SPACE-TIME REDUCED-ORDER MODELING FOR UNCERTAINTY QUANTIFICATION

RUHUI JIN [*], FRANCESCO RIZZI [†], AND ERIC PARISH [‡]

Abstract. This work focuses on the space-time reduced-order modeling (ROM) method for solving large-scale uncertainty quantification (UQ) problems with multiple random coefficients. In contrast with the traditional space ROM approach, which performs dimension reduction in the spatial dimension, the space-time ROM approach performs dimension reduction on both the spatial and temporal domains, and thus enables accurate approximate solutions at a low cost. We incorporate the space-time ROM strategy with various classical stochastic UQ propagation methods such as stochastic Galerkin and Monte Carlo. Numerical results demonstrate that our methodology has significant computational advantages compared to state-of-the-art ROM approaches. By testing the approximation errors, we show that there is no obvious loss of simulation accuracy for space-time ROM given its high computational efficiency.

1. Introduction. Quantifying uncertainties in physical systems plays an important role in numerous fields, including climate modeling [6], hypersonic aerodynamics [4, 8] and quantum mechanics [11]. It has long been a computational challenge to model and simulate large-scale dynamical and control systems with high-dimensional parametric uncertainties. Researchers have been developing model reduction methods [10, 3] to tackle this computational bottleneck. By building and working with a reduced-order model (ROM) as a qualified approximation to the full-oder model (FOM), the overall computational complexity is reduced significantly.

Current ROM studies mostly consider the space ROM method and focus on only spatial dimension reduction but maintain the full dimensionality of the temporal domain. As a result, space ROMs can have limited computational savings for unsteady problems characterized by, e.g., small required time steps or long simulation horizons. Regarding the UQ approach, on one hand, Monte Carlo (MC) is by far the most popular method applied in the ROM workflow [12, 13] due to its reliability and implementation simplicity. On the other hand, other types of UQ propagation methods, for example, the stochastic Galerkin (SG) technique [1, 9] based on polynomial chaos expansion, have advantages of good spectral accuracy and convergence over the classical MC.

In this work, we study the space-time ROM method [2, 5] constructed via Galerkin projection and space-time proper orthogonal decomposition. This novel approach is considered a variation of the space ROM. Its implementation simply stacks the space and time dimensions to achieve the model reduction by finding a lower-dimensional representation for both spatial and temporal domains. The method simultaneously approximates a large-scale PDE model for all points in space and time within a much faster computing time compared to the commonly used space ROM method. Additionally, the space-time ROM approach is often equipped with more favorable error bounds and stability properties than space ROMs [5].

We apply various UQ propagation techniques such as Monte Carlo and stochastic Galerkin in the space-time ROM framework. We test the described methodology on advection-diffusion PDE problems with multi-dimensional parametric uncertainties. Our numerical results show that the space-time approach can result in huge computational speed-ups while maintaining accurate approximated solutions.

---

[*]Department of Mathematics, University of Texas at Austin, rhjin@math.utexas.edu

[†]NexGen Analytics, fr.francescorizzi@gmail.com

[‡]Sandia National Laboratories, ejparis@sandia.gov

The main contributions of this work are:
1. We study the space-time ROM method and combine it with the well established stochastic Galerkin (SG) technique. By constructing polynomial basis on the reduced space-time domain, we demonstrate that the space-time ROM with SG has good accuracy and a faster computing time as compared to traditional ROM techniques (e.g., the space ROM with MC sampling).
2. We implement the proposed computational scheme for one and two-dimensional advection-diffusion-reaction PDE problems.
3. We provide thorough numerical assessments for the space-time ROM with respect to the computational time, approximation errors and convergence property given increasing number of samples (MC) and polynomial degrees (SG).

2. Mathematical background.

2.1. Full-order model. We consider the numerical solution to the parametrized dynamical system:

$$\dot{\boldsymbol{u}}(t, \boldsymbol{\mu}) = \boldsymbol{f}(\boldsymbol{u}(t, \boldsymbol{\mu}), t, \boldsymbol{\mu})), \quad \boldsymbol{u}(0, \boldsymbol{\mu}) = \boldsymbol{u_0}(\boldsymbol{\mu}) \tag{2.1}$$

where
1. $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^{N_\mu}$ denotes uncertain parameters;
2. $\boldsymbol{u} : [0, T] \times \mathcal{D} \to \mathbb{R}^{N_s}$ is the time-dependent, parametrized state as the solution to problem (2.1);
3. $\boldsymbol{f} : \mathbb{R}^{N_s} \times [0, T] \times \mathcal{D} \to \mathbb{R}^{N_s}$ is the velocity;
4. $\boldsymbol{u_0} \in \mathbb{R}^{N_s}$ is the initial state.

We aim to understand how the system state $\boldsymbol{u}(t, \boldsymbol{\mu})$ responds as a function of time t and uncertain parameters $\boldsymbol{\mu}$. To this end, we apply numerical simulation techniques to solve the UQ problem (2.1).

We now introduce the time-discretized form of the main problem (2.1). In particular, we discretize the temporal domain $[0, T]$ into N_t time instances characterized by $t^n = n \Delta t$ where Δt denotes the time step. For example, one classical time-discretization method is the Crank-Nicolson method which yields a sequence of discrete solutions $\mathbf{u}^n(\boldsymbol{\mu}) \approx \boldsymbol{u}(t^n, \boldsymbol{\mu}) \in \mathbb{R}^{N_s}$ as the implicit solution to the system of equations at each time step $n = 1, \ldots, N_t$:

$$\begin{aligned} \mathbf{r}^n(\mathbf{u}^n, \mathbf{u}^{n-1}, \boldsymbol{\mu}) \quad &: \mathbb{R}^{N_s} \otimes \mathbb{R}^{N_s} \otimes \mathbb{R}^{N_\mu} \to \mathbb{R}^{N_s} \\ &:= \frac{\mathbf{u}^n - \mathbf{u}^{n-1}}{\Delta t} - \frac{1}{2} \left(\boldsymbol{f}(\mathbf{u}^n, t^n, \boldsymbol{\mu}) - \boldsymbol{f}(\mathbf{u}^{n-1}, t^{n-1}, \boldsymbol{\mu}) \right) \end{aligned} \tag{2.2}$$

with initial condition $\mathbf{u}^0 = \boldsymbol{u_0}(\boldsymbol{\mu})$. Note the parametric dependence of the state has been suppressed in the above for simplicity. Thus a discrete representation of the FOM system is

$$[\mathbf{u}^1(\boldsymbol{\mu}), \mathbf{u}^2(\boldsymbol{\mu}), \ldots, \mathbf{u}^{N_t}(\boldsymbol{\mu})] \in \mathbb{R}^{N_s} \otimes \mathbb{R}^{N_t}.$$

2.2. Projection-based model reduction. The FOM solving process is computationally expensive in practice when the spatial dimension N_s and temporal dimension N_t are large. The reduced-order modeling technique is proposed to overcome this computational challenge. It follows an offline-online paradigm. Please see the workflow Figure 2.1 below.

In the offline phase, we sample and plug in a certain number of uncertain parameter instances into the full-order model and solve the system accordingly. The

obtained sample solutions are collected to form a snapshot matrix. We then identify a low-dimensional subspace by performing proper orthogonal decomposition (POD) for the state snapshots. The governing equation (2.1) is projected onto this trial subspace to create a reduced-order model. The result of this process is a reduced-order model which can be solved more efficiently.
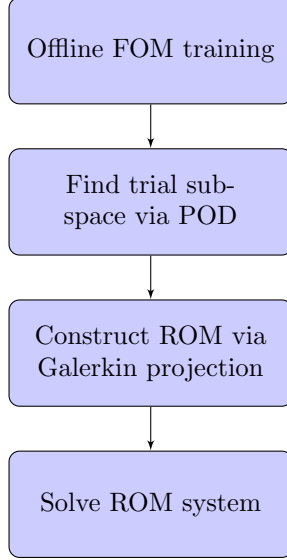
Fig. 2.1. *ROM workflow*

2.3. Trial subspace and POD.

2.3.1. Spatial trial subspace. Suppose in the offline training procedure, we obtain a collection of snapshot solutions for N_{train} randomly drawn parameter instances :

$$\mathbf{U}_{\text{train}} = \left[\mathbf{u}^1(\boldsymbol{\mu}_1), \ldots, \mathbf{u}^{N_t}(\boldsymbol{\mu}_1), \ldots, \mathbf{u}^1(\boldsymbol{\mu}_{N_{\text{train}}}), \ldots, \mathbf{u}^{N_t}(\boldsymbol{\mu}_{N_{\text{train}}})\right] \in \mathbb{R}^{N_s \times (N_t N_{\text{train}})}.$$

The proper orthogonal decomposition method identifies a lower-dimensional trial subspace represented by an orthonormal matrix $\boldsymbol{\Phi}$ from the above training solution set. In particular, we consider the optimization problem in a least squares sense:

$$\arg \min_{\boldsymbol{\Phi} \in \mathbb{R}^{N_s \times K}} \|\boldsymbol{\Phi}\boldsymbol{\Phi}^\top \mathbf{U}_{\text{train}} - \mathbf{U}_{\text{train}}\|_2^2, \quad \text{subject to } \boldsymbol{\Phi}^\top \boldsymbol{\Phi} = \mathbf{I}_K, \qquad (2.3)$$

where K is the subspace dimension. For the choice of K, we set a relative energy tolerance threshold e_{tol} and compute K such that the selected basis $\boldsymbol{\Phi} \in \mathbb{R}^{N_s \times K}$ preserves the amount of energy for the training solution set $\mathbf{U}_{\text{train}}$ that exceeds the threshold.

To solve (2.3), we compute the singular value decomposition of $\mathbf{U}_{\text{train}}$: $\mathbf{L}, \mathbf{s}, \_ = \text{SVD}(\mathbf{U}_{\text{train}})$. The subspace dimension K is determined by

$$K := \arg \min_{K \in \mathbb{N}} \left| e_{\text{tol}} - \frac{\sum_{i=1}^K s_i^2}{\|\mathbf{s}\|_2^2} \right|.$$

We then select the first K columns of the left singular vectors $\mathbf{L}$ to form the basis $\boldsymbol{\Phi} \in \mathbb{R}^{N_s \times K}$.

2.3.2. Space-time trial subspace. In the space-time formulation, instead of expressing residuals of all time steps as in (2.2), we formulate the residual in just one system:

$$\overrightarrow{\mathbf{r}}(\overrightarrow{\mathbf{u}}(\boldsymbol{\mu}), t, \boldsymbol{\mu}) = \mathbf{0} \in \mathbb{R}^{N_s N_t}, \tag{2.4}$$

where we concatenate solutions and residuals of all time steps along one dimension, i.e. $\overrightarrow{\mathbf{u}}(\boldsymbol{\mu}) = [\mathbf{u}^1(\boldsymbol{\mu})^\top, \ldots, \mathbf{u}^{N_t}(\boldsymbol{\mu})^\top]^\top \in \mathbb{R}^{N_s N_t}$.

Identifying a space-time trial subspace is rather similar to identifying the spatial subspace. To be more specific, given a collection of space-time training solutions

$$\overrightarrow{\mathbf{U}}_{\text{train}} = [\overrightarrow{\mathbf{u}}(\boldsymbol{\mu}_1), \ldots, \overrightarrow{\mathbf{u}}(\boldsymbol{\mu}_{\text{train}})] \in \mathbb{R}^{(N_s N_t) \times N_{\text{train}}},$$

we apply POD to find a lower-dimensional subspace that captures most of the energy of the above solution set.

2.4. Galerkin projection. After identifying a basis $\boldsymbol{\Phi}$ of the training solution set, we apply Galerkin projection to construct a reduced-order model. We denote the approximated low-dimensional solution $\hat{\mathbf{u}}(\boldsymbol{\mu}) \in \mathbb{R}^K$. By the assumption of $\mathbf{u}(\boldsymbol{\mu}) \approx \boldsymbol{\Phi}\hat{\mathbf{u}}(\boldsymbol{\mu})$, we impose the residual of the full-order model to be orthogonal to the basis:

$$\boldsymbol{\Phi}^\top \mathbf{r}(\boldsymbol{\Phi}\hat{\mathbf{u}}(\boldsymbol{\mu}), t, \boldsymbol{\mu}) = \mathbf{0} \in \mathbb{R}^K \tag{2.5}$$

and solve the above reduced system for $\hat{\mathbf{u}}(\boldsymbol{\mu})$.

Note that the above general forms of state solution $\hat{\mathbf{u}}$ and residual $\mathbf{r}$ in (2.5) can be replaced by $\overrightarrow{\hat{\mathbf{u}}}, \overrightarrow{\mathbf{r}}$ for the space-time ROM approach.

3. Uncertainty quantification methods. In the above section, we introduced the spatial-Galerkin and space-time-Galerkin ROMs to reduce the computational cost associated with solving the forward model. This section details how these ROMs can be combined with several classical UQ propagation methods to solve the underlying UQ problem. In particular, we consider Monte Carlo (MC) sampling and stochastic Galerkin projection.

3.1. Monte Carlo sampling. The MC methodology simply follows as:
1. draw samples of random parameters from certain probability distributions;
2. solve the system (2.1) based on these parameter instances;
3. compute quantities of interest (e.g., mean, variance) from the ensemble of solutions.

3.2. Stochastic Galerkin. We first provide some background of polynomial chaos expansion (PCE) which the stochastic Galerkin approach is built upon. We consider a parametrized linear system

$$\mathbf{A}(\boldsymbol{\mu})\mathbf{u}(\boldsymbol{\mu}) = \mathbf{b}(\boldsymbol{\mu}) \in \mathbb{R}^N, \tag{3.1}$$

where the linear operators $\mathbf{A} : \mathcal{D} \to \mathbb{R}^N$ and $\mathbf{b} : \mathcal{D} \to \mathbb{R}^N$ are constructed correspondingly from the residuals of spatial domain (2.2) ($N = N_s$) or space-time domain (2.4) ($N = N_s N_t$), with initial condition given by $\mathbf{u}^0 = \boldsymbol{u_0}(\boldsymbol{\mu}) \in \mathbb{R}^{N_s}$.

The idea is to approximate the numerical solution function $\mathbf{u}(\cdot) : \mathcal{D} \to \mathbb{R}^N$ by using a spectral approximation that lies in the span of a finite set of polynomials $\{\psi_j(\cdot)\}_{j=1}^{N_\psi} \subset L^2(\mathcal{D})$. The mathematical formulation is as follows:

$$\mathbf{u}(\boldsymbol{\mu}) \approx \tilde{\mathbf{u}}(\boldsymbol{\mu}) = \left(\boldsymbol{\psi}(\boldsymbol{\mu})^\top \otimes \mathbf{I}_N\right) \mathbf{m}, \tag{3.2}$$

where $\boldsymbol{\psi}(\boldsymbol{\mu}) = [\psi_1(\boldsymbol{\mu}), \dots, \psi_{N_\psi}(\boldsymbol{\mu})]^\top \in \mathbb{R}^{N_\psi}$ denotes the collection of the polynomial basis and $\mathbf{m} \in \mathbb{R}^{NN_\psi}$ is the coefficient vector .

Thus we can define the approximation residual with respect to polynomial coefficients and uncertainty parameters,

$$
\begin{aligned}
\mathbf{r}(\mathbf{m}, \boldsymbol{\mu}) \quad &: \mathbb{R}^{NN_\psi} \otimes \mathbb{R}^{N_\mu} \to \mathbb{R}^N \\
&:= \mathbf{A}(\boldsymbol{\mu}) \left(\boldsymbol{\psi}(\boldsymbol{\mu})^\top \otimes \mathbf{I}_N \right) \mathbf{m} - \mathbf{b}(\boldsymbol{\mu}) \\
&= \left(\boldsymbol{\psi}(\boldsymbol{\mu})^\top \otimes \mathbf{A}(\boldsymbol{\mu}) \right) \mathbf{m} - \mathbf{b}(\boldsymbol{\mu}).
\end{aligned}
\tag{3.3}
$$

We compute the unknown coefficients $\mathbf{m}$ by the residual formula.

We now formally introduce the stochastic Galerkin approach to solve for the coefficients $\mathbf{m}$ in (3.3). Given a density function ρ for the probability space $\mathcal{D}$, we define the inner product:

$$
\langle g(\boldsymbol{\mu}), h(\boldsymbol{\mu}) \rangle_\rho = \int_{\mathcal{D}} g(\boldsymbol{\mu}) \, h(\boldsymbol{\mu}) \, \rho(\boldsymbol{\mu}) d\boldsymbol{\mu},
\tag{3.4}
$$

where $g, h \in L^2(\mathcal{D})$ are functions. The expectation of a function g is given by:

$$
\mathbb{E}[g] = \int_{\mathcal{D}} g(\boldsymbol{\mu}) \, \rho(\boldsymbol{\mu}) d\boldsymbol{\mu}.
\tag{3.5}
$$

To solve for the coefficients in (3.3), the stochastic Galerkin method asks to impose orthogonality on the residual of the system (3.3) with respect to the inner product $\langle \cdot, \cdot \rangle$ (3.4). That is to say, we restrict the residual to be orthogonal to the polynomial bases, i.e.

$$
\langle \psi_j, r_i(\mathbf{m}) \rangle_\rho = \mathbb{E}[\psi_j \, r_i(\mathbf{m})] = 0,
$$

for all residual dimensions $i = 1, \dots, N$, and stochastic dimensions $j = 1, \dots, N_\psi$. An alternative vector expression is

$$
\mathbb{E}[\boldsymbol{\psi} \otimes \mathbf{r}(\mathbf{m})] = \mathbf{0} \in \mathbb{R}^{N N_\psi}.
$$

From the PCE residual formula given in (3.3), we end up with solving

$$
\mathbb{E}[\boldsymbol{\psi}\boldsymbol{\psi}^\top \otimes \mathbf{A}] \, \mathbf{m} = \mathbb{E}[\boldsymbol{\psi} \otimes \mathbf{b}]
\tag{3.6}
$$

and form an approximating function solution $\tilde{\mathbf{u}}(\cdot)$ by (3.2) at each time step $t = 1, \dots, N_t$.

Similarly, for the stochastic Galerkin ROM solution $\hat{\mathbf{u}}(\cdot) : \mathcal{D} \to \mathbb{R}^K$, based on the Galerkin projection approach shown in (2.5), we solve coefficients $\hat{\mathbf{m}} \in \mathbb{R}^{K N_\psi}$ in a reduced system:

$$
\mathbb{E}[\boldsymbol{\psi}\boldsymbol{\psi}^\top \otimes \boldsymbol{\Phi}^\top \mathbf{A} \boldsymbol{\Phi}] \, \hat{\mathbf{m}} = \mathbb{E}[\boldsymbol{\psi} \otimes \boldsymbol{\Phi}^\top \mathbf{b}]
\tag{3.7}
$$

and formulate $\hat{\mathbf{u}}(\cdot)$ by $\left(\boldsymbol{\psi}(\boldsymbol{\mu})^\top \otimes \mathbf{I}_K \right) \hat{\mathbf{m}}$.

We would like to remark that, for SG space ROM method, we solve (3.7) for function $\hat{\mathbf{u}}(\cdot)$ at each time step, while in the SG space-time ROM scheme, we solve (3.7) and obtain function solution $\overrightarrow{\hat{\mathbf{u}}}(\cdot)$ for all time steps at once. We additionally remark that both the SG space ROM and SG space-time ROM methods result in significantly smaller solution vectors than the standard SG approach and are thus significantly more computationally tractable.

4. Numerical experiment I: 1D parametrized advection-diffusion problem. We focus on the numerical solution of the one-dimensional parametrized advection -diffusion problem with initial and boundary conditions:

$$
\begin{cases}
\dfrac{\partial \boldsymbol{u}}{\partial t} + c\,\dfrac{\partial \boldsymbol{u}}{\partial x} = \nu\,\dfrac{\partial^2 u}{\partial x^2}, \\[2mm]
\boldsymbol{u}(0, t, (c, \nu)) = 0, \quad \forall\, t \in [0, 1], (c, \nu) \in \mathcal{D} \\[2mm]
\boldsymbol{u}(x, 0, (c, \nu)) = 0, \quad \forall\, x \in [0, 1], (c, \nu) \in \mathcal{D}
\end{cases}
\tag{4.1}
$$

where the state $\boldsymbol{u} : [0, 1] \times [0, 1] \times \mathcal{D} \to \mathbb{R}$. Here, the wave speed $c \sim \mathcal{N}(1, 0.15)$ and the diffusion coefficient $\nu \sim U[0.01, 0.02]$. We set the initial condition to be $\hat{\mathbf{u}}^0 = \mathbf{0}$ for the space ROM, which exactly enforces the homogeneous initial conditions of the original problem (4.1). The homogeneous initial conditions are trivially satisfied for the space-time ROM. It is noted that more complex initial conditions can be handled by building an affine trial subspace centered about the initial conditions.

We apply the backward difference scheme for spatial discretization with $N_s = 255$ spatial degrees of freedom. For time discretization, we employ the Crank-Nicolson method with uniform time step $\Delta t = 0.001$ and implicitly solve results for all $N_t = 1/\Delta t = 1000$ time instances. For consistency, we use the same time step in the online phase of both FOM and space ROM approaches and solve the system iteratively at each time step. For the space-time ROM method, we directly obtain the solution at all time steps.

In terms of the ROM solving workflow — see Figure 2.1, we set the number of training samples to be 20 in the offline phase and select a trial subspace that captures at least 99.9999% of the energy in the original snapshot solution set.

4.1. Numerical results. We first consider ROMs equipped with the Monte Carlo sampling approach for the UQ problem (4.1). In what follows, we discuss the computational efficiencies of space ROM and space-time ROM methods in comparison with the FOM solutions. The computation time of a ROM method is calculated as the total running time of:

$$
\text{finding trial subspace} + \text{building ROM system} + \text{solving ROM system.} \tag{4.2}
$$

We test FOM and ROM methods on $10,000$ MC samples and give a detailed time assessment for space and space-time ROMs in Table 4.1. From the table, we can see that the actual solving time (the third column) of the space-time ROM method is 8000 times lower than the time of space ROM, which shows the computational advantages of the space-time ROM approach.

TABLE 4.1
Computation time for each step in the workflow (unit: second).

	find trial subspace	build ROM	solve ROM
space ROM	0.606	0.005	**245.507**
space-time ROM	0.197	0.370	**0.323**

To evaluate ROM methods' efficiency, we employ the speed-up metric, i.e., the result of FOM's computation time divided by ROM's computation time given the same number of Monte Carlo samples. These time results are obtained by the same

experiment for Table 1. It is shown in Figure 4.1 that the MC space-time ROM method has much greater (3000 times) speed-up compared to the space ROM method.
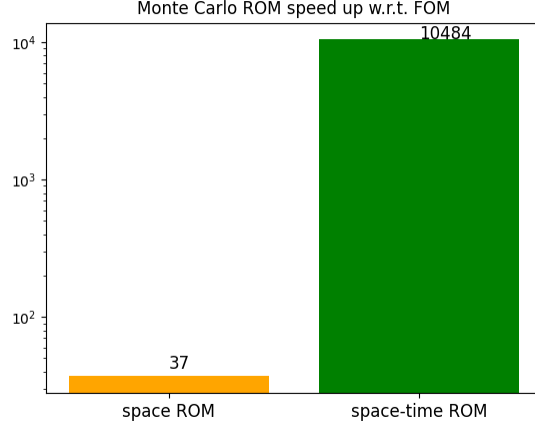

FIG. 4.1. *Monte Carlo ROM speed-up.*

We proceed to investigate the convergence of the space and space-time ROM methods. Given a certain number of Monte Carlo samples, we plot the relative errors of solution mean and variance respectively for FOM, space ROM and space-time ROM methods. The relative errors for solution mean and variance are defined as follows:

$$\frac{\left\|\mathbb{E}[\mathbf{u}] - \text{sample mean}[\mathbf{\Phi}\hat{\mathbf{u}}]\right\|_2}{\left\|\mathbb{E}[\mathbf{u}]\right\|_2}, \quad \frac{\left\|\text{var}[\mathbf{u}] - \text{sample variance}[\mathbf{\Phi}\hat{\mathbf{u}}]\right\|_2}{\left\|\text{var}[\mathbf{u}]\right\|_2},$$

where $\mathbb{E}[\mathbf{u}], \text{var}[\mathbf{u}]$ are the expectation and variance of true solution obtained by 12 million FOM MC samples. sample mean$[\mathbf{\Phi}\hat{\mathbf{u}}]$ and sample variance$[\mathbf{\Phi}\hat{\mathbf{u}}]$ denote the empirical evaluation for approximated solutions by MC sampling. Note that in the FOM setting, the projection basis $\mathbf{\Phi}$ is simply the identity matrix.

Regarding the reproducibility of the experiment, we set a random seed and draw $10,000$ pairs of random samples to run the test. The errors shown in Figure 4.2 are averaged results from 5 different repetitions of the experiment.

In Figure 4.2, all three methods converge well as the number of MC samples increases. The MC FOM method achieves the lowest solution errors, while the MC space-time ROM method in general has the largest errors, especially for the solution variance. We emphasize that the accuracy of both the space and space-time ROMs can be improved by using more basis vectors. Moreover, from the observation on the curves' tendency, the stability of the three methods are rather similar. These numerical results imply space-time approach in general owns good accuracy and stability properties.

Another important UQ propagation method is stochastic Galerkin based on polynomial chaos expansion (PCE) — see Section 3.2 for details. Following the same layout in the Monte Carlo case, we are interested in the computational speed-ups and convergency properties of stochastic Galerkin ROM approaches.

In order to demonstrate the computational efficiency, we plot the space and space-time ROM methods' speed-ups depending on various SG approximation polynomial degrees in Figure 4.3. The speed-up gradually increases as the polynomial degree
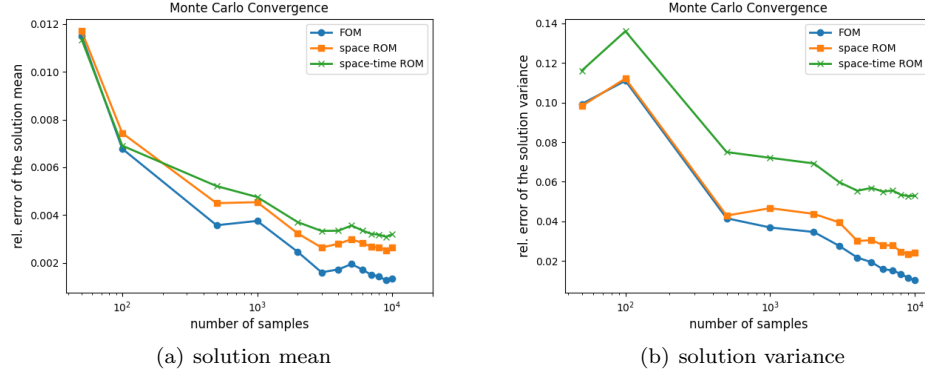
Fig. 4.2. *Monte Carlo convergence: relative error versus number of samples.*

grows. Moreover, the space-time ROM method achieves greater speed-ups than the space ROM.[1]
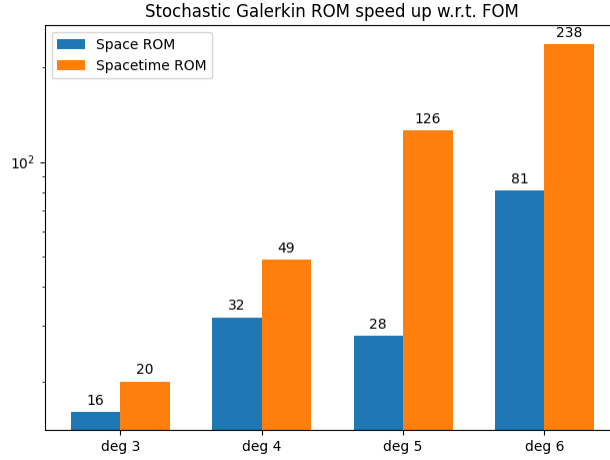

Fig. 4.3. *stochastic Galerkin ROM speed-up.*

We discuss the convergence performances of SG FOM and ROM methods. Similarly, we plot the relative errors of solution mean and variance as quantities of interests in Figure 4.4. The relative errors in the SG setting are defined as:

$$\frac{\left\|\mathbb{E}[\mathbf{u}] - \mathbb{E}[\mathbf{\Phi}\hat{\mathbf{u}}]\right\|_2}{\left\|\mathbb{E}[\mathbf{u}]\right\|_2}, \quad \frac{\left\|\mathrm{var}[\mathbf{u}] - \mathrm{var}[\mathbf{\Phi}\hat{\mathbf{u}}]\right\|_2}{\left\|\mathrm{var}[\mathbf{u}]\right\|_2},$$

where the expectation and variance of the true solution obtained by 12 millions FOM MC simulations. We use integrations like (3.5) to calculate the expectation and variance for approximated solution function $\mathbf{\Phi}\hat{\mathbf{u}}$.

Figure 4.4 shows that FOM, space ROM and space-time ROM implemented by the stochastic Galerkin strategy all converge smoothly as the approximation polynomial

---

[1]Computational times are reported from one run only. We expect that averaging over multiple runs will smooth the observed trends; in particular, the space ROM at polynomial degree 5.

degree increases. Similar to the MC case, space-time SG ROM has slightly larger errors than the other two methods. We again emphasize that the accuracy of the space and space-time ROM can be improved by including more basis vectors.
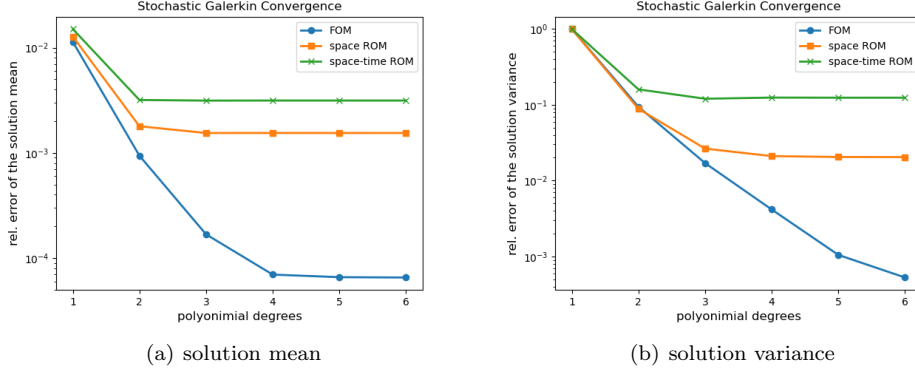

(a) solution mean   (b) solution variance

Fig. 4.4. *stochastic Galerkin convergence: relative error versus maximal polynomial degrees.*

5. Numerical experiment II: 2D parametrized advection-diffusion problem.
In this section, we consider a rather similar advection-diffusion problem to (4.1), but in higher dimensions and with more parameters. Specifically, we consider the following two-dimensional parametrized system with initial and boundary conditions:

$$
\begin{cases}
\dfrac{\partial \boldsymbol{u}}{\partial t} + b\,\cos(\dfrac{\pi}{3})\dfrac{\partial \boldsymbol{u}}{\partial x} + b\,\sin(\dfrac{\pi}{3})\dfrac{\partial \boldsymbol{u}}{\partial y} + \sigma\,\boldsymbol{u} = \nu\left(\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2}\right), \\[2mm]
\boldsymbol{u}((0,y),t,(b,\sigma,\nu)) = 0, \quad \forall\, y \in [0,1], t \in [0,2.5], (b,\sigma,\nu) \in \mathcal{D} \\[2mm]
\boldsymbol{u}((x,0),t,(b,\sigma,\nu)) = 0, \quad \forall\, x \in [0,1], t \in [0,2.5], (b,\sigma,\nu) \in \mathcal{D} \\[2mm]
\boldsymbol{u}((x,y),0,(b,\sigma,\nu)) = 0, \quad \forall\, (x,y) \in [0,1]\times[0,1], (b,\sigma,\nu) \in \mathcal{D}
\end{cases}
\tag{5.1}
$$

where the state $\boldsymbol{u} : [0,1] \times [0,1] \times \mathcal{D} \to \mathbb{R}$. Here, the speed $b \sim \mathcal{N}(0.5, 0.1)$, the reaction coefficient $\sigma \sim U[0.003, 0.005]$ and the diffusion coefficient $\nu \sim U[0.9, 1.1]$.

Similarly, we apply the second-order backward difference scheme for spatial discretization with $N_{s_x} = 63, N_{s_y} = 63$ nodes respectively in the x and y directions, hence total $N_s = N_{s_x} \times N_{s_y} = 3969$ degrees of freedom on the spatial domain. We still apply the Crank-Nicolson method with uniform time step $\Delta t = 0.005$ and implicitly solve results for all $N_t = 2.5/\Delta t = 500$ time instances.

We set the number of training samples to be 20 in the offline phase and select a trial subspace that captures at least 99.9999% of the energy in the original snapshot solution set.

We first show the computational efficiencies of the MC space ROM and space-time ROM methods. The computation time of a ROM method is calculated as same as in (4.2).

In order to identify a spatial trial subspace for the numerical solution of (5.1), one needs to do a singular value decomposition on a $N_s \times N_t N_{\text{train}} = 3969 \times 10000$ snapshot matrix that has a large number of columns. This procedure is so expensive

that it may even exceed the actual computation time of solving the reduced system. Therefore, we propose the random range finder (RRF) method [7] for the spatial subspace finding to reduce this overhead complexity. The methodology of RRF is introduced as follows:

1. fix the number of truncated columns $\hat{K}$;
2. right multiply a Gaussian testing matrix $\mathbf{G} \sim \mathcal{N}(0,1)^{N_t N_{\text{train}} \times \hat{K}}$ on the training solutions $\mathbf{U}_{\text{train}} \in \mathbb{R}^{N_s \times N_t N_{\text{train}}}$;
3. compute the svd for the resulting matrix with a much reduced number of columns $\mathbf{U}_{\text{train}}\mathbf{G} \in \mathbb{R}^{N_s \times \hat{K}}$ and keep the left singular vectors to be the basis.

5.1. Numerical results. In Table 5.1, we test FOM and ROM methods on 2000 MC samples and provide a time report for space and space-time ROMs. For the setup of RRF, we pick the number of truncated columns to be $\hat{K} = 20$ versus $K_{\text{true}} = 12$ applying a conventional svd on $\mathbf{U}_{\text{train}}$.

Some important observations from Table 5.1 are: (1) The solving time (the third table column) of the space-time ROM method is 300 times lower than the time of space ROM; (2) Applying RRF for spatial trial subspace reduces almost 1000 times of the computation time compared to the conventional spatial subspace finding — see the first table column.

TABLE 5.1
Computation time for each step in the workflow (unit: second).

	find trial subspace	build ROM	solve ROM
space ROM	**77.908**	0.006	63.059
space ROM (RRF)	**0.822**	0.007	76.351
space-time ROM	4.016	1.055	**0.227**

From the same experiment for Table 5.1, we report the ROM speed-ups in Figure 5.1. The speed-up's metric is the same as in Figure 4.1. We again observe that the space-time ROM method is the fastest among the three methods.
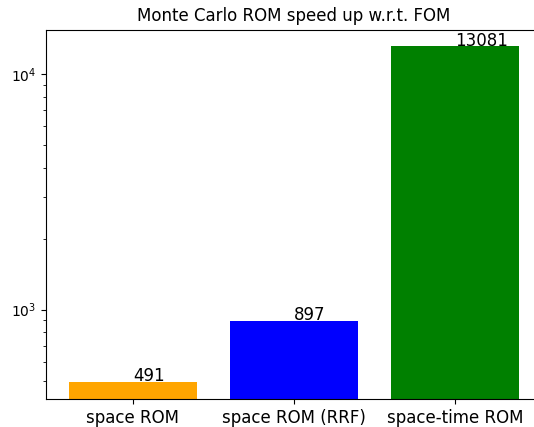

FIG. 5.1. *Monte Carlo ROM speed-up.*

We next discuss the convergence property. Figure 5.2 shows the convergence of

the FOM and ROM methods with respect to the number of MC samples. Similar to the presentation in Figure 4.2 for the 1D UQ problem, we make the y-axis to be the solution errors for relative mean and variance with respect to the number of MC samples in the x-axis. The experiment runs on a total 2000 random instances of $(b, \sigma, \nu) \in \mathcal{D}$ in eq. (5.1). The errors are averaged results from 3 repetitions of the test based on different random seeds.

In general, all four MC solving methods converge with the same trend. MC space-time ROM is the least accurate method and has the largest error means and variances. This can be considered as a trade-off of its computational efficiency and accuracy. The error results of FOM and both two variants of space ROM methods are very close and almost overlapping, which proves the high accuracy of MC space ROM.

(a) solution mean   (b) solution variance
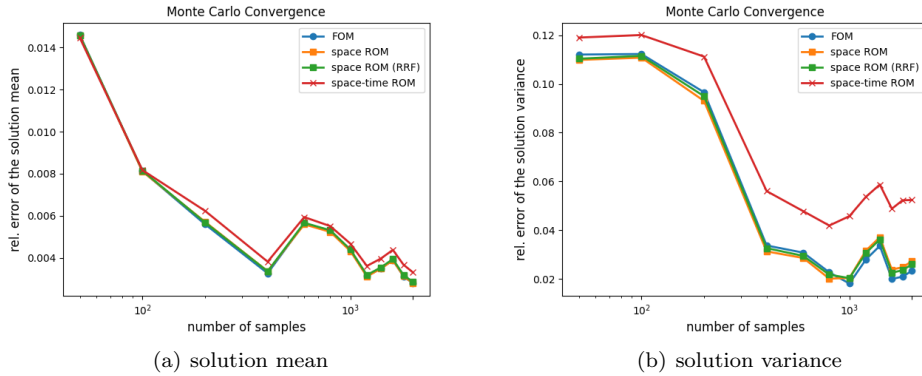
Fig. 5.2. *Monte Carlo convergence: relative error versus number of samples.*

We employ stochastic Galerkin ROM methods for the 2D advection-diffusion problem (5.1). Please note that it is commonly recognized that running a full-order model with stochastic Galerkin is too time consuming, especially in large-scale problems, thus we skip its implementation in this subsection.

We are interested in the computation time of space and space-time ROMs implemented with the stochastic Galerkin strategy. In Figure 5.3, we report the ROM solving times (the last step in the workflow Figure 2.1) — other workflow steps are already studied above in the MC case. We use the space ROM implemented by conventional subspace finding as the representative space ROM approach.

Figure 5.3 demonstrates the time comparisons of the SG space and space-time ROM. It is clearly shown that SG space-time ROM method is roughly $40 - 800$ times faster than SG space ROM. As the approximating polynomial degrees grows, this solving time discrepancy becomes more pronounced.

We finally discuss the convergence and accuracy properties of the SG ROMs. From Figure 5.4, we can observe that space ROM, space ROM with random range finder and space-time ROM methods all converge very smoothly as the approximating polynomial degree tends to increase. In terms of accuracy, the SG space ROM method with conventional trial subspace finding achieves the highest precision, while the SG space-time ROM overall has bigger error means and variances than space ROMs.

6. Conclusions and future directions. In this work, we have studied and showed the significant computational advantages of the space-time ROM method in-
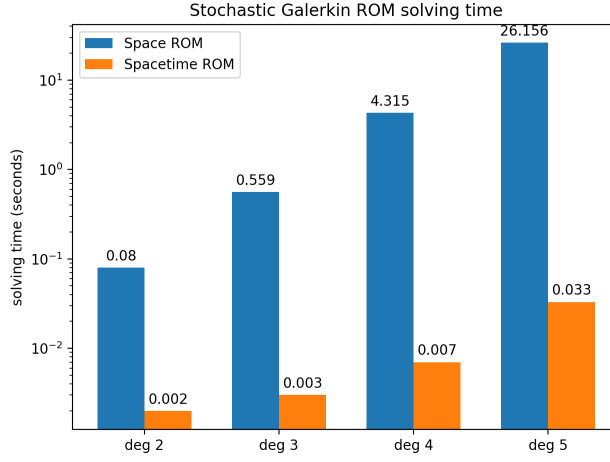
Fig. 5.3. *stochastic Galerkin ROM solving time.*

(a) solution mean   (b) solution variance
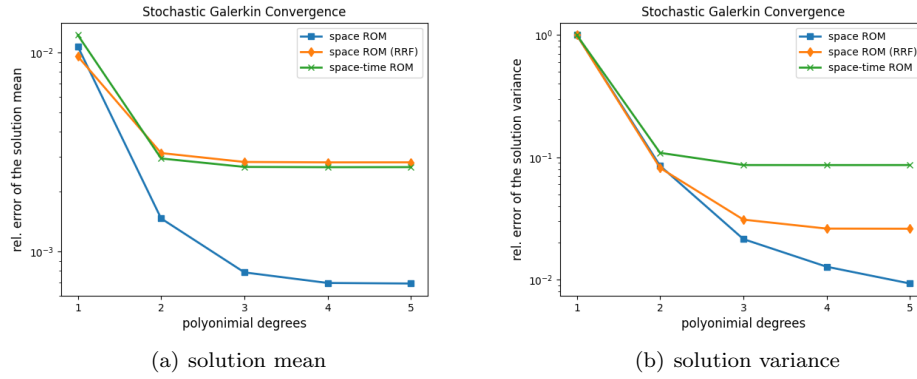
Fig. 5.4. *stochastic Galerkin convergence: relative error versus maximal polynomial degrees.*

corporated with Monte Carlo and stochastic Galerkin techniques. By testing our proposed method on parametrized 1D and 2D advection-diffusion problems, we provided thorough numerical experiments to demonstrate both computational and convergence properties of the space-time ROM method and compared them with the FOM and space ROM methods. The numerical performance showed that the space-time ROM method achieved remarkably high efficiency compared to the other two approaches. However, it also suffered a small loss of solution accuracy as suggested in convergence plots.

We finally lay out the remaining future works.

1. We hope to directly compare the computational cost and theoretical performance of SG ROMs to MC ROMs.
2. Since stochastic Galerkin is an intrusive method, implementing the space-time ROM by other non-intrusive UQ propagation methods such as stochastic collocation could be a promising direction.
3. We also want to extend the proposed method to challenging nonlinear systems where explicitly forming reduced operators is more difficult.

4. Developing theoretical error bounds for the proposed method is another sound future plan.

5. It is interesting to explore the advanced sparse grid strategy to further reduce the computational complexity.

REFERENCES

[1] I. Babuska, R. Tempone, and G. E. Zouraris, *Galerkin finite element approximations of stochastic elliptic partial differential equations*, SIAM Journal on Numerical Analysis, 42 (2004), pp. 800–825.

[2] M. Baumann, P. Benner, and J. Heiland, *Space-time galerkin pod with application in optimal control of semilinear partial differential equations*, SIAM Journal on Scientific Computing, 40 (2018), pp. A1611–A1641.

[3] P. Benner, M. Ohlberger, A. Cohen, and K. Willcox, *Model Reduction and Approximation*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017.

[4] T. Bui-Thanh, M. Damodaran, and K. Willcox, *Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition*, AIAA Journal, 42 (2004), pp. 1505–1516.

[5] Y. Choi and K. Carlberg, *Space–time least-squares Petrov–Galerkin projection for nonlinear model reduction*, SIAM Journal on Scientific Computing, 41 (2019), pp. A26–A58.

[6] B. Gershgorin and A. J. Majda, *Quantifying uncertainty for climate change and long-range forecasting scenarios with model errors. part i: Gaussian models*, Journal of Climate, 25 (2012), pp. 4523 – 4548.

[7] N. Halko, P. Martinsson, and J. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288.

[8] N. Lamorte, P. P. Friedmann, B. Glaz, A. J. Culler, A. R. Crowell, and J. J. McNamara, *Uncertainty propagation in hypersonic aerothermoelastic analysis*, Journal of Aircraft, 51 (2014), pp. 192–203.

[9] K. Lee, K. Carlberg, and H. C. Elman, *Stochastic least-squares petrov–galerkin method for parameterized linear systems*, SIAM/ASA Journal on Uncertainty Quantification, 6 (2018), pp. 374–396.

[10] D. J. Lucia, P. S. Beran, and W. A. Silva, *Reduced-order modeling: new approaches for computational physics*, Progress in Aerospace Sciences, 40 (2004), pp. 51–117.

[11] J. Oliver, M. Caicedo, A. Huespe, J. Hernandez, and E. Roubin, *Reduced order modeling strategies for computational multiscale fracture*, Computer Methods in Applied Mechanics and Engineering, 313 (2017), pp. 560–595.

[12] D. Pasetto, M. Putti, and W. Yeh, *A reduced-order model for groundwater flow equation with random hydraulic conductivity: Application to monte carlo methods*, Water Resources Research, 49 (2013).

[13] I. Udagedara, B. Helenbrook, A. Luttman, and S. Mitchell, *Reduced order modeling for accelerated monte carlo simulations in radiation transport*, Applied Mathematics and Computation, 267 (2015).