\theta;\theta_2)]_{\Theta,p}[\Psi(\theta_1)]^T_{\Theta_1,p}\, dPr_1(\theta_1)\right);$$

$$\Rightarrow \left[\hat{X}(\theta_2)\right]_{\Theta_1,p} = \left[\hat{X}\right]_{\Theta,p}\left(\int_{\Theta_1}[\Psi(\theta;\theta_2)]_{\Theta,p}[\Psi(\theta_1)]^T_{\Theta_1,p}\, dPr_1(\theta_1)\right). \tag{34}$$

We can verify that

$$\int_{\Theta_1} [\Psi(\theta;\theta_2)]_{\Theta,p} [\Psi(\theta_1)]^T_{\Theta_1,p} \, dPr_1(\theta_1) = \begin{bmatrix} \pi_0(\theta_2) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \pi_{P_1}(\theta_2) \end{bmatrix} = [\Pi_{1,2}(\theta_2)]_p; \tag{35}$$

As such, we obtain that:

$$\begin{aligned} \left\langle \left[\hat{X}\right]_{\Theta_1,p} [\Pi_{1,2}]^T_p \right\rangle_{\Theta_2} &= \int_{\Theta_2} \left[\hat{X}(\theta_2)\right]_{\Theta_1,p} [\Pi_{1,2}(\theta_2)]^T_p \, dPr_2(\theta_2) \\ &= \int_{\Theta_2} \left[\hat{X}\right]_{\Theta,p} [\Pi_{1,2}(\theta_2)]_p [\Pi_{1,2}(\theta_2)]^T_p \, dPr_2(\theta_2) \\ &= \left[\hat{X}\right]_{\Theta,p} \left(\int_{\Theta_2} [\Pi_{1,2}(\theta_2)]_p [\Pi_{1,2}(\theta_2)]^T_p \, dPr_2(\theta_2) \right) \\ &= \left[\hat{X}\right]_{\Theta,p}. \end{aligned} \tag{36}$$

$[\Pi_{2,1}]_p$ and $\left[\hat{X}_0\right]_{\Theta_2,p}$ can be defined in the similar manner. This enables us to compute the global mean and variance directly using the conditional PCE coefficients as follows.

$$\mu_p(X) = \left\langle \hat{X}_{\Theta_1,0} \right\rangle_{\Theta_2} = \left\langle \hat{X}_{\Theta_2,0} \right\rangle_{\Theta_1}; \tag{37}$$

$$\begin{aligned} \Sigma_p(X) = &\left\langle \sum_{j=1}^{P_1} \hat{X}_{\Theta_1,j} \hat{X}^T_{\Theta_1,j} \right\rangle_{\Theta_2} + \left\langle \hat{X}_{\Theta_1,0} \hat{X}^T_{\Theta_1,0} \right\rangle_{\Theta_2} - \\ &\left\langle \hat{X}_{\Theta_1,0} \right\rangle_{\Theta_2} \left\langle \hat{X}^T_{\Theta_1,0} \right\rangle_{\Theta_2} \left\langle \sum_{j=1}^{P_1} \hat{X}_{\Theta_2,j} \hat{X}^T_{\Theta_2,j} \right\rangle_{\Theta_1} + \left\langle \hat{X}_{\Theta_2,0} \hat{X}^T_{\Theta_2,0} \right\rangle_{\Theta_1} - \left\langle \hat{X}_{\Theta_2,0} \right\rangle_{\Theta_1} \left\langle \hat{X}^T_{\Theta_2,0} \right\rangle_{\Theta_1}. \end{aligned} \tag{38}$$

Lemma 2: Let X be a second order random variable defined in the conditional probability space above, then

$$\left\langle \hat{X}_{\Theta_1,0} \right\rangle_{\Theta_2} = \left\langle \hat{X}_{\Theta_2,0} \right\rangle_{\Theta_1} = \hat{X}_0; \tag{39}$$

$$\left\langle \sum_{j=1}^{P_1} \hat{X}_{\Theta_1,j} \hat{X}^T_{\Theta_1,j} \right\rangle_{\Theta_2} = \sum_{j=P_2+1}^{P} \hat{X}_j \hat{X}^T_j. \tag{40}$$

Proof: Let us define $\hat{e}_0 = \begin{bmatrix} 1 & 0 & \cdots \end{bmatrix}^T$. We have

$$\begin{aligned} \left\langle \hat{X}_{\Theta_1,0} \right\rangle_{\Theta_2} &= \int_{\Theta_2} \hat{X}_{\Theta_1,0}(\theta_2) \, dPr_2(\theta_2) \\ &= \int_{\Theta_2} \left(\left[\hat{X}\right]_{\Theta,p} [\Pi_{1,2}(\theta_2)]_p \hat{e}_0 \right) dPr_2(\theta_2) \\ &= \left[\hat{X}\right]_{\Theta,p} \int_{\Theta_2} \left([\Pi_{1,2}(\theta_2)]_p \hat{e}_0 \right) dPr_2(\theta_2) \\ &= \begin{bmatrix} \hat{X}_0 & \cdots & \hat{X}_P \end{bmatrix} \int_{\Theta_2} \left(\begin{bmatrix} \pi_0(\theta_2) \\ 0 \\ \vdots \end{bmatrix} \right) dPr_2(\theta_2) \\ &= \hat{X}_0. \end{aligned} \tag{41}$$

We can similarly show that $\left\langle \hat{X}_{\Theta_2,0} \right\rangle_{\Theta_1} = \hat{X}_0$, thereby we consider the following:

$$\begin{aligned}
\left\langle \sum_{j=1}^{Q_1} \hat{X}_{\Theta_1,j} \hat{X}_{\Theta_1,j}^T \right\rangle_{\Theta_2} &= \int_{\Theta_2} \left(\sum_{j=1}^{Q^*} \hat{X}_{\Theta_1,j}(\theta_2) \hat{X}_{\Theta_1,j}^T(\theta_2) \right) dPr_2(\theta_2) \\
&= \int_{\Theta_2} \left[\hat{X}\right]_{\Theta,p} \left[\Pi_{1,2}(\theta_2)\right]_p \left(I - \hat{e}_0 \hat{e}_0^T\right) \left[\Pi_{1,2}(\theta_2)\right]_p^T \left[\hat{X}\right]_{\Theta,p}^T dPr_2(\theta_2) \\
&= \left[\hat{X}\right]_{\Theta,p} \left(\int_{\Theta_2} \left[\Pi_{1,2}(\theta_2)\right]_p \left(I - \hat{e}_0 \hat{e}_0^T\right) \left[\Pi_{1,2}(\theta_2)\right]_p^T dPr_2(\theta_2) \right) \left[\hat{X}\right]_{\Theta,p}^T \\
&= \left[\hat{X}\right]_{\Theta,p} \left(\int_{\Theta_2} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \pi_1(\theta_2) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \pi_{P_1}(\theta_2) \end{bmatrix} dPr_2(\theta_2) \right) \left[\hat{X}\right]_{\Theta,p}^T \\
&= \sum_{j=0}^{Q} \hat{X}_j \hat{X}_j^T - \sum_{j=0}^{Q'} \hat{X}_j \hat{X}_j^T : \ 0 < Q' < Q.
\end{aligned}$$

Using the definition of $q(0)$, we obtain

$$Q' + 1 = \frac{(p + m_2)!}{p! m_2!}. \tag{42}$$

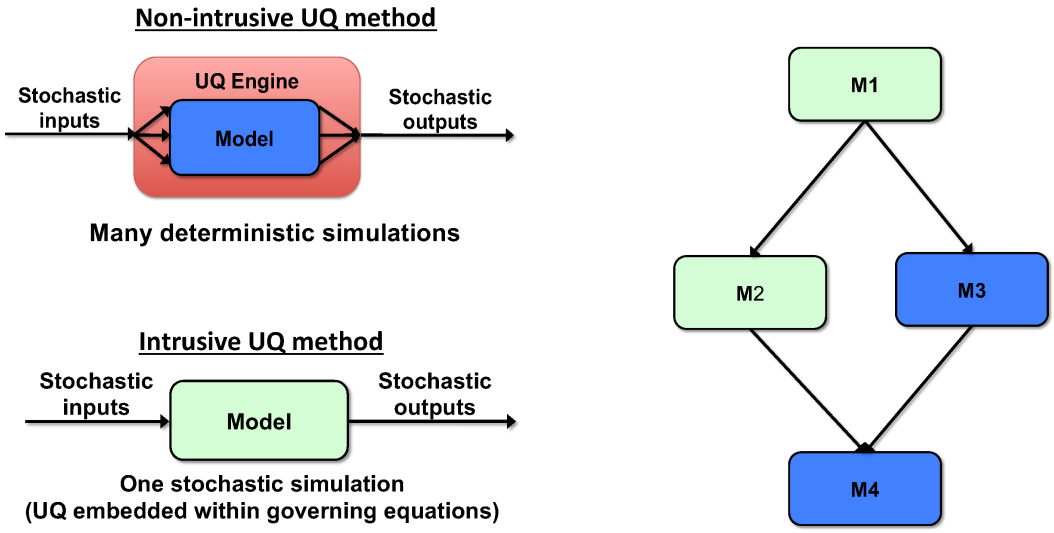


Figure 1: Hybrid (modular) UQ conceptual model

4.2. Modular Operators and Uncertainty Propagation

We continue by explaining the operators that handle information between the modules. For illustrative purposes, we will explain these operators within the context of forward uncertainty propagation for linearly coupled modules.

Let K be the number of independent physics modules in the multi-physics system. Each module has embedded UQ, meaning each module has a "stochastic" (intrusive or non-intrusive) solver that propagates

uncertainty from inputs to outputs at each time step. Let $\boldsymbol{\xi} = \{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \cdots, \boldsymbol{\xi}_K\}$ be K disjoint subsets of independent second-order random variables, where $\boldsymbol{\xi}_k$ denotes module k's internal random variables. Also, let $u(\boldsymbol{x}, \boldsymbol{\xi})$ be the unknown global quantities of interest for the full multi-physics system. At each time step, each module k contributes local information to update the global quantities of interest as follows:

$$\hat{C}(\boldsymbol{\xi}, \boldsymbol{x}, t) = M_k(\boldsymbol{\xi}_k, C(\boldsymbol{\xi}, \boldsymbol{x}, t), u(\boldsymbol{\xi})) \tag{43}$$

where C is the solution field updated by the function M_k to become $\hat{C}$; u is the auxiliary stochastic field (can be a temporal-spatial field) to be used in the local solver; and $\boldsymbol{x}$ and t are the spatial and temporal dimensions, respectively. Thus, in a multi-component environment, a number of these modules may be connected and represented by a directed graph to describe the order these modules are to be executed within a time step. Note that both C and $\hat{C}$ is a function of the global set of random variables $\boldsymbol{\xi}$, not just the local ones in each module. This is critical, since the goal is to propagate global uncertainties.

To make our discussion concrete, we explain our framework within the context of two linearly coupled modules. Using operator-splitting, we assume that the full multi-physics model has been decomposed into two modules that work in concert to update the state from time t to $t+1$:

$$\begin{aligned} \hat{C}^{t+1/2}(\boldsymbol{\xi}, \boldsymbol{x}, t) &= M_1(\boldsymbol{\xi}_1, C^t(\boldsymbol{\xi}, \boldsymbol{x}, t), u(\boldsymbol{\xi})) \\ \hat{C}^{t+1}(\boldsymbol{\xi}, \boldsymbol{x}, t) &= M_2(\boldsymbol{\xi}_2, C^{t+1/2}(\boldsymbol{\xi}, \boldsymbol{x}, t), u(\boldsymbol{\xi}) \end{aligned} \tag{44}$$

where M_1 and M_2 are the respective solvers for modules 1 and 2, and $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ are their respective internal random variables such that $\boldsymbol{\xi} = \{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2\}$. Eq. (44) reflects the status quo where the UQ treatment at each module has to deal with the entire uncertainty profile corresponding to all random variables in the system. Since $u^{t+1/2}(\boldsymbol{x}, \boldsymbol{\xi})$, the partial solution from M_1, is fed directly into M_2, changes introduced in M_1 will need to be reflected in M_2. This is because both M_1 and M_2 require knowledge of the global stochastic space $\boldsymbol{\xi}$. For example, if one adds a new variable to M_1, this new variable will be part of $\boldsymbol{\xi}$, and thus M_2 will need to be modified to handle the new $\boldsymbol{\xi}$ that has been augmented with the new variable.

We now introduce a new framework, the *modulary hybrid framework*, where changes in M_1 will *not* impact the M_2's implementation (and vice versa). The key is in making each module "self-inclusive" via the use of inter-module operators that translates each module's local information into a common format that can be understood by all modules. Let's assume that M_1 has a PCE-based solver and M_2 has a sampling-based solver. Furthermore, let's consider a PCE representation as the common format for the solution, so each module's partial solution will be expressed as:

$$u^*(\boldsymbol{x}, \boldsymbol{\xi}) = \sum_{i=0}^{Q} u_i(\boldsymbol{x}) \boldsymbol{\Psi}_i(\boldsymbol{\xi}) \tag{45}$$

where $(Q+1)$ is the number of PCE coefficients as defined in Eq. (6).

Our algorithmic goal is to develop techniques that require modules with intrusive UQ schemes to propagate uncertainties with respect to only the local random variables ($\boldsymbol{\xi}_k$) and the hybrid UQ framework will handle the transfer between local and global stochastic space. If only deterministic solvers are available in a module, our hybrid UQ framework will provide non-intrusive UQ capabilities to propagate uncertainties so that module developers are relieved of the chores to incorporate UQ capabilities in their modules. This concept is in contrast to Eq. (43) which implies that global uncertainties ($\boldsymbol{\xi}$) must be propagated even in module k. Towards this end, we introduce the *restriction* and *prolongation* operators. The restriction operator maps information from the global uncertain parameter space $\boldsymbol{\xi}$ to the module's local parameter space $\boldsymbol{\xi}_k$. The prolongation operator does the inverse mapping from the module's local parameter space $\boldsymbol{\xi}_k$ back to the global uncertain parameter space $\boldsymbol{\xi}$. As such, these operators are specific to the UQ method embedded in a given module. For linearly coupled modules with intrusive polynomial chaos method, we can, without any loss of information, decompose each module's updates into independent subproblems that can be solved in parallel. In other words, we can write:

$$u^*(\boldsymbol{x}, \boldsymbol{\xi}) = \sum_{i=1}^{n_k} P_{ki} \hat{M}_{ki}(\boldsymbol{\xi}_k, R_{ki} u(\boldsymbol{x}, \boldsymbol{\xi})) \tag{46}$$

where R_{ki} and P_{ki} are restriction and prolongation operators for module k's subproblem i, and n_k is the number of subproblems. We use the solver $\hat{M}_{ki}$ to ensure only local uncertainties are propagated within module k.

For PCE-based modules, the implementation of R_{ki} and P_{ki} depends on whether the modules are coupled linearly or nonlinearly. Recall that p is the PCE polynomial order, m is the number of random variables in $\boldsymbol{\xi}$, and m_k is the number of internal random variables in $\boldsymbol{\xi}_k$. For the linear case, the number of independent subproblems will be $n_k = (m - m_k + p)!/((m - m_k)!p!)$ (proof given in the next subsection). In each of these subproblems, R_{ki} and P_{ki} are analogous to the "scatter" and "gather" operations, respectively. For the nonlinear case, the construction of R_{ki} and P_{ki} are algorithmically more complex. A feasible approach would be to use sampling in conjunction with PCE, in particular, apply sampling to the external variables so that the PCE representations of the nonlinear coefficients for each sample point are expressed only in terms of the internal variables to the module at hand. The nonlinear case will be further outlined in 4.5.

For sampling-based modules, the treatment is the same for both linear and nonlinear cases. In this setting, $\hat{M}_{ki}$ is simply a deterministic solver for a fixed sample $\boldsymbol{\xi}$ and n_k is the sample size. R_{ki} maps $u(\boldsymbol{x}, \boldsymbol{\xi})$ onto a single point (can think of it as a delta function) in the random parameter space, and P_{ki} transforms the results back to polynomial coefficients in the PC space in a fashion analogous to polynomial regression.

Developing this hybrid UQ framework requires a software infrastructure that provides building blocks, such as the R_{ki}'s and P_{ki}'s for different scenarios, and thus permit reuse for different applications. The objective of this framework design is to enable independent module development while preserving the ability to propagate global uncertainty. We have implemented a prototype of this framework.

It should be noted that, in order to propagate uncertainties from one module to another without mutual interference (in the design and implementation for individual modules), a common uncertainty representation is needed. In our current study and without a loss of generality, we have selected the Legendre polynomial chaos expansion as the common representation. With this selection, each stochastic dependent random variable is characterized by a vector of polynomial chaos coefficients. There are two advantages to this choice (polynomial chaos and specifically, Legendre PCE): (1) variance-based sensitivity information are propagated in addition to uncertainties; and (2) the mathematical complexity can be greatly reduced by having a single chaos expansion and yet re-analysis of the model output uncertainties with different probability distributions of the stochastic variables is computationally inexpensive. A potential problem with this representation, however, is that it does not handle large nonlinearities and discontinuities (nonlinear/discontinuous behaviors in the stochastic variable space) well due to inherent smoothness assumptions for low-order PCE-based methods. In addition, this representation also suffers from the curse of dimensionality, namely, that the computational complexity grows exponentially with the total number of stochastic variables. These issues will be addressed in later sections.

4.3. Non-intrusive Methods for Modular Uncertainty Propogation

Given the input stochastic solution field $C(\boldsymbol{\xi}, \boldsymbol{x}, t)$, the auxiliary field $u(\boldsymbol{\xi})$; and a deterministic solver $F_k(\boldsymbol{x}, t, C(\boldsymbol{x}, t))$, module M_k (not the same as F_k) advances the stochastic solution to become $\hat{C}(\boldsymbol{\xi}, \boldsymbol{x}, t)$. This involves, at time t_i,

1. generating a sample of size N_k (which should be larger enough to enable reconstruction of stochastic information) for $\boldsymbol{\xi}$ in the parameter space;
2. evaluating the sample on $C(\boldsymbol{\xi}, \boldsymbol{x}, t_i)$ and $u(\boldsymbol{\xi})$ to produce initial conditions for each simulation;
3. running the N_k deterministic simulations with the sample points and the initial conditions; and
4. collecting the solver outputs and using regression techniques to compute new PCE coefficients $\hat{C}(\boldsymbol{\xi}, \boldsymbol{x}, t_i)$.

In mathematical notation, this process can be described by:

$$\hat{C}(\boldsymbol{\xi}, \boldsymbol{x}, t) = \sum_{i=1}^{N_k} P_{ki} F_k(\boldsymbol{\xi}_k = \boldsymbol{\xi}_{ki}, R_{ki} C(\boldsymbol{\xi}, \boldsymbol{x}, t), R_{ki} u(\boldsymbol{\xi})) \tag{47}$$

where R_{ki} evaluates the given PCE representation at the i-th sample point (which are drawn using collocation or design of experiments such as quasi-Monte Carlo or Latin hypercube); and $P_{ki}, i = 1, \cdots, N_k$ corresponds

to using Legendre regression (projection onto the Legendre bases or least-squares) to reconstruct the PCE coefficients. To use the least-squares approach, we first have the Legendre polynomial equation, say, of third order:

$$Y = L_0 + \sum_{i=1}^{m} \alpha_i L_1(\boldsymbol{\xi}_{ki}) + \sum_{i=1}^{m}\sum_{j\le i}^{m} \alpha_{ij} L_2(\boldsymbol{\xi}_{ki}, \boldsymbol{\xi}_{kj}) + \sum_{i=1}^{m}\sum_{j\le i}^{m}\sum_{l\le j}^{m} \alpha_{ijl} L_3(\boldsymbol{\xi}_{ki}, \boldsymbol{\xi}_{kj}, \boldsymbol{\xi}_{kl}) + \epsilon \tag{48}$$

where $L_i(*)$'s are Legendre polynomial of i-th order involving the random variables $(*)$; α_k's, α_{kj}'s, and α_{kjl}'s are the regression coefficients to be determined, and ϵ_i is the regression error. Subsituting the sample to this equation gives rise to N equations which can be expressed in matrix form as:

$$\boldsymbol{Y} = \boldsymbol{L}\boldsymbol{\alpha} + \boldsymbol{\epsilon} \tag{49}$$

where $\boldsymbol{L}$ is a $N \times ((m+2)(m+1)/2)$ matrix where $\boldsymbol{\alpha}$ consists of all α_i's, α_{ij}'s, and α_{ijl}'s. Minimizing the 2-norm of the discrepancies $\boldsymbol{\epsilon}$ is equivalent to applying the least-squares approach so that

$$\boldsymbol{L}^T\boldsymbol{L}\boldsymbol{\alpha} = \boldsymbol{L}^T\boldsymbol{Y} \tag{50}$$

giving

$$\boldsymbol{\alpha} = (\boldsymbol{L}^T\boldsymbol{L})^{-1}\boldsymbol{L}^T\boldsymbol{Y}. \tag{51}$$

The reconstructed PCE coefficients $\boldsymbol{\alpha}$ are then propagated to the next module.

4.4. Intrusive Methods for Linear or Linearly Coupled Modules

As alluded to previously, for the linear case, it is possible to decompose a PCE-based module's computation into parallel, independent subproblems. Here, we present a theoretical property for linearly coupled equations that validates this decomposition. This property is significant in that it demonstrates that our "separability" of uncertainty treatment preserves the "global" nature of uncertainty propagation. Thus, this property legitimizes the "plug-and-play"-motivated approach to uncertainty propagation, that is, a code developer for a PCE-based intrusive module need not be concerned about random variables related to external modules.

Theorem 1: suppose the function $F_{\xi_1} : X(\xi_1, \xi_2) \to \hat{X}(\xi_1, \xi_2)$ is a linear map from $X(\xi_1, \xi_2)$ to $\hat{X}(\xi_1, \xi_2)$, and is p times differentiable at $\xi_1 = 0$ where ξ_1 and ξ_2 are independent random variables. Let $X(\xi_1, \xi_2)$ and $\hat{X}(\xi_1, \xi_2)$ both have PCE representation with order p. Then, the evaluation of F can be decomposed into $p+1$ independent subproblems. A given subproblem k involves the PC terms of $X(\xi_1, \xi_2)$ and the PC terms of $\hat{X}(\xi_1, \xi_2)$ that correspond only to the k^{th} order in ξ_2.

Proof: Let the PC representation of $X(\xi_1, \xi_2)$ be:

$$X(\xi_1, \xi_2) = \sum_{j=0}^{p}\sum_{i=0}^{p-j} X_{ij}\psi_i(\xi_1)\psi_j(\xi_2) \tag{52}$$

where $\psi_i(\xi_1)$ and $\psi_j(\xi_2)$ are the Legendre polynomials of degree i and j for ξ_1 and ξ_2; and similarly, let the PC representation of $\hat{X}(\xi_1, \xi_2)$ be:

$$\hat{X}(\xi_1, \xi_2) = \sum_{k=0}^{p}\sum_{l=0}^{p-k} \hat{X}_{kl}\psi_k(\xi_1)\psi_l(\xi_2) \tag{53}$$

where $\Psi_k(\xi_1)$ and $\Psi_l(\xi_2)$ are the Legendre polynomials of degree k and l for ξ_1 and ξ_2.

Then, the coefficients $\hat{X}_{kl}$'s can be computed by:

$$\hat{X}_{kl} = \int_{\mathcal{R}}\int_{\mathcal{R}} F_{\xi_1}(X(\xi_1(\theta), \xi_2(\theta)))\psi_k(\xi_1(\theta))\psi_l(\xi_2(\theta))P(\xi_1, \xi_2)d\xi_1 d\xi_2. \tag{54}$$

Since the operator F_{ξ_1} is linear, it can be decomposed into:

$$\begin{aligned} F_{\xi_1}(X(\xi_1(\theta), \xi_2(\theta))) &= F_{\xi_1}\left(\sum_{j=0}^{p}\sum_{i=0}^{p-j} X_{ij}\psi_i(\xi_1(\theta))\psi_j(\xi_2(\theta))\right) \\ &= \sum_{j=0}^{p} F_{\xi_1}\left(\sum_{i=0}^{p-j} X_{ij}\psi_i(\xi_1(\theta))\psi_j(\xi_2(\theta))\right). \end{aligned} \tag{55}$$

Substituting this into Eq. (54), we obtain:

$$\hat{X}_{kl} = \sum_{j=0}^{p} \int_{\mathcal{R}} \int_{\mathcal{R}} F_{\xi_1} \left(\sum_{i=0}^{p-j} X_{ij} \psi_i \left(\xi_1(\theta) \right) \psi_j \left(\xi_2(\theta) \right) \right) \psi_k(\xi_1(\theta)) \psi_l(\xi_2(\theta)) P(\xi_1, \xi_2) d\xi_1 d\xi_2. \tag{56}$$

Since F_{ξ_1} does not depend on ξ_2 (i.e., F_{ξ_1} is random only in ξ_1), the terms in the summation corresponding to $j \neq l$ will vanish giving the following simplified form:

$$\begin{aligned} \hat{X}_{kl} &= \int_{\mathcal{R}} \int_{\mathcal{R}} F_{\xi_1} \left(\sum_{i=0}^{p-l} X_{il} \psi_i \left(\xi_1(\theta) \right) \right) \psi_l \left(\xi_2(\theta) \right) \psi_k \left(\xi_1(\theta) \right) \psi_l \left(\xi_2(\theta) \right) P(\xi_1, \xi_2) d\xi_1 d\xi_2 & (57) \\ &= \left\langle \psi_l^2 \right\rangle \int_{\mathcal{R}} F_{\xi_1} \left(\sum_{i=0}^{p-l} X_{il} \psi_i \left(\xi_1(\theta 0) \right) \right) \psi_k \left(\xi_1(\theta) \right) P(\xi_1) d\xi_1. & (58) \end{aligned}$$

It can be observed from Eq. (58) that computing $\hat{X}_{kl}$ only depends on X_{il} for $i = 0$ to $p - l$.

Theorem 2: Suppose the function $F_{\boldsymbol{\xi_1}} : X(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2) \rightarrow \hat{X}(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)$ is a linear map from $X(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)$ to $\hat{X}(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)$, and is p times differentiable at $\boldsymbol{\xi}_1 = 0$ where $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ are independent random variables with respective dimensions m_1 and m_2. Let $X(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)$ and $\hat{X}(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)$ both have polynomial chaos (PC) representations with order p. Then, the evaluation of F can be decomposed into $(m_2 + p)!/\left(m_2! p!\right)$ independent subproblems. A given subproblem k involves the PC terms of $X(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)$ and the PC terms of $\hat{X}(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)$ that correspond only to the k^{th} term in the single-index PC representation of $\boldsymbol{\xi}_2$).

Proof: without loss of generality, let $\boldsymbol{\xi}_2 = \{\xi_1, \ldots, \xi_{m_2}\}$, $\boldsymbol{\xi}_1 = \{\xi_{m_2+1}, \ldots, \xi_{m_2+m_1}\}$, where $m_1 + m_2 = m$.

Let's consider the last term of a multi-dimensional PC representation of $X\left(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2\right)$ be:

$$\begin{aligned} &\sum_{j_1=1}^{p} \cdots \sum_{j_{m_2}=1}^{j_{m_2-1}} \cdots \sum_{j_m=1}^{j_{m-1}} X_{j_1 \cdots j_m} \Psi\left(\boldsymbol{\xi}\right) \\ &= \sum_{j_1=1}^{p} \cdots \sum_{j_{m_2}=1}^{j_{m_2-1}} \cdots \sum_{j_m=1}^{j_{m-1}} X_{j_1 \cdots j_m} \psi_{j_1}\left(\xi_1\right) \cdots \psi_{j_m}\left(\xi_m\right) \end{aligned} \tag{59}$$

where we omit the index of $\Psi\left(\boldsymbol{\xi}\right)$.

Similarly, let the general term of a multi-dimensional PC representation of $\hat{X}\left(\boldsymbol{\xi}\right)$ be:

$$\begin{aligned} &\sum_{i_1=1}^{p} \cdots \sum_{i_{m_2}=1}^{i_{m_2-1}} \cdots \sum_{i_m=1}^{i_{m-1}} \hat{X}_{i_1 \cdots i_m} \Psi\left(\boldsymbol{\xi}\right) \\ &= \sum_{i_1=1}^{p} \cdots \sum_{i_{m_2}=1}^{i_{m_2-1}} \cdots \sum_{i_m=1}^{i_{m-1}} \hat{X}_{i_1 \cdots i_m} \psi_{i_1}\left(\xi_1\right) \cdots \psi_{i_m}\left(\xi_m\right) \end{aligned} \tag{60}$$

Then, the coefficients $\hat{X}_{i_1 \cdots i_m}$'s can be computed by:

$$
\begin{aligned}
\hat{X}_{i_1\cdots i_m} &= \int_{\Theta} F_{\boldsymbol{\xi}_1}(X(\boldsymbol{\xi}_1(\theta), \boldsymbol{\xi}_2(\theta)))\psi_{i_1}(\xi_1(\theta))\cdots\psi_{i_m}(\xi_m(\theta))\,\mathrm{d}Pr(\theta) \\
&= \int_{\Xi} F_{\boldsymbol{\xi}_1}(X(\boldsymbol{\eta}_1, \boldsymbol{\eta}_2))\psi_{i_1}(\eta_1)\cdots\psi_{i_m}(\eta_m)\prod_{\mathrm{k}=1}^{\mathrm{m}} Pr_{\xi_k}(\eta_k)\,\mathrm{d}\boldsymbol{\eta} \\
&= \int_{\Xi} F_{\boldsymbol{\xi}_1}(X(\boldsymbol{\eta}_1, \boldsymbol{\eta}_2))\prod_{\mathrm{k}=1}^{\mathrm{m}}\psi_{i_k}(\eta_k)\prod_{\mathrm{k}=1}^{\mathrm{m}} Pr_{\xi_k}(\eta_k)\,\mathrm{d}\boldsymbol{\eta} \\
&= \int_{\Xi_2}\int_{\Xi_1} F_{\boldsymbol{\xi}_1}(X(\boldsymbol{\eta}_1, \boldsymbol{\eta}_2))\prod_{\mathrm{k}=1}^{\mathrm{m}_1}(\psi_{i_k}(\eta_k) Pr_{\xi_k}(\eta_k))\,\mathrm{d}\boldsymbol{\eta}_1\prod_{\mathrm{k}=1}^{\mathrm{m}_2}(\psi_{\mathrm{i_k}}(\eta_{\mathrm{k}})\,\mathrm{Pr}_{\xi_{\mathrm{k}}}(\eta_{\mathrm{k}}))\,\mathrm{d}\boldsymbol{\eta}_2 \\
&= \int_{\Xi_2}\int_{\Xi_1} F_{\boldsymbol{\xi}_1}(X(\boldsymbol{\eta}_1, \boldsymbol{\eta}_2))\Psi_1(\boldsymbol{\eta}_1)\,\mathrm{d}Pr(\boldsymbol{\eta}_1)\,\Psi_2(\boldsymbol{\eta}_2)\,\mathrm{d}Pr(\boldsymbol{\eta}_2) \qquad (61)
\end{aligned}
$$

Thus, we obtain

$$
\begin{aligned}
\hat{X}_{i_1\cdots i_m} &= \int_{\Xi_2}\int_{\Xi_1} F_{\boldsymbol{\xi}_1}(\sum_{j_1=1}^{p}\cdots\sum_{j_{m_2}=1}^{j_{m_2-1}}\cdots\sum_{j_m=1}^{j_{m-1}} X_{j_1\cdots j_m}\Psi(\boldsymbol{\eta}))\Psi_1(\boldsymbol{\eta}_1)\,\mathrm{d}Pr(\boldsymbol{\eta}_1)\,\Psi_2(\boldsymbol{\eta}_2)\,\mathrm{d}Pr(\boldsymbol{\eta}_2) \\
&= \sum_{j_1=1}^{p}\cdots\sum_{j_{m_2}=1}^{j_{m_2-1}}\int_{\Xi_2}\int_{\Xi_1} F_{\boldsymbol{\xi}_1}(\sum_{j_{m_2+1}=1}^{j_{m_2}}\cdots\sum_{j_m=1}^{j_{m-1}} X_{j_1\cdots j_m}\Psi(\boldsymbol{\eta}))\Psi_1(\boldsymbol{\eta}_1)\,\mathrm{d}Pr(\boldsymbol{\eta}_1)\,\Psi_2(\boldsymbol{\eta}_2)\,\mathrm{d}Pr(\boldsymbol{\eta}_2) \\
&= \prod_{k=1}^{m_2}\langle\psi_{i_k}^2\rangle\int_{\Xi_1} F_{\boldsymbol{\xi}_1}(\sum_{j_{m_2+1}=1}^{j_{m_2}}\cdots\sum_{j_m=1}^{j_{m-1}} X_{i_1\cdots i_{m_2}\cdots j_m}\Psi_1(\boldsymbol{\eta}_1))\Psi_1(\boldsymbol{\eta}_1)\,\mathrm{d}Pr(\boldsymbol{\eta}_1) \qquad (62)
\end{aligned}
$$

Hence, the equation above holds the multi-dimensional PC representation of $X(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)$. It can be observed that the total number of independent problems is

$$
\sum_{k=1}^{m_2}\sum_{i_1=1}^{p}\cdots\sum_{i_k=1}^{i_k-1} = \frac{(m_2+p)!}{m_2!p!}. \qquad (63)
$$

Furthermore, each given subproblem k involves the

$$
\frac{(m_1+k+p)!}{m_1!\,(p-k)!} \qquad (64)
$$

number of PC terms of $X(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)$ and the

$$
\frac{(m_2+k)!}{m_2!k!} \qquad (65)
$$

number of PC terms of $\hat{X}(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)$ that correspond only to the k^{th} term in the single-index PC representation of $\boldsymbol{\xi}_2$. □

Corollary: for the linear problem, we obtain that

$$
\frac{(m_1+m_2+p)!}{(m_1+m_2)!p!} = \sum_{k=0}^{p}\left[\frac{(m_1+p-k)!}{m_1!\,(p-k)!}\right]\left[\frac{(m_2+k-1)!}{(m_2-1)!k!}\right] \qquad (66)
$$

Alternatively, the identity above can be verified by Rothe-Hagen Identity as follows:

$$
\sum_{k=0}^{n}\frac{x}{x+kz}\begin{pmatrix} x+kz \\ k \end{pmatrix}\frac{y}{y+(n-k)z}\begin{pmatrix} y+(n-k)z \\ n-k \end{pmatrix} = \frac{x+y}{x+y+nz}\begin{pmatrix} x+y+nz \\ n \end{pmatrix} \qquad (67)
$$

Proof: Let $z=1$, $n=p$, $x=m_2$, $y=m_1+1$ in the Rothe-Hagen Identity. □

As opposed to pure intrusive PCE, the intrusive PCE module in a hybrid framework needs to be formulated in the larger context of the entire multi-physics system. Specifically, there are random variables in the system that are external to this module. Let M_k be an intrusive PCE module with its internal random variables $\boldsymbol{\xi}_k$, then the external variables are $\boldsymbol{\xi} \setminus \boldsymbol{\xi}_k$. Since the goal of the hybrid framework is to hide the presence of non-local random variables in the incoming solution field (that is, the local solver should only need to advance the effect of local random variables) without sacrificing the global uncertainty propagation property, M_k needs to perform additional processing (for example, projection of the incoming field to the field involving only the local random variables) before and after calling the user-implemented local PCE solver. For linear problems (meaning that the discretization matrix does not depend on stochastic terms involving random variables external to the module), we show through a decomposition property that the additional processing involves operations analogous to scatter-and-gather. These operations can be represented by:

$$\hat{C}(\boldsymbol{\xi}, \boldsymbol{x}, t) = \sum_{i=0}^{p} P_{ki} F_{ki}(\boldsymbol{\xi}_k, R_{ki} C(\boldsymbol{\xi}, \boldsymbol{x}, t), R_{ki} u(\boldsymbol{\xi})) \tag{68}$$

where R_{ki} is the projection (scatter) operator onto order i, F_{ki} is the PCE solver of order i, and P_{ki} is the corresponding prolongation (gather) operator. These generic operators can be built into the hybrid UQ computational infrastructure.

We use the following example to illustrate these operations. Suppose a second-order ($p = 2$) PCE is used and full system has 2 random variables with one internal to the intrusive PCE module ($\boldsymbol{\xi}_1$), then the system PCE representation of the incoming solution has $(P + 1) = (2 + 2)!/(2!2!) = 6$ terms to capture the overall uncertainties. However, the user-provided PCE solvers (F_{ki}'s) has only the knowledge of their internal variables, which should have $(P + 1) = (1 + 2)!/(1!2!) = 3$ permutation terms. How do we bridge the mismatch between the internal and external uncertainty representations. For linear problems, it suffices to break up the incoming solution into 3 subproblems, each corresponding to a different permutation in the external random variables (see Table 1 below).

Table 1: Partition of PCE Permutation Into 3 Sub-problems

Internal Variable	External Variable	Sub-problem
0	0	0
1	0	0
0	1	1
2	0	0
1	1	1
0	2	2

Hence, the processing steps for this module are:

1. partition $C(\boldsymbol{\xi}, \cdots)$ into 3 parts (sub-problem 0, 1, and 2),
2. solve sub-problem i by F_{ki} corresponding to the user-provided PCE solver of order i, and
3. gather the solution from each sub-problem back to form $\hat{C}(\boldsymbol{\xi}, \cdots)$ (6-term).

A key computational kernel for this module is the solver for the linear systems arising from discretization of the stochastic equations, assuming that the physics for this module is governed by some partial differential equations that can be solved efficiently by fast matrix equation solvers. In the following we present a fast linear solver algorithm for stochastic matrices from using polynomial chaos expansions. In [37], Sousedik, Ghanem, and Phipps showed that the stochastic matrices exhibit a special nonzero block structure that facilitates a hierarchical decomposition. As such, the global matrix A_p of PCE order p can be represented by

$$A_p = \begin{bmatrix} A_{p-1} & B_p \\ C_p & D_p \end{bmatrix}$$

where D_p is a block diagonal matrix that can be easily parallelized, and A_{p-1} is the corresponding global matrix of polynomial order $p-1$. Hence, A_p can also be represented hierarachically by

$$A_l = \begin{bmatrix} A_{l-1} & B_l \\ C_l & D_l \end{bmatrix}, \quad l = p, \cdots, 1.$$

This hierarchical structure can be exploited in designing efficient solver algorithms. Specifically, it enables a construction of the hierarchical preconditioner M_p:

$$A_p \approx M_p = \begin{bmatrix} M_{p-1} & B_l \\ C_l & D_l \end{bmatrix}$$

where $M_l, l = 1, \cdots, p-1$ are obtained from the corresponding A_l with the $(1,1)$ block replaced by M_{l-1} except for $M_0 = A_0$. Several options are available for designing the preconditioners. While Sousedik *el al.* presented a Schur complemented-based preconditioner, an alternative preconditioning algorithm based on block Gauss Seidel is given below:

Algorithm 1 (Hierarchical Block Gauss Seidel) $u_p = M_p^{-1} z_p$
for $l = p, \cdots, 1$
  Let $z_l = [(z_l^{l-1})^T \ (z_l^l)^T]^T$ and $u_l = [(u_l^{l-1})^T \ (u_l^l)^T]^T$, then $z_{l-1} = z_l^{l-1} - B_l u_l^l$.
end for
$u_0 = M_0^{-1} z_0$
for $l = 1, \cdots, p$
  $u_l^l = D_l^{-1}(z_l^l - C_l u_l^{l-1})$
end for

The most time-consuming operation in this algorithm is the inversion (D_l^{-1}). Fortunately, D_l is block diagonal (each block is analogous to the discretized matrix of a different partial different equation) which can be solved efficiently in parallel.

There is an additional efficiency gain of employing this hierarchical structure to the matrix solution in our intrusive PCE module. As described previously, during an intrusive PCE solve, the global problem is partitioned into $p+1$ subproblems, each corresponding to a different polynomial order. It is clear from the hierarchical structure that the subproblems of lower polynomial orders are principal submatrices of the those of the higher polynomial orders. As such, we can coalese these solves for higher efficiency and lower matrix storage resulting in the following modified algorithm:

Algorithm 2 (Modified Hierarchical Block Gauss Seidel for $U_l = M_l^{-1} Z_l, l = 0, \cdots, p$)
Let $Y_p = Z_p$ and $V_p = U_p$
for $l = p, \cdots, 1$
  Split $Y_l = [(Y_l^{l-1})^T \ (Y_l^l)^T]^T$ and $V_l = [(V_l^{l-1})^T \ (V_l^l)^T]^T$
  Update $Y_l^{l-1} = (Y_l^{l-1} - B_l V_l^l)^T]$.
  Coalesce $Y_{l-1} = [(Y_l^{l-1})^T \quad Z_l^T]^T$.
end for
Mean solve: $V_0 = A_0^{-1} Y_0$
Un-coalesce: split V_0 into U_0 and V_1^0.
for $l = 1, \cdots, p$
  $V_l^l = D_l^{-1}(Y_l^l - C_l V_l^{l-1})$
  Un-coalesce: split V_l into U_l and V_{l+1}^l.
end for

In this algorithm, Z_p and U_p are such that $M_p U_p = Z_p$ for the p-order subproblem. The first loop first splits both $Y_p = Z_p$ into two parts corresponding to the right hand sides of the p-th order as well as the (p-1)-th or lower orders. The same splitting is applied to $V_p = U_p$. The part corresponding to the lower order data is updated and then combined with the data Z_{p-1} (right hand sides of the subproblem for polynomial order $p-1$). The combined information are transferred to the next level (lower order) for similar processing until $l = 0$. At this point, Y_0 has the updated information from all subproblems. This multiple right hand

side problem is solved ($A_0 V_0 = Y_0$) and the processing steps are performed in the reverse direction (increasing order). It can be seen that the lower order solves (D_l and A_0) benefit from fast solvers for multiple right hand sides. In addition, efficiency improvement is expected since now $B_l V_l^l$ and $C_l V_l^{l-1}$ are matrix-matrix multiplications instead of matrix-vector multiplications.

4.5. Intrusive Methods for Nonlinear or Non-linearly Coupled Modules

Since non-intrusive sampling is trivially applicable for both linear and nonlinear equations, we will focus on intrusive PCE within our modularly hybrid framework. The methodology for propagating global uncertainty through PCE-based modules with nonlinear partial differential equations, as opposed to those with linear equations, requires generating a sample with the random variables external to the modules and running the sample on the user-provided stochastic nonlinear solver. This amounts to conditional evaluation of the global uncertainty representation with respect to external random variables. In other words, each sampled value of the external variables is used to construct a set of PCE coefficients with respect to the internal random variables. Hence, each subproblem corresponds to an independent evaluation of PCE terms. So if there are n_k sampled values of the external random variables, then we will end up with n_k sets of "partial" PCE coefficients, which are then assembled to recover the global uncertainty representation. Here, n_k needs to be sufficiently large to allow for unique reconstruction of the global representation. In summary, R_{ki} corresponds to the restriction of PCE format to the internal variables by sampling the external variables, and P_{ki} corresponds to the global PCE reconstructions from the partial PCE representations.

For nonlinear problems (that is, coefficients of the local equations are also functions of random variables external to the module), the same decomposition property for linear problems does not apply and neither are the simple scatter-gather operations. Nonetheless, generic projection and prolongation operators can be formulated based on the familiar mathematical representation:

$$\hat{C}(\boldsymbol{\xi}, \boldsymbol{x}, t) = \sum_{i=1}^{N_k} P_{ki} F_k(\boldsymbol{\xi}_k, R_{ki} C(\boldsymbol{\xi}, \boldsymbol{x}, t), R_{ki} u(\boldsymbol{\xi})). \tag{69}$$

Here R_{ki} is the partial sampling operator onto the external random variables while leaving the information with respect to the internal random variables in the PCE space. We thus call this "conditional" PCE corresponding to partial evaluation of polynomial chaos expansions. F_k is the user-provided PCE nonlinear solver of order p. Note that, unlike the linear case, all instances (sample points) are now solved using the p-th order PCE solver. P_{ki} is a prolongation operator combining all N_k instances back into the original format. This operation involves Legendre regression similar to those in the non-intrusive module but restricted only to the external random variables (partial reconstruction). Hence, this module can be viewed as a combination of non-intrusive and linear PCE operations. The processing steps for this module are:

1. generate N global sample points on all the random variables $\boldsymbol{\xi}$;
2. restrict N_k local sample points from the external random variables $\boldsymbol{\xi} \setminus \boldsymbol{\xi}_k$;
3. perform partial evaluation of $C(\boldsymbol{\xi}, \cdots)$ and $u(\boldsymbol{\xi})$ on the sample local N_k sample points;
4. propogate uncertainty of the all N_k instances via user-provided conditional PCE nonlinear solver of order p;
5. use Legendre regression to partially reconstruct PCE coefficients for the external random variables $\boldsymbol{\xi} \setminus \boldsymbol{\xi}_k$

Finally, we combine all N_k instances back into the original global PCE format for $\hat{C}(\boldsymbol{\xi}, \boldsymbol{x}, t)$. Unlike the linear module that can exploit the decomposition property and the fast solvers, here each of the N_k instances are independently discretized and solved from each other. The derivation of "conditional" PCE reconstruction is straightforward and thus not to be included here.

4.6. Semi-intrusive Methods for Linear or Nonlinear Modules

Given the input stochastic solution field $C(\boldsymbol{\xi}, \boldsymbol{x}, t)$, the auxiliary field $u(\boldsymbol{\xi})$; and a derivative-based solver which computes $F_k(\boldsymbol{x}, t, C(\boldsymbol{x}, t))$ as well as its derivatives $\frac{\partial F_k}{\partial \boldsymbol{\xi}_{kj}}; j = 1, \cdots, m_k$; module M_k advances the stochastic solution to become $\hat{C}(\boldsymbol{\xi}, \boldsymbol{x}, t)$ such that

$$\hat{C}(\boldsymbol{\xi}, \boldsymbol{x}, t) = \sum_{i=1}^{N_k} P_{ki} \boldsymbol{F}_k(\boldsymbol{\xi}_k = \boldsymbol{\xi}_{ki}, R_{ki} C(\boldsymbol{\xi}, \boldsymbol{x}, t), R_{ki} u(\boldsymbol{\xi})) \tag{70}$$

where $\boldsymbol{F}_k$ is a vector of functions comprising $F_k(\boldsymbol{x}, t, C(\boldsymbol{x}, t))$ and $\frac{\partial F_k}{\partial \boldsymbol{\xi}_{kj}}; j = 1, \cdots, m_k$; R_{ki}, same as in the non-intrusive case, evaluates the given PCE representation at the i-th sample point; and $P_{ki}, i = 1, \cdots, N_k$ corresponds to using regression to reconstruct the PCE coefficients. The equations have been given previously as in Eqns (14) and (18).

There is an additional consideration in the propagation of uncertainties through this module. Specifically, for reason of simplication, we leave out $u(\boldsymbol{\xi})$ and also the spatial and temporal notations. Then,

$$\frac{\partial F_k(\boldsymbol{\xi}_k, C(\boldsymbol{\xi}))}{\partial \boldsymbol{\xi}_{kj}} = \frac{dF_k(\boldsymbol{\xi}_k, C(\boldsymbol{\xi}))}{d\boldsymbol{\xi}_{kj}} + \frac{dF_k(\boldsymbol{\xi}_k, C(\boldsymbol{\xi}))}{dC(\boldsymbol{\xi}_{kj})} \frac{dC(\boldsymbol{\xi}_{kj})}{d\boldsymbol{\xi}_{kj}}. \tag{71}$$

Here the first term and the first part of first part of the second term on the right hand side are calculated by the user-provided semi-intrusive solver, while the second part of the second term is to be computed by the hybrid computational framework by computing the derivatives of the solution field with respect to the random variables.

4.7. Stochastic Coupling

Let $\boldsymbol{\xi} = \{\xi_1, \xi_2, \xi_3, \xi_4\}$ be a set of random variables uniformly distributed in some stochastic space Ξ in the following system of equations:

$$\begin{aligned} u_1(\boldsymbol{\xi}) &= f_{11}(\xi_1, u_1, u_2) + f_{12}(\xi_2, u_1, u_2) \\ u_2(\boldsymbol{\xi}) &= f_{21}(\xi_3, u_1, u_2) + f_{22}(\xi_4, u_1, u_2) \end{aligned} \tag{72}$$

where $f_{ij}; i, j = 1, \cdots, 2$ are stochastic functions; and let $u_1(\boldsymbol{\xi})$ and $u_2(\boldsymbol{\xi})$ be the corresponding stochastic solution fields. Suppose that, after proper discretization, we obtain the following linear system:

$$\begin{bmatrix} F_{11}(\xi_1) & F_{12}(\xi_2) \\ F_{21}(\xi_3) & F_{22}(\xi_4) \end{bmatrix} \begin{bmatrix} U_1(\boldsymbol{\xi}) \\ U_2(\boldsymbol{\xi}) \end{bmatrix} = \begin{bmatrix} B_1(\boldsymbol{\xi}) \\ B_2(\boldsymbol{\xi}) \end{bmatrix}$$

where $B_i; i = 1, 2$ are some right hand sides that are also functions of all the random variables $\boldsymbol{\xi}$.

In the following we show how to solve this coupled system using hybrid UQ.

Let F_{11} and F_{12} be modules equipped with linear or non-linear intrusive UQ methods; and F_{21} and F_{22} with non-intrusive UQ methods. Two possible ways of solving this system:

1. Iterate on the following system:

$$\begin{bmatrix} F_{11}(\xi_1) & 0 \\ 0 & F_{22}(\xi_4) \end{bmatrix} \begin{bmatrix} U_1^{n+1}(\boldsymbol{\xi}) \\ U_2^{n+1}(\boldsymbol{\xi}) \end{bmatrix} = \begin{bmatrix} B_1(\boldsymbol{\xi}) \\ B_2(\boldsymbol{\xi}) \end{bmatrix} - \begin{bmatrix} 0 & F_{12}(\xi_2) \\ F_{21}(\xi_3) & 0 \end{bmatrix} \begin{bmatrix} U_1^{n}(\boldsymbol{\xi}) \\ U_2^{n}(\boldsymbol{\xi}) \end{bmatrix}$$

   where n is the iteration index.

2. Apply iterative solver for the entire system and use the block diagonal matrix as the preconditioning.

In both cases, we will have to create four stochastic modules within the hybrid UQ framework where F_{11} and F_{12} will be using the linear or non-linear intrusive module, while F_{21} and F_{22} will be using the non-intrusive sampling module. Case 1 above will involve a 'solve' operation for F_{11} and F_{22}, and a 'matvec' (matrix vector multiply) for F_{12} and F_{21}. For case 2, additional 'matvec' operations are needed for F_{11} and F_{22}. Both cases can be captured well by our hybrid UQ framework.

4.8. Applicability to Modules with High Dimensions (With Correlated Inputs)

Our modularly hybrid framework can also be applied to multi-physics stochastic equations involving spatially distributed random parameters (i.e., $\boldsymbol{x}$ with spatial dependence). To do so, we would simply need another specialized module whose solver is equipped to handle uncertainty propagation for spatially distributed random parameters. In principle, spatial dependence increases the dimensionality of the random variable space, so our concepts of hybrid modularization (i.e., subproblem decomposition, restriction and prolongation) naturally apply to the spatial case as well. In essence, a spatial module can be construed as a high-dimensional linear module. To address the issue of high dimensionality, Kahunen-Loeve expansion (KLE) [38, 39] can be leveraged to de-correlate the spatial process for dimension reduction. KLE has the attractive property that the mean-square error introduced by truncating the expansion is minimized. In addition, one can also decide on the number of terms to truncate based on an acceptable threshold for the total variance of the truncated expansion. For these reasons, KLE has been widely applied to problems involving spatial heterogeneity, and is becoming more prevalent in environmental applications involving subsurface flow problems [40, 41, 42, 43, 44]. An application example will be given later in this report.

4.9. Dimension Reduction in General Settings

As with all PCE-based methods, our hybrid UQ methodolgy also suffers from the 'curse-of-dimensionality' problem. As such, we also investigate possible strategies for dimension reduction (DR). The previous section addresses high dimensionality induced by correlated random field, for which methods based on singular value decomposition suffice. For input spatial process with relative low correlation length, the dimensionality of the random input represented by the truncated KLE can be very large. To deal with the "curse of dimensionality", we apply non-parametric screening method [45] to efficiently identify the most sensitive random variables for truncated KLE with many random variables. Other dimension-reduction techniques have been Other dimension-reduction techniques have been proposed, such as (1) multi-element, generalized polynomial chaos [46]; and (2) decomposition methods motivated by analysis of variance (ANOVA) techniques [47] and used for high-dimensional UQ problems in [48, 49].

Since our hybrid UQ methodology facilitates the propagation of not just the uncertainty, but also global sensitivities (Sobol' indices), we may be able to reduce uncertain parameter dimension dynamically by examining the global sensitivity information of each parameter so that parameters with small Sobol' index are candidates for removal. Alternatively, two subsets of parameters that have no coupling terms can be treated more efficiently by decoupling them. This scheme, however, is at best heuristic, since an insensitive parameter at a certain time step may become sensitive in future time steps. Nevertheless, equipped with expert judgment in the course of simulation, this may be a viable approach.

4.10. Bayesian Calibration

Uncertainty analysis begins with a prescription of probability distributions (called 'priors') of the uncertain parameters. Samples are drawn from these priors and propagated through the simulation models to produce an ensemble of model outputs for analysis. These output statistics are often compared against results from physical experiments, which can help 'correct' the priors using a technique called Bayesian inference after Bayes. The Bayes' rule says that

$$\pi(\theta|D) \propto P(\theta)L(D|\theta) \tag{73}$$

where $P(\theta)$ is the prior distribution, $L(D|\theta)$ is called the likelihood function, and $\pi(\theta|D)$ is the posterior distribution describing how the input 'priors' have been modified in view of the observational data.

Bayesian inference is thus an inverse operation analogous to numerical optimization, only that it provides more informative optimization results as they generate posterior distributions for the input parameters instead of a single (or a few) optimal input points. However, this advantage comes with additional computational cost, usually requiring thousands of function evaluations in its Markov Chain Monte Carlo (MCMC) iterations. As such, this method is mostly used with response surface models rather than directly with the simulators.

As mentioned above, two pieces of information are needed to set up and run Bayesian inferences: ¶(θ) and $L(D|\theta)$; and the latter is derived by a function in terms of some discrepancy metric between simulation and data. A popular likelihodd function is:

$$L(D|\theta) = N_L \exp(-E) \tag{74}$$

where E is some discrepancy function and N_L is the normalization factor. To compute E at different locations in the parameter space, many function evaluations may be needed. In the context of our hybrid framework, E can be estimated inexpensively by probing the 'polynomial' response surfaces in PCE representation, which are readily available from the module outputs at every time step. We demonstrate this analysis in our numerical study section.

5. A Computational Framework for Hybrid UQ Methods

The mathematical operations for our hybrid UQ framework can be encapsulated in a generic computational framework. In this section, we summarize the capabilities needed in each type of modules. We then discuss the design of a software infrastructure for hybrid UQ.

5.1. Basic Operators in Modules

The basic components in each module are R_{ki} and P_{ki}.

1. Non-intrusive Methods for Modular Uncertainty Propogation

   R_{ki}**:** This corresponds to generating a sample for all random variables in the system. Sampling designs available are Latin hypercube and quasi-Monte Carlo. This operator involves evaluating the incoming field $C(\boldsymbol{\xi})$ at the sample points.

   P_{ki}**:** This corresponds to least-squares regression given in Eqn (48 or collocation to reconstruct $\hat{C}(\boldsymbol{\xi})$.

2. Intrusive Methods for Linear or Linearly Coupled Modules

   R_{ki}**:** This corresponds to scattering the incoming field $C(\boldsymbol{\xi})$ into PCE solvers of different orders (i's).

   P_{ki}**:** This corresponds to gathering the solutions from PCE solvers of different orders to reconstruct $\hat{C}(\boldsymbol{\xi})$.

3. Intrusive Methods for Nonlinear or Non-linearly Coupled Modules

   R_{ki}**:** This corresponds to first generating a sample for only the random variables external to the current module. The incoming $C(\boldsymbol{\xi})$ is then evaluated partially at the sample points and the "reduced" PCE (only expressed in local random variable space) are created.

   P_{ki}**:** This corresponds to least-squares regression given in Eqn (48 only for the random variables external to the module.

4. Semi-intrusive Methods for Linear or Nonlinear Modules

   R_{ki}**:** This corresponds to generating a sample for all random variables in the system. Sampling designs available are Latin hypercube and quasi-Monte Carlo. This operator involves evaluating the incoming field $C(\boldsymbol{\xi})$ at the sample points.

   P_{ki}**:** This corresponds to a special least-squares regression given in Eqn (48 and (18. In additon, since the user-provided derivative-based solver only computes part of the total derivative (Eqn (71), additional processing is needed to append the needed terms.

5.2. Hybrid UQ Software Infrastructure

In this section we give a brief description of the software framework to capture the features of the hybrid computational framework. There are four basic modules in the framework:

1. ModuleSampling,
2. ModuleLinPCE,
3. ModuleNonLinPCE, and

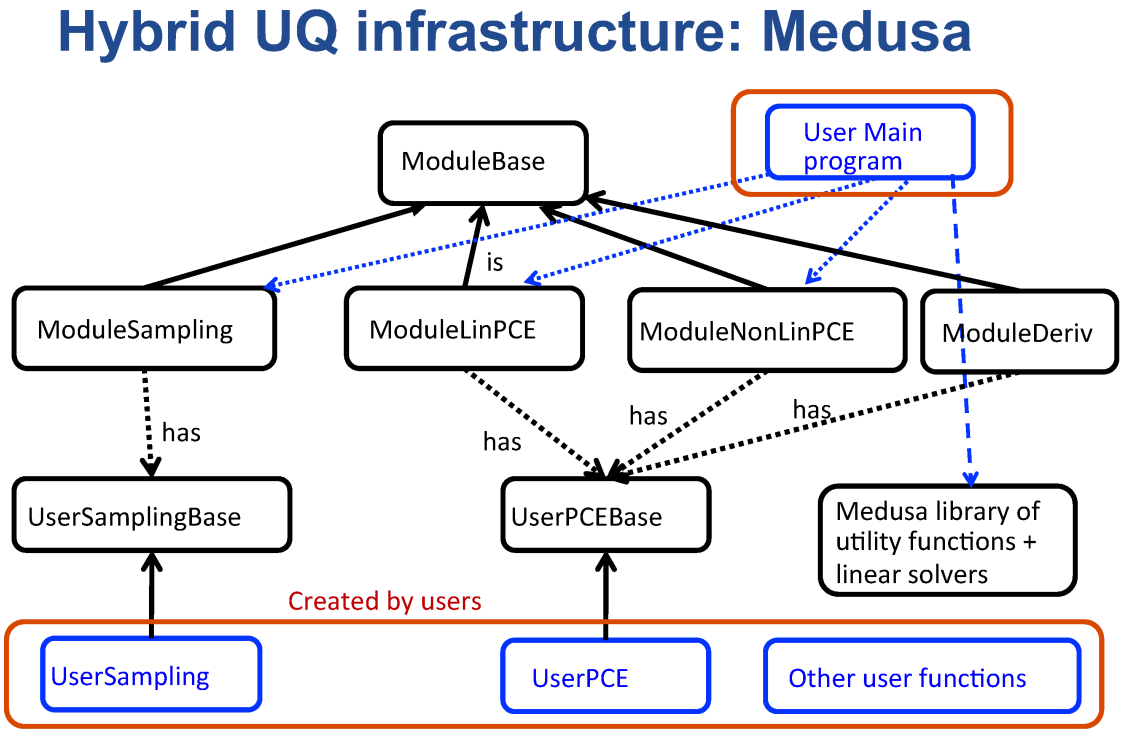


Figure 2: Hybrid UQ Software Infrastructure.

4. ModuleDerivative

all of which inherit from an abstract **ModuleBase** class. Each module is characterized by seven functions:

- a constructor,
- a destructor,
- a 'setup' function,
- a 'solve' function,
- a 'matvec' function,
- a 'setParam' function (for setting internal parameters), and
- a 'setSolver' function for setting the user-provided solvers.

The R_{ki} and P_{ki} operators are embedded in the 'setup' and 'solve' functions. An example of a user program using the non-intrusive module is as follow:

```
int nRVs=2, pOrder=2;
PCEPermutations pcePerms;
PCEDataStreams  VecPD;
ModuleSampling  Module;
SolverContext   Context;
UserSampling    *userSolver = new UserSampling();
sparam = "nRVs 2";   Module.setParam(sparam);
sparam = "pOrder 2"; Module.setParam(sparam);
sparam = "setRV 1";  Module.setParam(sparam);
sparam = "setRV 2";  Module.setParam(sparam);
```

```
Module.setSolver((UserSamplingBase *) userSolver);
pcePerms.genPermutationTable(nRVs, pOrder);
nPerms = pcePerms.numPermutations();
VecPD.resize(nPerms, 2);
//**/ set values for the incoming field VecPD
Module.setup(VecPD, Context);
Module.solve(VecPD, Context);
```

6. Numerical Study

In this section we demonstrate the application of hybrid UQ to a few example problems. We begin with a two-module reactive transport solver comprising two modules: intrusive transport and non-intrusive reaction. The second example is a three-module flow-reactive-transport solver for showing dimension reduction of a random field. The third example is a simple predator-prey model to demonstrate the non-linear intrusive UQ capability in our hybrid framework. The fourth example is a more realistic non-linear spring model decomposed into two modules. The fifth example is a non-linear cuopled contaminant flow solver. The final example demonstrates the use of Bayesian inference on one of our reactive transport examples.

6.1. Two-dimensional Reactive Transport in Isotropic and Homogeneous Medium

In this section, we demonstrates how our proposed framework can be used in the UQ analyses of sequential multi-step transport reaction modules.

Sequential multi-step reaction models are central to the understanding of various biologic and chemical processes that occur in the subsurface [50, 51]. For example, during denitrification (process used to remove nitrogen from sewage and wastewater), nitrate reacts to produce nitrite, and subsequently ammonia or nitrogen gas [52]. Similarly, reductive anaerobic degradation of tetrachloroethylene (common soil contaminant) to trichloroethylene, to dichloroethylene, and eventually to vinyl chloride, may be modeled using a sequential first-order degradation kinetic model [53]. Being able to apply UQ on these types of sequential multi-step reaction models has tremendous potential to facilitate assessment and control of various biochemical processes that greatly impact the environment.

In the latter case, reductive anaerobic degradation falls under the general category of multi-species reactive transport in porous media, which can be described by the following system of time-dependent partial differential equations:

$$R_i \frac{\partial c_i}{\partial t} = D^x \frac{\partial^2 c_i}{\partial x^2} + D^y \frac{\partial^2 c_i}{\partial y^2} - v^x \frac{\partial c_i}{\partial x} - v^y \frac{\partial c_i}{\partial y} + R_{i-1} k_{i-1} c_{i-1} - R_i k_i c_i, \;\; \forall i = 1, 2, \ldots, n \tag{75}$$

where c_i [ML^{-3}] is the concentration of the i^{th} species; t is time [T]; v^x and v^y [LT^{-1}] are the velocity components for x- and y-direction respectively; D^x and D^y [$\mathrm{L}^2\mathrm{T}^{-1}$] are the dispersion coefficients for x- and y-direction respectively; k_i is the first-order reaction rate [T^{-1}]; R_i is the retardation factor of the i^{th} species with $R_0 = 0$; and n is the total number of species. Dividing both sides by R_i, we obtain:

$$\frac{\partial c_i}{\partial t} = \frac{D^x}{R_i} \frac{\partial^2 c_i}{\partial x^2} + \frac{D^y}{R_i} \frac{\partial^2 c_i}{\partial y^2} - \frac{v^x}{R_i} \frac{\partial c_i}{\partial x} - \frac{v^y}{R_i} \frac{\partial c_i}{\partial y} + \frac{R_{i-1}}{R_i} k_{i-1} c_{i-1} - k_i c_i, \;\; \forall i = 1, 2, \ldots, n. \tag{76}$$

In matrix form, the system of multi-species equations can be rewritten as:

$$\frac{\partial \mathbf{c}}{\partial t} = \mathbf{D}^x \frac{\partial^2 \mathbf{c}}{\partial x^2} + \mathbf{D}^y \frac{\partial^2 \mathbf{c}}{\partial y^2} - \mathbf{V}^x \frac{\partial \mathbf{c}}{\partial x} - \mathbf{V}^y \frac{\partial \mathbf{c}}{\partial y} + \mathbf{A}\mathbf{c} \tag{77}$$

where:

$$\begin{aligned} \mathbf{c} &= \begin{pmatrix} c_1 & \ldots & c_n \end{pmatrix}^{\mathbf{T}}, \\ \begin{pmatrix} \mathbf{D}^x \\ \mathbf{D}^y \\ \mathbf{V}^x \\ \mathbf{V}^y \end{pmatrix} &= \begin{pmatrix} D^x/R_1 & \ldots & D^x/R_n \\ D^y/R_1 & \ldots & D^y/R_n \\ v^x/R_1 & \ldots & v^x/R_n \\ v^y/R_1 & \ldots & v^y/R_n \end{pmatrix}, \end{aligned} \tag{78}$$

and the structure of the matrix $\mathbf{A}$ depends on the reaction network [54].

In this work, we apply Eq. (77) to model the sequential first-order network that represents the biodegradation of trichloroethylene (TCE). TCE reacts to produce daughter species, dichloroethylene (DCE), while DCE further reacts to produce vinyl chloride (VC), and finally VC reacts to produce ethylene (ETH). That is, there are four species (i.e., $n = 4$) and the process is depicted as follows:

$$\text{TCE} \xrightarrow[k_1]{l_1} \text{DCE} \xrightarrow[k_2]{l_2} \text{VC} \xrightarrow[k_3]{l_3} \text{ETH} \xrightarrow[k_4]{l_4} \tag{79}$$

where l_i is the yield coefficient of the i^{th} reaction, with $l_i = 1$ for unimolecular reactions. Following Eq. (77), the reaction matrix $\mathbf{A}$ for this network has the following form:

$$\mathbf{A} = \begin{pmatrix} -k_1 & 0 & 0 & 0 \\ \frac{R_1}{R_2} l_1 k_1 & -k_2 & 0 & 0 \\ 0 & \frac{R_2}{R_3} l_2 k_2 & -k_3 & 0 \\ 0 & 0 & \frac{R_3}{R_4} l_3 k_3 & -k_4 \end{pmatrix}. \tag{80}$$

To solve Eq. (77), one popular solution approach is to use "operator splitting", namely, to split the solution process in two stages to mimic the equation structure which consists of independent transport and reaction terms. At each time step, the "transport equation" is solved for the species concentration, followed by the solution of the "reaction equation" solved by an ODE solver. In the special case of sequential networks with first-order reaction, the reaction equation has an analytical solution [54].

Under operator splitting of Eq. (77), the (vectorized) transport equation is:

$$\frac{\partial \mathbf{c}^{t+\frac{1}{2}}}{\partial t} = \mathbf{D}^x \frac{\partial^2 \mathbf{c}^{t+\frac{1}{2}}}{\partial x^2} + \mathbf{D}^y \frac{\partial^2 \mathbf{c}^{t+\frac{1}{2}}}{\partial y^2} - \mathbf{V}^x \frac{\partial \mathbf{c}^{t+\frac{1}{2}}}{\partial x} - \mathbf{V}^y \frac{\partial \mathbf{c}^{t+\frac{1}{2}}}{\partial y} \tag{81}$$

which consists of four independent equations, one per species, since $\mathbf{D}^x$, $\mathbf{D}^y$, $\mathbf{V}^x$ and $\mathbf{V}^y$ are all diagonal matrices. Then, we apply $\mathbf{c}^{t+1/2}$ (i.e., the solution to Eq. (81)) as the initial conditions to the (vectorized) reaction equation:

$$\frac{\partial \mathbf{c}^{t+1}}{\partial t} = \mathbf{A}\mathbf{c}^{t+1} \tag{82}$$

which can be solved analytically [54].

In reactive transport problems, there are many sources of uncertainties. In our model (cf. Eq. (75)), we treat the following parameters as uncertain: the dispersivities (α^x, α^y), the velocities (v^x, v^y), and the first-order reaction rates k_i. The dispersion coefficient can be prescribed as a linear function of dispersivity, i.e., $D^x = \alpha^x v^x$ and $D^y = \alpha^y v^x$ in the x- and y-direction, respectively. We assume α^x, α^y, v^x, v^y and $\{k_i\}_{i=1}^4$ as independent second-order random variables. Hence, the uncertainty of the concentration for the first species involves the uncertain parameters α^x, α^y, v^x, v^y and k_1. The uncertainty of the concentration for the second species involves the uncertain parameters α^x, α^y, v^x, k_1 and k_2. The uncertainty of the concentration for the third species involves the uncertain parameters α^x, α^y, v^x, v^y, k_1, k_2 and k_3. Finally, the uncertainty of the concentration for the fourth species involves all the uncertain parameters α^x, α^y, v^x, v^y, k_1, k_2, k_3 and k_4. While previous works [55, 56] assumed either the transport parameters (α^x, α^y, v^x, v^y) or the reactions parameters ($\{k_i\}_{i=1}^n$) are uncertain, this work is the first in examining the uncertainties in both transport and reaction parameters jointly.

6.1.1. Stochastic Galerkin Method for Transport Module

The transport system consists of four transport equations can be independently solved for each species. Hence, we will discuss the PCE formulation of a generic species' scalar transport equation. As such, we can simplify notation by removing both the species and time indexing. Denote $\mathbf{x} = (x, y) \in \Omega$ and the scalar transport equation becomes:

$$\frac{\partial c(\mathbf{x}, t)}{\partial t} = \frac{\alpha^x v^x}{R} \frac{\partial^2 c(\mathbf{x}, t)}{\partial x^2} + \frac{\alpha^y v^x}{R} \frac{\partial^2 c(\mathbf{x}, t)}{\partial y^2} - \frac{v^x}{R} \frac{\partial c(\mathbf{x}, t)}{\partial x} - \frac{v^y}{R} \frac{\partial c(\mathbf{x}, t)}{\partial y} \quad \text{in } \Omega \tag{83}$$

where Ω is an anisotropic two-dimensional domain, with boundary $\partial\Omega$.

When α^x, α^y, v^x and v^y are fixed parameters, Eq. (83) describes a deterministic problem. Its solution requires solving for the concentration field:

$$c : (\mathbf{x}, t) \in \Omega \times [t_0, t_f] \mapsto c(\mathbf{x}, t) \in \mathcal{R}. \tag{84}$$

We consider non-homogeneous Dirichlet conditions and null Neumann boundary conditions over the respective portions Γ_d and Γ_n of the domain boundary $\partial\Omega = \Gamma_d \cup \Gamma_n$, i.e., $c(\mathbf{x}, \cdot) = c_d(\mathbf{x}), \quad \mathbf{x} \in \Gamma_d = \partial\Omega$ and $\Gamma_n = \emptyset$, where $c_d(x, y)$ is a given function on Γ_d.

When α^x, α^y, v^x and v^y are spatially dependent (i.e., for inhomogeneous domains), there is no exact solution for Eq. (83). Hence, we reformulate the problem in its weak or variational form so we may apply approximation methods to compute its solution.

Let $\mathcal{V}$ be the set of functionals on Ω such that

$$\mathcal{V} = \left\{ c \in H_0^1(\Omega) : \ c = c_d(\mathbf{x}) \ \text{ on } \Gamma_d \right\} \tag{85}$$

where $H_0^1(\Omega)$ with $\Gamma_n = \emptyset$ is the Sobolev space of square-integrable functionals whose first-order derivatives are also square-integrable. Thus, the problem above can be expressed in the following variational form: "Find $c \in \mathcal{V}$ such that

$$a(c, w) = 0 \quad \forall w \in \mathcal{V} \tag{86}$$

where

$$\begin{aligned} a(c, w) &= \int_\Omega w(\mathbf{x}) \frac{\partial c(\mathbf{x}, t)}{\partial t} \mathrm{d}\mathbf{x} + \frac{1}{R} \int_\Omega \left(\alpha^x v^x \frac{\partial w(\mathbf{x})}{\partial x} \frac{\partial c(\mathbf{x}, t)}{\partial x} + \alpha^y v^x \frac{\partial w(\mathbf{x})}{\partial y} \frac{\partial c(\mathbf{x}, t)}{\partial y} \right) \mathrm{d}\mathbf{x} \\ &- \frac{1}{R} \int_\Omega \left(v^x w(\mathbf{x}) \frac{\partial c(\mathbf{x}, t)}{\partial x} + v^y w(\mathbf{x}) \frac{\partial c(\mathbf{x}, t)}{\partial y} \right) \mathrm{d}\mathbf{x} \end{aligned} \tag{87}$$

and $w(\mathbf{x}) \in \mathcal{V}$ is a test (or weighting) function."

To solve the variational form of the problem, we apply Galerkin finite-element discretization involving $\mathcal{T} = \{\Omega_k^e\}_{k=1}^\eta$, a triangulation of Ω with η non-overlapping triangular elements Ω_k^e. Locally on each element Ω_k^e, we assume the finite element approximation c^h of the functional $c \in \mathcal{V}$ is linear.

Let $\mathcal{N}$ is the set of nodes of the finite-element mesh which are not lying on Γ_d. Denote $\mathcal{V}^h$ as the finite-element approximation of $\mathcal{V}$ from Eq. (85):

$$\mathcal{V}^h = \mathrm{span}\{\Phi_i\}_{i \in \mathcal{N}}. \tag{88}$$

The trial function c^h and the test function w^h are discretized accordingly:

$$\begin{aligned} c^h(\mathbf{x}, t) &= \sum_{i \in \mathcal{N}} c_i(t) \Phi_i(\mathbf{x}) \in \mathcal{V}^h \\ w^h(\mathbf{x}, t) &= \sum_{j \in \mathcal{N}} w_j(t) \Phi_j(\mathbf{x}) \in \mathcal{V}^h \end{aligned} \tag{89}$$

where $\Phi_i(\mathbf{x})$ is the corresponding shape function associated with the nodes, and $c_i(t)$ is the transient nodal values at nodal point i.

c and w can now be approximated by their respective finite-element counterparts, c^h and w^h. Substituting c^h and w^h into Eqs. (86-87) yields a finite-element approximation of the variational form of the problem:

$$\sum_{i,j \in \mathcal{N}} (m_{ij} \dot{c}_i + a_{ij} c_i) w_j = 0 \tag{90}$$

where

$$m_{ij} = \int_{\Omega} \Phi_i(\mathbf{x}) \Phi_j(\mathbf{x}) \, \mathrm{d}\mathbf{x} \tag{91}$$

and

$$\begin{aligned} a_{ij} &= \frac{1}{R} \int_{\Omega} \left(\alpha^x v^x \frac{\partial \Phi_i(\mathbf{x})}{\partial x} \frac{\partial \Phi_j(\mathbf{x})}{\partial x} + \alpha^y v^x \frac{\partial \Phi_i(\mathbf{x})}{\partial y} \frac{\partial \Phi_j(\mathbf{x})}{\partial y} \right) \mathrm{d}\mathbf{x} \\ &- \frac{1}{R} \int_{\Omega} \left(v^x \frac{\partial \Phi_i(\mathbf{x})}{\partial x} \Phi_j(\mathbf{x}) + v^y \frac{\partial \Phi_i(\mathbf{x})}{\partial y} \Phi_j(\mathbf{x}) \right) \mathrm{d}\mathbf{x}. \end{aligned} \tag{92}$$

The single term from Eq. (91) can be rewritten as the *element matrix* $\mathbf{M}^e$, and the two terms from Eq. (92) can also be rewritten as *element matrices* $\mathbf{K}_1^e$ and $\mathbf{K}_2^e$ respectively (see 8.1 for details). Subsequently, these element matrices can be assembled into the (full-mesh) *global matrices* $\mathbf{M}$, $\mathbf{K}_1$ and $\mathbf{K}_2$.

Using an appropriate indexing of the nodes in $\mathcal{N}$, the finite-element approximation of the variational form (cf. Eqs. (90-92)) can be rewritten as a set of linear equations for the set of transient nodal values c_i of c^h, as follows:

$$\mathbf{M}\dot{\mathbf{c}} + \mathbf{K}\mathbf{c} = 0 \tag{93}$$

where $\mathbf{K} = \mathbf{K}_1 - \mathbf{K}_2$, and the dimension of $\mathbf{c}$ equates to the cardinality of $\mathcal{N}$ (i.e., $\dim(\mathbf{c}) = |\mathcal{N}|$). Here, $\dim(\mathbf{c})$ represents the number of unknowns or degrees of freedom in the finite-element problem.

To solve Eq. (93), we apply the backward Euler method for time integration:

$$(\mathbf{M} + \mathbf{K}\Delta t)\, \mathbf{c}^{t+1/2} = \mathbf{M}\mathbf{c}^t. \tag{94}$$

The global matrices $\mathbf{M}$ and $\mathbf{K}$, individually of size $(|\mathcal{N}| \times |\mathcal{N}|)$ (because they were assembled from the element matrices), are very sparse and can be stored compactly. For example, since we used linear triangular elements in the assembly process, we were able to store these global matrices as $(|\mathcal{N}| \times 7|)$ matrices.

So far, we have described how to go about solving the deterministic problem (cf. Eq. (83)), in which α^x, α^y, v^x and v^y are assumed fixed. However, in practical applications, α^x, α^y, v^x and v^y have uncertainties. Thus, we proceed to describe how to solve the stochastic version of this same problem, where α^x, α^y, v^x and v^y are assumed to be independent second-order random variables.

By definition, α^x, α^y, v^x and v^y are functions of random event θ in an abstract probability space (Θ, Σ, P):

$$\alpha^x = \alpha^x(\theta); \quad \alpha^y = \alpha^y(\theta); \quad v^x = v^x(\theta); \quad v^y = v^y(\theta). \tag{95}$$

As a result, the solution (i.e., the concentration field c) is also random and satisfies almost surely the stochastic problem as follows:

$$\begin{aligned} \frac{\partial c(\mathbf{x}, \theta, t)}{\partial t} &= \frac{\alpha^x(\theta) v^x}{R} \frac{\partial^2 c(\mathbf{x}, \theta, t)}{\partial x^2} + \frac{\alpha^y(\theta) v^x}{R} \frac{\partial^2 c(\mathbf{x}, \theta, t)}{\partial y^2} \\ &- \frac{v^x(\theta)}{R} \frac{\partial c(\mathbf{x}, \theta, t)}{\partial x} - \frac{v^y(\theta)}{R} \frac{\partial c(\mathbf{x}, \theta, t)}{\partial y} \quad \mathbf{x} \in \Omega \end{aligned} \tag{96}$$

such that $c(\mathbf{x}, \theta, t) = c_d(\mathbf{x})$, $\mathbf{x} \in \Gamma_d = \partial\Omega$. The random solution $c(\mathbf{x}, \theta, t)$ lies in the space $\mathcal{V} \otimes L_2(\Theta, P)$, where $\otimes$ denotes the Cartesian product, and:

$$c(\cdot, \theta, t) \in \mathcal{V}, \quad c(x, \cdot, t) \in L_2(\Theta, P) \tag{97}$$

Here, the deterministic space for the random solution is $\mathcal{V}$ (from Eq. (85)) and the stochastic space is $L_2(\Theta, P)$, the set of second-order random variables.

The *stochastic* variational formulation of the problem is to find $c \in \mathcal{V} \otimes L_2(\Theta, P)$ such that:

$$A(c, w) = \mathrm{E}[a(c, w)] = \int_{\Theta} a(c, w) \, \mathrm{d}P(\theta), \quad \forall w(\mathbf{x}, \theta) \in \mathcal{V} \otimes L_2(\Theta, P) \tag{98}$$

by taking the expectation of a (cf. Eq. (87)), where $w(\mathbf{x},\theta) \in \mathcal{V} \otimes L_2(\Theta, P)$ is a test (or weighting) function of random event θ.

Recall the deterministic finite-element space $\mathcal{V}^h$ defined in Eq. (88). Similarly, the *semi*-discrete stochastic solution lies in:

$$c^h(\mathbf{x},\theta,t) = \sum_{i\in\mathcal{N}} c_i(\theta,t)\,\Phi_i(\mathbf{x}) \in \left(\mathcal{V}^h \otimes L_2(\Theta,P)\right) \tag{99}$$

(The solution is semi-discrete because we have not yet discretized the stochastic space $L_2(\Theta, P)$.) Next, we apply finite-element approximation to the stochastic variational form of the problem, derived from taking the expectation of the finite-element approximation of the deterministic variational form given in Eq. (90). This involves solving for a set of N transient random variables $c_i(\theta,t)$ that satisfy Eq. (99) for $\forall w_i(\theta) \in L_2(\Theta,P)\,,\ i \in \mathcal{N}$. This is equivalent to solving for $c_i(\theta,t)$ that satisfy:

$$\sum_{i,j\in\mathcal{N}} \mathrm{E}\left[\left(m_{ij}\dot{c}_i(\theta,t) + A_{ij}(\theta)\,c_i(\theta,t)\right) w_j(\theta)\right] = 0 \tag{100}$$

where

$$A_{ij}(\theta) \triangleq a_{ij}(\theta) \tag{101}$$

based on Eqs. (95) and (97).

To apply PCE-based Galerkin projection (to the finite-element approximation of the stochastic variational form), we introduce the "polynomial chaos space" $\mathcal{W}^Q$ onto which we will be projecting the stochastic space $L_2(\Theta, P)$:

$$\mathcal{W}^Q \equiv \mathrm{span}\{\Psi_0,\ldots,\Psi_Q\} \subset L_2(\Theta,P) \tag{102}$$

where $\{\Psi_k\}_{k=0}^Q$ are polynomial chaoses of some prespecified order with its type and domain depending on the distributions imposed by the independent second-order random variables α^x, α^y, v^x and v^y. The number of polynomials in this multi-dimensional expansion is denoted by $(Q+1)$, which is defined in Eq. (6). Thus, α^x, α^y, v^x and v^y can now be approximated by the following truncated $(Q+1)$-term PCE:

$$\begin{aligned}
\alpha^x(\boldsymbol{\xi},\theta) &= \sum_{i=0}^{Q} \alpha_i^x \Psi_i(\boldsymbol{\xi}(\theta)) & \alpha^y(\boldsymbol{\xi},\theta) &= \sum_{i=0}^{Q} \alpha_i^y \Psi_i(\boldsymbol{\xi}(\theta)) \\
v^x(\boldsymbol{\xi},\theta) &= \sum_{i=0}^{Q} v_i^x \Psi_i(\boldsymbol{\xi}(\theta)) & v^y(\boldsymbol{\xi},\theta) &= \sum_{i=0}^{Q} v_i^y \Psi_i(\boldsymbol{\xi}(\theta))
\end{aligned} \tag{103}$$

Analogously, the solution $c^h(\mathbf{x},\boldsymbol{\xi},t)$ can be approximated by substituting the corresponding truncated PCE to Eq. (99):

$$c^h(\mathbf{x},\boldsymbol{\xi},t) = \sum_{i\in\mathcal{N}} \left(\sum_{k=0}^{Q} c_{i,k}(t)\,\Psi_k(\boldsymbol{\xi})\right) \Phi_i(\mathbf{x}) \in \left(\mathcal{V}^h \otimes \mathcal{W}^Q\right). \tag{104}$$

Similarly, the same can be done for the test functions $w^h(\mathbf{x},\boldsymbol{\xi},t)$:

$$w^h(\mathbf{x},\boldsymbol{\xi},t) = \sum_{i\in\mathcal{N}} \left(\sum_{k=0}^{Q} w_{i,k}(t)\,\Psi_k(\boldsymbol{\xi})\right) \Phi_i(\mathbf{x}) \in \left(\mathcal{V}^h \otimes \mathcal{W}^Q\right). \tag{105}$$

At this point, we can express the semi-discrete stochastic variational formulation in terms of the expansions for α^x, α^y, v^x, v^y, c^h and w^h.

Next, we apply stochastic Galerkin projection, which involves solving for the coefficients

$$\{c_{ik}\}_{i\in\mathcal{N},0\le k\le\kappa} \quad \forall\{w_{jk}\}_{j\in\mathcal{N},0\le k\le\kappa} \tag{106}$$

that satisfy:

$$
\begin{aligned}
0 \quad = \quad & \sum_{i\in\mathcal{N}}\sum_{k=0}^{Q}\left[\int_{\Omega}\Phi_i\left(\mathbf{x}\right)\Phi_j\left(\mathbf{x}\right)\mathrm{d}\mathbf{x}\right]\dot{c}_{ik}w_{jk} + \frac{1}{R}\Bigg\{\sum_{i,j\in\mathcal{N}} \\
+\Bigg[& \sum_{s,k,l,m=0}^{Q}\alpha_s^x v_k^x\left\langle\Psi_s\Psi_k\Psi_l\Psi_m\right\rangle\left(\int_{\Omega}\frac{\partial\Phi_i\left(\mathbf{x}\right)}{\partial x}\frac{\partial\Phi_j\left(\mathbf{x}\right)}{\partial x}\mathrm{d}\mathbf{x}\right)c_{il}w_{jm} \\
+ & \sum_{s,k,l,m=0}^{Q}\alpha_s^y v_k^x\left\langle\Psi_s\Psi_k\Psi_l\Psi_m\right\rangle\left(\int_{\Omega}\frac{\partial\Phi_i\left(\mathbf{x}\right)}{\partial y}\frac{\partial\Phi_j\left(\mathbf{x}\right)}{\partial y}\mathrm{d}\mathbf{x}\right)c_{il}w_{jm} \\
- & \sum_{k,l,m=0}^{Q}v_k^x\left\langle\Psi_k\Psi_l\Psi_m\right\rangle\left(\int_{\Omega}\frac{\partial\Phi_i\left(\mathbf{x}\right)}{\partial x}\Phi_j\left(\mathbf{x}\right)\mathrm{d}\mathbf{x}\right)c_{il}w_{jm} \\
- & \sum_{k,l,m=0}^{Q}v_k^y\left\langle\Psi_k\Psi_l\Psi_m\right\rangle\left(\int_{\Omega}\frac{\partial\Phi_i\left(\mathbf{x}\right)}{\partial y}\Phi_j\left(\mathbf{x}\right)\mathrm{d}\mathbf{x}\right)\Bigg]c_{il}w_{jm}\Bigg\}.
\end{aligned}
\tag{107}
$$

which was derived by expanding Eq. (100) by applying Eqs. (91), (101), (103), and (104).

Due to the orthogonality of the stochastic expansion bases $\{\Psi\}$, Eq. (107) can be simplified and rewritten in terms of the *stochastic global matrices* $[\mathbf{M}]$ and $[\mathbf{K}]$:

$$
[\mathbf{M}]\left[\dot{\mathbf{c}}\right] + [\mathbf{K}]\left[\mathbf{c}\right] = 0 \tag{108}
$$

In essence, Eq. (108) is the stochastic version of Eq. (93). Due to the stochastic Galerkin projection, we now have a specific instance of Eq. (93) per each of the $(Q+1)$ stochastic modes that resulted from projection onto the stochastic expansion bases.

Thus, the stochastic global matrices $[\mathbf{M}]$ and $[\mathbf{K}]$ are constructed from the deterministic global matrices $\mathbf{M}$ and $\mathbf{K}$ (from Eq. (100)) as follows:

$$
[\mathbf{M}] = \begin{pmatrix} \mathbf{M} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{M} \end{pmatrix}, \quad [\mathbf{K}] = \begin{pmatrix} \mathbf{K}_{0,0} & \dots & \mathbf{K}_{0,Q} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{Q,0} & \cdots & \mathbf{K}_{Q,Q} \end{pmatrix}. \tag{109}
$$

Moreover, the solution to Eq. (108) (i.e., the stochastic concentration field) is:

$$
[\mathbf{c}] = \begin{pmatrix} \mathbf{c}_0 & \dots & \mathbf{c}_Q \end{pmatrix}^{\mathbf{T}} \tag{110}
$$

where $\mathbf{c}_k = \begin{pmatrix} c_{1,k} & \dots & c_{|\mathcal{N}|,k} \end{pmatrix}^{\mathbf{T}}$, $0 \le k \le \kappa$, denotes the vector of nodal values of the k^{th} stochastic mode of the solution.

To explain in more details the construction of $\mathbf{M}$ and $\mathbf{K}_{kl}$, we show Eq. (108) in its expanded form:

$$
\begin{pmatrix} \mathbf{M} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{M} \end{pmatrix}\begin{pmatrix} \dot{\mathbf{c}}_0 \\ \vdots \\ \dot{\mathbf{c}}_Q \end{pmatrix} + \begin{pmatrix} \mathbf{K}_{0,0} & \dots & \mathbf{K}_{0,Q} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{Q,0} & \cdots & \mathbf{K}_{Q,Q} \end{pmatrix}\begin{pmatrix} \mathbf{c}_0 \\ \vdots \\ \mathbf{c}_Q \end{pmatrix} = 0 \tag{111}
$$

Let $\mathbf{K}_{kl} = \mathbf{K}_{kl}^1 - \mathbf{K}_{kl}^2$ for $0 \le k, l \le \kappa$. The construction of $\mathbf{M}$, $\mathbf{K}_{kl}^1$ and $\mathbf{K}_{kl}^2$ is analogous to that in Section 6.1.1; the only difference is that these matrices are now assembled from *stochastic element matrices* instead of their deterministic counterparts.

Using the backward Euler method, we can discretize Eq. (111) in time and rewrite it as:

$$
\begin{pmatrix} \mathbf{M}+\mathbf{K}_{0,0}\Delta t & \dots & \mathbf{K}_{0,Q}\Delta t \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{Q,0}\Delta t & \cdots & \mathbf{M}+\mathbf{K}_{Q,Q}\Delta t \end{pmatrix}\begin{pmatrix} \mathbf{c}_0 \\ \vdots \\ \mathbf{c}_Q \end{pmatrix}^{t+1/2} = \begin{pmatrix} \mathbf{M} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{M} \end{pmatrix}\begin{pmatrix} \mathbf{c}_0 \\ \vdots \\ \mathbf{c}_Q \end{pmatrix}^{t}. \tag{112}
$$

Assuming uniform probability distributions for the random variables $\{\alpha^x, \alpha^y, v^x, v^y\}$, we can exploit the orthogonality of Legendre polynomial chaos to approximate the mean and variance of the concentration at each grid point i by:

$$\bar{c}_i\left(t+\frac{1}{2}\right) = c_{i,0}^{t+\frac{1}{2}} \tag{113}$$

$$\sigma_i^2\left(t+\frac{1}{2}\right) = \sum_{k=1}^{Q}\left(c_{ik}^{t+\frac{1}{2}}\right)^2 \tag{114}$$

In summary, after applying stochastic Galerkin projection to our stochastic problem, we arrive at Eq. (112). Eq. (112) represents a set of $(Q+1)$ coupled deterministic systems of partial differential equations (PDEs) to be solved by preconditioned Krylov methods Golub and Loan [57]. The number of PDE systems, $(Q+1) = (p+m)!/(p!m!)$, is determined by the number of random variables m and the desired order p of the polynomials. Each such PDE system is defined over a grid consisting of $|\mathcal{N}|$ points. Thus, the total number of algebraic equations encapsulated in Eq. (112) is $(p+m)!/(p!m!) \times |\mathcal{N}|$.

6.1.2. Sampling-based Methods for Reaction Module

While the transport system (cf. Eq. (81)) can be solved independently per species, the reaction system (cf. Eq. (82)) is coupled so it must be solved in its vectorized form. Reproduced here for completeness, the reaction problem is to apply $\mathbf{c}^{t+\frac{1}{2}}$ (the transport solution) as the initial condition for solving $\mathbf{c}^{t+1}$ in:

$$\frac{\partial \mathbf{c}^{t+1}}{\partial t} = \mathbf{A}\mathbf{c}^{t+1} \tag{115}$$

where $\mathbf{A}$ depends on the structure of the reaction network, which for our specific TCE biodegradation problem, takes on the form of Eq. (80). The random variables under consideration are the first-order reaction rates $\{k_i\}_{i=1}^4$ that comprise $\mathbf{A}$.

The reaction module is implemented using non-intrusive UQ methods. Assuming each k_i is uniformly distributed on a prespecified range, we apply Latin hypercube or sparse grid sampling to derive an ensemble of sample points, then compute $\mathbf{c}^{t+1}$ for each sample point analytically.

Following [54], the analytic procedure starts with a decomposition of $\mathbf{A}$:

$$\mathbf{A} = \mathbf{S}\Lambda\mathbf{S}^{-1} \tag{116}$$

where Λ is a diagonal matrix with diagonal $\begin{pmatrix} -k_1 & -k_2 & -k_3 & -k_4 \end{pmatrix}$,

$$\mathbf{S} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{R_1}{R_2}\frac{l_1 k_1}{k_2-k_1} & 1 & 0 & 0 \\ \frac{R_1}{R_3}\prod_{i=2}^{3}\frac{l_{i-1}k_{i-1}}{k_i-k_1} & \frac{R_2}{R_3}\frac{l_2 k_2}{k_3-k_2} & 1 & 0 \\ \frac{R_1}{R_4}\prod_{i=2}^{4}\frac{l_{i-1}k_{i-1}}{k_i-k_1} & \frac{R_2}{R_4}\prod_{i=3}^{4}\frac{l_{i-1}k_{i-1}}{k_i-k_2} & \frac{R_3}{R_4}\frac{l_3 k_3}{k_4-k_3} & 1 \end{pmatrix}, \tag{117}$$

and

$$\mathbf{S}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{R_1}{R_2}\frac{l_1 k_1}{k_1-k_2} & 1 & 0 & 0 \\ \frac{R_1}{R_3}\prod_{i=1}^{2}\frac{l_i k_i}{k_i-k_3} & \frac{R_2}{R_3}\frac{l_2 k_2}{k_2-k_3} & 1 & 0 \\ \frac{R_1}{R_4}\prod_{i=1}^{3}\frac{l_i k_i}{k_i-k_4} & \frac{R_2}{R_4}\prod_{i=2}^{3}\frac{l_i k_i}{k_i-k_4} & \frac{R_3}{R_4}\frac{l_3 k_3}{k_3-k_4} & 1 \end{pmatrix}. \tag{118}$$

Then, $\mathbf{c}^{t+1}$ is computed as follows:

(i) Compute $\tilde{\mathbf{c}} = \mathbf{S^{-1}}\mathbf{c^{t+\frac{1}{2}}}$, then let $\tilde{\mathbf{c}} = \begin{pmatrix} \tilde{c}_1 & \dots & \tilde{c}_4 \end{pmatrix}^{\mathbf{T}}$.

(ii) For $i = 1$ to 4, compute $c_i^* = \tilde{c}_i \exp(-k_i \Delta t)$, then let $\mathbf{c}^* = \begin{pmatrix} c_1^* & \dots & c_4^* \end{pmatrix}^{\mathbf{T}}$.

(iii) Finally, compute $\mathbf{c}^{t+1} = \mathbf{S}\mathbf{c}^*$.

The analytic solver can be implemented as a black box, since the only requirement is that it be used to evaluate $\mathbf{c}^{t+1}$ for each input sample of $\{k_i\}_{i=1}^4$. Once all sample points have been evaluated, the sample mean and variance for $\mathbf{c}^{t+1}$ can be computed from Eqs. (11-12).

6.1.3. Propagation of Uncertainties through the Transport and Reaction Modules

To propagate uncertainty information between modules, one needs to decide on a common format with which to represent uncertainties as they are passed from one module to another. In our implementation, we have chosen to use PCE coefficients, but samples may also be a viable alternative.

The full reaction-transport system with embedded UQ can now be solved by calling each module sequentially at each time step. At the end of the simulation, the overall uncertainty and sensitivity information can be extracted from the PC coefficients as prescribed in Section 3. The PC formulation for the transport equation needs to be described in the larger context of the full reaction-transport system. The reason for this requirement is due to the fact that, at the beginning of each time step, the incoming species concentrations encompass uncertainties from both the transport and reaction systems from previous time steps. As such, the transport module needs the knowledge of all uncertain parameters in the entire system, not just the ones solely pertaining to transport. But nonetheless, the task of managing the global propogation of uncertainties through transport module can be still handled elegantly so that the transport solver can be developed independently from the reaction module (as validated by the decomposition property given in Section 4.4). The decomposition and recombination of the global PCE information inside the stochastic transport module are achieved through the restriction and prolongation operators available through a generic library of utility functions provided as part of our modularly hybrid UQ framework.

At each time step, we iterate between the transport solve and the reaction solve. Since the transport module is implemented using PCE-based method, its outputs are already expressed in the common PCE format, as $(Q+1)$ coefficients for each grid point. To propagate these coefficients through the reaction module, an ensemble of at least $(Q+1)$ sample points is required as input into the reaction module. The sample generation is performed by the framework interface so that the reaction solver does not need the knowledge of all parameters in the system. The sample points are then evaluated, analyzed and converted back to the global PC coefficients by polynomial regression.

Here, we quickly summarize the implementation details for the modularly hybrid framework as applied to the reaction-transport problem. We assume uniform distributions for all random variables under consideration (i.e., α^x, α^y, v^x, v^y and $\{k_i\}_{i=1}^4$), as well as linear triangular elements in our approximations.

The objective is to solve Eq. (112). To do so, we need to form both the "left hand side" and "right hand side" matrices in that equation before applying the equation to a linear equation solver. This equation needs to be instantiated and solved for each species.

1. Enumerate the permutations corresponding to the mapping of the multi-dimensional form of the PC expansion to a single-index form. The resulting number of permutations is equivalent to $(Q+1)$, which is the number of required polynomials for the stochastic Galerkin projection.

2. Compute the PCE representations of α^x, α^y, v^x and v^y in terms of random variables $\boldsymbol{\xi}$.

3. For each (k,l) of the $(Q+1)\times(Q+1)$ blocks:

   (a) Compute the effective dispersion coefficients D^x_{kl} and D^x_{kl}:

   $$D^x_{kl} = \sum_{i=0}^{Q}\sum_{j=0}^{Q} \alpha^x_i v^x_j e_{ijkl} \qquad D^y_{kl} = \sum_{i=0}^{Q}\sum_{j=0}^{Q} \alpha^y_i v^x_j e_{ijkl}. \tag{119}$$

   (b) Compute the effective velocities V^x_{kl} and V^y_{kl}:

   $$V^x_{kl} = \sum_{i=0}^{Q} v^x_i e_{ikl} \qquad V^y_{kl} = \sum_{i=0}^{Q} v^y_i e_{ikl} \tag{120}$$

   where $e_{skl} = \langle \Psi_s \Psi_k \Psi_l \rangle$, $0 \le s,k,l \le Q$, and $e_{smkl} = \langle \Psi_s \Psi_m \Psi_k \Psi_l \rangle$, $0 \le s,m,k,l \le Q$.

4. Form the stochastic element matrices $\mathbf{M}^e$, $\mathbf{K}^{1e}_{kl}$, and $\mathbf{K}^{2e}_{kl}$ by applying Eqs. (169), (176) and (177) respectively.

5. Form $\mathbf{M}$, $\mathbf{K}^1_{kl}$, and $\mathbf{K}^2_{kl}$ by block assembly of $\mathbf{M}^e$, $\mathbf{K}^{1e}_{kl}$, and $\mathbf{K}^{2e}_{kl}$, respectively. Compute $\mathbf{K}_{kl} = \mathbf{K}^1_{kl} - \mathbf{K}^2_{kl}$ for $0 \le k,l \le Q$.

6. Let $[\mathbf{A}]$ denote the "left hand side" matrix of Eq. (112). Form $[\mathbf{A}]$ by block assembly of $\mathbf{M}$ and $\mathbf{K}_{kl}$ as follows:

$$[\mathbf{A}]_{kl} = \begin{cases} \mathbf{M} + \mathbf{K}_{kl}\Delta t & \text{if } k = l \\ \mathbf{K}_{kl}\Delta t & \text{otherwise} \end{cases} \tag{121}$$

7. Denote by $\mathcal{D}$ the set of nodes lying on Γ_d, the portion of the domain boundary with Dirichlet conditions. For $m \in \mathcal{D}$, specify boundary conditions for each $[\mathbf{A}]$ sub-block, denoted by $\mathbf{B} = [\mathbf{A}]_{kl} \in R^{|\mathcal{N}|\times|\mathcal{N}|}$:

$$\begin{cases} \mathbf{B}_{mm} = 1 & \text{if } k = l \\ \mathbf{B}_{mn} = 0 & \text{if } k = l, n \neq m \\ \mathbf{B}_{mn} = 0 & \text{otherwise} \end{cases} \tag{122}$$

8. Let $[\mathbf{b}]$ denote the "right hand side" matrix of Eq. (112). Form $[\mathbf{b}] = \begin{pmatrix} \mathbf{b}_0 & \dots & \mathbf{b}_Q \end{pmatrix}^{\mathbf{T}}$ where $\mathbf{b}_k = \mathbf{M}\mathbf{c}^t$ for $0 \leq k \leq Q$, and $\mathbf{c}^t$ contains the computed concentrations from the previous time step.

9. For $l \in \mathcal{D}$, we specify boundary conditions for $[\mathbf{b}]$:

$$\begin{cases} b_{l,k} = (c_d)_l & \text{if } k = 0 \\ b_{l,k} = 0 & \text{otherwise} \end{cases}$$

   where $(c_d)_l$ is the l^{th} nodal value of the given function $c_d(\mathbf{x})$ on Γ_d.

10. Finally, we compute $\mathbf{c}^{t+1/2}$ by solving the spectral linear system:

$$[\mathbf{A}]\,\mathbf{c}^{t+1/2} = [\mathbf{b}]$$

    where $\mathbf{c}_k^{t+1/2} = \begin{pmatrix} c_{1,k}^{t+1/2} & \dots & c_{|\mathcal{N}|,k}^{t+1/2} \end{pmatrix}^{\mathbf{T}}$ for $0 \leq k \leq Q$.

The objective is to solve Eq. (115) using $\mathbf{c}^{t+1/2}$ as the initial condition. We proceed as follows:

1. Generate an ensemble of $(Q+1)$ or more samples of $\{k_i\}_{i=1}^4$.
2. Convert $\mathbf{c}^{t+1/2}$ from the PCE format to the "ensemble" format. Recall that:

$$c_j^{t+1/2}(\boldsymbol{\xi}) = \sum_{i=0}^{Q} c_j^{t+1/2}\Psi_i(\boldsymbol{\xi})$$

   for each grid point j. Hence, the PC coefficients $c_j^{t+1/2}$ for each grid point are used with the sample coordinates $\boldsymbol{\xi}$ (generated by Latin hypercube and optionally sparse sampling in our implementation) to compute $c_j^{t+1/2}(\boldsymbol{\xi})$.

3. Evaluate each sample point using the analytical formulae given in Section 6.1.2 with $c_j^{t+1/2}(\boldsymbol{\xi})$ as the initial condition.
4. Convert the sample outputs $c_j^{t+1}(\boldsymbol{\xi})$ back to PC coefficients using Legendre regression.

To validate our modularly hybrid UQ approach, we applied the framework to a reaction-transport system with sequential first-order reactions involving four species. We used a two-dimensional domain of size 50 meters by 30 meters, with 50 evenly spaced elements in x-direction and 30 evenly spaced elements in y-direction. Initial concentrations for all four species were assumed to be zero throughout the computational domain. A Gaussian boundary concentration $c(0, y, t) = \exp\left(-(y-15)^2/45\right)$ was imposed at the inlet of x-direction. The retardation coefficients and yield coefficients in the reaction matrix (cf. Eq. (80)) are treated as deterministic and are set to $\mathbf{R} = \begin{pmatrix} 2.9, & 2.8, & 1.4, & 5.3 \end{pmatrix}$, and $\mathbf{l} = \begin{pmatrix} 0.7927 & 0.7385 & 0.6458 & 0.4516 \end{pmatrix}$. In this test scenario, we have $n_x = 51$ grid points in the x-direction and $n_y = 31$ grid points in the y-direction; the time step is set to $\Delta t = 1$ (day). We used the HYPRE iterative solver package [58] for solving the matrix equations required to compute the deterministic and stochastic finite-element solutions.

We assumed that the velocity in x-direction was dominant over the y-direction, so the randomness in the y-direction was negligible and $v^y = 0$. Subsequently, we have $\{\alpha^x, \alpha^y, v^x, k_1, k_2, k_3, k_4\}$ as our random variables, which we imposed as uniformly distributed with ranges given in Table 2. Each of these variables are mapped onto the Legendre interval $[-1, 1]$. We denote this transformed variable as ξ. For example, $v^x \sim \mathrm{U}([0.4, 0.6])$, so setting $\gamma_v^x = 0.5$ and $\beta_v^x = 0.1$ would ensure that ξ is mapped back to v^x via $v^x(\boldsymbol{\xi}) = \gamma_v^x + \beta_v^x \xi$. Since there are actually six independent variables (α^y is α^x scaled), there are also six transformed variables $\boldsymbol{\xi} = \{\xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_6\}$.

Table 2: Uncertain parameters

Parameter	Minimum	Maximum	Distribution	Scale	Randomness
v^x (m d^{-1})	0.4	0.6	Uniform	Linear	ξ_1
α^x (m)	8.0	12.0	Uniform	Linear	ξ_2
$\alpha^y = 0.1\alpha^x$ (m)	0.8	1.2	Uniform	Linear	ξ_2
k_1 (day^{-1})	0.04	0.06	Uniform	Linear	ξ_3
k_2 (day^{-1})	0.024	0.036	Uniform	Linear	ξ_4
k_3 (day^{-1})	0.016	0.024	Uniform	Linear	ξ_5
k_4 (day^{-1})	0.004	0.006	Uniform	Linear	ξ_6

As basis for comparison, we approximated the ground truth by quasi-Monte Carlo sampling (sample size of 1000) using a non-intrusive UQ software called PSUADE [59]. We studied the convergence of our method with respect to the polynomial order by examining the concentration mean and standard deviation at $c\,(25, 15)$, which corresponds to the output at the center of the physical domain. We observed from Fig. 3 that the polynomial order $p = 3$ provides reasonable accuracy for this problem.

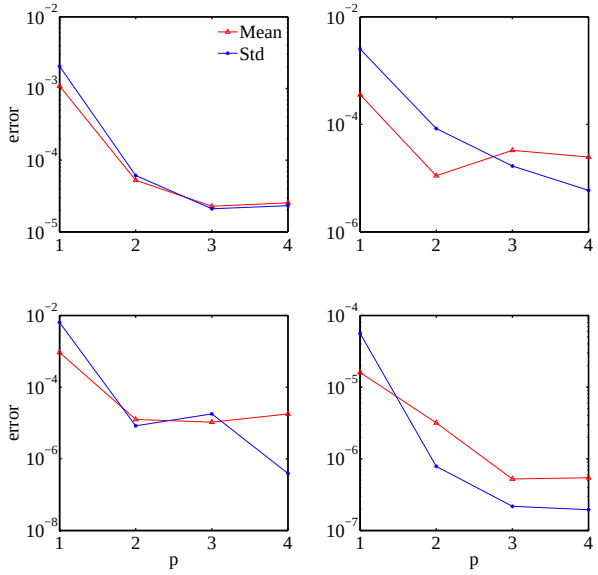


Figure 3: Errors of concentration mean and standard deviation computed by the modularly hybrid UQ method compared to pure sampling using 1000 samples. Concentrations for species 1, 2, 3 and 4 are shown at top left, top right, bottom left and bottom right respectively. All concentrations shown are taken from the center of the physical domain at $t = 30$ days.

For subsequent experiments, we used the "optimal" polynomial order of $p = 3$ for the PCE implementation. Figs. 4 and 5 show the computed means μ and standard deviations σ for the species concentrations at $t = 30$ days. Fig. 4 shows, for each species, the uncertainty bands of $\mu \pm \sigma$ along the center line $y = 15$.

Fig. 5 shows μ and σ over the entire two-dimensional domain. Lastly, we compare these results against the ground truth (approximated by sampling) in Fig. 6.

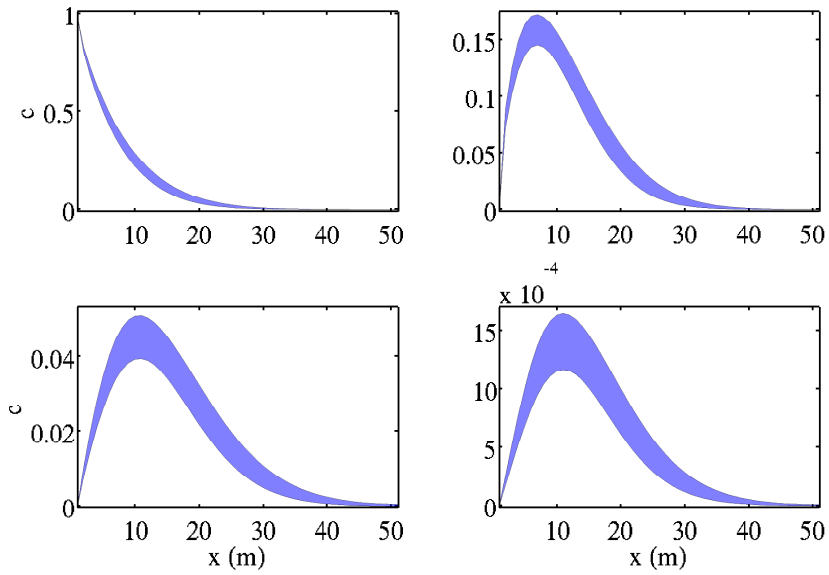
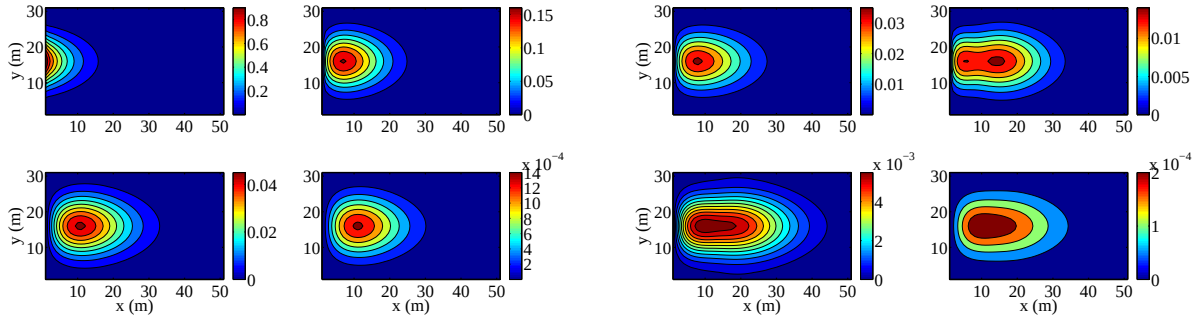


Figure 4: Uncertainty bands ($\mu \pm \sigma$) derived from the concentration mean μ and the standard deviation σ computed by the modular hybrid UQ method using $p = 3$ for the PCE order. Concentrations for species 1, 2, 3 and 4 are shown at top left, top right, bottom left and bottom right respectively. All concentrations shown are taken from the center line $y = 15$ of the physical domain at $t = 30$ days.



(a) Mean values

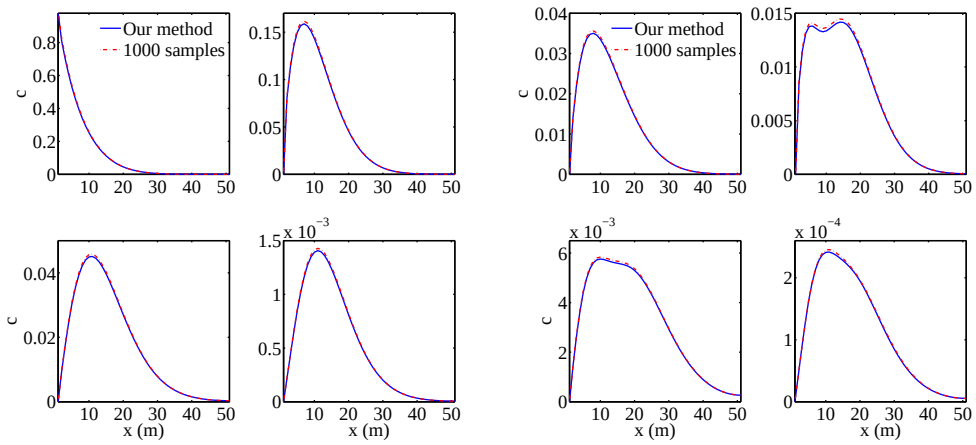
(b) Standard deviations

Figure 5: Contours of the concentration mean μ and the standard deviation σ computed by the modular hybrid UQ method using $p = 3$ for the PCE order. Concentrations for species 1, 2, 3 and 4 are shown at top left, top right, bottom left and bottom right respectively. All concentrations are from $t = 30$ days.

Recall from Section 6.1 that, due to the structure of the sequential network, subsequent species' concentrations depend on incrementally more variables than the precedent species. In particular, species 1 depends on three random variables ($\{v_x, \alpha_x, k_1\}$); species 2 depends on four ($\{v_x, \alpha_x, k_1, k_2\}$); species 3 depends on five ($\{v_x, \alpha_x, k_1, k_2, k_3\}$); and species 4 depends on six ($\{v_x, \alpha_x, k_1, k_2, k_3, k_4\}$).

If one were to apply third-order PCE to the full problem, in order to capture the uncertainties at each grid point, according to Eq. (6):

- species 1 would require $(3 + 3)!/(3!3!) = 20$ terms;



(a) Mean values

(b) Standard deviations

Figure 6: Concentration mean and standard deviation computed by the modularly hybrid UQ method (using $p = 3$ for the PCE order) compared to pure sampling using 1000 samples. Concentrations for species 1, 2, 3 and 4 are shown at top left, top right, bottom left and bottom right respectively. All concentrations shown are taken from the center line $y = 15$ of the physical domain at $t = 30$ days.

- species 2 would require $(4+3)!/(4!3!) = 35$ terms;
- species 3 would require $(5+3)!/(5!3!) = 56$ terms;
- species 4 would require $(6+3)!/(6!3!) = 84$ terms.

Although our method does not *solve* the problem using 20-, 35-, 56- and 84-terms PCE, our method does use this format to *store* the global uncertainty information from both modules. Recall that inference of v_x and α_x takes place in the transport module (via third-order PCE) and inference of $\{k_i\}_{i=1}^4$ takes place in the reaction module (via sampling). To match the incoming 20-, 35-, 56- and 84-terms PCE to the transport module's internal format, the incoming uncertainty information from the reaction module are decomposed into subproblems. Let m_1 denote the number of variables in the transport module, m_2 denote the number of variables in the reaction module, and $m = m_1 + m_2$ denote the number of joint variables. Each species' $((m+p)!/m!p!)$-terms PCE will be decomposed into $((m_2+p)!/m_2!p!)$ subproblems, one for each term in the single-index PC representation of the reaction variables. For example, species 1 has $m_1 = 2$ transport variables and $m_2 = 1$ reaction variable. Its 20-terms PCE will be decomposed into 4 smaller subproblems. The first subproblem corresponds to zeroth order of k_1, which contains $(m_1+p)!/m_1!p! = 10$ terms involving only v_x and α_x. The second subproblem corresponds to first order of k_1, which contains $(m_1+p-1)!/m_1!(p-1)! = 6$ terms. The third subproblem corresponds to second order of k_1, which contains $(m_1+p-2)!/m_1!(p-2)! = 3$ terms. The fourth subproblem corresponds to third order of k_1, which contains $(m_1+p-3)!/m_1!(p-3)! = 1$ term. Similarly:

- species 2's 35-terms PCE will be decomposed into 10 subproblems (one containing 10-terms, two 6-terms, three 3-terms, and four 1-term);
- species 3's 56-terms PCE into 20 subproblems (one containing 10-terms, three 6-terms, six 3-terms, and ten 1-term);
- species 4's 84-terms PCE into 35 subproblems (one containing 10-terms, six 6-terms, three 3-terms, and twenty 1-term).

These subproblems are then solved by the transport module, then the results are subsequently re-packaged back into the 20-, 35-, 56- and 84-terms PCE format (for each of four species, respectively) for subsequent

processing by the reaction module. For the reaction module, the incoming uncertainty information (in PCE format) would have 84 coefficients for each grid point. Thus, to propagate and reconstruct the coefficients, a minimum sample size of 84 is required as input into the reaction module.

6.2. Two-dimensional Flow and Reactive Transport in Heterogeneous Porous Media

Let $K(\mathbf{x},\theta)$, the stochastic hydraulic conductivity, be represented as a random process in the space $\Omega \otimes L_2(\Theta, P)$, where $\otimes$ denotes the Cartesian product and $\mathbf{x} = (x, y) \in \Omega$, $\theta \in \Theta$. Applying the logarithmic transformation $Y(\mathbf{x},\theta) = \ln K(\mathbf{x},\theta)$, we arrive at the following stochastic partial differential equations to describe a stochastic flow:

$$\frac{\partial}{\partial x}\left\{\exp\left[Y(\mathbf{x},\theta)\right]\frac{\partial h(\mathbf{x},\theta)}{\partial x}\right\} + \frac{\partial}{\partial y}\left\{\exp\left[Y(\mathbf{x},\theta)\right]\frac{\partial h(\mathbf{x},\theta)}{\partial y}\right\} = 0, \tag{123}$$

where $h(\mathbf{x},\theta)$ is the stochastic hydraulic head, which will be used to compute the stochastic velocity field required by the transport module.

In this section, we describe the steps to compute the stochastic velocity field (i.e., the solution to the stochastic flow equation given in equation (123)) via a doubly-nested dimension reduction scheme that combines the KLE and proper orthogonal decomposition (POD) methods. Section 6.2.1 describes how KLE is used to reduce the infinite-dimensional random field $Y(\mathbf{x},\theta)$ to an approximate finite-dimensional representation. Section 6.2.2 presents a non-intrusive PCE-based method (the implementation of which is greatly facilitated by the modular UQ framework) to solve for the stochastic hydraulic head $h(\mathbf{x},\theta)$ from the stochastic flow equation. Section 6.2.3 describes a POD reduced-order modeling technique based on singular value decomposition (SVD) to generate a low-dimensional stochastic velocity field from the stochastic hydraulic head.

6.2.1. Kahunen-Loeve Expansion (KLE) of Logarithmic Hydraulic Conductivity

For the random process $Y(\mathbf{x},\theta)$ in equation (123), the covariance function [40, 41, 42, 43]:

$$C_Y(\mathbf{x},\mathbf{y}) = \langle Y(\mathbf{x},\theta), Y(\mathbf{y},\theta)\rangle_\Omega \tag{124}$$

is bounded, symmetric and positive definite with $\mathbf{x} = (x_1, y_1) \in \Omega$ and $\mathbf{y} = (x_2, y_2) \in \Omega$ ($\langle\cdot,\cdot\rangle_\Omega$ in equation (124) denotes the inner product in the space Ω). The covariance function can be decomposed into:

$$C_Y(\mathbf{x},\mathbf{y}) = \sum_{n=1}^{\infty} \lambda_n f_n(\mathbf{x}) f_n(\mathbf{y}) \tag{125}$$

where λ_n and $f_n(\mathbf{x})$ are the eigenvalues and eigenfunctions, respectively. Here, $f_n(\mathbf{x})$ are the orthogonal and deterministic functions that form a complete set such that:

$$\int_\Omega f_n(\mathbf{x}) f_m(\mathbf{x})\, d\mathbf{x} = \delta_{nm}, \quad n, m \geq 1$$

where δ_{nm} is the Kronecker product.

The random process $Y(\mathbf{x},\theta)$ can be expressed via KLE as:

$$Y(\mathbf{x},\theta) = \bar{Y}(\mathbf{x}) + \sum_{n=1}^{\infty} \sqrt{\lambda_n} f_n(\mathbf{x}) \xi_n(\theta) \tag{126}$$

where $\bar{Y}(\mathbf{x})$ is the mean of the stochastic process $Y(\mathbf{x},\theta)$, and $\xi_n(\theta)$ are orthogonal zero-mean random variables. Formally, $\langle \xi_n(\theta)\rangle_{L_2(\Theta,P)} = 0$, and $\langle \xi_n(\theta), \xi_m(\theta)\rangle_{L_2(\Theta,P)} = \delta_{nm}$, where $\langle\cdot,\cdot\rangle_{L_2(\Theta,P)}$ denotes the inner product in the space $L_2(\Theta, P)$.

Eigenvalues and eigenfunctions of the covariance function $C_Y(\mathbf{x},\mathbf{y})$ can be solved from the following Fredholm equation:

$$\int_\Omega C_Y(\mathbf{x},\mathbf{y}) f(\mathbf{x})\, d\mathbf{x} = \lambda f(\mathbf{y}). \tag{127}$$

Table 3: Ranges of physical parameters in validation experiments

Physical Parameters	Case I	Case II	Physical Parameters	Case I	Case II
v^x (m d^{-1})	$0.2 \pm \epsilon$	$[0.1, 0.3]$	k_1	$0.05 \pm \epsilon$	$[0.04, 0.06]$
v^y (m d^{-1})	0.0	0.0	k_2	$0.03 \pm \epsilon$	$[0.024, 0.036]$
α^x (m)	$1.25 \pm \epsilon$	$[1.0, 1.5]$	k_3	$0.02 \pm \epsilon$	$[0.016, 0.024]$
α^y (m)	0.0	0.0	k_4	$0.005 \pm \epsilon$	$[0.004, 0.006]$

For example, consider a one-dimensional stochastic process with the following covariance function:

$$C_Y(x_1, x_2) = \sigma_Y^2 \exp\left(-|x_1 - x_2|/\zeta\right) \tag{128}$$

where x and y are scalars, and σ_Y^2 and ζ are the variance and correlation length of the random process. The eigenvalues λ_n and eigenfunctions $f_n(\mathbf{x}, \mathbf{y})$ can be solved analytically [40]. For problems in two dimensions, we consider a separable covariance function:

$$C_Y(\mathbf{x}, \mathbf{y}) = \sigma_Y^2 \exp\left(-|x_1 - x_2|/\zeta_1 - |y_1 - y_2|/\zeta_2\right) \tag{129}$$

in a domain $\Omega = \{(x, y) : 0 \le x \le L_1; 0 \le y \le L_2\}$, and the eigenvalues and eigenfunctions can be obtained by combining those from the one-dimensional formulations. We can truncate the KLE to a finite number of terms by inspecting the spectral decay rate of λ_n. The higher the rate of spectral decay is, the smaller the number of terms is needed in the truncated KLE. In fact, the rate of spectral decay depends on the correlation function of the stochastic process. The more correlated the process is, the higher is the rate of spectral decay and fewer terms are needed in the truncated KLE to account for the same fraction of the total variance.

6.2.2. Non-intrusive PC-based Stochastic Solution of Flow Equation

By substituting the truncated ($\mathcal{N}$-term) KLE of $Y(\mathbf{x}, \theta)$ (assuming $\bar{Y}(\mathbf{x}) = 0$ and omitting θ in $\xi_n(\theta)$) into the stochastic flow equation, we obtain:

$$\begin{aligned} &\frac{\partial}{\partial x}\left[\exp\left(\sum_{n=1}^{\mathcal{N}} \sqrt{\lambda_n} f_n(\mathbf{x})\, \xi_n\right) \frac{\partial h(\mathbf{x}, \theta)}{\partial x}\right] \\ &+ \frac{\partial}{\partial y}\left[\exp\left(\sum_{n=1}^{\mathcal{N}} \sqrt{\lambda_n} f_n(\mathbf{x})\, \xi_n\right) \frac{\partial h(\mathbf{x}, \theta)}{\partial y}\right] = 0. \end{aligned} \tag{130}$$

Next, we expand $h(\mathbf{x}, \theta)$ in terms of PCE:

$$h(\mathbf{x}, \theta) = \sum_{i=0}^{Q} h_i(\mathbf{x})\, \Psi_i(\boldsymbol{\xi}) \tag{131}$$

where Q is defined in equation (6), and $\boldsymbol{\xi}$ is the vector of random variables from equation (130):

$$\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_{\mathcal{N}})^{\mathbf{T}}. \tag{132}$$

To solve the stochastic flow equation, we first generate a set of the $N \ge Q + 1$ sample points, that is:

$$\boldsymbol{\xi}_l = (\xi_{1l}, \xi_{2l}, \ldots, \xi_{\mathcal{N}l})^{\mathbf{T}}, \quad l = 1, \ldots, N. \tag{133}$$

For each sample point $\boldsymbol{\xi}_l$, we solve the following deterministic equation:

$$\begin{aligned} &\frac{\partial}{\partial x}\left\{\exp\left[\sum_{n=1}^{\mathcal{N}} \sqrt{\lambda_n} f_n(\mathbf{x})\, \xi_{nl}\right] \frac{\partial h_l(\mathbf{x})}{\partial x}\right\} \\ &+ \frac{\partial}{\partial y}\left\{\exp\left[\sum_{n=1}^{\mathcal{N}} \sqrt{\lambda_n} f_n(\mathbf{x})\, \xi_{nl}\right] \frac{\partial h_l(\mathbf{x})}{\partial y}\right\} = 0 \end{aligned} \tag{134}$$

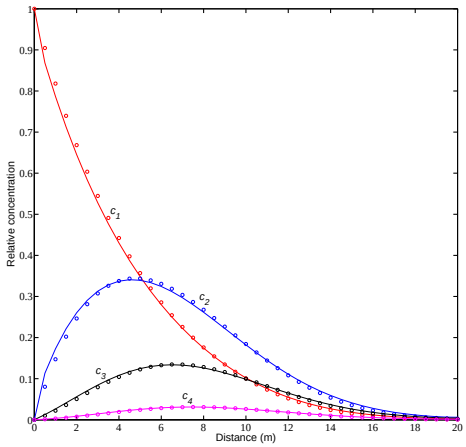


Figure 7: Comparison of relative concentrations at $y = 10$ m and $t = 40$ d. The relative concentration means computed from the modular UQ method with polynomial order of $p = 3$ are shown as solid lines, and those computed from the analytical procedure are shown as circles. For the modular UQ method, parameters correspond to Case I in Table 3.

by applying a Galerkin finite-element discretization approach.

Let $\mathbf{H}(\mathbf{x}) = [h(\mathbf{x}, \xi_1), h(\mathbf{x}, \xi_2), \ldots, h(\mathbf{x}, \xi_{\mathcal{N}})]^{\mathbf{T}}$ be the set of hydraulic heads obtained by solving the corresponding $\mathcal{N}$ deterministic equations above. The objective is to compute the PCE coefficients for the hydraulic head $h(\mathbf{x}, \theta)$, that is, $[h_0(\mathbf{x}), h_1(\mathbf{x}), \ldots, h_Q(\mathbf{x})]^{\mathbf{T}}$, such that:

$$h_l(\mathbf{x}, \theta_l) = \sum_{i=0}^{Q} h_i(\mathbf{x}) \Psi_i(\boldsymbol{\xi}_l), \quad l = 1, \ldots, \mathcal{N}. \tag{135}$$

In matrix form, equation (135) can be rewritten as:

$$\mathbf{Z}\mathbf{h}(\mathbf{x}) = \mathbf{H}(\mathbf{x}) \tag{136}$$

where $\mathbf{Z} = (z_{li})_{\mathcal{N} \times (Q+1)}$ with $z_{li} = \Psi_i(\boldsymbol{\xi}_l)$ consisting of Legendre polynomials evaluated at the sample points $\boldsymbol{\xi}_l$, $l = 1, \ldots, \mathcal{N}$. Since $\mathbf{Z}$ is either a square matrix or an overdetermined matrix, $\mathbf{h}(\mathbf{x})$ can be solved in a least-square sense by:

$$\mathbf{h}(\mathbf{x}) = \left(\mathbf{Z}^{\mathbf{T}}\mathbf{Z}\right)^{-1} \mathbf{Z}^{\mathbf{T}}\mathbf{H}(\mathbf{x}). \tag{137}$$

The sample generation and the computation of $\mathbf{h}(\mathbf{x})$ are automatically handled by the non-intrusive module in the modular UQ framework.

6.2.3. Generation of Stochastic Velocity Field

The stochastic velocity can now be computed using the KLE of hydraulic conductivity and the stochastic hydraulic head. The respective x- and y-direction velocities are:

$$\begin{aligned} v^x(\mathbf{x}, \theta) &= \frac{K(\mathbf{x}, \theta)}{\phi} \frac{\partial h(\mathbf{x}, \theta)}{\partial x} \\ &= \frac{1}{\phi} \exp\left[\sum_{n=1}^{m} \sqrt{\lambda_n} f_n(\mathbf{x}) \xi_n\right] \frac{\partial h(\mathbf{x}, \boldsymbol{\xi})}{\partial x} \\ &= \frac{1}{\phi} \exp\left[\sum_{n=1}^{m} \sqrt{\lambda_n} f_n(\mathbf{x}) \xi_n\right] \sum_{i=0}^{Q} \frac{\partial h_i(\mathbf{x})}{\partial x} \Psi_i(\boldsymbol{\xi}) \end{aligned} \tag{138}$$

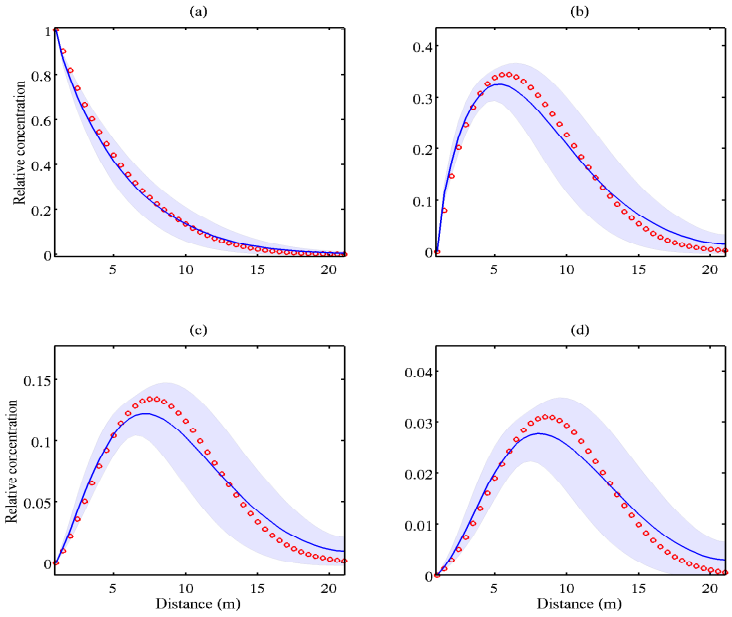


Figure 8: Uncertainty bands ($\mu \pm \sigma$) showing the relative concentration means (solid lines) and the standard deviations (blue bands) computed by the modular UQ method with polynomial order of $p = 3$ for $y = 10$ m. Exact solutions (circles) are shown for completeness. (a), (b), (c), and (d) correspond to species 1, 2, 3, 4, respectively. For the modular UQ method, parameters are from Case II in Table 3.

$$\begin{aligned} v^y(\mathbf{x},\theta) &= \frac{K(\mathbf{x},\theta)}{\phi}\frac{\partial h(\mathbf{x},\theta)}{\partial y} \\ &= \frac{1}{\phi}\exp\left[\sum_{n=1}^{m}\sqrt{\lambda_n}f_n(\mathbf{x})\xi_n\right]\sum_{i=0}^{Q}\frac{\partial h_i(\mathbf{x})}{\partial y}\Psi_i(\boldsymbol{\xi}) \end{aligned} \tag{139}$$

where ϕ is the porosity, which is assumed to be a known constant. Therefore, the overall velocity field is given by:

$$\mathbf{v}(\mathbf{x},\theta) = \left(v^x\left[\mathbf{x},\boldsymbol{\xi}(\theta)\right], v^y\left[\mathbf{x},\boldsymbol{\xi}(\theta)\right]\right)^{\mathbf{T}}. \tag{140}$$

In equation (140), the bi-orthogonality structure of the KLE representation of $K(\mathbf{x},\theta)$ is destroyed in $\mathbf{v}(\mathbf{x},\theta)$. This introduces high stochastic dimension and thus computational inefficiency in the solution of the stochastic transport equation. To mitigate this situation, we use a numerical covariance quadrature method to approximate the analytical covariance function (the method of snapshots [60]), then apply singular value decomposition. To do so, we generate a set of samples $\{\theta\}_{i=1}^{N}$ in the stochastic space Θ and solve equations (138) and (139) at each sample point θ_i to compute $\mathbf{v}_s = \{\mathbf{v}(\mathbf{x},\theta_i)\}_{i=1}^{N}$, where $N \geq Q+1$. Subsequently, the numerical covariance matrix can be constructed from $(\mathbf{v}_s - \overline{\mathbf{v}}_s)^{\mathbf{T}}(\mathbf{v}_s - \overline{\mathbf{v}}_s)$, where $\overline{\mathbf{v}}_s$ is the sample mean of $\mathbf{v}_s$.

It can be shown that the eigenvalue decomposition of the covariance matrix is equivalent to applying SVD [57] to $(\mathbf{v}_s - \overline{\mathbf{v}}_s)$:

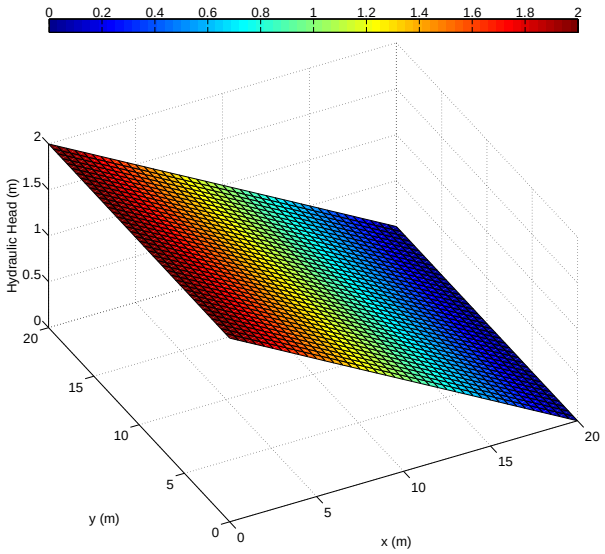


Figure 9: Hydraulic head mean computed by a PCE-based non-intrusive method using least squares fit with polynomial order of $p = 3$.

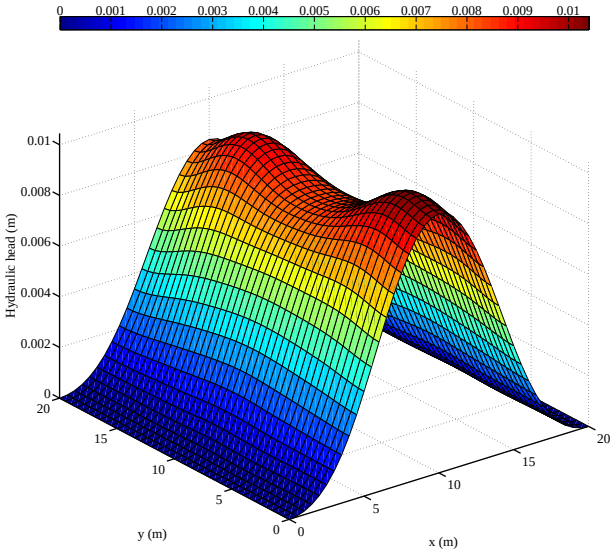


Figure 10: Hydraulic head variance computed by a PCE-based non-intrusive method using least squares fit with polynomial order of $p = 3$.

$$\mathbf{v}_s - \overline{\mathbf{v}}_s = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathbf{T}} = \underbrace{\begin{pmatrix} f_1 & \cdots & f_r & \cdots & f_{|\mathcal{N}|} \end{pmatrix}_{\aleph\times\aleph}}_{\mathbf{U}} \underbrace{\begin{pmatrix} s_1 & & & & 0 \\ & \ddots & & & \\ & & s_r & & \\ & & & \ddots & \\ & & & & s_N \\ 0 & & \cdots & & 0 \end{pmatrix}_{\aleph\times N}}_{\mathbf{S}} \underbrace{\begin{pmatrix} g_1 & \cdots & g_N \end{pmatrix}^{\mathbf{T}}_{N\times N}}_{\mathbf{V}^{\mathbf{T}}}$$

where $\overline{\mathbf{v}}_s = f_0(\mathbf{x})$ is the sample mean of $\mathbf{v}_s$ representing the average velocity field; r is the number of truncated singular values; and $\aleph$ is the number of nodes on the finite-element mesh that are not lying on $\partial\Omega$. Let $f_n = (f_{n1}, \ldots, f_{n\aleph})^{\mathbf{T}}$, and $g_n = (g_{n1}, \ldots, g_{nN})^{\mathbf{T}}$. Then, $\mathbf{v}_s$ can now be represented by an optimal rank-r approximation using the eigen-velocity field $f_n(\mathbf{x})$:

$$\mathbf{v}_s(\mathbf{x}) = \overline{\mathbf{v}}_s(\mathbf{x}) + \sum_{n=1}^{r} s_n f_n(\mathbf{x}) g_n^{\mathbf{T}}$$

where $f_n(\mathbf{x}) = (f_n^x(\mathbf{x}), f_n^y(\mathbf{x}))$. If instead of using the samples $g_n^{\mathbf{T}}$, we consider the sampled variables themselves (denote the first r of those variables as $\boldsymbol{\eta} = \{\eta_n\}_{n=1}^r$), then we have the general (non-sample-based) expression for the velocity field:

$$\mathbf{v}(\mathbf{x}, \theta) = f_0(\mathbf{x}) + \sum_{n=1}^{r} s_n f_n(\mathbf{x}) \eta_n \tag{141}$$

Thus:

$$\begin{aligned} v^x(\mathbf{x}, \theta) &= f_0^x(\mathbf{x}) + \sum_{n=1}^{r} s_n f_n^x(\mathbf{x}) \eta_n \\ v^y(\mathbf{x}, \theta) &= f_0^y(\mathbf{x}) + \sum_{n=1}^{r} s_n f_n^y(\mathbf{x}) \eta_n. \end{aligned} \tag{142}$$

Note that $v^x(\boldsymbol{x}, \boldsymbol{\eta})$ and $v^y(\boldsymbol{x}, \boldsymbol{\eta})$ now share a new set of random variables $\boldsymbol{\eta}$, and their ranges can be specified by the variability of their realizations from the coefficient matrix $\mathbf{V}$ computed from SVD. Let $R_{n1} = \min\{g_{nj}\}_{j=1}^N$ and $R_{n2} = \max\{g_{nj}\}_{j=1}^N$, $n = 1, \ldots, r$, and define:

$$R_n = \min\{|R_{n1}|, |R_{n2}|\}, \quad n = 1, \ldots, r. \tag{143}$$

Using R_n, we can normalize the random variables $\{\eta_n\}_{n=1}^r$ by $s_n' = s_n / R_n$ and finally represent the spatially correlated random field $\mathbf{v}(\mathbf{x}, \boldsymbol{\xi})$ in terms of the stochastic expansion of the spatially uncorrelated normalized random variables $\boldsymbol{\eta}$:

$$\mathbf{v}(\mathbf{x}, \theta) = f_0(\mathbf{x}) + \sum_{n=1}^{r} s_n' f_n(\mathbf{x}) \eta_n, \tag{144}$$

which is applied as an (uncertain) input to the transport module (to be discussed in Section 6.2). Here, $\eta_n(\theta)$ follows a standard uniform distribution: $\eta_n(\theta) \sim U[-1, 1]$, $n = 1, \ldots, r$.

In this section, we describe the steps to solve a multi-species reactive transport system using the velocity field obtained in Section 6.2. Here, we apply the modular UQ methodology described in Section 4. In the following, we first introduce the formulation of a two-dimensional reactive transport system in heterogeneous porous media, then describe the stochastic solution of the transport and reaction modules, where the transport system is solved by the PCE-based intrusive method and the reaction system is solved by a sampling-based non-intrusive method. Finally, we show how to "glue" these two independently-implemented modules together using the modular UQ framework to propagate global uncertainties and sensitivities.

6.2.4. PC-based Stochastic Transport

The transport equation can be solved independently for each species. Hence, it suffices to discuss the PCE formulation of one generic species' (scalar) transport equation. Removing the species and time indices, the scalar transport equation becomes:

$$\frac{\partial c}{\partial t} = \frac{1}{R}\frac{\partial}{\partial x}\left(\alpha^x v^x \frac{\partial c}{\partial x}\right) + \frac{1}{R}\frac{\partial}{\partial y}\left(\alpha^y v^x \frac{\partial c}{\partial y}\right) - \frac{1}{R}\frac{\partial\left(v^x c\right)}{\partial x} - \frac{1}{R}\frac{\partial\left(v^y c\right)}{\partial y} \quad \in \Omega \tag{145}$$

where again, Ω is an anisotropic two-dimensional domain with boundary $\partial\Omega$.

When α^x, α^y, v^x and v^y are fixed parameters, equation (145) describes a deterministic problem with the species' concentration field as its solution:

$$c : (\mathbf{x}, t) \in \Omega \times [t_0, t_f] \mapsto c\left(\mathbf{x}, t\right) \in \mathcal{R}. \tag{146}$$

where $c\left(\mathbf{x}, \cdot\right) = c_d\left(\mathbf{x}\right)$, $\mathbf{x} \in \partial\Omega$, and $c_d\left(\mathbf{x}\right)$ is a given function on $\partial\Omega$.

We apply PCE-based stochastic Galerkin projection with the finite-element approximation of the stochastic variational form and we introduce the "polynomial chaos space" $\mathcal{W}^Q$ onto which we project the stochastic space $L_2(\Theta, P)$:

$$\mathcal{W}^Q \equiv \text{span}\left\{\Psi_0, \ldots, \Psi_Q\right\} \subset L_2\left(\Theta, P\right) \tag{147}$$

where $\{\Psi_k\}_{k=0}^Q$ are the polynomial chaoses of some pre-specified order p, and the number of terms $(Q+1)$ is defined in equation (6). After applying stochastic Galerkin projection (Appendix A) to the stochastic problem, we arrive at equation (112), which is a sparse matrix system that can be solved by preconditioned Krylov methods [57].

6.2.5. Propagation of Uncertainties through the Transport and Reaction Modules

While the transport system (equation (81)) can be solved independently for each species, the reaction system (equation (82)) is coupled so it must be solved in its vectorized form. Reproduced here for clarity, the reaction problem is to apply $\mathbf{c}^{t+\frac{1}{2}}$ (the transport solution) as the initial condition for solving $\mathbf{c}^{t+1}$ in:

$$\frac{\partial \mathbf{c}^{t+1}}{\partial t} = \mathbf{A}\mathbf{c}^{t+1} \tag{148}$$

where $\mathbf{A}$ has the form of equation (80). To solve this equation, we follow the analytical procedure in [54]. This equation is solved for each sample of $\{k_i\}_{i=1}^4$, the first-order reaction rates which are the random variables in the reaction module.

In terms of implementation, the reaction solver is "wrapped" around by a non-intrusive module in the modular UQ framework, which provides generic tools for sampling and regression analysis. Thus, the reaction solver is treated as a black box to run an ensemble of sample points drawn from the probability distributions of $\{k_i\}_{i=1}^4$, the results of which are used to construct uncertainty and sensitivity information to be propagated to the next module.

The full reaction-transport system with embedded UQ can now be solved by calling each module sequentially at each time step. At the end of the simulation, the overall uncertainty and sensitivity information can be extracted from the PCE coefficients as prescribed in Section 3. The task of global uncertainty propagation is handled by the modular UQ framework, freeing the user to develop the transport or reaction solver independently. The computational framework to handle the decomposition and recombination of the global PCE information is generic so that it can be applied to other similar "multi-physics" problems.

In summary, we have described the multi-species reactive transport problem as a two-module system in the modular UQ framework, in which the propagation of global uncertainties through the transport module is performed intrusively while the propagation of global uncertainties through the reaction module is performed non-intrusively. By applying the stochastic velocity field from the flow module as input into the transport module, spatial uncertainty associated with the randomly heterogeneous media is propagated through the system.

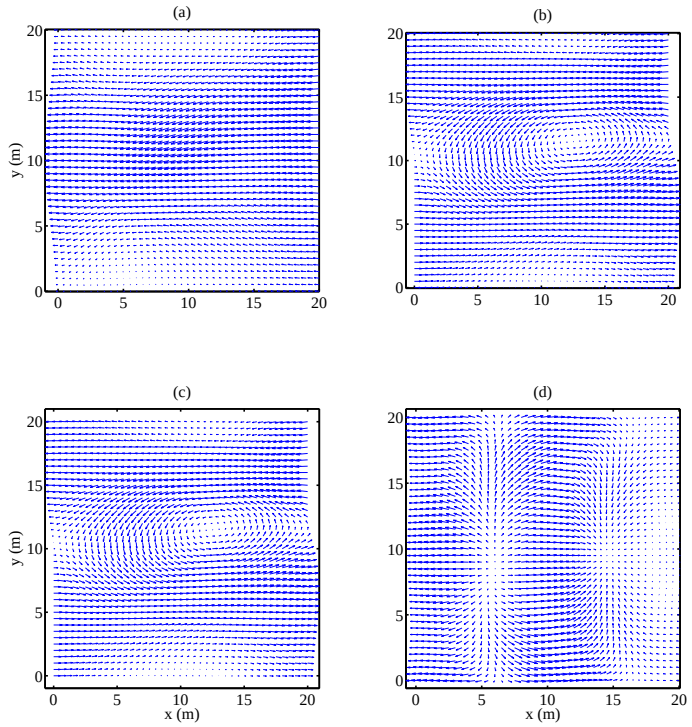


Figure 11: Eigen-velocity fields computed by our doubly-nested dimension reduction scheme. (a), (b), (c), (d) correspond to the first, second, fifth, and seventh mode of the eigen-velocity fields, respectively.

Table 4: Description of physical flow and transport parameters in numerical experiments

Physical Parameters	Definitions	Internal Representations
$v = (v^x, v^y)$ $(\mathrm{m\,d^{-1}})$	$v = f_0(\mathbf{x}) + \sum_{n=1}^{r} \lambda_n' f_n(\mathbf{x})\,\eta_n(\theta)$	$\{\eta_i\}_{i=1}^{r}$
α^x, α^y (m)	$\alpha^y(\xi_1) = 0.1\alpha^x(\xi_1)$	ξ_1
D^x, D^y $(\mathrm{m^2\,d^{-1}})$	$D^x = \alpha^x v^x$, $D^y = \alpha^y v^x$	$\xi_1, \{\eta_i\}_{i=1}^{r}$

Table 5: Description and ranges of physical reaction parameters in numerical experiments

Physical Parameters	Internal Representations	Case I	Case II
k_1 $(\mathrm{day^{-1}})$	ξ_2	$[0.04999, 0.05001]$	$[0.04, 0.06]$
k_2 $(\mathrm{day^{-1}})$	ξ_3	$[0.02999, 0.03001]$	$[0.024, 0.036]$
k_3 $(\mathrm{day^{-1}})$	ξ_4	$[0.01999, 0.02001]$	$[0.016, 0.024]$
k_4 $(\mathrm{day^{-1}})$	ξ_5	$[0.004999, 0.005001]$	$[0.004, 0.006]$

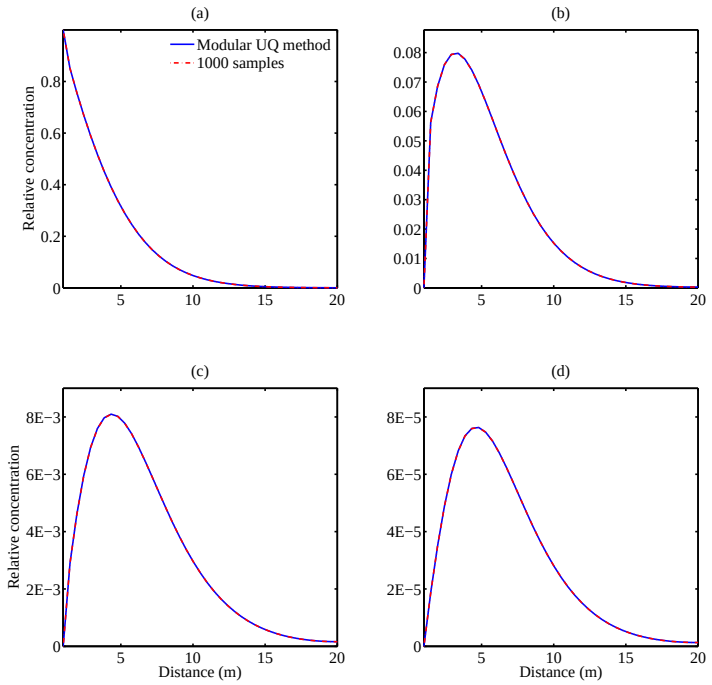


Figure 12: Comparison of relative concentration means at $y = 10$ m and $t = 10$ d. The results computed from the modular UQ method with polynomial order of $p = 3$ are shown as solid lines, and those computed from sampling with 1000 samples are shown as dashed lines. Transport and reaction parameters are from Tables 4 and 5 (Case II). (a), (b), (c), and (d) correspond to species 1, 2, 3, 4, respectively.

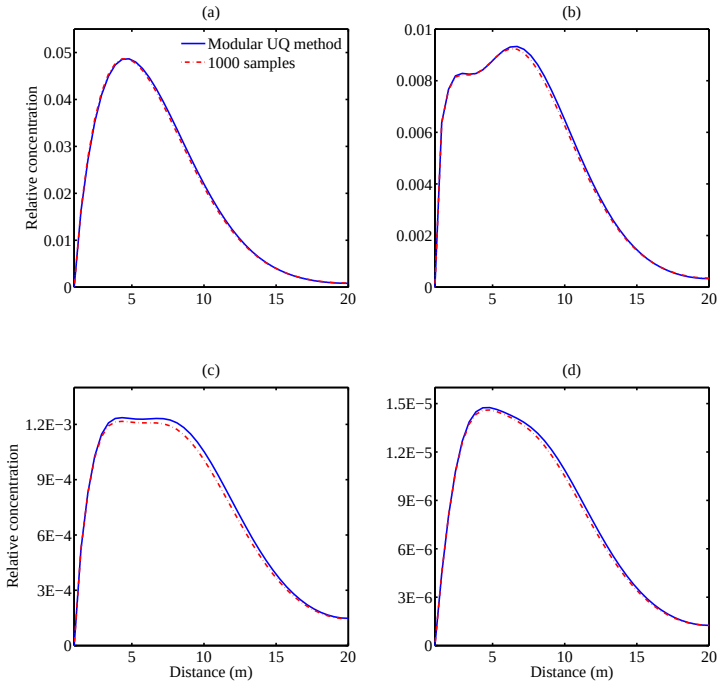


Figure 13: Comparison of relative concentration standard deviations at $y = 10$ m and $t = 10$ d. The results computed from the modular UQ method with polynomial order of $p = 3$ are shown as solid lines, and those computed from sampling with 1000 samples are shown as dashed lines. Transport and reaction parameters are from Tables 4 and 5 (Case II). (a), (b), (c), and (d) correspond to species 1, 2, 3, 4, respectively.

In numerical experiments, we consider a two-dimensional square domain Ω of size $L_1 = L_2 = 20$ (m), with 40 evenly spaced elements in each direction. The time step is set to $\Delta t = 1$ (d). We used a polynomial order of $p = 3$ in the PCE representation which has been verified to give results with sufficient accuracy.

First, we make sure that our previously described two-dimensional methodology agrees well with the analytical solution [54] for one-dimensional multi-species reactive transport model problem. This validation step is important to ensure the methodology does not introduce numerical bias to the solution. In this problem setup, dispersivity is $\alpha = 1.25$ (m) and velocity is $v = 0.2$ (m d^{-1}). The reaction rates for the four species are $k_1 = 0.05$, $k_2 = 0.03$, $k_3 = 0.02$, and $k_4 = 0.005$ (d^{-1}), respectively. The retardation factors and yield coefficients are all assumed to be 1.0. The homogeneous velocity is given by (v^x, v^y) (m d^{-1}), where both v^x and v^y are homogeneous random variables with ranges defined in Table 3. Since we are using a one-dimensional problem to verify the two-dimensional implementation, we set the dispersivity and velocity in the y-direction to zero. The total simulation time is $t = 40$ (days) with $\Delta t = 1$ (day). We used the HYPRE iterative solver package [58] to solve the matrix equations arising from the discretization of the stochastic finite-element problem.

We considered two scenarios: one in which the uncertainty is negligible (Case I) and another with increased uncertainty (Case II). In both scenarios, we compare the analytic solution against our solution at the center line $y = 10$ (m) since the solution is computed for a two-dimensional stochastic domain. For the first scenario, we assumed no variability in dispersivity and velocity (as defined in Case I of Table 3) by setting very small uncertainty ranges to velocity parameters and using a polynomial order of $p = 0$. As shown in Fig. 7, the concentration means computed by the modular UQ method (solid lines) agree well with the exact solutions (circles) given in [53]. For the second scenario, the ranges of dispersivity and velocity are defined in Case II of Table 3. As shown in Fig. 4, the exact solutions (circles) of the concentration profiles lie inside of the uncertainty bands of $\mu \pm \sigma$ computed by our method. Here, the μ lines are the same as the ones from Fig. 7. These two scenarios demonstrate that our method is well validated against analytic methods and that no bias is introduced by our method.

In the next numerical experiment, we consider a two-dimensional four-species flow and reactive transport system in randomly heterogeneous porous media. We first used KLE to reduce the dimension of the logarithmic hydraulic conductivity.We then applied a PCE-based non-intrusive approach (i.e., used sampling to construct the PCE coefficients) to solve the stochastic flow equation. In this setup, no-flow boundary conditions are assumed at $y = 0.0$ m and $y = 20$ m; and constant hydraulic heads are prescribed to be 2.0 m and 0.0 m at $x = 0.0$ m and $x = 20$ m, respectively. The mean of the logarithmic hydraulic conductivity is prescribed as $\overline{Y} = 0$. We used a separable covariance function of the logarithmic hydraulic conductivity $C_Y(\mathbf{x}, \mathbf{y}) = \sigma_Y^2 \exp\left(-|x_1 - x_2|/\zeta_1 - |y_1 - y_2|/\zeta_2\right)$ in the square domain $\Omega = \{(x, y) : 0 \le x \le 20; 0 \le y \le 20\}$, with variance $\sigma_Y^2 = 1.0$ and correlation length $\zeta_1 = \zeta_2 = 1.0$. Figs. 9 and 10 show the computed mean and variance of the hydraulic head, which have been verified by using a large Monte Carlo sample. The stochastic velocity field was thus computed based on the KLE of hydraulic conductivity field and the computed stochastic hydraulic head field, which introduces high stochastic dimension to the velocity field. We then obtain orthogonal eigen-velocity fields by our doubly nested dimension reduction scheme. Fig. 11 displays the first, second, fifth and seventh eigen-velocity field in the two-dimensional domain.

Next, we applied the solution to the stochastic flow equation for solving the reactive transport system (equation 75). Initial concentrations for all four species are assumed to be zero throughout the computational domain. A Gaussian boundary concentration $c(0, y, t) = \exp\left(-(y - 15)^2/45\right)$ is imposed at the inlet in the x-direction. The retardation factors and yield coefficients in the reaction matrix $\mathbf{A}$ (equation (80)) are treated as deterministic and are given by: $\mathbf{R} = (2.9, 2.8, 1.4, 5.3)$ and $\mathbf{l} = (0.7927, 0.7385, 0.6458, 0.4516)$. Results are presented for $t = 10$ (d).

In this experiment, the porosity is specified by a deterministic constant $\phi = 0.3$. The independent *physical* parameters are: $\{v^x, v^y\}$ from the flow module, α^x from the transport module (because $\alpha^y = 0.1\alpha^x$, $D^x = \alpha^x v^x$, and $D^y = \alpha^y v^x$ are dependent random variables), and $\{k_1, k_2, k_3, k_4\}$ from the reaction module. These parameters are assumed to follow uniform distributions and their ranges are given in Table 4 and Table 5. These tables also show how each of these physical parameters are internally represented by the *generic* random variables in the modular UQ framework: v^x, v^y by $\{\eta_i\}_{i=1}^r$, where r is the number of KLE functions taken from the stochastic flow module; α^x is internally represented by ξ_1; D^x, D^y by $\{\eta_i\}_{i=1}^r$ and

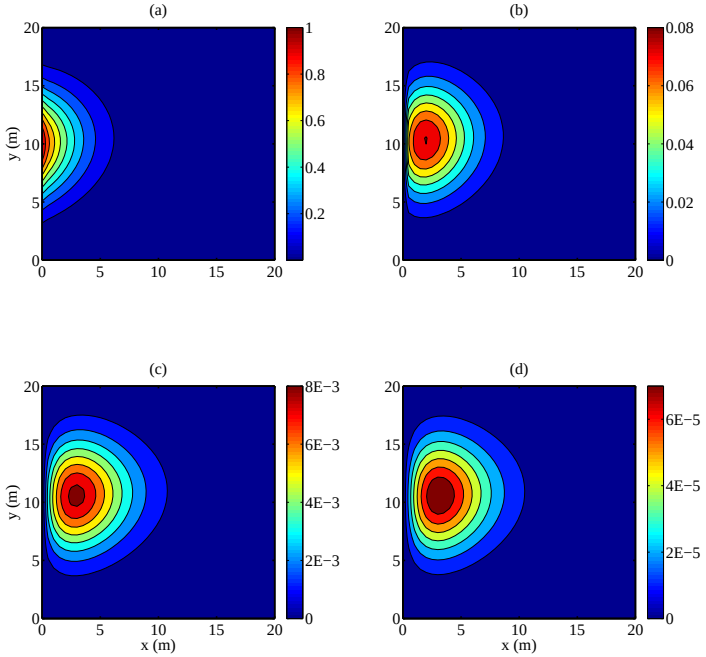


Figure 14: Contours of relative concentration means computed by the modular UQ method with polynomial order of $p = 3$ at $t = 10$ d. Transport and reaction parameters are from Tables 4 and 5 (Case II). (a), (b), (c), and (d) correspond to species 1, 2, 3, 4, respectively.

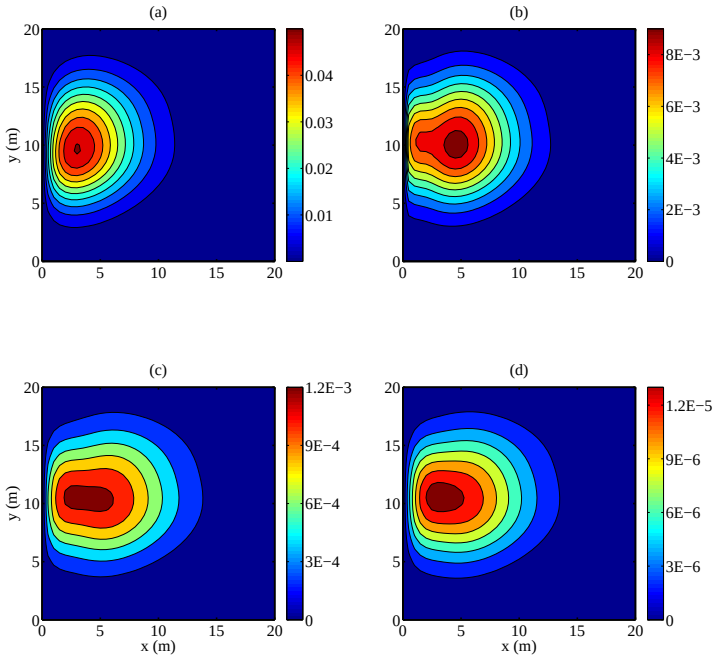


Figure 15: Contours of relative concentration standard deviations computed by the modular UQ method with polynomial order of $p = 3$ at $t = 10$ d. Transport and reaction parameters are from Tables 4 and 5 (Case II). (a), (b), (c), and (d) correspond to species 1, 2, 3, 4, respectively.

ξ_1; and k_i by $\{\xi_i\}_{i=1}^5$. The total number of uncertain parameter is $r+5$ where r is the number of KLE functions taken from the stochastic flow module and 5 is the number of ξ_i's. We use $r=2$, for the reason that, when we varied r from 2 to 7, we noticed only a negligible change in the standard deviations. In sum, the physical uncertain parameters are represented by the random vector $\boldsymbol{\xi} = \{\eta_1, \ldots, \eta_r, \xi_1, \xi_2, \xi_3, \xi_4, \xi_5\}$.

We ran the numerical experiment on an iMac computer (with 3.4 GHz Intel Core i7 processor and 8 GB 1333 MHz DDR3 memories). We compare our results against the "ground truth" results approximated by a large number of samples, as shown in Figs. 12 and 13. We observe that the modular UQ method demonstrates a similar accuracy as the ground truth results. Figs. 14 and 15 show the contours of μ (mean) and σ (standard deviation) over the entire two-dimensional domain, respectively. We also compare the performance of our modular UQ method with a pure sampling-based method (using 1000 samples, which is the minimum required for convergence of statistical moments). The overall CPU run times are 91 (s) and 821 (s), respectively. The speedup of the modular UQ can be attributed to the reuse of the assembled stochastic matrices in the intrusive transport module. Specially, the CPU run times for matrix setup are 15 (s) and 389 (s), respectively, while the CPU run time for linear system solves are 76 (s) and 432 (s), respectively. Further speedup of our modular UQ can be obtained by using more efficient solvers for linear system with multiple right hand sides.

6.3. Predator-prey Model

In this section we demonstrate the use of our hybrid framework in the simulation of a predator-prey model characterized by the following system of Lotka-Volterra equations.

$$\frac{dx}{dt} = x(\alpha - \beta y) \tag{149}$$

$$\frac{dy}{dt} = -y(\gamma - \delta y)$$

where x is the number of preys, y is the number of predators, and $\alpha, \beta, \gamma, \delta$ are parameters describing the interaction of the two species. This system can be solved using the Runge-Kutta method at each time step.

In this example, all $\alpha, \beta, \gamma, \delta$ are uncertain parameters such that $0.95 \leq \alpha \leq 1.05$, $0.19 \leq \beta \leq 0.21$, $0.48 \leq \gamma \leq 0.52$, and $0.038 \leq \delta \leq 0.042$. The equations are solved sequentially in a given time step using polynomial chaos expansions. The main program using our hybrid UQ framework is as follow:

```
int nRVs=4, pOrder=2;
PCEPermutations pcePerms;
PCEDataStreams  VecPD;
ModuleNonLinPCE xModule, yModule;
sparam = "nRVs 4";   xModule.setParam(sparam); yModule.setParam(sparam);
sparam = "pOrder 6"; xModule.setParam(sparam); yModule.setParam(sparam);
sparam = "setRV 1";  xModule.setParam(sparam);
sparam = "setRV 2";  xModule.setParam(sparam);
sparam = "setRV 3";  yModule.setParam(sparam);
sparam = "setRV 4";  yModule.setParam(sparam);
xuserSolver = new UserPCEPrey();
yuserSolver = new UserPCEPredator();
//**/ set up initial conditions
VecPD[0][0] = 20;
VecPD[0][1] = 8;
for (iT = 0; iT < numTimeSteps; iT++)
{
   printf("Lotka-Volterra: time step = %d\n", iT);
   //**/ solve for prey
   xModule.setup(VecPD, xContext);
   xModule.solve(VecPD, xContext);

   //**/ solve for prey
```

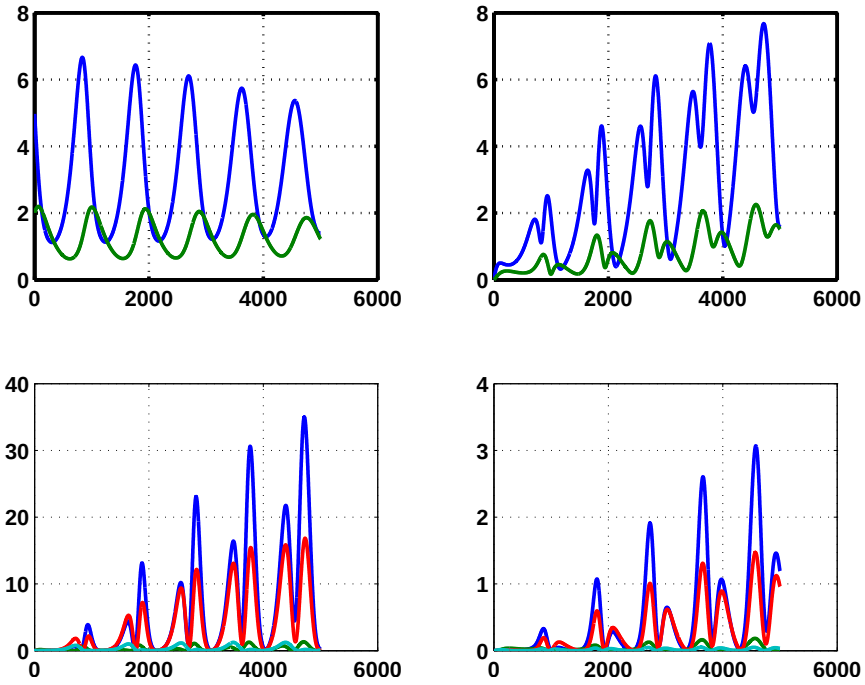


Figure 16: Stochastic Solution of the Lotka-Volterra Equation: (Upper left) Mean (blue) and standard deviation (green) for prey; (Upper right) Mean and standard deviation for predator; (Lower left) Sobol' indices for the 4 random variables for prey; (Lower right) Sobol' indices for the 4 random variables for predator (blue-α, green-β, red-γ, cyan-δ).

```
        yModule.setup(VecPD, dContext);
        yModule.solve(VecPD, dContext);
    }
```

Other than this main program, users need to implement the two PCE modules for prey and predator, respectively. Figure 16 shows the time evolution of the prey and predator population after 5000 time steps.

6.4. Non-linearly Coupled Contaminant Flow

We consider the a reactive-transport model for contaminant flow where x_1 and x_2 are the velocity components of the fluid, x_3, x_4 and x_5 are the respective concentration of the chemical species. The physical domain is defined as $\Omega \equiv (0,\tau)\times[0,1]^2$. The formulation of the entire system is given as follows.

$$\frac{\partial}{\partial t}x_1 + x_1\frac{\partial}{\partial s_1}x_1 + x_2\frac{\partial}{\partial s_2}x_1 = (\kappa_0+\kappa_1x_3+\kappa_2x_4+\kappa_3x_5)\left(\frac{\partial^2}{\partial s_1^2}x_1+\frac{\partial^2}{\partial s_2^2}x_1\right), \tag{150}$$

$$\frac{\partial}{\partial t}x_2 + x_1\frac{\partial}{\partial s_1}x_2 + x_2\frac{\partial}{\partial s_2}x_2 = (\kappa_0+\kappa_1x_3+\kappa_2x_4+\kappa_3x_5)\left(\frac{\partial^2}{\partial s_1^2}x_2+\frac{\partial^2}{\partial s_2^2}x_2\right), \tag{151}$$

$$\frac{\partial}{\partial t}x_3 + x_1\frac{\partial}{\partial s_1}x_3 + x_2\frac{\partial}{\partial s_2}x_3 = -\left(\theta_4x_4^2+\theta_5x_5\right)x_3 + 2\theta_6x_4x_5; \tag{152}$$

$$\frac{\partial}{\partial t}x_4 + x_1\frac{\partial}{\partial s_1}x_4 + x_2\frac{\partial}{\partial s_2}x_4 = -\left(\theta_6x_5+\theta_4x_3x_4\right)x_4 + 2\theta_5x_5x_3; \tag{153}$$

$$\frac{\partial}{\partial t}x_5 + x_1\frac{\partial}{\partial s_1}x_5 + x_2\frac{\partial}{\partial s_2}x_5 = -\left(\theta_5x_3+\theta_6x_4\right)x_5 + 2\theta_4x_3x_4^2. \tag{154}$$

The initial conditions are as follows.

$$x_1(0,s_1,s_2) = \begin{cases}\theta_1v_0(1-4s_1) & s_1<0.25\\ 0 & s_1\geq 0.25\end{cases} \text{ and } x_2(0,s_1,s_2) = \theta_2x_1(0,s_1,s_2)\,s_2^2(1-s_2)^2;$$

$x_3(0,s_1,s_2) = \begin{cases} 1 & s_1 = 0 \\ 0 & s_1 > 0 \end{cases}$, $x_4(0,s_1,s_2) = 0$, and $x_5(0,s_1,s_2) = \theta_3 x_3(0,s_1,s_2)$.

The boundary conditions at $s_2 = 0$ and $s_2 = 1$ are periodic.

At $s_1 = 0$, we have Dirichlet boundary conditions:

$x_1(t,0,s_2) = \theta_1 v_0$, $x_2(t,0,s_2) = \theta_2 x_1(t,0,s_2) s_2^2 (1-s_2)^2$;

$x_3(t,0,s_2) = 1$, $x_4(t,0,s_2) = 0$, $x_5(t,0,s_2) = \theta_3$.

At $s_1 = 1$, we have Neumann boundary conditions:

$\partial_{s_1} x_1(t,1,s_2) = \partial_{s_1} x_2(t,1,s_2) = \partial_{s_1} x_3(t,1,s_2) = \partial_{s_1} x_4(t,1,s_2) = \partial_{s_1} x_5(t,1,s_2) = 0$. The above model is a simplified case of unsteady, multi-phase flows with non-linear interactions between the quantities. We are interested in the mean and variance of x_3. For our analysis, we set $v_0 = 10$, $\kappa_0 = 0.01$, $\kappa_1 = 0.01$, $\kappa_2 = 0.5$. and $\kappa_3 = 0.02$. The 6-dimensional stochastic domain Θ has uniform probability density such that $\theta_1 \sim \mathsf{U}[0.5, 1.5]$, $\theta_2 \sim \mathsf{U}[-0.2, 0.2]$, $\theta_3 \sim \mathsf{U}[0.1, 0.3]$, $\theta_4 \sim \mathsf{U}[0.2, 0.5]$, $\theta_5 \sim \mathsf{U}[0.3, 0.6]$ and $\theta_6 \sim \mathsf{U}[0.1, 0.2]$; We derive a global discrete formulation for the quantities as follows.

$$\left[I + \Delta t \left(T_1\left(X_1^{\ell-1}\right) + T_2\left(X_2^{\ell-1}\right) - T_{11}\left(X_3^{\ell-1}, X_4^{\ell-1}, X_5^{\ell-1}\right) - T_{22}\left(X_3^{\ell-1}, X_4^{\ell-1}, X_5^{\ell-1}\right)\right)\right]; \tag{155}$$

$$X_1^{\ell} = X_1^{\ell-1} \tag{156}$$

$$\left[I + \Delta t \left(T_1\left(X_1^{\ell}\right) + T_2\left(X_2^{\ell-1}\right) - T_{11}\left(X_3^{\ell-1}, X_4^{\ell-1}, X_5^{\ell-1}\right) - T_{22}\left(X_3^{\ell-1}, X_4^{\ell-1}, X_5^{\ell-1}\right)\right)\right]. \tag{157}$$

$$X_2^{\ell} = X_2^{\ell-1} \tag{158}$$

$$\left[I + \Delta t \left(T_1\left(X_1^{\ell}\right) + T_2\left(X_2^{\ell}\right) - R_1\left(X_4^{\ell-1}, X_5^{\ell-1}\right)\right)\right] X_3^{\ell} = X_3^{\ell-1} + \Delta t F_1\left(X_4^{\ell-1}, X_5^{\ell-1}\right); \tag{159}$$

$$\left[I + \Delta t \left(T_1\left(X_1^{\ell}\right) + T_2\left(X_2^{\ell}\right) - R_2\left(X_3^{\ell}, X_4^{\ell-1}, X_5^{\ell-1}\right)\right)\right] X_4^{\ell} = X_4^{\ell-1} + \Delta t F_2\left(X_3^{\ell}, X_5^{\ell-1}\right); \tag{160}$$

and

$$\left[I + \Delta t \left(T_1\left(X_1^{\ell}\right) + T_2\left(X_2^{\ell}\right) - R_3\left(X_3^{\ell}, X_4^{\ell}\right)\right)\right] X_5^{\ell} = X_5^{\ell-1} + \Delta t F_3\left(X_3^{\ell}, X_4^{\ell}\right). \tag{161}$$

with $T_1, T_2, T_{11}, T_{22}, R_1, R_2, R_3, F_1, F_2$ and F_3 defined using the a spatial discretization of the terms in (35)-(39) and associated boundary conditions. The modular formulation is as follows.

- Module 1: (Flow-field)

$$\left[I + \Delta t \left(T_1\left(X_1^{\ell-1}\right) + T_2\left(X_2^{\ell-1}\right) - T_{11}\left(X_3^{\ell-1}, X_4^{\ell-1}, X_5^{\ell-1}\right) - T_{22}\left(X_3^{\ell-1}, X_4^{\ell-1}, X_5^{\ell-1}\right)\right)\right]; \tag{162}$$

$$X_1^{\ell} = X_1^{\ell-1} \tag{163}$$

$$\left[I + \Delta t \left(T_1\left(X_1^{\ell}\right) + T_2\left(X_2^{\ell-1}\right) - T_{11}\left(X_3^{\ell-1}, X_4^{\ell-1}, X_5^{\ell-1}\right) - T_{22}\left(X_3^{\ell-1}, X_4^{\ell-1}, X_5^{\ell-1}\right)\right)\right]; \tag{164}$$

$$X_2^{\ell} = X_2^{\ell-1} \tag{165}$$

- Module 2: (Reaction)

$$\left[I + \Delta t \left(T_1\left(X_1^{\ell}\right) + T_2\left(X_2^{\ell}\right) - R_1\left(X_4^{\ell-1}, X_5^{\ell-1}\right)\right)\right] X_3^{\ell} = X_3^{\ell-1} + \Delta t F_1\left(X_4^{\ell-1}, X_5^{\ell-1}\right); \tag{166}$$

$$\left[I + \Delta t \left(T_1\left(X_1^{\ell}\right) + T_2\left(X_2^{\ell}\right) - R_2\left(X_3^{\ell}, X_4^{\ell-1}, X_5^{\ell-1}\right)\right)\right] X_4^{\ell} = X_4^{\ell-1} + \Delta t F_2\left(X_3^{\ell}, X_5^{\ell-1}\right); \tag{167}$$

$$\left[I + \Delta t \left(T_1\left(X_1^{\ell}\right) + T_2\left(X_2^{\ell}\right) - R_3\left(X_3^{\ell}, X_4^{\ell}\right)\right)\right] X_5^{\ell} = X_5^{\ell-1} + \Delta t F_3\left(X_3^{\ell}, X_4^{\ell}\right). \tag{168}$$

The coupling terms in each module are polynomial with $k = 1$ in module 1 and $k = 2$ in module 2. Therefore, based on Lemma 3, we can implement an intrusive Galerkin method or non-intrusive Least-squares method in either modules. These different combinations of UQ methods are compared against the global intrusive and non-intrusive formulations. The local parameters in module 1 are θ_1 and θ_2, whereas the local parameters in module 2 are θ_3, θ_4, θ_5 and θ_6.

In Figure 17, 18 and 19, we plot a comparison of the results obtained using the various modularly hybrid methods with the global intrusive or non-intrusive methods. Figure 19 shows the convergence history of variance estimates.

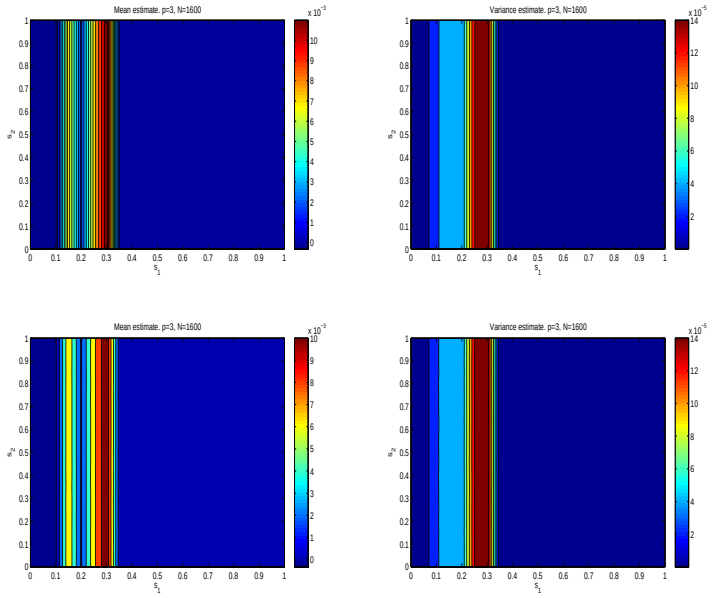


Figure 17: Mean and variance estimates of x_4 obtained using global PC methods. (Top) Global factorized Galerkin method. (Bottom) Global Collocation method.

6.5. Bayesian Inference of Flow and Reactive Transport in Heterogeneous Porous Media

In this study we apply Bayesian inference to the two-dimensional flow and reactive transport example (Equation 123 and 145) with the parameter ranges (assuming all uniform distributions) given by Table 6.

Table 6: Uncertain Parameter description

Parameter	Range	Description
$v^x, v^y (md^{-1})$	$\xi_1 \sim U[-S_1, S_1]$	hydraulic head 1
$v^x, v^y (md^{-1})$	$\xi_2 \sim U[-S_2, S_2]$	hydraulic head 2
$\alpha^x, \alpha^y = 0.1\xi_3(m)$	$\xi_3 \sim U[8, 12]$	dispersivity
k_1 (day^{-1}	$\xi_4 \sim U[0.04, 0.06]$	reaction rate 1
k_2 (day^{-1}	$\xi_5 \sim U[0.024, 0.036]$	reaction rate 2
k_3 (day^{-1}	$\xi_6 \sim U[0.016, 0.024]$	reaction rate 3
k_4 (day^{-1}	$\xi_7 \sim U[0.004, 0.006]$	reaction rate 4

In this scenario we selected two KL modes from the solution of the flow equation. These modes, together with dispersivity and four reaction rates, form the set of uncertain parameteres to be calibrated. The observation data is given in Table 7.

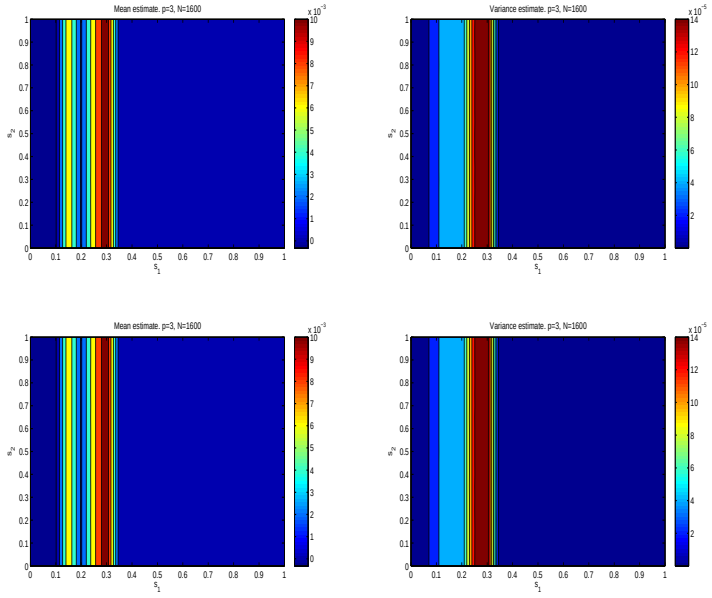


Figure 18: Mean and variance estimates of x_4 obtained using modular UQ methods. The methods are labeled based on the strategy used in the respective modules. (Top) Galerkin+Galerkin. (Bottom) Galerkin+Collocation.

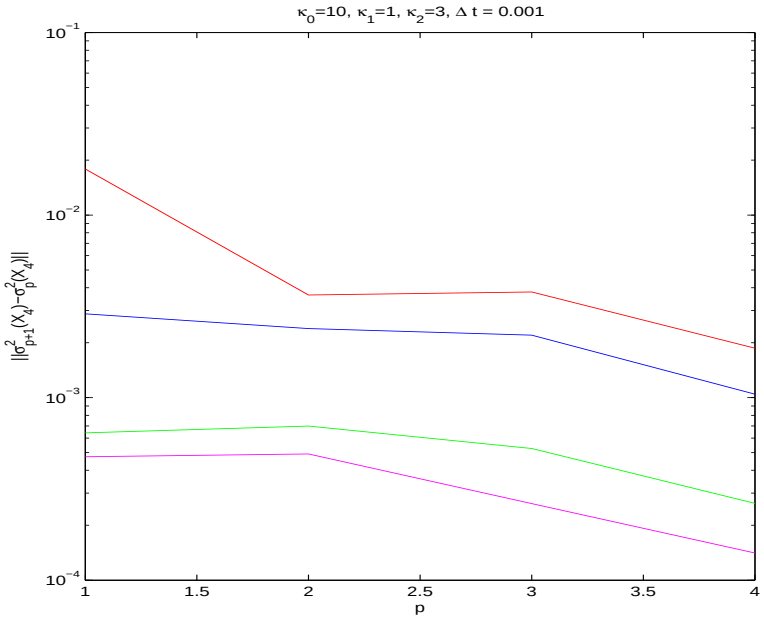


Figure 19: Convergence of variance estimates versus polynomial order of trunctation p.

Table 7: Experimental Observations

Species	Mean at (20,10)	Std. Dev at (20.10)	Mean at (20,30)	Std. Dev at (20,30)
1	$6.436e-2$	$6e-3$	$4.671e-6$	$5e-7$
2	$2.214e-2$	$2e-3$	$2.419e-6$	$2e-7$
3	$4.563e-3$	$4e-4$	$5.661e-6$	$6e-7$
4	$4.807e-5$	$5e-6$	$4.468e-8$	$4e-9$

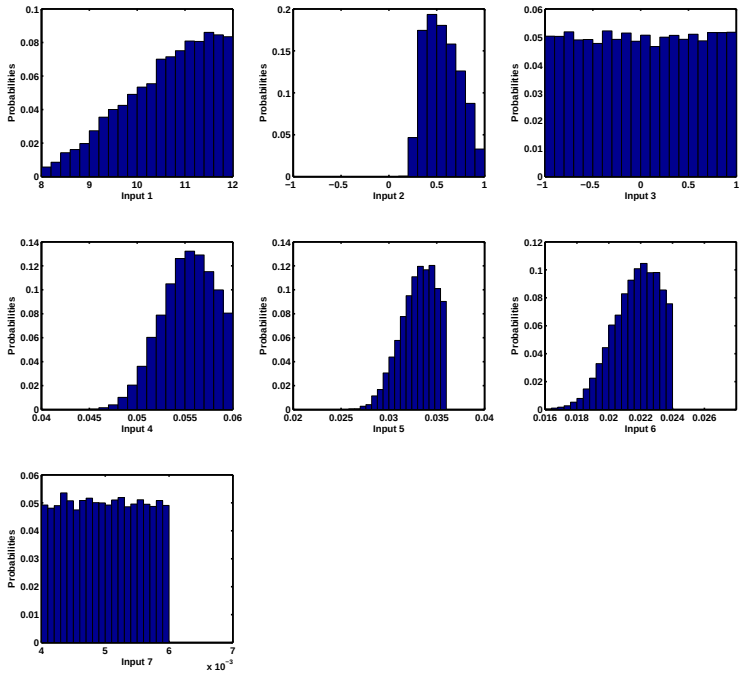


Figure 20: Posterior distributions for the 7 uncertain parameters

The results (posteriors) for the 7 parameters are given in Figure 20. We observe from the posterior plots that the best values of the second KL model (input 2), the first three reaction rates (input 4, 5, 6) lie somewhere inside the prior distributions (which explain the humps), while the first KL mode (input 1) has its best value in its far upper end. Finally, the dispersivity and reaction rate 4 do not seem to be sensitive parameters.

7. Summary and Future Work

This report describes a new methodology to facilitate the propagation of uncertainty and sensitivity through a multi-module simulation model where each module may be equipped with either intrusive, non-intrusive, or semi-intrusive UQ schemes. We have developed the mathematics and a computational framework to support the accurate coupling of these modules with diverse UQ schemes. We have demonstrated this methodology on a few multi-physics multi-physics models.

Our experience with this methodology is that it has given us confidence to be a viable methodology for practical applications. In the following we outline several areas of research and development that will further advance this methodology.

- Investigate parallelization strategies on high performance computers:

Due to the tight coupling of hybrid UQ and the size of the stochastic model (orders of magnitude larger than its deterministic counterpart), it is an excellent candidate for exascale computing. Since intrusive and non-intrusive UQ modules exhibit different characteristics in the context of high performance computing, effective parallelization will require detailed investigation into communication requirements, degree of parallelism, memory requirement, and etc. Additionally, as solution of matrix systems may take up much of the computational time, parallelization of the solution methods should also be closely examined.

- Investigate the interplay between uncertainties and numerical errors

  One of our findings is that numerical errors (discretization, algorithmic) can have a significant effect on the accuracy of the computed uncertainty measure. In addition, there may be tight coupling between uncertainties and numerical sense, in the sense of second order effects. As such, the interplay between uncertainties and numerical errors will (or has already) become an important issue. We maintain that our modular UQ framework is ideal for such an investigation because this framework facilitates the management (estimation and correction) of uncertainties and error at the module level. An effective error mitigation scheme can also be used to detect and correct soft errors (for example, the flipping of a bit in the computer hardware due to environmental factors).

- Resolving discontinuities in the uncertain parameter space

  As alluded to earlier in this report, our PCE-based hybrid UQ methodology has the deficiency, as with all PCE-based methods, that it works best for smooth problems (smooth in the parameter space). When discontinuities exist, a feasible remedy is to divide the parameter space into subdomains using schemes such as multi-element, generalized polynomial chaos [46].

Acknowledgements

This research was funded by U. S. Department of Energy Office of Advanced Computing Research Applied Mathematics Program and partially funded by the the U. S. Department NNSA ASC Program. It is performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract No. DE-AC52-07NA27344.

8. Appendix

8.1. Element matrices in homogeneous media

In this appendix, we provide the formula for computing the element matrices $\mathbf{M}^e$, $\mathbf{K}_1^e$ and $\mathbf{K}_2^e$.

$\mathbf{M}^e$ is derived from Eq. (91) as follows:

$$\mathbf{M}^e = \int_{\Omega_k^e} \Phi_i(\mathbf{x})\,\Phi_j(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_{\Omega_k^e} \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \end{bmatrix} \begin{bmatrix} \Phi_1 & \Phi_2 & \Phi_3 \end{bmatrix} \mathrm{d}\mathbf{x} = \frac{A}{12}\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \tag{169}$$

where the $\Phi_i(x,y)$ is the shape function for the linear triangular element:

$$\Phi_1 = \frac{1}{2A}\left[(x_2y_3 - x_3y_2) + (y_2 - y_3)\,x + (x_3 - x_2)\,y\right] \tag{170}$$

$$\Phi_2 = \frac{1}{2A}\left[(x_3y_1 - x_1y_3) + (y_3 - y_1)\,x + (x_1 - x_3)\,y\right] \tag{171}$$

$$\Phi_3 = \frac{1}{2A}\left[(x_1y_2 - x_2y_1) + (y_1 - y_2)\,x + (x_2 - x_1)\,y\right] \tag{172}$$

and

$$A = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} \tag{173}$$

is the area of each triangular element, where x_i and y_i be the coordinate value at the i^{th} node.

$\mathbf{K}_1^e$ and $\mathbf{K}_2^e$ are derived from the two terms in Eq. (92):

$$\mathbf{K}_1^e = (\mathbf{K}_1^e)_{ij} = \frac{1}{R}\int_{\Omega_k^e}\left(\alpha^x v^x \frac{\partial\Phi_i}{\partial x}\frac{\partial\Phi_j}{\partial x} + \alpha^y v^x \frac{\partial\Phi_i}{\partial y}\frac{\partial\Phi_j}{\partial y}\right)\mathrm{d}\mathbf{x} \quad 1 \le i,j \le 3. \tag{174}$$

$$\mathbf{K}_2^e = (\mathbf{K}_2^e)_{ij} = \frac{1}{R}\int_{\Omega_k^e}\left(v^x\Phi_i\frac{\partial\Phi_j}{\partial x} + v^y\Phi_i\frac{\partial\Phi_j}{\partial y}\right)\mathrm{d}\mathbf{x} \quad 1 \le i,j \le 3. \tag{175}$$

which simplify to:

$$\mathbf{K}_1^e = \frac{1}{4AR}\begin{pmatrix} \alpha^x v^x a_1^2 + \alpha^y v^x b_1^2 & \alpha^x v^x a_1a_2 + \alpha^y v^x b_1b_2 & \alpha^x v^x a_1a_3 + \alpha^y v^x b_1b_3 \\ \alpha^x v^x a_1a_2 + \alpha^y v^x b_1b_2 & \alpha^x v^x a_2^2 + \alpha^y v^x b_2^2 & \alpha^x v^x a_2a_3 + \alpha^y v^x b_2b_3 \\ \alpha^x v^x a_1a_3 + \alpha^y v^x b_1b_3 & \alpha^x v^x a_2a_3 + \alpha^y v^x b_2b_3 & \alpha^x v^x a_3^2 + \alpha^y v^x b_3^2 \end{pmatrix} \tag{176}$$

$$\mathbf{K}_2^e = \frac{1}{6R}\begin{pmatrix} v^x a_1 + v^y b_1 & v^x a_2 + v^y b_2 & v^x a_3 + v^y b_3 \\ v^x a_1 + v^y b_1 & v^x a_2 + v^y b_2 & v^x a_3 + v^y b_3 \\ v^x a_1 + v^y b_1 & v^x a_2 + v^y b_2 & v^x a_3 + v^y b_3 \end{pmatrix} \tag{177}$$

where $a_1 = y_1 - y_3$, $a_2 = x_3 - x_1$, $a_3 = x_1 - x_2$, $b_1 = x_3 - x_2$, $b_2 = x_1 - x_3$ and $b_3 = x_2 - x_1$.

8.2. Stochastic Galerkin projection

The stochastic Galerkin projection of the $Q+1$ polynomial chaos bases onto equation (81) gives rise to a $(Q+1)\times(Q+1)$ system, where each sub-problem (i,j) has the following form:

$$\begin{aligned}\frac{\partial c}{\partial t} = {} & \frac{1}{R}\frac{\partial}{\partial x}\left[D_{ij}^x(\mathbf{x})\,\frac{\partial c(\mathbf{x})}{\partial x}\right] + \frac{1}{R}\frac{\partial}{\partial y}\left[D_{ij}^y(\mathbf{x})\,\frac{\partial c(\mathbf{x})}{\partial y}\right] \\ & -\frac{1}{R}\frac{\partial\left[V_{ij}^x c(\mathbf{x},t)\right]}{\partial x} - \frac{1}{R}\frac{\partial\left[V_{ij}^y c(\mathbf{x},t)\right]}{\partial y} \ \in \Omega.\end{aligned} \tag{178}$$

Here, V_{ij}^x, V_{ij}^y, D_{ij}^x and D_{ij}^y are the effective velocity and dispersion coefficients corresponding to the sub-problem (i,j). These quantities can be computed by substituting equation (144) in x- and y- direction separately as follows:

$$\begin{aligned} V_{ij} &= \langle v\Psi_i\Psi_j\rangle \big/ \langle \Psi_i^2\rangle \\ &= \left\langle \left(f_0 + \sum_{k=1}^{r} s_k' f_k \eta_k \right) \Psi_i\Psi_j \right\rangle \Big/ \langle \Psi_i^2\rangle \\ &= \left[f_0 \langle \Psi_i\Psi_j\rangle + \left\langle \left(\sum_{k=1}^{r} s_k' f_k \eta_k \right) \Psi_i\Psi_j \right\rangle \right] \Big/ \langle \Psi_i^2\rangle \\ &= \left(f_0 \langle \Psi_i\Psi_j\rangle + \sum_{k=1}^{r} s_k' f_k \langle \eta_k\Psi_i\Psi_j\rangle \right) \Big/ \langle \Psi_i^2\rangle \end{aligned} \tag{179}$$

We assume the dispersivity α to be linear function of a random variable ξ. That is, $\alpha = \gamma + \beta\xi$, in which γ and β are constants, and their values depend on α's uncertainty range. For the dispersion coefficients, we have:

$$\begin{aligned} D_{ij} &= \langle \alpha v\Psi_i\Psi_j\rangle \big/ \langle \Psi_i^2\rangle \\ &= \left\langle (\gamma+\beta\xi) \left(f_0 + \sum_{k=1}^{r} s_k' f_k \eta_k \right) \Psi_i\Psi_j \right\rangle \Big/ \langle \Psi_i^2\rangle \\ &= \left[f_0 \langle (\gamma+\beta\xi)\, \Psi_i\Psi_j\rangle + \sum_{k=1}^{r} s_k' f_k \langle \eta_k (\gamma+\beta\xi)\, \Psi_i\Psi_j\rangle \right] \Big/ \langle \Psi_i^2\rangle . \end{aligned} \tag{180}$$

Due to the orthogonality of the stochastic expansion bases in $\mathcal{W}^Q$, the PCE-based stochastic variational problem can be simplified and rewritten in terms of the *stochastic global sparse matrices* $[\mathbf{M}]$ and $[\mathbf{K}]$:

$$[\mathbf{M}]\,[\dot{\mathbf{c}}] + [\mathbf{K}]\,[\mathbf{c}] = 0 \tag{181}$$

where the stochastic global matrices $[\mathbf{M}]$ and $[\mathbf{K}]$ are constructed from the deterministic global mass matrices $\mathbf{M}$ and stiffness matrices $\mathbf{K}$ (assembled from the finite-element method) as follows:

$$[\mathbf{M}] = \begin{pmatrix} \mathbf{M} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{M} \end{pmatrix}, \quad [\mathbf{K}] = \begin{pmatrix} \mathbf{K}_{0,0} & \dots & \mathbf{K}_{0,Q} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{Q,0} & \cdots & \mathbf{K}_{Q,Q} \end{pmatrix}. \tag{182}$$

Here, $\mathbf{K}_{i,j}$ is the stiffness matrix corresponding to each sub-problem (i,j) from the $(Q+1)\times(Q+1)$ stochastic global problem, and:

$$\mathbf{c} = (\mathbf{c}_0, \dots, \mathbf{c}_Q)^{\mathbf{T}}$$

is the stochastic concentration field, where $\mathbf{c}_k = (c_{1,k}, \dots, c_{\aleph,k})^{\mathbf{T}}$, $0 \le k \le Q$, denotes the vector of nodal values of the kth stochastic mode of the solution.

Using the backward Euler method, we can discretize equation (181) in time and rewrite it as:

$$\begin{aligned} &\begin{pmatrix} \mathbf{M}+\mathbf{K}_{0,0}\Delta t & \dots & \mathbf{K}_{0,Q}\Delta t \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{Q,0}\Delta t & \cdots & \mathbf{M}+\mathbf{K}_{Q,Q}\Delta t \end{pmatrix} \begin{pmatrix} \mathbf{c}_0 \\ \vdots \\ \mathbf{c}_Q \end{pmatrix}^{t+1/2} \\ &\qquad = \begin{pmatrix} \mathbf{M} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{M} \end{pmatrix} \begin{pmatrix} \mathbf{c}_0 \\ \vdots \\ \mathbf{c}_Q \end{pmatrix}^{t}. \end{aligned} \tag{183}$$

Assuming uniform probability distributions for the random variables $\{\alpha^x, \alpha^y\}$ and $\{\eta_i\}_{i=1}^r$, we can exploit the orthogonality of the Legendre polynomial chaos to approximate the mean and variance of the concentration at each grid point i by:

$$\bar{c}_i\left(t+\frac{1}{2}\right) = c_{i,0}^{t+\frac{1}{2}}$$
$$\sigma_i^2\left(t+\frac{1}{2}\right) = \sum_{k=1}^{Q}\left(c_{ik}^{t+\frac{1}{2}}\right)^2. \tag{184}$$

8.3. Element matrices in heterogeneous media

Again, denote by $a_1 = y_1 - y_3$, $a_2 = x_3 - x_1$, $a_3 = x_1 - x_2$, $b_1 = x_3 - x_2$, $b_2 = x_1 - x_3$ and $b_3 = x_2 - x_1$, we can define that

$$\frac{\partial \Phi_i(\mathbf{x})}{\partial x} = a_i \quad \frac{\partial \Phi_i(\mathbf{x})}{\partial y} = b_i \quad 1 \le i \le 3 \tag{185}$$

Similarly, apply it to the second term and we have

$$(\mathbf{K}_l)_{\Omega_k^e} = \left(\mathbf{K}_{ij}^l\right)_{\Omega_k^e} = \left(\beta^x a_i a_j + \beta^y b_i b_j\right)\int_{\Omega_k^e} Y_l(\mathbf{x})\, d\mathbf{x} \quad 1 \le i,j \le 3 \quad l = 1,\ldots,P \quad k = 1,\ldots,ne \tag{186}$$

Denote by

$$\begin{aligned}
(I_l)_{\Omega_k^e} &= \int_{\Omega_k^e} Y_l(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
&= \int_{\Omega_k^e} \exp\left(\bar{Y}(\mathbf{x}) + \sum_{n=1}^{N}\sqrt{\lambda_n} f_n(\mathbf{x})\,\xi_{nl}(\theta)\right)\mathrm{d}\mathbf{x} \\
&= \sum_{i=1}^{3}\int_{\Omega_k^e} \exp\left(\bar{Y}(\mathbf{x}_i) + \sum_{n=1}^{N}\sqrt{\lambda_n} f_n(\mathbf{x}_i)\,\xi_{nl}(\theta)\right)\Phi_{\mathrm{i}}(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
&= \frac{A}{3}\sum_{i=1}^{3}\exp\left(\bar{Y}(\mathbf{x}_i) + \sum_{n=1}^{N}\sqrt{\lambda_n} f_n(\mathbf{x}_i)\,\xi_{nl}(\theta)\right)
\end{aligned} \tag{187}$$

and we obtain

$$(I_l)_{\Omega_k^e} = \frac{A}{3}\sum_{i=1}^{3}\exp\left(\bar{Y}(\mathbf{x}_i) + \sum_{n=1}^{N}\sqrt{\lambda_n} f_n(\mathbf{x}_i)\,\xi_{nl}(\theta)\right) \quad k = 1,\ldots,ne \quad l = 1,\ldots,P \tag{188}$$

Hence, we obtain

$$\left(\mathbf{K}_{ij}^l\right)_{\Omega_k^e} = \left(\beta^x a_i a_j + \beta^y b_i b_j\right)(I_l)_{\Omega_k^e} \quad 1 \le i,j \le 3, \quad k = 1,\ldots,ne \quad l = 1,\ldots,P \tag{189}$$

To compute the effective flow velocity and dispersion coefficient for each sub-block (k,l), the following quantity needs to be evaluated:

$$V_{kl}^x(\mathbf{x}) = \sum_{i=0}^{P} v_i^x(\mathbf{x})\, e_{ikl} \quad V_{kl}^y(\mathbf{x}) = \sum_{i=0}^{P} v_i^y(\mathbf{x})\, e_{ikl} \tag{190}$$

Hence, for each each sub-block (k,l), we have

$$\begin{aligned}
\frac{\partial c}{\partial t} = \alpha^x\frac{\partial}{\partial x}\left(V^x(\mathbf{x})\frac{\partial c(\mathbf{x})}{\partial x}\right) + \alpha^y\frac{\partial}{\partial y}\left(V^y(\mathbf{x})\frac{\partial c(\mathbf{x})}{\partial y}\right) \\
-V^x\frac{\partial c(\mathbf{x},t)}{\partial x} - V^y\frac{\partial c(\mathbf{x},t)}{\partial y} \quad \text{in } \Omega
\end{aligned} \tag{191}$$

The trial function c^h and the test function w^h can be written as

$$\begin{aligned} c^h(\mathbf{x},t) &= \sum_{i\in\mathcal{N}} c_i(t)\,\Phi_i(\mathbf{x}) \in \mathcal{V}^h \\ w^h(\mathbf{x},t) &= \sum_{i\in\mathcal{N}} w_i(t)\,\Phi_i(\mathbf{x}) \in \mathcal{V}^h \end{aligned} \tag{192}$$

By plugging the finite-element approximation c^h and w^h of c and w into the variational form, we have

$$\sum_{i,j\in\mathcal{N}} a_{ij} c_i w_j = 0 \tag{193}$$

where

$$\begin{aligned} a_{i,j} &= \int_\Omega \frac{\partial\Phi_i(\mathbf{x},t)}{\partial t}\Phi_j(\mathbf{x})\,\mathrm{d}\mathbf{x} + \int_\Omega \left(\alpha^x V^x(\mathbf{x})\frac{\partial\Phi_i(\mathbf{x},t)}{\partial x}\frac{\partial\Phi_j(\mathbf{x})}{\partial x} + \alpha^y V^y(\mathbf{x})\frac{\partial\Phi_i(\mathbf{x},t)}{\partial y}\frac{\partial\Phi_j(\mathbf{x})}{\partial y}\right)\mathrm{d}\mathbf{x} \\ &- \int_\Omega \left(V^x(\mathbf{x})\frac{\partial\Phi_i(\mathbf{x},t)}{\partial x}\Phi_j(\mathbf{x}) + V^y(\mathbf{x})\frac{\partial\Phi_i(\mathbf{x},t)}{\partial y}\Phi_j(\mathbf{x})\right)\mathrm{d}\mathbf{x}. \end{aligned} \tag{194}$$

Apply it to the first term on the right hand side of the equation above, we obtain

$$\mathbf{M}^e = \int_{\Omega_k^e} \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \end{bmatrix} \begin{bmatrix} \Phi_1 & \Phi_2 & \Phi_3 \end{bmatrix} \mathrm{d}\mathbf{x} = \frac{A}{12}\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix} \tag{195}$$

where

$$A = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} \tag{196}$$

is the area of each triangular element, where x_i and y_i be the coordinate value at the i^{th} node.

Similarly, apply it to the second term and we have

$$\begin{aligned} (\mathbf{K}_1)_{\Omega_k^e} &= \left(\mathbf{K}_{ij}^1\right)_{\Omega_k^e} = \int_{\Omega_k^e} \left(\alpha^x V^x(\mathbf{x})\frac{\partial\Phi_i}{\partial x}\frac{\partial\Phi_j}{\partial x} + \alpha^y V^y(\mathbf{x})\frac{\partial\Phi_i}{\partial y}\frac{\partial\Phi_j}{\partial y}\right)\mathrm{d}\mathbf{x} \\ &= \alpha^x a_i a_j \int_{\Omega_k^e} V^x(\mathbf{x})\,\mathrm{d}\mathbf{x} + \alpha^y b_i b_j \int_{\Omega_k^e} V^y(\mathbf{x})\,\mathrm{d}\mathbf{x} \quad 1\le i,j\le 3 \end{aligned} \tag{197}$$

Apply it to the third term and we have

$$(\mathbf{K}_2)_{\Omega_k^e} = \left(\mathbf{K}_{ij}^2\right)_{\Omega_k^e} = a_j \int_{\Omega_k^e} \left(V^x(\mathbf{x})\,\Phi_i\right)\mathrm{d}\mathbf{x} + b_j \int_{\Omega_k^e} \left(V^y(\mathbf{x})\,\Phi_i\right)\mathrm{d}\mathbf{x} \quad 1\le i,j\le 3. \tag{198}$$

Denote by

$$(I_{11})_{\Omega_k^e} = \int_{\Omega_k^e} V^x(\mathbf{x})\,\mathrm{d}\mathbf{x} = \frac{A}{3}\sum_{i=1}^{3} V_i^x \quad (I_{12})_{\Omega_k^e} = \int_{\Omega_k^e} V^y(\mathbf{x})\,\mathrm{d}\mathbf{x} = \frac{A}{3}\sum_{i=1}^{3} V_i^y \tag{199}$$

and

$$\begin{aligned}\left(I_{21}^{i}\right)_{\Omega_k^e} &= \int_{\Omega_k^e}\left(\sum_{j=1}^{3} V_j^x \Phi_j \Phi_i\right) \mathrm{d}\mathbf{x} \\ &= \sum_{j=1}^{3} V_j^x \int_{\Omega_k^e} \Phi_j \Phi_i \mathrm{d}\mathbf{x} \\ &= \sum_{j=1, j\neq i}^{3} V_j^x \frac{A}{12} + V_i^x \frac{A}{6} \end{aligned} \tag{200}$$

$$\begin{aligned}\left(I_{22}^{i}\right)_{\Omega_k^e} &= \int_{\Omega_k^e} \left(V^y\left(\mathbf{x}\right)\Phi_i\right) \mathrm{d}\mathbf{x} \\ &= \sum_{j=1, j\neq i}^{3} V_j^y \frac{A}{12} + V_i^y \frac{A}{6} \end{aligned} \tag{201}$$

We obtain

$$\left(\mathbf{K}_1\right)_{\Omega_k^e} = \left(\mathbf{K}_{ij}^1\right)_{\Omega_k^e} = \alpha^x a_i a_j \left(I_{11}\right)_{\Omega_k^e} + \alpha^y b_i b_j \left(I_{12}\right)_{\Omega_k^e} \quad 1 \leq i,j \leq 3 \quad k = 1, \ldots, ne \tag{202}$$

and

$$\left(\mathbf{K}_2\right)_{\Omega_k^e} = \left(\mathbf{K}_{ij}^2\right)_{\Omega_k^e} = a_j \left(I_{21}^i\right)_{\Omega_k^e} + b_j \left(I_{22}^i\right)_{\Omega_k^e} \quad 1 \leq i,j \leq 3 \quad k = 1, \ldots, ne \tag{203}$$

All the elementary matrix and vectors can be assembled into the global matrix $\mathbf{M}$, $\mathbf{K}_1$, $\mathbf{K}_2$, and $\mathbf{K}$.

References

[1] W. Morokoff, R. Caflisch, Quasi-Monte Carlo integration, J. Comput. Phys. 122 (1985) 218–230.

[2] M. C. McKay, R. Beckman, W. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 21 (1979) 239–245.

[3] R. Ghanem, P. D. Spanos, Stochastic calculus: applications in sciences and engineering, Birkhauser, 2002.

[4] J. Liu, Monte Carlo strategies in scientific computing, Springer, 2001.

[5] R. Ghanem, P. D. Spanos, Stochastic finite elements: a spectral approach, Springer-Verlag, 1991.

[6] R. Ghanem, P. D. Spanos, Spectral stochastic finite-element formulation for reliability analysis, J. Eng. Mech. 117 (1991) 2351–2372.

[7] G. Schueller, Computational stochastic mechanics: recent advances, Comput. Struct. 79 (2001) 2225–2234.

[8] R. Ghanem, Probabilistic characterization of transport in heterogeneous media, Comput. Method Appl. M. 158 (1998) 199–220.

[9] O. M. Knio, O. P. Maître, Uncertainty propagation in cfd using polynomial chaos decomposition, Fluid Dyn. Res. 38 (2006) 616–640.

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

[11] D. Phenix, J. Dinaro, M. Tatagg, J. Test, J. Howard, G. McRae, Incorporation of parametric uncertainty into complex kinetic mechanisms: application to hydrogen oxidation in supercritical water, Combust. and Flame 112 (1998) 132–146.

[12] J. Li, D. Xiu, A generalized polynomial chaos based ensemble kalman filter with high accuracy, J. Comput. Phys. 228 (2009) 5454–5469.

[13] H. C. Elman, C. W. Miller, E. T. Phipps, R. S. Tuminaro, Assessment of collocation and galerkin approaches to linear diffusion equations with random data, Int. J. Uncertainty Quantification 1 (2011) 19–33.

[14] H. N. Najm, Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics, Annual Review of Fluid Mechanics 41 (2009) 35–42.

[15] S. Hosder, R. Walters, R. Perez, A non-intrusive polynomial chaos method for uncertainty propagation in CFD simulations, in: 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, pp. 891–820.

[16] G. E. K. X. Wan, A multi-element generalized polynomial chaos for arbitrary probability measures, SIAM J. Sci. Comput. 28 (2006) 901–928.

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

[18] C. G. W. F. Nobile, R. Tempone, A sparse grid stochastic collocation method for partial differential equations with random input data, article number: Trita-na 2007:07, SINUM (2007).

[19] R. G. S. Das, S. Finette, Polynomial chaos representation of spatio-temporal random fields from experimental measurements, J. Comput. Phys. 228 (2009) 8726–8751.

[20] X. Emory, Uncertainty modeling and spatial prediction by multi-gaussian kriging: Accounting for an unknown mean value, Computers & Geosciences 34 (2008) 1431–1442.

[21] H. B. A. Loeven, Radial basis functions for uncertainty quantification in cfd, WCCM8 (2008).

[22] G. I. T. Chantrasmi, A. Doostan, Uncertainty modeling and spatial prediction by multi-gaussian kriging: Accounting for an unknown mean value, J. Comput. Phys. 228 (2009) 7159–7180.

[23] D. G. Cacuci, Sensitivity & uncertainty analysis, volume 1:Theory, Chapman and Hall, 2003.

[24] M. S. E. G. Tang, L. P. Swiler, I. Gianluca, Using stochastic expansion methods in evidence theory for mixed aleatory-epistemic uncertainty quantification.

[25] D. Y. G. Medic, G. Kalitzin, An approach for coupling rans and les in integrated computations of jet engines, CTR Annual Research Briefs at Stanford University (2006) 287–298.

[26] S. Wiener, The homogeneous chaos, Amer. J. Math. 60 (1993) 897–936.

[27] R. Askey, J. Wilson, Some basic hypergeometric polynomials that generalize jacobi polynomials, Mem. Am. Math. Soc. 54 (1985) 375–377.

[28] I. Sobol, Sensitivity estimates for nonlinear mathematical models, MMCE 1 (1993) 407–414.

[29] M. C. McKay, Evaluating Prediction Uncertainty, Technical Report NUREG/CR-6311 LA-12915-MS, Los Alamos National Laboratory, 1995.

[30] H. S. Abdel-Khalik, Hybrid uncertainty and sensitivity algorithms for high dimensional nonlinear models, Trans. Am. Nucl. Soc. 103 (2010) 375–377.

[31] C. Joslyn, S. Ferson, Approximate representations of random intervals for hybrid uncertainty quantification, in: 4th Int. Conf. on Sensitivity Analysis of Model Output (SAMO), pp. 453–469.

[32] P. G. Constantine, A. Doostan, G. Iaccarino, A hybrid collocation/galerkin scheme for convective heat transfer problems with stochastic boundary conditions, Int. J. Numer. Meth. Eng. 80 (2009) 868–880.

[33] A. Surana, A. Banaszuk, Scalable uncertainty quantification in complex dynamic networks, in: 49th IEEE Conference on Decision and Control (CDC), Atlanta, GA, pp. 7278–7285.

[34] R. Abgrall, P. M. Congedo, S. Galera, G. Geraci, Semi-intrusive and non-intrusive stochastic methods for aerospace applications, in: 4th European Conf. for Aerospace Sciences (EUCASS), St Petersburg, Russia, pp. 1–8.

[35] P. G. Constantine, A. Doostan, G. Iaccarino, A hybrid uncertainty propagation scheme for convective heat transfer problems, in: 49th AIAA Structures, Structural Dynamics and Materials Conf., Schaumburg, IL, pp. 1723–1781.

[36] A. Guadagnini, L. Guadagnini, D. M. Tartakovsky, C. L. Winter, Random domain decomposition for flow in heterogeneous stratified aquifers, Stoch Environ Res Risk Assess 17 (2003) 394–407.

[37] R. G. B. Sousekik, E. Phipps, Hierachical schur complement preconditioner for the stochastic galerkin finite element methods, Numerical Analysis (2013).

[38] K. Kuhunen, Uber lineare methoden in der wahrscheinlichkeitsrechnung, Am. Acad. Sci. 37 (1947) 3–79.

[39] M. Kac, A. Siegert, An explicit representation of a stationary gaussian process, Ann. Math. Stat. 18 (1947) 438–442.

[40] D. Zhang, Z. Lu, An efficient, high-order perturbation approach for flow in random porous media via karhunen-loeve and polynomial expansions, J. Comput. Phys. 194 (2004) 773–794.

[41] H. Li, D. Zhang, Probabilistic collocation method for flow in porous media: Comparisons with other stochastic methods, Water Resour. Res. 43 (2007). W09409, doi:10.1029/2006WR005673.

[42] H. Li, D. Zhang, Stochastic analysis of unsaturated flow with probabilistic collocation method, Water Resour. Res. 45 (2009). W08425, doi:10.1029/2008WR007530.

[43] G. Liu, Z. Lu, D. Zhang, Stochastic uncertainty analysis for solute transport in randomly heterogeneous media using a karhunen-loeve based moment equation approach, Water Resour. Res. 43 (2007). W07427, doi:10.1029/2006WR005193.

[44] G. Lin, A. M. Tartakovsky, An efficient, high-order probabilistic collocation method on sparse grids for three-dimensional flow and solute transport in randomly heterogeneous porous media, Adv. Water Resour 32 (2009) 712–722.

[45] M. Morris, Factorial sampling plans for preliminary computational experiments, Technometrics 33 (1991) 161–174.

[46] X. Wan, G. E. Karniadakis, Multi-element generalized polynomial chaos for arbitrary probability measures, SIAM J. Sci. Comput. 28 (2006) 901–928.

[47] R. Fisher, Statistical methods for research workers, Oliver and Boyd (1925).

[48] M. Griebel, Sparse grids and related approximation schemes for higher-dimensional problems, in: Proceedings of the conference on Foundations of Computational Mathematics, Santander, Spain.

[49] J. Foo, G. E. Karniadakis, Multi-element probabilistic collocation method in high dimensions, J. Comput. Phys. 229 (2010) 1536–1557.

[50] T. P. Clement, Y. Sun, B. S. Hooker, J. N. Petersen, Modeling multi-species reactive transport in groundwater aquifers, Ground Water Monit. Remediat. 18 (1998) 79–92.

[51] Y. Sun, J. N. Petersen, T. P. Clement, B. S. Hooker, Effect of reaction kinetics on predicted concentration profiles during subsurface bioremediation, J. Contam. Hydrol. 31 (1998) 359–372.

[52] C. M. Cho, Convective transport of ammonium with nitrification in soil, Can. J. Soil Sci. 51 (1971) 339–350.

[53] Y. Sun, X. Lu, J. Petersen, T. Buscheck, An analytical solution of tetrachloroethylene transport and biodegradation, Transport Porous Med. 55 (2004) 301–308.

[54] Y. Sun, T. Buscheck, Y. Hao, An analytic method for modeling first-order decay networks, Comput. Geosci. 39 (2012) 86–97.

[55] S. Balakrishna, A. Roy, M. G. Ierapetritou, G. P. Flach, P. G. Georgopoulos, Uncertainty reduction and characterization for complex environmental fate and transport models: An empirical bayesian framework incorporating the stochastic response surface method, Water Res. Res. 39 (2003) 1350–1362.

[56] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, S. Tarantola, Global Sensitivity Analysis: The Primer, Wiley-Interscience, 2008.

[57] G. H. Golub, C. F. V. Loan, Matrix computations (3rd ed.), Johns Hopkins University Press, 1996.

[58] R. Falgout, HYPRE User's Manual (Version 2.8.0b), Lawrence Livermore National Laboratory, 2011.

[59] C. Tong, PSUADE User's Manual (Version 1.2.0), Lawrence Livermore National Laboratory, 2009. LLNL-SM-407882.

[60] L. Sirovich, Berkooz, Turbulence and the dynamics of coherent structures, part iii: dynamics and scaling, Quart. Appl. Math. 45 (2006) 583–590.