SAND2022-13286

SANDIA REPORT

SAND2022-xxxx
Printed September 2022

Sandia National Laboratories

Comprehensive uncertainty quantification (UQ) for full engineering models by solving probability density function (PDF) equation

Hemanth Kolla, Saibal De, Reese E. Jones, Michael A. Hansen, Julia A. Plews

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@osti.gov
Online ordering:	http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.gov
Online order:	https://classic.ntis.gov/help/order-methods

Comprehensive uncertainty quantification (UQ) for full engineering models by solving probability density function (PDF) equation

Hemanth Kolla
Scalable Modeling and Analysis
Sandia National Laboratories
P.O. Box 0969
Livermore, CA 94550
hnkolla@sandia.gov

Saibal De
Scalable Modeling and Analysis
Sandia National Laboratories
P.O. Box 0969
Livermore, CA 94550
sde@sandia.gov

Reese E. Jones
Mechanics of Materials
Sandia National Laboratories
P.O. Box 0969
Livermore, CA 94550
rjones@sandia.gov

Michael A. Hansen
Computational Thermal and Fluid Mechanics
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185
mahanse@sandia.gov

Julia A. Plews
Computational Solid Mechanics and Structural Dynamics
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185
japlews@sandia.gov

SAND2022-xxxx

ABSTRACT

This report details a new method for propagating parameter uncertainty (forward uncertainty quantification) in partial differential equations (PDE) based computational mechanics applications. The method provides full-field quantities of interest by solving for the joint probability density function (PDF) equations which are implied by the PDEs with uncertain parameters. Full-field uncertainty quantification enables the design of complex systems where quantities of interest, such as failure points, are not known *apriori*. The method, motivated by the well-known probability density function (PDF) propagation method of turbulence modeling, uses an ensemble of solutions to provide the joint PDF of desired quantities at every point in the domain. A small subset of the ensemble is computed exactly, and the remainder of the samples are computed with approximation of the driving (dynamics) term of the PDEs based on those exact solutions. Although the proposed method has commonalities with traditional interpolatory stochastic collocation methods applied directly to quantities of interest, it is distinct and exploits the parameter dependence and smoothness of the dynamics term of the governing PDEs. The efficacy of the method is demonstrated by applying it to two target problems: solid mechanics explicit dynamics with uncertain material model parameters, and reacting hypersonic fluid mechanics with uncertain chemical kinetic rate parameters. A minimally invasive implementation of the method for representative codes SPARC (reacting hypersonics) and NimbleSM (finite- element solid mechanics) and associated software details are described. For solid mechanics demonstration problems the method shows order of magnitudes improvement in accuracy over traditional stochastic collocation. For the reacting hypersonics problem, the method is implemented as a streamline integration and results show very good accuracy for the approximate sample solutions of re-entry flow past the Apollo capsule geometry at Mach 30.

ACKNOWLEDGMENT

We are grateful to the contributions of Michael Redle of North Carolina State University, who worked on this project as part of a graduate summer internship. We are also grateful to Ross Wagnild for early guidance on SPARC and useful target problems for uncertainty quantification. This work was funded by the Laboratory Directed Research & Development (LDRD) program, under the Engineering Science Research Foundation (ESRF) investment area.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Table 0-1.

Abbreviation	Definition
UQ	Uncertainty Quantification
PDE	Partial Differential Equations
PDF	Probability Density Function
SPARC	Sandia Parallel Aerodynamics and Reentry Code
SC	Stochastic Collocation

1. INTRODUCTION

Design of high consequence systems relies on robust uncertainty quantification (UQ) of modeling and simulation of coupled physics and multi-scale phenomena. Quantifying uncertainty from myriad physical and algorithmic sources is challenging and, currently, computationally costly. The influence of uncertainties is particularly severe under extreme conditions and abnormal environments, precisely where reliability requirements are most important. Mission relevant engineering simulations involve complex models for which existing UQ approaches are so infeasible as to prevent even being attempted. The focus of this project is a novel approach for propagating parameter uncertainty, *i.e.* forward uncertainty quantification, in mission relevant engineering simulation codes that solve computational mechanics problems. In this chapter we first discuss the importance and challenges of forward UQ and present a high-level overview of existing UQ approaches. We then motivate the need for an alternative approach that can overcome the computational difficulties and discuss the goals of our project.

1.1. Forward UQ and existing methods

Generally speaking, UQ is a relatively mature field of study with well-established theory and methods [2]. Determining the uncertainty of model parameters, given calibration data, and propagating the parameter uncertainty to design quantities of interest (QoIs) are two of the primary tasks in UQ. In this work we focus on forward propagation of uncertainty given model parameter uncertainties. Brute force sampling of the input parameter distributions and propagating through the simulation-based forward map to obtain samples of the QoIs is a straightforward albeit expensive method of forward UQ. Given the computational expense of simulations used for engineering analysis and design, UQ has traditionally relied heavily on efficient and sufficiently accurate surrogate models of specific QoIs as functions of the uncertain parameters. These surrogate models fall into broad categories: (a) regression, such as Gaussian processes [3] and radial basis functions [4], which are relatively expensive; (b) projection, such as polynomial chaos expansions (PCE) [5], which require the computation of integral-based inner products; and (c) interpolation, such as stochastic collocation [6], which can be relatively inexpensive and constructed independently of particular values of output. Given the need to model time-evolving quantities the expense of constructing a surrogate model can be considerable. The text by R. Smith [7] provides survey of alternative methods.

In particular, stochastic collocation (SC) is a widely-used method. A number of reviews survey the state of the art. For instance, Eldred [8] compared non-intrusive polynomial chaos expansion (PCE) with stochastic collocation for the tasks of uncertainty quantification, design under uncertainty, and reliability analysis. Various methods of constructing these representations, such as regression and projection, were explored, and their advantages and costs were compared. More recently, Xiu [6] gave a comprehensive overview of SC in interpolation, regression, and pseudo-projection modes; and outlined tensor product and sparse grid schemes. The work also described the use of unstructured samples via "least interpolation." The work of Babuška *et al.* [9] provides a thorough analysis of convergence and correspondence with stochastic Galerkin techniques for elliptic PDEs such as equilibrium elasticity with random coefficients.

An alternate category of surrogate models–reduced order models (ROMs) [10, 11, 12]–aim to directly reduce the complexity of the full computational model by projecting onto a spatio-temporal subspace. This can make a brute force sampling of the input parameter distributions more feasible since the ROM surrogates are less expensive than the full order model by orders of magnitude. Further cost savings can be realized by also projecting onto a subspace of the stochastic dimensions corresponding to the uncertain parameters [13]. Projection based ROMs typically rely on an *offline* stage for identifying the subspace, which can be expensive, and PDE non-linearities pose challenges for the degree of reduction achievable for satisfactory accuracy. Efficient methods to handle PDE non-linearities while still saving computational expense is an active research topic [14].

1.2. Goals of the project

Forward UQ is challenging when the computational model is expensive owing to non-linear physics and/or the sources of uncertainty are many (large stochastic dimension). The current UQ paradigms offer a choice between extreme computational burden or simplification of a complex computational model; the current methods either require numerous solves of the complex model or a transformation of the governing partial differential equations (PDEs) resulting in deep and pervasive code overhaul. The goal of this project is a novel paradigm for forward UQ that overcomes these and other challenges while remaining computationally feasible and providing comprehensive uncertainty information. *This is achieved by directly solving the probability density function (PDF) equations resulting from the governing physics: partial differential equations (PDEs) with uncertain parameters.* This provides full probabilistic field information, i.e., the joint PDF of the solution variables at every point in the computational domain, which is comprehensive and more information-rich than traditional methods since all statistics of interest are derivable from the full field joint PDF.

The goals of this project are as follows:

- To expound on the PDF equation approach for forward UQ.
- To develop computationally efficient approaches for solving the PDF equations for forward UQ.
- To tailor the approach for two target applications–computational solid mechanics and reacting hypersonic fluid mechanics–and study the cost versus accuracy tradeoffs.
- To present software implementations of the method compatible with exemplar Sandia codes: (1) Sandia Parallel Aerodynamics and Reentry Code (SPARC) [15], and (2) NimbleSM [16], the open-source Lagrangian finite element code.

2. TECHNICAL APPROACH

The motivation for solving PDF equations comes primarily from the field of turbulent combustion. Turbulence, an inherently stochastic physical process, can be seen as a source of aleatory uncertainty. The governing equations for turbulent reacting flow are highly non-linear PDEs: Navier-Stokes equations governing velocity, and convection- diffusion-reaction balance equations governing chemical species and enthalpy conservation. The most common approaches for computing turbulent reacting flows adopt a deterministic viewpoint, solving the governing PDEs in the original (direct numerical simulations (DNS)) or slightly modified forms (large eddy simulations (LES), Reynolds-averaged Navier-Stokes (RANS)); an alternative, fundamentally stochastic, approach solves a transport equation for the joint PDF of the solution variables (velocity, species composition, enthalpy) which is derived exactly from the original PDEs. Pope [17] describes the computational advantages of the PDF equation approach, including ability to handle stiff non-linear physics and favorable scaling for large-dimensional PDE systems. In contrast parameter uncertainty is a source of epistemic uncertainty, but the same advantages of the PDF equation approach can be brought to bear for forward UQ.

2.1. PDF equations for forward UQ

Computational mechanics are typically governed by a set of PDEs of the form

$$\frac{D\boldsymbol{\phi}}{Dt} = \boldsymbol{R}(\boldsymbol{\phi},\boldsymbol{\lambda};\boldsymbol{X},t), \tag{2.1}$$

where D/Dt is the material derivative, $\boldsymbol{X}$ are the spatial coordinates, t is time, $\boldsymbol{\lambda}$ are model parameters, $\boldsymbol{\phi}(\boldsymbol{\lambda};\boldsymbol{X},t)$ is the solution vector, and $\boldsymbol{R}(\boldsymbol{\phi},\boldsymbol{\lambda};\boldsymbol{X},t)$ is the driving term encapsulating all the dynamics governing conservation. If the model parameters are uncertain they can be represented by a finite dimensional random vector $\boldsymbol{\lambda}$ with probability distribution function (PDF) $p_{\boldsymbol{\lambda}}(\boldsymbol{\lambda})$. The stochasticity of the model parameters imply the state vector $\boldsymbol{\phi}$ is also a random variable. Its PDF, $p_{\boldsymbol{\phi}}(\boldsymbol{\phi};\boldsymbol{X},t)$, is defined at each material point $\boldsymbol{X}$ and time t. In forward uncertainty propagation, we seek to learn this state distribution $p_{\boldsymbol{\phi}}$ for a given parameter distribution $p_{\boldsymbol{\lambda}}$. The notation here follows that of Pope [18]: the semicolon distinguishes the random variables, which are placed on the left of the semicolon, from independent deterministic variables which are on the right. Accordingly, $\boldsymbol{R}(\boldsymbol{\phi},\boldsymbol{\lambda};\boldsymbol{X},t)$ denotes $\boldsymbol{R}$ is a function of the random variables $(\boldsymbol{\phi},\boldsymbol{\lambda})$ and independent deterministic variables $(\boldsymbol{X},t)$, and $p_{\boldsymbol{\phi}}(\boldsymbol{\phi};\boldsymbol{X},t)$ denotes the PDF of $\boldsymbol{\phi}$ as a function of $(\boldsymbol{X},t)$. Moreover, $\boldsymbol{\phi}$ is typically a vector with multiple components, and hence $p_{\boldsymbol{\phi}}(\boldsymbol{\phi};\boldsymbol{X},t)$ denotes a joint PDF of all the components of $\boldsymbol{\phi}$ at any given $(\boldsymbol{X},t)$.

The exact evolution equation for the state PDF, $p_{\boldsymbol{\phi}}$, can be derived formally [18];

$$\frac{Dp_{\boldsymbol{\phi}}}{Dt} = -\nabla_{\boldsymbol{\phi}} \cdot \left[\langle \boldsymbol{R} | \boldsymbol{\phi}, \boldsymbol{\lambda}; \boldsymbol{X}, t \rangle p_{\boldsymbol{\phi}} \right], \tag{2.2}$$

where $\nabla_{\boldsymbol{\phi}}$ denotes the gradient operator with respect to $\boldsymbol{\phi}$, $\langle \boldsymbol{R} | \boldsymbol{\phi}, \boldsymbol{\lambda}; \boldsymbol{X}, t \rangle$ is the expectation of $\boldsymbol{R}$ conditioned on $\boldsymbol{\phi}$ and $\boldsymbol{\lambda}$, and the ensemble average $\langle \bullet \rangle$ is marginalization over other stochastic variables, *e.g.* an uncertain boundary condition or inherent chaotic dynamics (*e.g.* turbulence). Note that this is for the case of parameter PDF, $p_{\boldsymbol{\lambda}}$, having no dependence on $\boldsymbol{X}$ and t. Following the steps in [17] one can first derive an evolution equation for the joint PDF, $p_{\boldsymbol{\phi},\boldsymbol{\lambda}}$, in a straightforward manner by recognizing that $\boldsymbol{\lambda}$ is not governed by any dynamics (derivatives of $\boldsymbol{\lambda}$ with respect to $\boldsymbol{X}$, t are zero). By marginalizing the $p_{\boldsymbol{\phi},\boldsymbol{\lambda}}$ evolution equation over $\boldsymbol{\lambda}$ one obtains Eq. (2.2). The derivation, and the resulting equation, would be slightly different, but no more complicated, if $p_{\boldsymbol{\lambda}}$ were a function of $\boldsymbol{X}$ or t. In other words the PDF equation approach can naturally admit the case of model parameters being spatio-temporal random fields. For this project we only focus on the case of a universal $p_{\boldsymbol{\lambda}}$. The PDF equation approach for random field parameters have been studied in previous works [19, 20, 21].

For otherwise deterministic systems (where the only stochasticity is due to parameter uncertainty) the expectation simplifies to $\boldsymbol{R}$ evaluated at the state $\boldsymbol{\phi}$ for parameter value $\boldsymbol{\lambda}$ at position $\boldsymbol{X}$ and time t. So (2.2) becomes

$$\frac{Dp_{\boldsymbol{\phi}}}{Dt} = -\nabla_{\boldsymbol{\phi}} \cdot \left[\boldsymbol{R}(\boldsymbol{\phi}, \boldsymbol{\lambda}; \boldsymbol{X}, t) p_{\boldsymbol{\phi}} \right], \tag{2.3}$$

where the right-hand side is the sensitivity of the probability weighted mean of $\boldsymbol{R}$ to the solution $\boldsymbol{\phi}$. Note that the sensitivity of $\boldsymbol{R}$ to $\boldsymbol{\lambda}$ is from two sources: (a) directly from the explicit dependence of $\boldsymbol{R}$ on $\boldsymbol{\lambda}$ through the embedded constitutive/closure model, and (b) from the trajectory $\boldsymbol{\phi}(\boldsymbol{\lambda}; \boldsymbol{X}, t)$ that evolves differently depending on the sample $\boldsymbol{\lambda}$ of $\boldsymbol{\Lambda}$, as the chain rule

$$\boldsymbol{\nabla}_{\boldsymbol{\lambda}} \boldsymbol{R}(\boldsymbol{\phi}, \boldsymbol{\lambda}; \boldsymbol{X}, t) = \boldsymbol{\partial}_{\boldsymbol{\phi}} \boldsymbol{R}(\boldsymbol{\phi}, \boldsymbol{\lambda}; \boldsymbol{X}, t)\, \boldsymbol{\partial}_{\boldsymbol{\lambda}} \boldsymbol{\phi} + \boldsymbol{\partial}_{\boldsymbol{\lambda}} \boldsymbol{R}(\boldsymbol{\phi}, \boldsymbol{\lambda}; \boldsymbol{X}, t) \tag{2.4}$$

illustrates. Unfortunately, the evolution equations (2.2) and (2.3), themselves PDES, are high dimensional even for small-scale problems, and conventional PDE discretization techniques to solve these becomes computationally intractable.

2.2. Solving PDF equations

For turbulent flows Pope [17] proposes a hybrid Eulerian-Lagrangian strategy for solving a stochastically equivalent system of the PDF equations. The approach, shown in Figure 2-1, proceeds as follows:

1. Sample the stochastic dimension *i.e.* drawing samples $\{\boldsymbol{\lambda}_i^s : 1 \leq i \leq N_s\}$ from the parameter distribution $p_{\boldsymbol{\lambda}}$,
2. For each sample initialize a Lagrangian state $\{\hat{\boldsymbol{x}}_i^s, \boldsymbol{\phi}_i^s, \boldsymbol{\lambda}_i^s\}$ where the initial position, $\hat{\boldsymbol{x}}_i^s(t=0)$, and state, $\boldsymbol{\phi}_i^s(t=0)$, are consistent with the initial and boundary conditions of the problem.

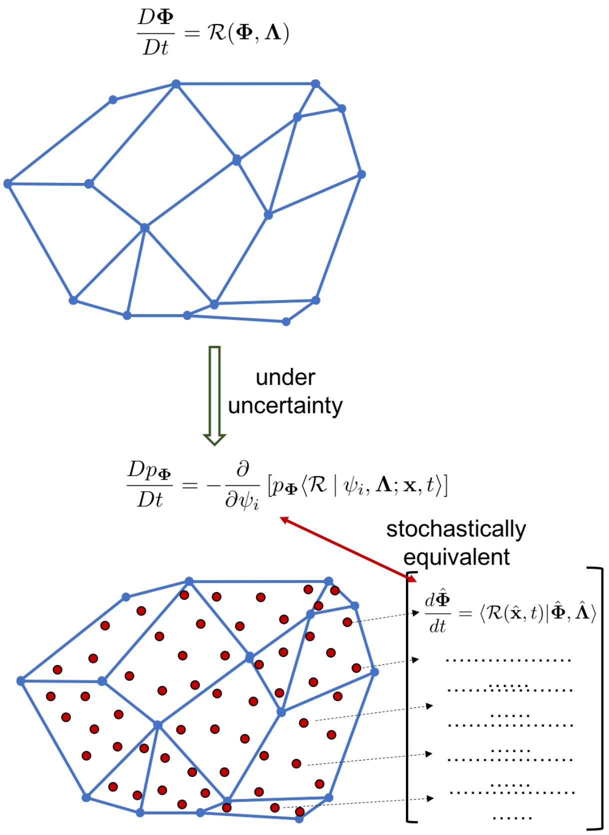

$\frac{D\mathbf{\Phi}}{Dt} = \mathcal{R}(\mathbf{\Phi}, \mathbf{\Lambda})$
under uncertainty
$\frac{Dp_{\mathbf{\Phi}}}{Dt} = -\frac{\partial}{\partial \psi_i}\left[p_{\mathbf{\Phi}}\langle \mathcal{R} \mid \psi_i, \mathbf{\Lambda}; \mathbf{x}, t\rangle\right]$
stochastically equivalent
$\frac{d\hat{\mathbf{\Phi}}}{dt} = \langle \mathcal{R}(\hat{\mathbf{x}}, t) | \hat{\mathbf{\Phi}}, \hat{\mathbf{\Lambda}}\rangle$

Figure 2-1. A hybrid Eulerian-Lagrangian PDF equation framework. A traditional mesh discretized deterministic PDE system (top) can be augmented, under uncertainty, by particles sampling the stochastic dimensions (bottom) to evolve the joint PDF equation by evolving particle ODEs.

3. Compute the corresponding sample solution trajectories $\{\boldsymbol{\phi}_i^s(\boldsymbol{X},t) \equiv \boldsymbol{\phi}(\boldsymbol{\lambda}_i^s;\boldsymbol{X},t)\}$ by time stepping Eq. (2.1). The position of each sample $\hat{\boldsymbol{x}}_i^s$ is updated with velocity which is usually a subset of $\boldsymbol{\phi}_i^s$ for mechanics.

4. Construct the state PDF $p_{\boldsymbol{\phi}}(\boldsymbol{\phi};\boldsymbol{X},t)$ from these sample solutions. This is typically done over the ensemble of solution trajectories for which $\boldsymbol{X} = \boldsymbol{x}_i^s$.

This route represents a means to approximate the state PDF, and has the flavor of traditional Monte-Carlo (MC) methods. Hence, we encounter the familiar slow convergence rate $O(1/\sqrt{N_s})$ of the empirical PDF to the true distribution. However, the advantage is that the number of samples scales linearly with the dimensionality of the joint PDF $p_{\boldsymbol{\phi}\boldsymbol{\lambda}}$, which makes it attractive if $\boldsymbol{\phi}$ and/or $\boldsymbol{\lambda}$ are large. Given the slow convergence, the main challenge is to generate the sample trajectories $\{\boldsymbol{\phi}_i^s(\boldsymbol{X},t) : 1 \leq i \leq N_s\}$ more efficiently than directly computing the trajectories, while maintaining sufficient accuracy.

Two aspects of this solution strategy present the main research challenges that need to be addressed:

- Sampling: Even for the simplest problems computing the solutions *exactly* for all N_s samples will be infeasible; it will be tantamount to solving Eq. (2.1) N_s times. Our proposed solution is to *solve a very small subset exactly and all the remaining samples approximately.* We denote the exact samples by $\{\boldsymbol{\lambda}_j^c : 1 \leq j \leq N_c, N_c \ll N_s\}$, and the approximate samples by $\{\boldsymbol{\lambda}^a : \boldsymbol{\lambda}^a \in \boldsymbol{\lambda}_i^s, \boldsymbol{\lambda}^a \neq \boldsymbol{\lambda}_j^c\}$. This will make the overall cost of the method sub-linear with respect to N_s and, by extension, sub-linear with respect to the dimensionality of the uncertain parameters which is key for forward UQ to be cost-effective.

- Closure: A related challenge is "closure" *i.e.* providing the information required to time advance all the sample solutions. Our proposal is to *use the machinery of the parent code to compute the dynamics term of the exact solution samples, $\boldsymbol{R}(\boldsymbol{\phi}_j^c, \boldsymbol{\lambda}_j^c, \hat{\boldsymbol{x}}_j^c)$, and provide necessary approximations from these for the dynamics terms of the approximate samples (corresponding to $\boldsymbol{\lambda}^a$).*

To summarize, for a small subset ($\ll N_s$) of samples we compute the exact $\boldsymbol{R}(\boldsymbol{\phi}, \boldsymbol{\lambda}; \boldsymbol{X}, t)$ using the parent code machinery (space/time discretizations, constitutive models) and approximate the $\boldsymbol{R}$ for the remaining samples, and time advance all with (2.1). The main distinction between our method and existing methods is simple. While existing methods attempt to approximate the function $\boldsymbol{\phi}_a(\boldsymbol{X}, t) = \boldsymbol{\phi}(\boldsymbol{\lambda}_a; \boldsymbol{X}, t)$ via projection/regression/collocation in the stochastic dimension ($\boldsymbol{\lambda}$), we propose to approximate the functional dependence of $\boldsymbol{R}$ on $\boldsymbol{\lambda}$, and obtain $\boldsymbol{\phi}_a(\boldsymbol{X}, t)$ by integrating (2.1). We hypothesize that since the functional dependence of $\boldsymbol{\phi}$ on $\boldsymbol{\lambda}$ includes non-linearities stemming from terms involving $\boldsymbol{\lambda}$ as well as other non-linearities in the problem, it is more difficult to approximate than the functional dependence of $\boldsymbol{R}$ on $\boldsymbol{\lambda}$. Furthermore, our approximation will preserve boundary conditions and momentum conservation in the approximate solutions. Since most codes typically compute $\boldsymbol{R}$ but do not save it, our method needs to be intrusive. Nevertheless, the prospect of solving for additional samples on-line with the primary solution is not as drastic as it might seem, and lends itself to many computational savings. Recent work on ensemble stochastic propagation methods [22, 23] has shown that aspects like contiguous memory access, data reuse, better vectorization, and aggregation of interprocessor communication yield significant cost savings over the whole ensemble.

A fundamental difference between computational solid and fluid mechanics is worth pointing out. Typically, computational fluid mechanics adopts an Eulerian viewpoint and the PDEs are solved in an Eulerian frame of reference. The schematic in Figure 2-1 hews closer to this scenario. In contrast solid mechanics adopts a Lagrangian formulation where the displacement of a material point is itself part of the solution vector and the vector $\boldsymbol{X}$ denotes a reference (typically initial) configuration or position. Regardless, the PDF equation approach can be consistently employed for both applications.

3. RESEARCH METHODOLOGY

The main thrust of our research is simply to sample efficiently in the stochastic dimensions (uncertain parameter, $\boldsymbol{\lambda}$, space) such that a bare minimum number of sample solutions are computed exactly and use these to approximate the dynamics term, $\boldsymbol{R}$, for the rest of the samples for time advancing. Of the approaches listed in section 1.1 stochastic collocation is the most natural starting point for our purpose. In this chapter we first present a general, broadly applicable, set of approximation methods we investigated based off of stochastic collocation. Separately, we also present an alternative approximation method that exploits the specifics of the hypersonic re-entry problem of interest. This is to demonstrate that the PDF equation approach, by virtue of adhering closely to the original governing PDEs, can be tailored to the physics of the target problem.

3.1. An Overview of Stochastic Collocation

Stochastic collocation (SC) is a simple technique for approximating in the stochastic dimensions from a few samples. In this framework, we start with a few collocation points $\{\boldsymbol{\lambda}_j^c : 1 \leq j \leq N_c\}$ that are typically constructed based on the range of the parameter distribution $p_{\boldsymbol{\lambda}}$. For any general quantity of interest, $\boldsymbol{Q}$, that is a function of $\boldsymbol{\lambda}$, exact computations at the collocation points $\{\boldsymbol{Q}_j^c \equiv \boldsymbol{Q}(\boldsymbol{\lambda}_j^c) : 1 \leq j \leq N_s\}$ and interpolation coefficients $\{c_{ij} = c(\boldsymbol{\lambda}_i^s, \boldsymbol{\lambda}_j^c) : 1 \leq i \leq N_s, 1 \leq j \leq N_c\}$ computed from the two parameter sets, provide an approximation for any value of $\boldsymbol{\lambda}_i^s$

$$\boldsymbol{Q}(\boldsymbol{\lambda}_i^s) \approx \widetilde{\boldsymbol{Q}}_i^s = \sum_j c_{ij} \boldsymbol{Q}_j^c. \tag{3.1}$$

Since the interpolation is completely independent of $\boldsymbol{Q}$, the coefficients $\{c_{ij} : 1 \leq i \leq N_s, 1 \leq j \leq N_c\}$ can be computed *a priori* and reused throughout a simulation. This makes SC a very low-cost and online surrogate model, especially when the number of collocation points is far smaller than the number of sample points ($N_c \ll N_s$).

In one-dimensional parameter spaces $\lambda \in [\lambda_{\min}, \lambda_{\max}]$, the collocation points are typically chosen from regular grids, *e.g.* the popular nested Clenshaw-Curtis grid of level $l \in \{0, 1, 2, \ldots\}$ is given by

$$\lambda_j^c = \lambda_{\max} - (\lambda_{\max} - \lambda_{\min}) \cos \frac{(j-1)\pi}{2^l}, \quad 1 \leq j \leq 2^l + 1 \tag{3.2}$$

and the interpolation coefficients are computed using Lagrange's scheme:

$$c_{ij} = c(\lambda_i^s, \lambda_j^c) = \prod_{\substack{j'=1 \\ j' \neq j}}^{N_c} \frac{\lambda_i^s - \lambda_{j'}^c}{\lambda_j^c - \lambda_{j'}^c} \tag{3.3}$$

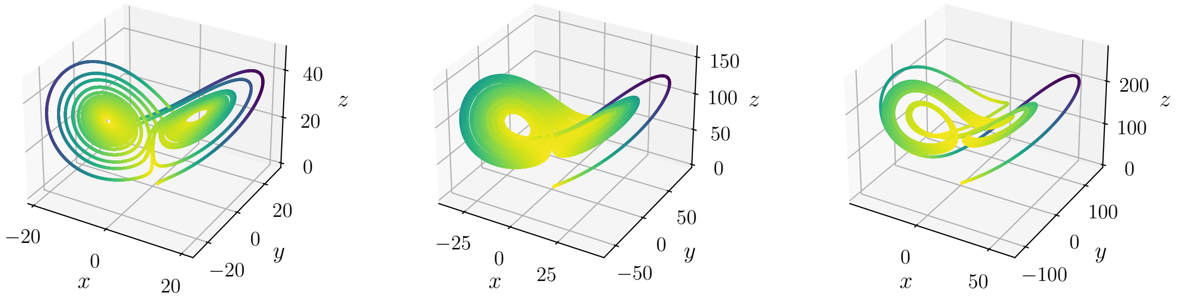
In higher dimensions, tensor products of these one-dimensional grids provide a simple way to construct the collocation points; however the number N_c of collocation points grows exponentially with the dimensionality of the parameter space. Fast interpolation schemes, such as sparse grids [24], alleviate this by systematically discarding specific nodes in the tensor product grid.

3.2. Stochastic Collocation over Dynamics

Traditional application of stochastic collocation has been directly to the state or some functions of it *i.e.* $\widetilde{\boldsymbol{\phi}}_i^s(\boldsymbol{X},t) = \sum_j c_{ij}\boldsymbol{\phi}_j^c(\boldsymbol{X},t)$. Despite the advantages (cheap computational cost, non-intrusiveness), the surrogates generated this way are typically inaccurate, especially when the parameter-to-state map is highly nonlinear. We demonstrate this in the context of the Lorenz system:

$$\dot{x} = \sigma(y-x), \quad \dot{y} = x(\rho - z) - y, \quad \dot{z} = xy - \beta z. \tag{3.4}$$

It is a three-dimensional system of nonlinear ordinary differential equations with state vector $\boldsymbol{\phi} = (x,y,z)$ and parameters $\boldsymbol{\lambda} = (\rho,\sigma,\beta)$. We fix two of the three parameters of the Lorenz system at $\sigma = 10$ and $\beta = 8/3$, and treat the third parameter ρ as a random variable with a uniform distribution in range $\rho \in [28,144]$. For this range, the system goes through a period-doubling bifurcation; this can be seen in Figure 3-1 where we plot a single trajectory of the Lorenz system starting at $\boldsymbol{\phi}_0 = (1,1,-1)$ for different values of ρ.

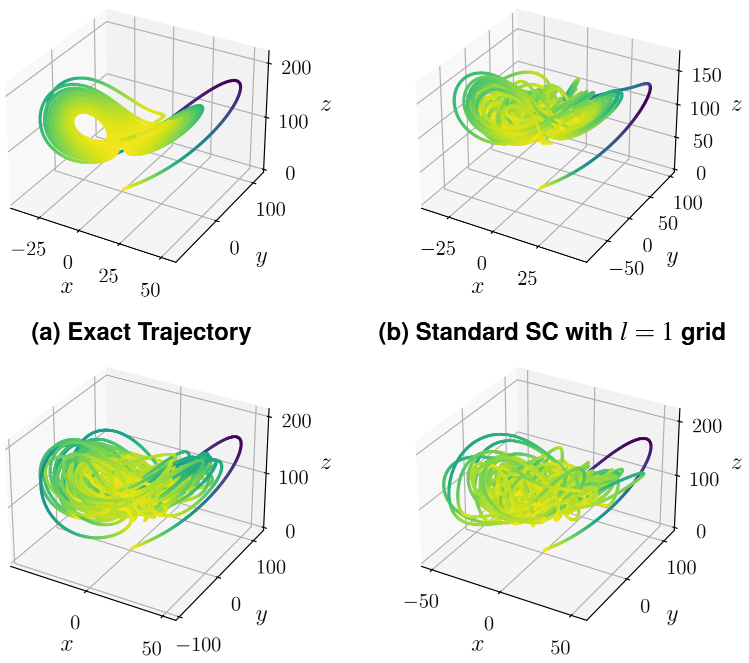


(a) $\rho = 28$ **(b)** $\rho = 86$ **(c)** $\rho = 144$

Figure 3-1. Trajectories of the Lorenz system starting at $(1,1,-1)$ for different values of the parameter ρ. The remaining two parameters are kept fixed at $\sigma = 10$ and $\beta = 8/3$. As we increase the value of ρ, the system undergoes a period-doubling bifurcation, and the trajectories at the two extremes are drastically different.

We attempt to approximate the trajectory of the Lorenz system with the same initial condition for parameter value $\rho = 115$ using stochastic collocation with Clenshaw-Curtis grid at different levels, and show the results in Figure 3-2. We note that none of the stochastic collocation approximations comes remotely is close to the exact trajectory. We can attribute this poor approximation to the highly nonlinear nature of the parameter-to-state map in the Lorenz system.

Our proposal is simple: *instead of directly approximating $\boldsymbol{\phi}$ as a function of the model parameters $\boldsymbol{\lambda}$ (traditional SC), we approximate the forcing/dynamics term $\boldsymbol{R}$ and integrate over it to obtain the approximate sample solutions.* We hypothesize that the dependence of the dynamics on the



(a) Exact Trajectory

(b) Standard SC with $l = 1$ grid

(c) Standard SC with $l = 2$ grid

(d) Standard SC with $l = 3$ grid

Figure 3-2. Exact and approximate trajectories of the Lorenz system starting at $(1,1,-1)$ at parameter values $\rho = 115$, $\sigma = 10$ and $\beta = 8/3$. The approximations are constructed using standard stochastic collocation using nested Clenshaw-Curtis grids of different levels l. We observe visually that all the approximations are fairly inaccurate.

parameters will be much simpler than that of the state; indeed, in the example of the Lorenz system, the dynamics are a linear function of the parameters:

$$\begin{aligned}\boldsymbol{R}^{\top}(\boldsymbol{\phi},\boldsymbol{\lambda}) &= \begin{bmatrix} R_x(\boldsymbol{\phi},\boldsymbol{\lambda}) & R_y(\boldsymbol{\phi},\boldsymbol{\lambda}) & R_z(\boldsymbol{\phi},\boldsymbol{\lambda}) \end{bmatrix} \\ &= \begin{bmatrix} x & y & z & xy & xz \end{bmatrix} \begin{bmatrix} -\sigma & \rho & 0 \\ \sigma & -1 & 0 \\ 0 & 0 & -\beta \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}\end{aligned} \tag{3.5}$$

where "⊤" indicates a transpose operation converting the column vector $\boldsymbol{R}$ into a row-vector.

However, a straightforward application of SC for $\boldsymbol{R}$

$$\boldsymbol{R}(\boldsymbol{\lambda}_i^s) \approx \widetilde{\boldsymbol{R}}_i^s = \sum_j c_{ij} \boldsymbol{R}_j^c \tag{3.6}$$

will yield no advantage over performing SC over $\boldsymbol{\phi}$ under certain conditions. Specifically, if such an approximation were to be used and Eq. (2.1) integrated with linear constant coefficient time integrators, the approximation and the time integration commute and approximating the dynamics yields no accuracy advantage [25]. Rather, the dependence of $\boldsymbol{R}$ on $\boldsymbol{\lambda}$ as well as $\boldsymbol{\phi}$ needs to be teased out for accuracy gains.

In general ODE systems, especially in the context of our problems of interest where the ODE systems are derived from spatial discretizations such as finite-element or finite-volume of the underlying PDE, we do not know the exact form of the dynamics and how it depends on the state or model parameters in advance. In this context, in order to design an application agnostic approach, we employ dynamics discovery techniques. In this report, we focus on the sparse identification of non-linear dynamics (SINDy) architecture developed in [26], and extend it to enable prediction at parameter values where we do not have access to any direct simulation data.

The centerpiece of the SINDy approach is based on a factorization of the dynamics term

$$\underbrace{\boldsymbol{R}^{\top}(\boldsymbol{\phi},\boldsymbol{\lambda})}_{1\times N_n} \approx \underbrace{\boldsymbol{\theta}(\boldsymbol{\phi})}_{1\times N_b}\, \underbrace{\boldsymbol{\Xi}_{\boldsymbol{\lambda}}}_{N_b\times N_n} \tag{3.7}$$

Here N_n is the number of components in the state $\boldsymbol{\phi}$, and

$$\boldsymbol{\theta}(\boldsymbol{\phi}) = \begin{bmatrix}\theta_1(\boldsymbol{\phi}) & \cdots & \theta_{N_b}(\boldsymbol{\phi})\end{bmatrix} \tag{3.8}$$

is a set of *user chosen* basis functions that can be linearly combined using the coefficients from $\boldsymbol{\Xi}_{\boldsymbol{\lambda}}$ to approximate the dynamics. For instance, a very common choice involves using monomials in $\boldsymbol{\phi}$ of degree up to some $N_p \geq 0$; in this case we can write

$$\boldsymbol{\theta}(\boldsymbol{\phi}) = \begin{bmatrix}\cdots & \boldsymbol{\phi}^{\boldsymbol{p}} & \cdots\end{bmatrix}, \quad \boldsymbol{p} = (p_1,\ldots,p_{N_n}), \quad p_n \geq 0, \quad \sum_{n=1}^{N_n} p_n \leq N_p \tag{3.9}$$

where we denote

$$\boldsymbol{\phi}^{\boldsymbol{p}} = \phi_1^{p_1}\cdots\phi_{N_n}^{p_{N_n}} \tag{3.10}$$

This construction leads to a total of $N_b = \binom{N_n+N_p}{N_p}$ basis functions.

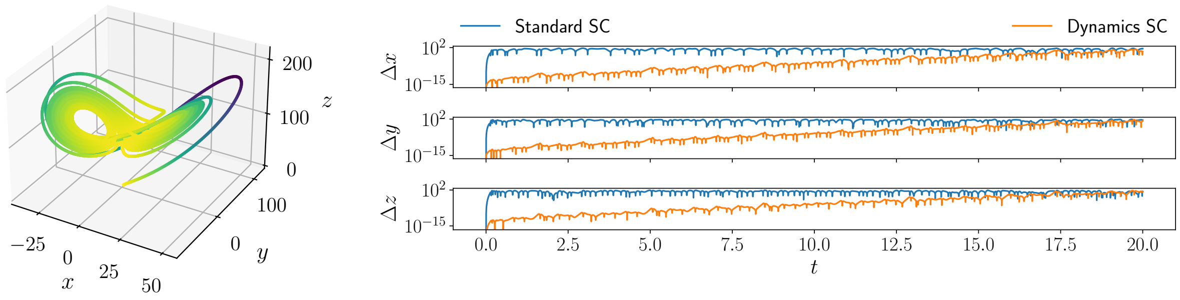
Once we have designed the basis functions depending on the problem of interest, we learn the unknown coefficients $\boldsymbol{\Xi}_{\boldsymbol{\lambda}}$ at a parameter collocation point $\boldsymbol{\lambda} = \boldsymbol{\lambda}_j^c$ from the data generated from the exact simulations of the trajectories. In particular, we collect the states $\boldsymbol{\phi}_{jk}^c \equiv \boldsymbol{\phi}_j^c(t_k)$ and the corresponding dynamics evaluations $\boldsymbol{R}_{jk}^c \equiv \boldsymbol{R}(\boldsymbol{\phi}_{jk}^c,\boldsymbol{\lambda}_j^c)$ along the trajectory at timesteps t_k to set up the linear system

$$\underbrace{\begin{bmatrix}(\boldsymbol{R}_{j1}^c)^{\top}\\ \vdots \\ (\boldsymbol{R}_{jT_t}^c)^{\top}\end{bmatrix}}_{T_t\times N_n} \approx \underbrace{\begin{bmatrix}\boldsymbol{\theta}(\boldsymbol{\phi}_{j1}^c)\\ \vdots \\ \boldsymbol{\theta}(\boldsymbol{\phi}_{jT_t}^c)\end{bmatrix}}_{T_t\times N_b} \underbrace{\boldsymbol{\Xi}_j^c}_{N_b\times N_n} \tag{3.11}$$

where we denote $\boldsymbol{\Xi}_j^c = \boldsymbol{\Xi}_{\boldsymbol{\lambda}_j^c}$. After solving this linear system to obtain the coefficients $\boldsymbol{\Xi}_j^c$ corresponding to the dynamics at each of the collocation points, we use stochastic collocation to construct approximate dynamics of the system at sample points as:

$$\widehat{\boldsymbol{R}}(\boldsymbol{\phi},\boldsymbol{\lambda}_i^s) = \boldsymbol{\theta}(\boldsymbol{\phi})\widehat{\boldsymbol{\Xi}}_i^s, \quad \widehat{\boldsymbol{\Xi}}_i^s = \sum_j c_{ij}\boldsymbol{\Xi}_j^c \tag{3.12}$$

We then numerically integrate this surrogate dynamics to obtain the approximate trajectories $\widehat{\boldsymbol{\phi}}_i^s(t)$. As proposed in the original SINDy formulation, we use a sequentially thresholded least squares



(a) Dynamics SC

(b) Error comparison between standard and dynamics SC

Figure 3-3. Approximate trajectory of the Lorenz system at parameter values $\rho = 115$, $\sigma = 10$ and $\beta = 8/3$ starting at $\phi_0 = (1, 1, -1)$ using the dynamics stochastic collocation approach. We see that this reconstruction is significantly better than the standard stochastic collocation reconstructions in Figure 3-2. On the right, we compare the errors in approximating the exact trajectory; we see that the Dynamics SC approach performs significantly better than Standard SC.

algorithm to solve (3.11). This imposes sparsity restrictions on the coefficients and allows us to identify the most important terms of the dynamics. See [26] for more details about this algorithm.

In Figure 3-3, we plot the approximate trajectory for the Lorenz system at parameter value $\rho = 115$ using our dynamics stochastic collocation approach. We see that visually this reconstruction is significantly better than the standard stochastic collocation approximations in Figure 3-2. We also plotted the divergence of the approximate trajectories using these two approaches from the exact trajectory — while we see an exponential growth in error in both cases as simulation progresses (due to the chaotic nature of the system), the growth rate in case of the dynamics SC is much milder than that for standard SC.

3.3. Hypersonic reacting flow: A special case

One of our target applications–hypersonic reacting flow for re-entry conditions with SPARC– offers a special case. Typically $\boldsymbol{R}$ comprises of terms involving spatial derivatives of $\boldsymbol{\phi}$ *e.g.* advection, diffusion. This is what makes the closure of this term challenging; since each Lagrangian sample carries state, $\{\hat{\boldsymbol{x}}_i^s, \boldsymbol{\phi}_i^s, \boldsymbol{\lambda}_i^s\}$, at a specific spatial location the spatial derivatives in $\boldsymbol{R}(\boldsymbol{\phi}, \boldsymbol{\lambda}; \boldsymbol{X}, t)$ introduce a dependence on non-local state ($\boldsymbol{\phi}$ in some neighbourhood of $\hat{\boldsymbol{x}}_i^s$) that each sample does not have access to. However, certain terms like chemical reaction sources are functions purely of local state $\{\boldsymbol{\phi}_i^s, \boldsymbol{\lambda}_i^s\}$ and these can be evaluated exactly for each sample. If the uncertain parameters $\boldsymbol{\lambda}$ pertain to such terms then the closure of $\boldsymbol{R}$ can be made to take advantage of this[1].

We make the following observations about the SPARC target problem:

- SPARC solves the governing PDEs in an Eulerian frame of reference

[1] Note that $\boldsymbol{R}$ cannot be comprised only of terms that have pure local dependence, for if it were then Eq. (2.1) would be ordinary differential equations (ODEs) and not PDEs.

- Steady state, and therefore, partial derivatives w.r.t. t (∂_t) are zero,
- The uncertain parameters $\boldsymbol{\lambda}$ are those pertaining to thermo-chemical source terms, and

The governing PDEs solved by SPARC are of the form [15]:

$$\frac{\partial \boldsymbol{\phi}}{\partial t} \;=\; \underbrace{\boldsymbol{C} \,+\, \boldsymbol{P} \,+\, \boldsymbol{D}}_{\text{non}-\text{local}} \;+\; \omega, \tag{3.13}$$

where $\boldsymbol{C}$ is convective flux, $\boldsymbol{P}$ is pressure flux, and $\boldsymbol{D}$ is diffusive flux, and these terms are non-local in nature since they involve spatial derivatives. The reaction term, ω, is purely a function of local $\boldsymbol{\phi}$, as well as the uncertain parameters $\boldsymbol{\lambda}$. The sample solutions being in a Lagrangian reference frame, the material derivative in Eq. (2.1) can be written as

$$\frac{D\boldsymbol{\phi}}{Dt} = \cancelto{0}{\frac{\partial \boldsymbol{\phi}}{\partial t}} + \mathbf{v}\cdot\nabla\boldsymbol{\phi} = \boldsymbol{R}, \tag{3.14}$$

where $\mathbf{v}$ is the velocity (derived from density and momenta which are part of $\boldsymbol{\phi}$) and the first term is zero for steady problems.

For such steady problems the $\boldsymbol{R}$ for each streamline is a field quantity and depends explicitly only on spatial location. However, for the approximate trajectories corresponding to $\{\boldsymbol{\lambda}^a\}$, the dynamics term is split into 'local' and 'non-local' components

$$\boldsymbol{R}^a \equiv \boldsymbol{R}(\hat{\boldsymbol{x}}^a, \boldsymbol{\phi}^a, \boldsymbol{\lambda}^a) = \boldsymbol{R}^a_{nl}(\hat{\boldsymbol{x}}^a) + \boldsymbol{R}^a_l(\boldsymbol{\phi}^a, \boldsymbol{\lambda}^a). \tag{3.15}$$

For the non-local component we perform the usual approximation using the collocation samples

$$\boldsymbol{R}^a_{nl}(\hat{\boldsymbol{x}}^a) \approx \sum_j c_{aj} \boldsymbol{R}^a_{nl}(\hat{\boldsymbol{x}}^a, \boldsymbol{\lambda}^c_j), \tag{3.16}$$

noting only the explicit dependence on $\hat{\boldsymbol{x}}^a$. The local component, the chemical reaction source term, can be evaluated exactly as $\boldsymbol{R}^a_l \equiv \omega^a = \omega(\boldsymbol{\phi}^a, \boldsymbol{\lambda}^a)$ since it has no explicit dependence on spatial location but rather on $\{\boldsymbol{\phi}^a, \boldsymbol{\lambda}^a\}$. The rationale here is that this should be more accurate since the explicit dependence on the parameter value of the sample is being accounted for. It is easy to show that the non-local component in Eq. (3.15) is

$$\boldsymbol{R}^a_{nl}(\hat{\boldsymbol{x}}^a, \boldsymbol{\lambda}^c_j) = (\mathbf{v}\cdot\nabla\boldsymbol{\phi} - \omega)(\hat{\boldsymbol{x}}^a, \boldsymbol{\lambda}^c_j), \tag{3.17}$$

i.e. the field quantity $(\mathbf{v}\cdot\nabla\boldsymbol{\phi} - \omega)$ corresponding to the collocation samples at query location $\hat{\boldsymbol{x}}^a$.

4. RESULTS

We present results of application of our method to the two target problems. Since the target problems pertain to different applications, our methodology was tailored to take advantage of the specifics of the applications. We also describe the software resulting from these implementations.

4.1. Solid mechanics explicit dynamics

Our first target application is solid mechanics explicit dynamics problems with uncertain material model parameters. Keeping in mind the finite element spatial discretization, used by most solid mechanics codes, and corresponding temporal discretization for explicit dynamics problems, we made refinements to accommodate the spatial and and temporal complexity of the problem for method feasibility.

4.1.1. Improving efficiency by exploiting FEM mesh structure

Since the ODE system results finite element method (FEM) discretization of the domain, the state vector $\boldsymbol{\phi}$ can be very high dimensional ($N_n \gg 1$). In this context, the number of basis functions can grow exponentially fast: if we are using monomial basis functions of degree up to N_p, then

$$N_b = \binom{N_n+N_p}{N_p} \sim \sqrt{\frac{1}{2\pi}}\sqrt{\frac{N_n+N_p}{N_nN_p}}\frac{(N_n+N_p)^{N_n+N_p}}{N_n^{N_n}N_p^{N_p}} \tag{4.1}$$

from Stirling's approximation. As a result, the linear system (3.11) can become prohibitively costly to solve and obtain the coefficients.

In this setup, we can exploit the local nature of the FEM solution method to reduce the number of effective state variables: we note that for each node in the FEM mesh, the force evaluation depends only on those nodes which are part of the same FEM element. Thus, the corresponding component of the dynamics, identified by a material reference point $\boldsymbol{X}$, can be simplified to

$$\boldsymbol{R_X}(\boldsymbol{\phi},\boldsymbol{\lambda}) = \boldsymbol{R_X}(\boldsymbol{\phi}_{\boldsymbol{X}_{\text{neighbor}}},\boldsymbol{\lambda}) \tag{4.2}$$

where $\boldsymbol{X}_{\text{neighbor}}$ is the collection of material reference points that are part of an FEM element with node $\boldsymbol{X}$. This reduces the number of basis functions necessary to represent the dynamics at any reference point, making the least squares problem tractable.

Beyond reducing the number of effective variables in the nodal dynamics, this local nature of the constructed surrogate models can also be helpful in other ways. For instance, in multi-material solid mechanics simulations, the response of one region of the solid can be constructed completely independently of another region — potentially leading to more accurate surrogate model for the overall material model.

4.1.2. *Adaptive dynamics stochastic collocation*

Our dynamics stochastic collocation method introduced above is not fully online in the same way standard stochastic collocation is. Learning the dynamics requires state and forcing data from several timesteps of the exact simulations at the collocation points. On the other hand, once the dynamics is approximated well enough, there is no need to learn it again (unless the dynamics of the system suddenly changes, which is rare in practical applications). This suggests an adaptive approach, where we

1. First reset the learned dynamics by setting the coefficients to zero.
2. Then collect state and forcing data from several timesteps of the exact simulations and learn the coefficients.
3. Next test the prediction of the learned dynamics against the true dynamics in the exact simulations; if they do not match within some specified tolerance, we go to step 2 and repeat.
4. Once the learned dynamics has converged to the true dynamics, interpolate the coefficients and simulate the approximate trajectories at the sample points.
5. Periodically check if the learned dynamics stays consistent with the true dynamics using the exact simulations; if not we go to step 1 and repeat.

To test the convergence criteria, the user can specify a relative tolerance τ_{rel} and an absolute tolerance τ_{abs}. We say that the learned dynamics has converged at timestep t_k and can be used to construct the states at timestep t_{k+1} if

$$\|\widehat{\boldsymbol{R}}(\boldsymbol{\phi}_j^c(t_{k+1}),\boldsymbol{\lambda}_j^c)-\boldsymbol{R}(\boldsymbol{\phi}_j^c(t_{k+1}),\lambda_j^c)\| \leq \max\{\tau_{\text{abs}},\tau_{\text{rel}}\|\boldsymbol{R}(\boldsymbol{\phi}_j^c(t_{k+1}),\lambda_j^c)\|\} \tag{4.3}$$

for all $1 \leq j \leq N_c$.

4.1.3. *Software implementation details*

We outline the flow of data in constructing the approximate trajectories using dynamics stochastic collocation in Figure 4-1. We note that apart from collecting the states $\boldsymbol{\phi}_j^c(t_k)$ and corresponding dynamics evaluations $\boldsymbol{R}_j^c(t_k)$ at different time steps t_k for the collocation parameter values $\boldsymbol{\lambda} = \boldsymbol{\lambda}_j^c$, $1 \leq j \leq N_c$ to construct the history, the surrogate model generation process is largely independent from the full simulation model. Consequently, we can write a mostly non-intrusive supplement code to existing simulation models; we only need access to the data arrays storing the state and force evaluations in the original simulation code.

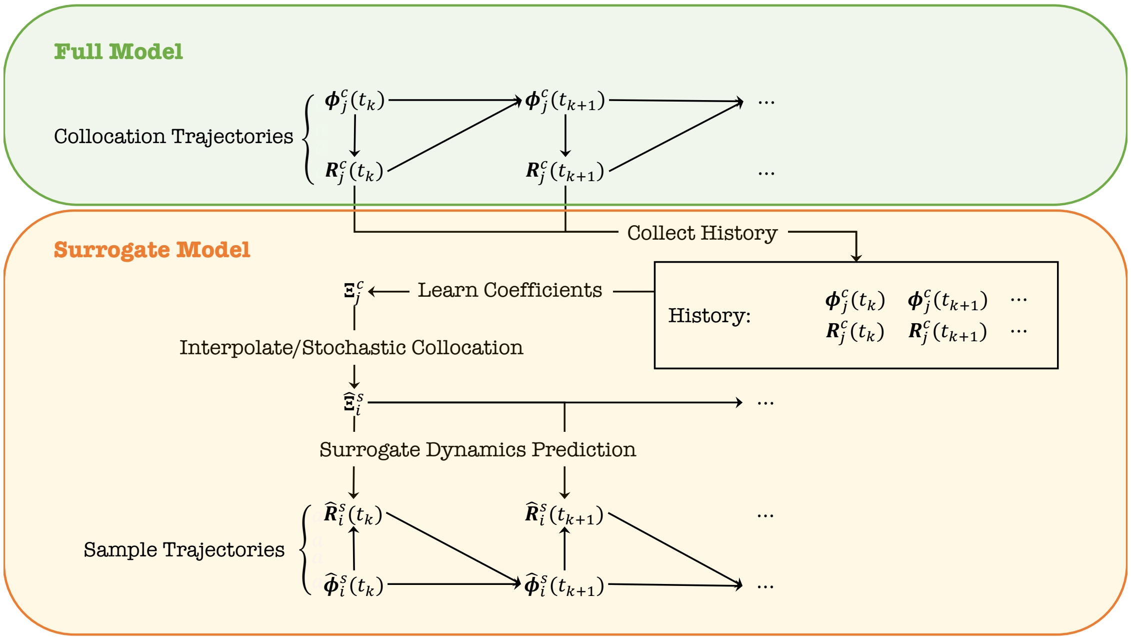


Figure 4-1. Flow of data in constructing the dynamics stochastic collocation surrogate models for a dynamical system. On the top, we evolve the state $\phi_j^c(t_k)$ of the ODE system corresponding to collocation parameter values $\lambda = \lambda_j^c$ for all $1 \leq j \leq N_c$. To build the surrogate models, we record the evolution history (external to the original simulation) consisting of the states and dynamics evaluations, and use SINDy to learn the dynamics coefficients Ξ_j^c. Interpolating these coefficients (using stochastic collocation) leads us to the functional representation of the surrogate dynamics $\widehat{R}_i^s(\cdot)$, which we integrate to construct the approximate states $\phi_i^s(t_k)$.

We implemented this dynamics stochastic collocation scheme as a C++ library named `ODEUQ`. It has the following build dependencies:

- `BLAS` and `LAPACK` for linear algebra (in particular, the SINDy least squares solve),
- `yaml-cpp` for parsing input parameters,
- `HDF5` for basic simulation output, and
- Optionally `MPI` if `HDF5` was built with parallel IO support.

The `ODEUQ` library provides two main setups for constructing surrogate dynamics and evolving a dynamical system:

1. The `odeuq::OdeSystem` class is suitable for simple systems; it creates an abstraction for the underlying first/second order ODE system. All the bookkeeping tasks, such as memory allocations and evolving the exact and approximate trajectory ensembles, are handled automatically. The user can use an YAML file to feed model inputs, collocation and sample parameter values, interpolation coefficients etc. to the generated `odeuq_run` executable and obtain HDF5 files consisting of full-field quantities of interest.

2. The odeuq::UqModelBase class is provided for more complicated use cases, e.g. the existing numerical simulation code is using custom data management and the ODE evolution cannot be easily encapsulated. We provide an example of this below.

In either case, an ODEUQ user will write a child class and implement the necessary methods. This modular design will allow the user to essentially create a wrapper around existing simulation codes with UQ capabilities.

For a concrete example of the second approach, consider adding dynamics SC uncertainty propagation capabilities to the NimbleSM library [16] — it is an open source C++ code for Lagrangian finite-element code for solid mechanics simulations with support for multi-material solids. We can specify density, bulk modulus and shear modulus of each of the materials as program input — these are our dynamics parameters. The code can run multiple exact simulations at specified collocation parameter values, and implements a data manager object that exposes the corresponding displacement, velocity and acceleration data arrays for each exact simulation. In this setup, we wrote a nimble_odeuq::UqModel class derived from the odeuq::UqModelBase implementing:

- *Model initialization,* where we retrieve the NimbleSM data manager, allocate memory for storing the displacement and acceleration history, and initialize separate surrogate dynamics models for each exact and approximate trajectories.

- *Simulation initialization*, where we prepare for time integration by computing lumped masses, setting up boundary condition enforces etc.

- *Exact time stepping*, where we evolve the system state corresponding to a single exact trajectory; much of this code is a copied from the NimbleSM implementation.

- *Approximate time stepping*, where we evolve system state corresponding to a single approximate trajectory; this code is the same as exact time stepping except that the acceleration is computed from the surrogate dynamics.

Rest of the bookkeeping tasks, such as testing for surrogate convergence, training the exact dynamics surrogates and interpolating the approximate surrogate dynamics, are handled by methods of the odeuq::UqModelBase class. Other than these adaptations, we needed to write/modify a few additional lines of code for

- Retrieving the node-neighbor structure from nimble::GenesisMesh (FEM mesh); this is used by nimble_odeuq::UqModel.

- Adjusting the main() entry point and nimble::NimbleMain() driver routine to properly parse additional parameters for dynamics SC learning and call the nimble_odeuq::UqModel object.

Putting all of this together, we obtain a nimble_odeuq executable which takes the standard NimbleSM input file as well as an additional YAML file describing dynamics SC configurations, and generates HDF5 files consisting of the full-field quantities of interest.

4.1.4. 1D Solid Bar Impact Simulation

We first consider a simple solid mechanics problem in one dimension. A solid bar moving at a constant velocity impacts a wall; then the displacement $u(X,t)$ is governed by

$$\rho_0 u_{tt} = \sigma_X \tag{4.4}$$

where ρ_0 is the density of the bar and σ is the stress. Let L be the length of the bar, then the displacement field satisfies the initial conditions:

$$u(X,t=0)=0, \quad u_t(X,t=0) = \begin{cases} 0 & \text{if} \quad x=0 \\ v_0 & \text{if} \quad 0 < x \leq L \end{cases} \tag{4.5}$$

where v_0 is the initial velocity of the bar. We assume the left end of the bar is fixed and the right end is free:

$$u(X=0,t)=0, \quad u_X(X=L,t)=0 \tag{4.6}$$

The constitutive relation connecting the stress and displacement is given by

$$\sigma = \varepsilon_0 u_X (1+\varepsilon_1 u_X^2)(1+\varepsilon_2 u_X^2)\cdots \tag{4.7}$$

Here $\boldsymbol{\lambda} = (\varepsilon_0, \varepsilon_1, \varepsilon_2, \ldots)$ are the model parameters.

We discretize the spatial domain $[0,L]$ into (N_n-1) uniform elements of length $h = L/(N_n-1)$ and denote the nodes as $x_n = (n-1)h$ for $1 \leq n \leq N_n$. Then imposing linear basis functions within each element, we approximate the PDE solution as

$$u(X,t) \approx \sum_{n=1}^{N_n} \phi_n(t)\omega_n(X), \quad \omega_n(X) = \begin{cases} (X-x_{n-1})/h & \text{if} \quad 0 \leq x_{n-1} \leq X \leq x_n \\ (x_{n+1}-X)/h & \text{if} \quad x_n \leq X \leq x_{n+1} \leq L \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

where $\boldsymbol{\phi}(t) = (\phi_1(t), \ldots, \phi_{N_n}(t))$ is governed by an ODE of the form

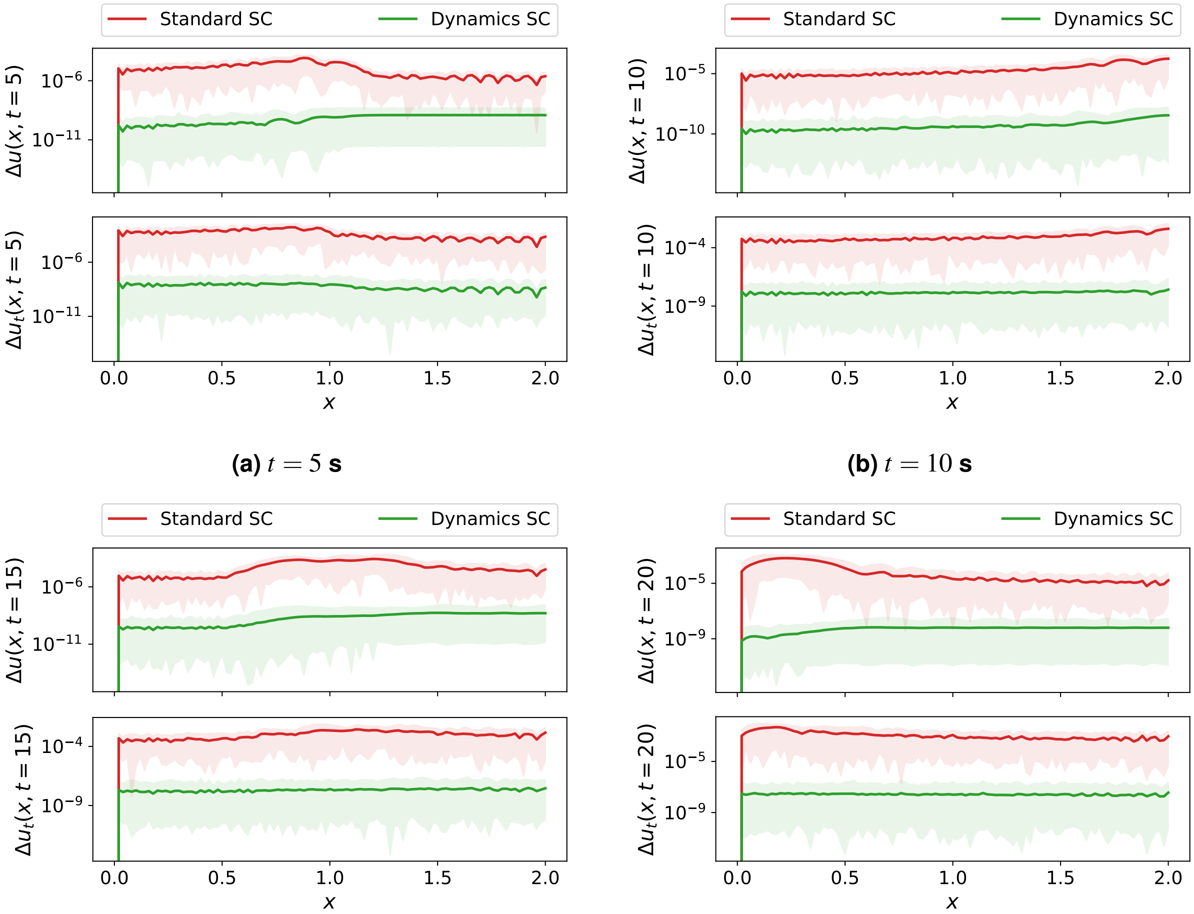
$$\ddot{\boldsymbol{\phi}} = \boldsymbol{R}(\boldsymbol{\phi}, \boldsymbol{\lambda}). \tag{4.9}$$

To construct the surrogate models, we apply the Dynamics SC approach described in Section 3.2 to each component of this dynamics and assume

$$R_n(\boldsymbol{\phi}, \boldsymbol{\lambda}) = R_n(\phi_{n-1}, \phi_n, \phi_{n+1}, \boldsymbol{\lambda}) \approx \boldsymbol{\theta}_n(\phi_{n-1}, \phi_n, \phi_{n+1})\boldsymbol{\xi}_{n,\boldsymbol{\lambda}} \tag{4.10}$$

with monomial basis functions $\boldsymbol{\theta}_n$.

In our numerical experiments, we discretize the bar of length $L=2$ and density $\rho_0 = 1$ using $N_n = 101$ uniformly placed FEM nodes and impose an initial velocity $v_0 = -0.1$. We choose a two-parameter constitutive relation with $\varepsilon_1, \varepsilon_2 \sim$ Uniform$[1.5, 2.5]$ and draw $N_s = 128$ random samples where we would like to construct the approximate trajectories. We choose the collocation points using a sparse grid constructed from level $l=1$ nested Clenshaw-Curtis grid, and use up to fifth degree monomials in constructing the SINDy basis functions. For timestepping, we use



(a) $t = 5$ **s**

(b) $t = 10$ **s**

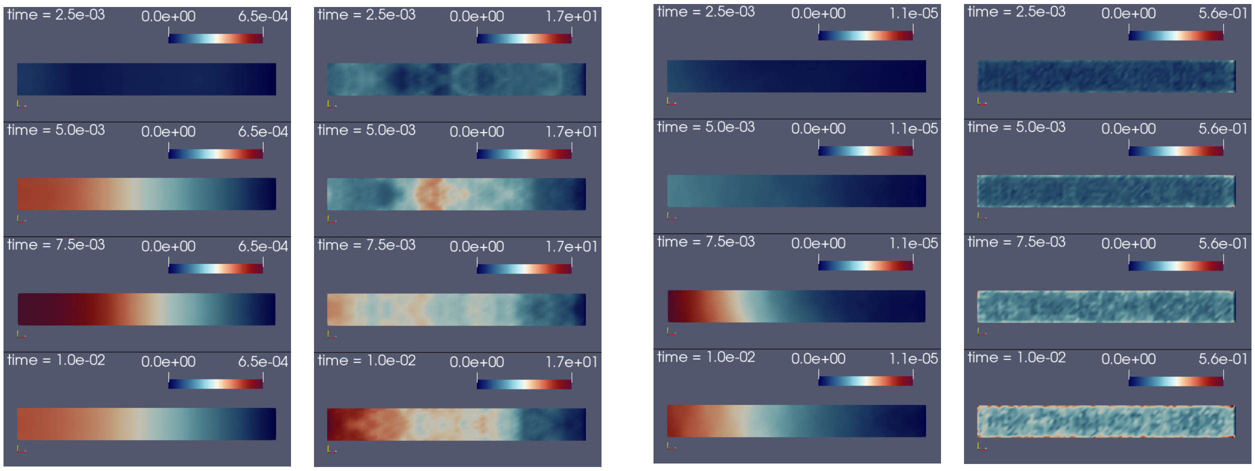
(c) $t = 15$ **s**

(d) $t = 20$ **s**

Figure 4-2. Comparison of the Standard SC and Dynamics SC surrogates for the two-parameter bar impact system. Error is computed as the difference in $u(x,t)$ and $u_t(x,t)$ between the approximate and the corresponding exact trajectories. The shaded regions encompass the range of errors over 128 approximate trajectories, and the solid lines represent the mean over the trajectories.

a Strömer-Verlet integrator for 2000 timesteps of size $\Delta t = 10^{-2}$ s. To construct the surrogate dynamics, we use data from only the first 10 s of the collocation trajectories; the second half of the sample trajectories are then constructed using this learned dynamics.

In Figure 4-2, we compare the errors in the approximate states $u(x,t)$ and their time derivatives $u_t(x,t)$ constructed using the standard and dynamics stochastic collocation methods at different times t. We observe that the errors using the dynamics stochastic collocation scheme is several orders of magnitude smaller than those using the standard stochastic collocation scheme.



Displacement Error
Standard SC

Velocity Error
Standard SC

Displacement Error
Dynamics SC

Velocity Error
Dynamics SC

Figure 4-3. Comparison of average errors over multiple approximate trajectories constructed using the standard stochastic collocation and dynamics stochastic collocation surrogates. The nominal `NimbleSM` simulation models a solid pseudo-3D bar moving with uniform velocity that is clamped at time $t = 0$ at the right end. Along each column, we plot the evolution of the error at times $t = 2.5 \times 10^{-3}$, 5.0×10^{-3}, 7.5×10^{-3}, and 1.0×10^{-2} seconds using the same color scale for one state variable (displacement or velocity) and one surrogate model (standard SC or dynamics SC). The upper limit of the color scale varies across the columns. Note that the errors from the dynamics SC surrogates are orders of magnitude smaller than those from the standard SC surrogates.

4.1.5. NimbleSM 2D Bar Impact Simulation

We now consider a pseudo-3D variant of the bar impact simulation using the `NimbleSM` software package. The displacement $u(X,t) \in \mathbb{R}^3$ evolves as

$$\rho_0 u_{tt} = \nabla \cdot \sigma \tag{4.11}$$

We assume the material is elastic, i.e. the dependence of stress σ the strain $\varepsilon = \frac{1}{2}(\nabla u + \nabla u^T)$ is expressed as

$$\sigma = 2\mu\varepsilon + \lambda \operatorname{tr}(\varepsilon) I \tag{4.12}$$

where μ and λ are the Lame parameters; for a material with bulk modulus K and shear modulus G we have

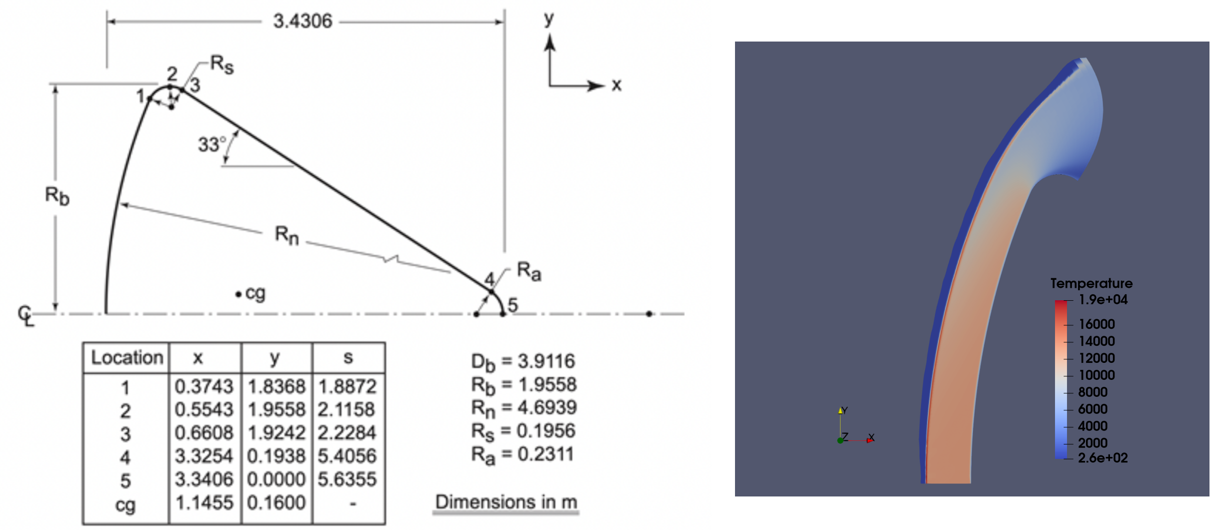
$$\mu = G, \quad \lambda = K - \frac{2}{3}G \tag{4.13}$$

The solid bar extends 0.40 m along the x-direction, 0.05 m along the y-direction and 0.01 m along the z-dimension. The material has density $\rho_0 = 7{,}850$ kg/m^3 and shear modulus $G = 77$ GPa; we treat the bulk modulus as stochastic with distribution $K \sim \text{Uniform}[80\text{ GPa}, 240\text{ GPa}]$. We discretize the simulation domain using $1{,}600$ hexahedral FEM elements organized in a $80 \times 10 \times 2$ grid; each element is a cube with side length of 0.01 m. The bar is moving with an initial uniform velocity of $v_0 = (10.0, 0.1, 0.0)$ m/s, and at time $t = 0$, the right end surface is clamped. We use the Strömer-Verlet integrator simulating $10{,}000$ time steps of this system using a step size of 10^{-6} s.

In our experiments, we simulate three trajectories exactly (corresponding to $K = 80$, 160, and 240 GPa values), and construct approximate trajectories at 16 other parameter values using our dynamics stochastic collocation approach. In Figure 4-3, we compare the accuracy of the approximate trajectories generated using our approach against those generated using standard SC. We see that our errors are orders of magnitudes smaller than those from standard SC.

4.2. Reacting hypersonic fluid mechanics: Apollo re-entry simulation

Our second target problem is hypersonic flow with thermo-chemical non-equilibrium involving dissociation reactions and shocks. We considered re-entry simulation of the well known Apollo capsule geometry at Mach 30 with SPARC. A schematic of the geometry is shown in Fig. 4-4. Flow over only the forebody was considered, and a 2D axisymmetric domain covering an upstream region bounded in the y direction by the symmetry axis and the uppermost location (labelled ‘2’ in the geometry schematic) was used. A 2D curvi-linear structured mesh of 65536 quadrilateral elements was employed along with the cell-centered finite volume spatial discretization in SPARC. Inflow conditions specified at the left boundary were a density of 5×10^{-4} kg/m^3, an x velocity of 9.798×10^3 m/s, and translational and vibrational temperatures of 257.86 K. The top boundary was specified as an outflow, the right boundary as an isothermal no-slip wall at 2500 K, and symmetry conditions at the lower boundary. No turbulence models were activated. Figure 4-4 shows the steady state solution in terms of the contours of translational temperature over the domain.



Location	x	y	s
1	0.3743	1.8368	1.8872
2	0.5543	1.9558	2.1158
3	0.6608	1.9242	2.2284
4	3.3254	0.1938	5.4056
5	3.3406	0.0000	5.6355
cg	1.1455	0.1600	-

Figure 4-4. Schematic of the Apollo capsule geometry is shown on the left. The plot on the right shows contours of translational temperature from a SPARC simulation at Mach 30.

We used the two temperature formulation of Park [1], solving for translational and vibrational modes of temperature, to model thermal non-equilibrium and a 5-species chemical mechanism consisting of N_2, O_2, NO, N, O to model dissociation chemistry. The parameters of the Arrhenius type kinetics rate expressions have considerable uncertainty and variations of orders of magnitude have been reported in literature for some of them [27]. We considered two of these parameters for UQ: (1) the collision coefficients for the dissociation reaction of $N2$, and (2) the pre-exponential factor for the exchange reaction $N_2 + O \Leftrightarrow NO + N$.

4.2.1. Implementation of sampled streamlines integration

As outlined in section 3.3, this case offers a specialization in terms of propagating parameter uncertainty. Since its a steady fluid mechanics problem the Lagrangian sample solutions are effectively streamlines traversing the domain. To keep the method implementation non-intrusive to SPARC we implemented SAMURAI, a stand-alone python library for streamline integration for steady reacting flow problems. The main design features of SAMURAI are:

- A class hierarchy to setup a generic 2D flow domain.
- Class methods to load SPARC solution data saved in `exodus` format to set up the field quantities.
- A bilinear interpolant to perform interpolation of field values from the corners of a quadrilateral element to any interior location.
- A integrator class that uses SciPy's `solve_ivp` function to solve ODEs and time advance a streamline state from an initial condition.
- A class to define a reaction mechanism and compute reaction source terms for prescribed values of rate parameters.

Using SAMURAI we compute the streamline solutions for all the approximate samples corresponding to $\boldsymbol{\lambda}^a$. Since we are using a tailored, and more accurate treatment, of the driving term $\boldsymbol{R}^a$ as outlined in section 3.3, we use only two collocation samples, $\lambda_{\min}$ and $\lambda_{\max}$, to provide the approximation for the 'non-local' part $\boldsymbol{R}^a_{nl}$ in Eq. 3.15. Effectively, since every sample $\lambda_{\min} < \lambda^a < \lambda_{\max}$, we are doing an interpolation for $\boldsymbol{R}^a_{nl}$.

Our streamline integration proceeds in the following steps:

1. Perform SPARC simulations corresponding to $\lambda_{\min}$ and $\lambda_{\max}$ offline. Save the solution data to exodus files.
2. Use the class hierarchy in SAMURAI to read the two solution exodus files. Set up domain in terms of quadrilateral element nodes, and the bilinear interpolant to compute field quantities.
3. Setup the reaction model.
4. Initialize streamlines with initial parameter values sampled within $(\lambda_{\min}, \lambda_{\max})$, and desired initial location and state consistent with the boundary conditions.
5. Integrate the streamlines using SAMURAI's integrator class, *i.e.* solve Eq. (2.1). For the two components of the ODE dynamics:

- The non-local component is defined in Eq. 3.17. Compute the term $(\mathbf{v}\cdot\nabla\boldsymbol{\phi}-\boldsymbol{\omega})$ for each of the collocation sample solutions, and use the bilinear interpolant to interpolate to the current streamline location $\hat{\boldsymbol{x}}^a$. Then interpolate along the λ dimension corresponding to the streamline sample λ^a.
- The local component is computed exactly using the reaction mechanism class of SAMURAI with the streamline state $\{\boldsymbol{\phi}^a,\lambda^a\}$.

4.2.2. *Accuracy of streamline integration*

The methodology described allows us to compute streamlines to trace the solution through the domain and its variability with respect to the uncertain parameter. The streamlines can be initialized in a select region to focus the UQ study on specific portions of the domain as desired. The streamlines ensemble spanning the parameter range provide the PDFs of the solution quantities at any location. As with the solid mechanics problem, our main objective is to assess the acuracy of streamline solution methodology along with the closure outlined in section 3.3. Since the approximation in the stochastic dimension is a simple interpolation between λ_{min} and λ_{max}, *i.e.* all approximate samples satisfy $\lambda_{\text{min}} < \lambda^a < \lambda_{\text{max}}$, for the accuracy assessment we present streamlines results corresponding to the median parameter value $\lambda^a = 0.5(\lambda_{\text{min}} + \lambda_{\text{max}})$. For comparison with the (approximate) streamlines we perform a corresponding SPARC simulation *a posteriori* for this parameter value to provide the benchmark data.

Table 4-1. The parameters of the UQ study for the Apollo re-entry problem, both specified in a logarithmic scale. The first parameter is based on K, a factor multiplying all the collision coefficients for the N_2 dissociation reaction (there is one coefficient for each species, ranging from 7×10^{18} to 3×10^{19}). The * denotes the nominal value prescribed by Park [1]; for the first parameter the nominal value is λ_{max}. The second parameter is based on C, the pre-exponential factor, for the exchange reaction. Here the nominal parameter value, 6.4×10^{14}, corresponds to λ_{min}.

λ	Description	Reaction	λ_{min}	λ_{max}
$\log_{10}(K)$	K: collision coefficients factor	$N_2 + M \Leftrightarrow N + N + M$	-3.0	1.0^*
$\log_{10}(C)$	C: pre-exp factor	$N_2 + O \Leftrightarrow NO + N$	14.80618^*	15.80618

The two parameters considered for the UQ study, and their range of uncertainty, are listed in Table 4-1. Both are prescribed in a logarithmic scale to account for the orders of magnitude variation in the UQ study. For the first parameter, corresponding to the collision coefficients of the N_2 dissociation reaction, its value was reduced from the nominal by three orders of magnitude while for the second parameter, corresponding to the pre-exponential factor of the exchange reaction, its value was increased from the nominal by one order of magnitude. Figure 4-5 illustrates the sensitivity of the SPARC solution to the two UQ parameters studied, by plotting quantities along a x line segment spanning the shock close to the symmetry axis ($y = 0.1$). As to be expected, decreasing the collision coefficients for N_2 dissociation (case-1) results in a higher concentration relative to the nominal case. Similarly, increasing the pre-exponential factor of the exchange reaction (case-2) leads to more NO being formed at the shock. The trends are also interesting for the two components of temperature, with a higher sensitivity of the temperature at the shock in case-1, leading to overall higher temperature values at the shock for this case. On the other hand case-2 results in a decrease in temperature relative to the nominal case.

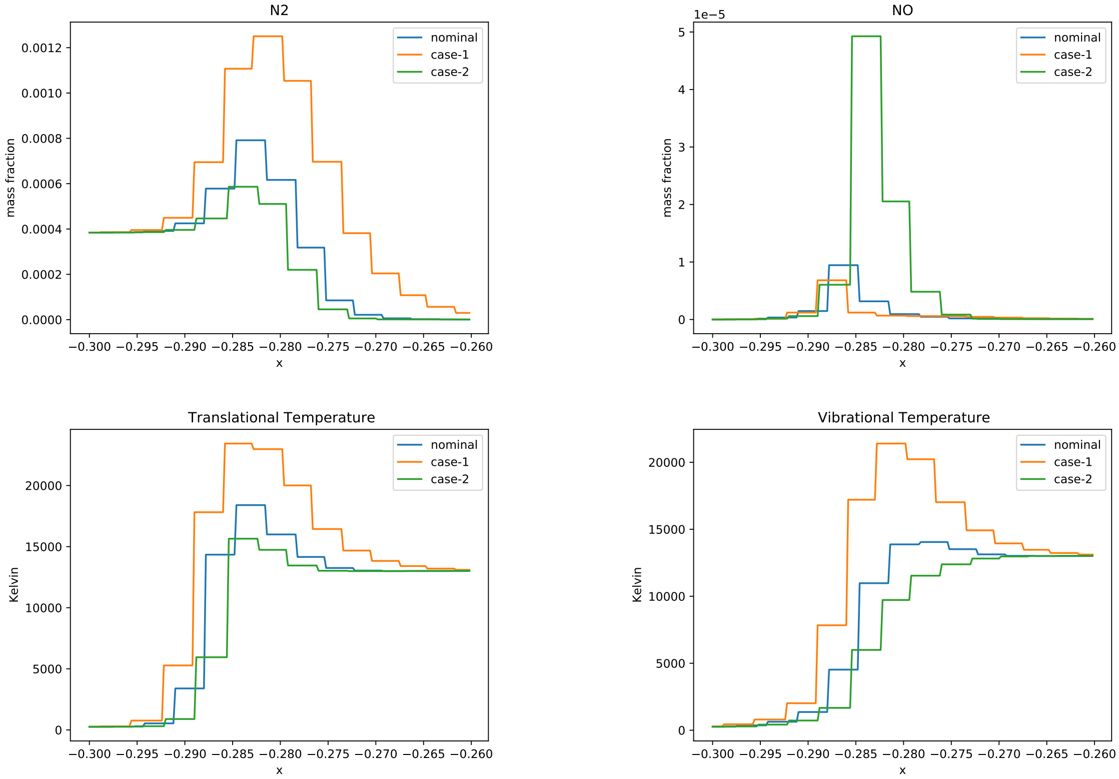


Figure 4-5. Plots of solution quantities along a x line segment spanning the shock, at $y = 0.1$, illustrate the solution sensitivity to the two UQ parameters. Since case-1 pertains to a parameter affecting N_2 reaction, and case-2 pertains to one affecting NO, only these species are shown, along with translational and vibrational temperatures.

For both cases streamlines were initialized close to the inflow boundary, but upstream of the shock, at various locations along a tangential coordinate, with the streamline spacing varying exponentially with tangential distance from the axis of symmetry. Figure 4-6 shows the streamlines, coloured by enthalpy, for validation case-1 ($\lambda^a = 0.5(\lambda_{\text{min}} + \lambda_{\text{max}})$), along with enthalpy from the true SPARC solution. The plot on the right shows the relative error of enthalpy over all the streamlines. Considering that enthalpy is not a part of the solution state $\boldsymbol{\phi}$, but rather a non-linear function of all elements of $\boldsymbol{\phi}$ (species mass fractions, velocity, density, temperature), a maximum relative error of 20% indicates a high accuracy of the overall streamline solutions. Note also that the error is high only near the vicinity of the shock, as expected, and quite low away from it.

Figure 4-7 show the streamlines enthalpy for case-2, which shows a slightly higher relative error, 40%, compared to case-1. The rationale behind showing enthalpy is that it is a key quantity used in the modelling of wall heat transfer from SPARC solutions, which is a key quantity of interest for modelling thermal ablation. Another key quantity is kinetic energy, which is also a derived quantity, and this is shown for both cases, along with the relative error in Figure 4-8. It is evident that kinetic energy is also recovered accurately from the streamline solutions, with a relative error of only a maximum of 12%.

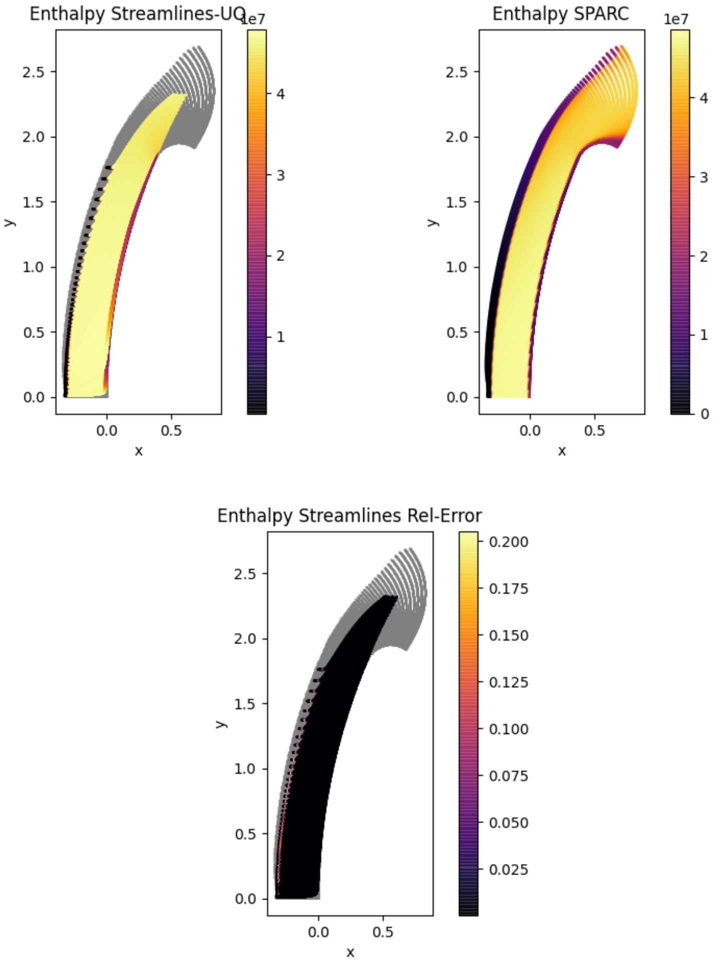


Figure 4-6. Plots of enthalpy computed from the streamline solutions for case-1 (top-left). The streamlines are initialized just upstream of the shock close to the inflow boundary at locations increasingly spaced along a tangential coordinate. The corresponding enthalpy at each node from an actual SPARC solution is shown (top-right), along with the relative error (bottom).

Finally, the two chemical species most sensitive to the two parameters varied are N_2 and NO and these quantities from the streamlines are shown in Figures 4-9 and 4-10 for case-1 and -2, respectively. For these comparisons, to avoid division by zero, we show the absolute error instead of the relative error. Still, comparing the color scales of the actual quantities with the absolute error, it is evident that the error is smaller by an order of magnitude.

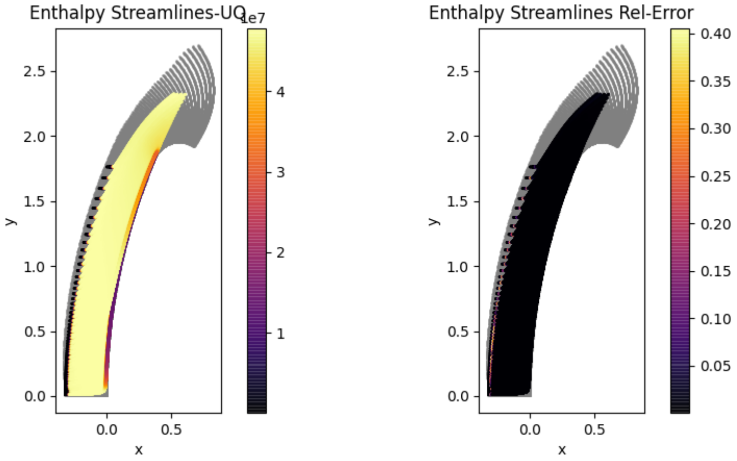
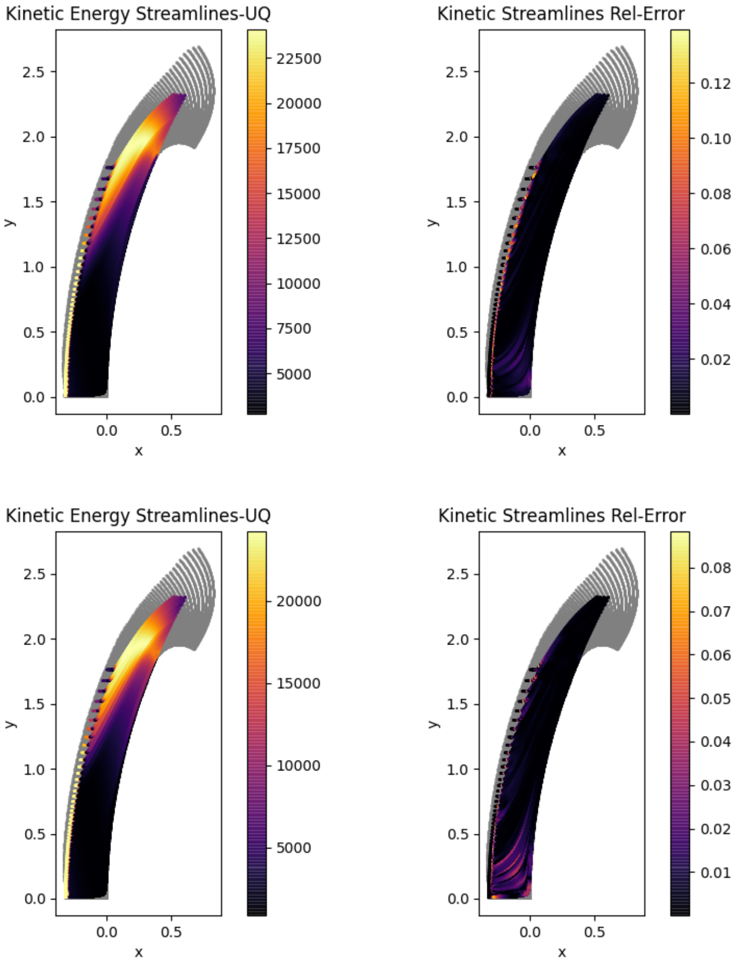


Figure 4-7. Plots of enthalpy computed from the streamline solutions for case-2 (left), and the corresponding relative error (right) with respect to a full SPARC solution.


Kinetic Energy Streamlines-UQ

Kinetic Streamlines Rel-Error

Kinetic Energy Streamlines-UQ

Kinetic Streamlines Rel-Error

Figure 4-8. Plots of kinetic energy computed from the streamline solutions for (left), and the corresponding relative error (right) with respect to a full SPARC solution. The plots on the top are for case-1, and the bottom for case-2.

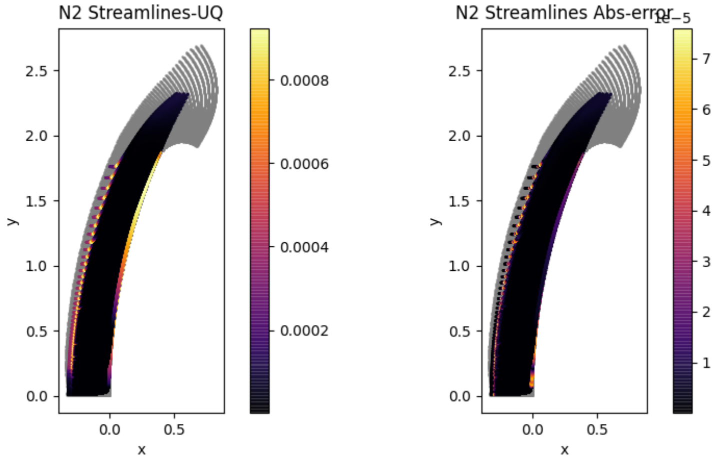


Figure 4-9. Plots of $N2$ mass fraction computed from the streamline solutions for case-1 (left), and the corresponding relative error (right) with respect to a full SPARC solution.

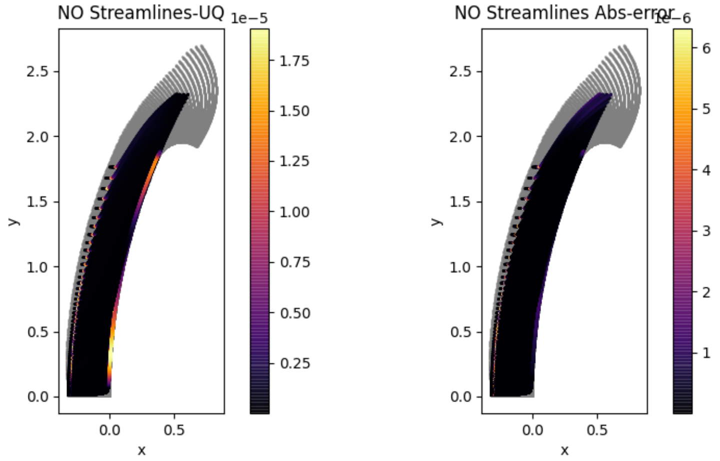


Figure 4-10. Plots of NO mass fraction computed from the streamline solutions for case-2 (left), and the corresponding relative error (right) with respect to a full SPARC solution.

5. CONCLUSIONS AND FUTURE WORK

With numerical models for high consequence applications becomes commonplace, the need for extensive UQ of these computational models is self-evident. At the same time, potentially high computational cost of these simulation models requires us to rely on surrogate models, such as stochastic collocation, for efficient uncertainty propagation. We presented a method of improving the accurate of stochastic collocation surrogate models in the context of ODE systems by introducing a separation between state variables and stochastic parameters in the dynamics evaluation. The dynamics surrogates are constructed in a data-driven fashion and are minimally invasive, leading to easy integration with existing simulation codes.

We note that in the current form of dynamics surrogate construction, we do not impose any constraints such as conservation of energy. In future, we will utilize such conservation laws to construct physically robust surrogates. We also wish to compare the performance of our method against existing surrogate construction frameworks such as physics-informed neural networks (PINNs) [28] and neural ODEs [29]. An associated issue that needs to be addressed is ensuring stability of the temporal integration of the samples with approximate dynamics, since this is not explicitly guaranteed. Approaches that deal with approximation of dynamics, e.g. discrete empirical interpolation of manifold (DEIM) used in conjunction with projection-based reduced order models (ROMs), have a similar issue and literature in this field could be used to address stability of approximation. Our long-term objective is to provide this UQ capability in Sandia codes. To that end we will chart a path for integrating SAMURAI with SPARC+ARIA, and ODEUQ with SIERRA-SM.

REFERENCES

[1] C. Park. *Nonequilibrium Hypersonic Aerothermodynamics*. Wiley, 1990.

[2] Roger Ghanem, David Higdon, and Houman Owhadi. *Handbook of uncertainty quantification*, volume 6. Springer, 2017.

[3] Carl Edward Rasmussen and Christoper K. I. Williams. *Gaussian processes in machine learning*. MIT Press, 2006.

[4] Martin D Buhmann. *Radial basis functions: theory and implementations*, volume 12. Cambridge university press, 2003.

[5] Dongbin Xiu. *Numerical methods for stochastic computations: a spectral method approach*. Princeton university press, 2010.

[6] Dongbin Xiu. Stochastic collocation methods: a survey. *Handbook of uncertainty quantification*, pages 699–716, 2016.

[7] Ralph C Smith. *Uncertainty quantification: theory, implementation, and applications*, volume 12. Siam, 2013.

[8] Michael Eldred and John Burkardt. Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification. In *47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition*, page 976, 2009.

[9] Ivo Babuška, Fabio Nobile, and Raul Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 45(3):1005–1034, 2007.

[10] Kenneth C Hall, Jeffrey P Thomas, and Earl H Dowell. Proper orthogonal decomposition technique for transonic unsteady aerodynamic flows. *AIAA journal*, 38(10):1853–1862, 2000.

[11] Kevin Carlberg, Charbel Farhat, Julien Cortial, and David Amsallem. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. *Journal of Computational Physics*, 242:623–647, 2013.

[12] Charbel Farhat, Radek Tezaur, Todd Chapman, Philip Avery, and Christian Soize. Feasible probabilistic learning method for model-form uncertainty quantification in vibration analysis. *AIAA Journal*, 57(11):4978–4991, 2019.

[13] Martin Drohmann and Kevin Carlberg. The ROMES Method for Statistical Modeling of Reduced-Order-Model Error. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):116–145, 2015.

[14] Patrick J. Blonigan, Francesco Rizzi, Micah Howard, Jeffrey A. Fike, and Kevin T. Carlberg. Model Reduction for Steady Hypersonic Aerodynamics via Conservative Manifold Least-Squares Petrov–Galerkin Projection. *AIAA Journal*, 59(4):1296–1312, 2021.

[15] Brian A. Freno, Brian R. Carnes, and V. Gregory Weirs. Code-verification techniques for hypersonic reacting flows in thermochemical nonequilibrium. *Journal of Computational Physics*, 425:109752, 2021.

[16] NimbleSM:Lagrangian finite-element code for solid mechanics on next-generation computing platforms. `https://github.com/NimbleSM/NimbleSM`.

[17] S.B. Pope. PDF methods for turbulent reactive flows. *Progress in Energy and Combustion Science*, 11(2):119–192, 1985.

[18] Stephen B Pope. Lagrangian modelling for turbulent flows. In *Theoretical Approaches to Turbulence*, pages 369–373. Springer, 1985.

[19] D. M. Tartakovsky and S. Broyda. Pdf equations for advective-reactive transport in heterogeneous porous media with uncertain properties. *Journal of Contaminant Hydrology*, 120-121:129–140, 2011.

[20] D. Venturi, D. M. Tartakovsky, A. M. Tartakovsky, and G. E. Karniadakis. Exact PDF equations and closure approximations for advective-reactive transport. *Journal of Computational Physics*, 243:323–343, 2013.

[21] F. Boso, S. V. Broyda, and D.M. Tartakovsky. Cumulative distribution function solutions of advection-reaction equations with uncertain parameters. *Proceedings of The Royal Society A*, 470(20140189), 2014.

[22] E. Phipps, M. D'Elia, H. C. Edwards, M. Hoemmen, J. Hu, and S. Rajamanickam. Embedded ensemble propagation for improving performance, portability, and scalability of uncertainty quantification on emerging computational architectures. *SIAM Journal on Scientific Computing*, 39(2):C162–C193, 2017.

[23] M. D'Elia, H. C. Edwards, J. Hu, E. Phipps, and S. Rajamanickam. Ensemble grouping strategies for embedded stochastic collocation methods applied to anisotropic diffusion problems. *SIAM/ASA Journal on Uncertainty Quantification*, 6(1):87–117, 2018.

[24] Jochen Garcke. Sparse grids in a nutshell. In *Sparse grids and applications*, pages 57–80. Springer, 2012.

[25] R.E. Jones, M.T. Redle, H. Kolla, and J.A. Plews. Erratum to "a minimally invasive, efficient method for propagation of full-field uncertainty in solid dynamics" (int. j. numer. meth. engng. 2021; https://doi.org/10.1002/nme.6818). *International Journal for Numerical Methods in Engineering*, 123(7):1710–1711, 2022.

[26] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

[27] Kevin Neitzel, Daniil Andrienko, and Iain D. Boyd. Aerothermochemical nonequilibrium modeling for oxygen flows. *Journal of Thermophysics and Heat Transfer*, 31(3):634–645, 2017.

[28] Shengze Cai, Zhiping Mao, Zhicheng Wang, Minglang Yin, and George Em Karniadakis. Physics-informed neural networks (pinns) for fluid mechanics: A review. *Acta Mechanica Sinica*, pages 1–12, 2022.

[29] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural odes. *Advances in Neural Information Processing Systems*, 32, 2019.

DISTRIBUTION

Hardcopy—External

Number of Copies	Name(s)	Company Name and Company Mailing Address

Hardcopy—Internal

Number of Copies	Name	Org.	Mailstop

Email—Internal (encrypt for OUO)

Name	Org.	Sandia Email Address
Technical Library	1911	sanddocs@sandia.gov

Sandia National Laboratories