

Intrusive Uncertainty Quantification for NeK5000: Development of Intrusive Uncertainty Quantification for High-Dimensional, High-Fidelity Codes

Mathematics and Computer Division

About Argonne National Laboratory

Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

Availability of This Report

This report is available, at no cost, at <http://www.osti.gov/bridge>. It is also available on paper to the U.S. Department of Energy and its contractors, for a processing fee, from:

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
phone (865) 576-8401
fax (865) 576-5728
reports@adonis.osti.gov

Disclaimer

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 UChicago Argonne, LLC, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents 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 or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or UChicago Argonne, LLC.

Intrusive Uncertainty Quantification for NeK5000: Development of Intrusive Uncertainty Quantification for High-Dimensional, High-Fidelity Codes

prepared by
O. Roderick and M. Anitescu
Mathematics and Computer Science Division, Argonne National Laboratory

September 30, 2012

ABSTRACT

We study the use of lower-fidelity training data for uncertainty quantification of complex simulation models. In our approach, computationally expensive full-model outputs are approximated applying proper orthogonal-decomposition-based dimensionality reduction to the full model.

A Gaussian-processes-based machine learning approach is then used to model the difference (or, rather, the correspondence) between the higher-fidelity and the lower-fidelity data. This stochastic model can be constructed by using few additional full code evaluations; it is used to calibrate arbitrarily many lower-quality outputs in order to create additional training data for sampling-based analysis. In effect, an adequate approximation of the simulation model's response to uncertainty can be constructed at a modest computational cost. In fact, we aim at the number of full-model evaluations comparable to that required for a linear approximation (as opposed to numbers traditionally associated with sampling in high-dimensional spaces).

In this report, we explain the basic algorithm, suggest some performance tuning options, and give a first characterization for the class of simulation models of nuclear engineering for which the suggested report is effective.

The primarily goal of our work was to demonstrate the effectiveness of multifidelity analysis on high-performance fluid dynamics simulation code Nek5000. We have been successful at estimating statistics for an output of interest at low cost. While further demonstration exercises may be helpful, the information provided here clearly argues for the effectiveness of the approach. Logically, the next stage of work is to explore the full range of tasks related to Nek5000 code development and verification that could be made more computationally accessible with the use of calibrated lower-fidelity data.

The suggested approach can be generalized to multiple levels of fidelity, enabling uncertainty analysis based on many different model approximations. The long-term goal of our research is to fit the developed method into a larger context of advanced uncertainty analysis tools for nuclear engineering applications. We foresee implementation as a suite of analysis tools that can become a component of SHARP / NEAMS codes.

TABLE OF CONTENTS

Abstract	i
Table of Contents	iii
List of Figures	v
List of Tables.....	v
1 Introduction	1
2 Method Description	3
2.1 Uncertainty Analysis in the Context of Previous Work.....	3
2.2 POD-Based Dimensionality Reduction of Model Dynamics	5
2.3 Gaussian Processes-Based Inference of Covariance.....	6
2.4 Method Algorithm and Tuning Options.....	7
3 Results	10
3.1 Prototype Models and Nek5000.....	10
3.2 Numerical Results	12
3.3 Discussion	16
4 Summary.....	17
Acknowledgments.....	18
References	18

LIST OF FIGURES

Figure 1: Reduced-order model sampling and calibration process	9
Figure 2: Navier-Stokes flow, uncertainty in geometry	12
Figure 3: Model 1: prediction on reduced model outputs; 90% confidence interval	13
Figure 4: Model 2: prediction on reduced model outputs; uncertainty dimension 4	14
Figure 5: Nek5000: prediction and confidence interval, training set of 5 points	15
Figure 6: Nek5000: prediction and confidence interval, training set of 3 points	15
Figure 7: Nek5000: 95-95 estimate	16

LIST OF TABLES

Table 1: Model 1: prediction on reduced model data	13
Table 2: Nek5000: prediction on reduced model data	14

1 Introduction

The modern field of nuclear engineering relies on numerical simulations of physical systems for tasks of reactor prototype development, performance optimization, and safety analysis. The current state of development of hardware and computing techniques allows models of high geometric resolution and physical fidelity, with a large number of parameters taken into account by simulation. However, with the use of high-fidelity simulations, all analysis tasks for such models require increasingly large amounts of computational and development resources. When a single simulation run may take hours, days, or weeks of computation, any assessment method that relies on traditional sampling is no longer feasible.

We investigate the task of propagating uncertainty through a complex simulation model in the situation of extremely tight computational budget constraints. We aim to develop hybrid, intrusive methods of uncertainty quantification that are effective in a high dimension of uncertainty without large-scale sampling. This task can be achieved through intrusive analysis that extracts additional data from each of the few model runs. We expanded on this idea by introducing a method that *relates* additional data to each of the expensive model runs. That is, each full-model evaluation that produces high-fidelity outputs is augmented with many (perhaps arbitrarily many) simplified model runs that produce lower-fidelity outputs.

The work was completed in fulfillment of milestone M4MS-12AN0603242, under the Nuclear Energy Advanced Modeling and Simulation (NEAMS) program of the U.S. Department of Energy Office of Nuclear Energy.

As a central exercise, we demonstrated effectiveness of the suggested approach on an analysis of a simulation performed using the SHARP component code Nek5000 [1]. This code is an industry-level high-fidelity scalable fluid mechanics solver. It includes the capability to accept user-defined routines and scenarios, but it is not transparent to intrusive analysis. Results of our analysis can be reproduced only by using large-scale sampling (hundreds of runs or more). Our work, while still in the initial stages, has made practical those tasks of code verification and confidence assessment that previously were not performed for simulation models of higher complexity.

The existing work on uncertainty quantification mainly follows the same scheme, variously called surrogate modeling, response surface learning, or multivariate regression [2]. An uncertainty-influenced output is fitted to an a priori chosen algebraic structure on the inputs, based on extensive sampling. In the development of our intrusive uncertainty quantification methods, we always aimed for a significant, as opposed to incremental, improvement of the existing methodology. In the past, we have shown that additional data (proportionally to the dimension of uncertainty) can be extracted from the model if model derivatives are also available [3,4]. That work significantly accelerated multivariate regression and stochastic-processes-based techniques for quantifying response to uncertainty. Now we introduce another way to obtain additional data that helps with response model construction.

Generally, simulation models do not exist alone, as sole examples in their class of codes. During development, there will be simplified versions of the code that are mostly consistent in the input-output format but run much faster because of lower resolution, lower-precision requirements, and fewer enabled routines. If access to such simplified models is no longer

available, a simplified approximation can be constructed at modest additional development effort. Proper orthogonal decomposition (POD) based dimensionality reduction is a particularly convenient choice, and we use it in our work.

By necessity, model simplifications and approximations are used as a rough tool to explore the model, either as direct replacement of the more expensive data or as a source of first guess at the model's properties. Intuition-based use of model simplifications for intermediate verification is unavoidable in applied studies, if not always documented. The consequences of simplification are tolerated as long as the essential, global properties of the model are preserved. As an important step toward systematic use of multifidelity data, Kennedy and O'Hagan [5] suggested constructing a statistical model to describe the imperfection in the cheaper data (error-prone computer simulations versus almost perfect physical observational data, in their case).

We argue for use of an approach based on the following understanding:

- Lower-quality data differs from high-quality data by a stochastic response surface that can be learned.
- This learning, or calibration, process can have a small number of degrees of freedom.
- Calibrated lower-quality data can be used to replace the unavailable high-quality data in the analysis.
- Lower-quality data is available from model simplification techniques such as POD-based dimensionality reduction; construction of simplifications is possible at modest development effort.
- In particular, lower-quality data preserves some characteristics of high-quality data in thermohydraulic models of interest for the nuclear engineering field, as demonstrated by the recent work in extrapolation of behavior in time by using reduced-order modeling [6].

The attractiveness of the proposed approach is not so much in pointwise precision in predicting the model's response to uncertainty as in the enabled assessment of model's global properties. In particular, we show how to output confidence intervals for the statistics of the model, even for order statistics (percentiles). This task would normally require extensive sampling, whereas we aim to accomplish analysis at a computational cost comparable to or lower than that of linear approximation.

We note that the suggested approach can be generalized to using multiple sources of lower-fidelity data. That has implications for our recent and ongoing work in intrusive uncertainty analysis of simulation codes (derivative-augmented regression techniques). The implications also point in both directions: if derivative information on the model is available, it can be used to accelerate model reduction and calibration components of the present work.

Based on our success with analysis of Nek5000 simulations, we suggest that the next stage of work is to extend the approach to the full range of tasks related to code development and verification, which could be improved through the use of lower-fidelity data, and to implement the approach as an analysis tool within the NEAMS software suite.

2 Method Description

We begin this section with a review of uncertainty analysis. We then discuss POD-based analysis and Gaussian-processes-based inference of covariance. We conclude the section with a description of our algorithm.

2.1 Uncertainty Analysis in the Context of Previous Work

Consider a simulation model as a system of differential-algebraic equations

$$F(T, P, x) = 0 \quad (1)$$

implemented by a computer code. Here, $T = (T_1, T_2, \dots, T_n)$ is a model state vector, $P = (P_1, P_2, \dots, P_N)$ is a set of model parameters explicitly present in the inputs or in the body of the code, and $x = (x_1, x_2, \dots, x_m)$ are the additional parameters that quantify uncertainty in the code. Such uncertainty quantifiers may be understood as direct variations on the nominal values of parameters, $P_i := P_{i,0} + x_i$, but we also accept any other deterministic manner of inserting uncertainty into the code; the number of uncertainty quantifiers may be greater than the number of parameters. It may be better to think of model equations with uncertainty in the coupled form $F(T, P(T, x), x) = 0$.

The essential task of uncertainty analysis amounts to describing an uncertainty-influenced scalar output $\mathfrak{J}(T)$ by creating an approximation $J(x) \approx \mathfrak{J}(T(x))$ with some capability for extrapolation. A standard technique is multivariate regression. Given a family of functions (often, multivariate polynomials) $\{\Psi\}: R^m \rightarrow R$, an uncertain output is represented by an expansion

$$J(x) \approx R(x) = \sum_j a_j \Psi_j(x) \quad (2)$$

with expansion coefficients a_j found by regression. This representation is chosen based on a tradeoff between computational cost and quality: we would like to use a larger basis to represent nonlinear dynamics in the model, but at the same time we would like to fit a regression model based on an affordable amount of training data. The computational cost grows rapidly with the dimension of the uncertainty space. For example, a polynomial expansion on 20 variables requires over 200 model evaluations if second-order polynomials are used, and over 1500 evaluations in third-order polynomials [7].

For improved prediction, in order to avoid overtraining on the available data and to gain an estimate of confidence in prediction, the regression model (2) can be treated as a first version of the response function and enhanced with a correction term: a deterministic or a stochastic representation of the error. In the absence of an analytical solution of the model, a posteriori error estimation techniques will carry an additional computational cost, in the form of additional direct, or adjoint, model evaluations.

One can avoid constructing a regression model altogether and estimate the statistics of the output directly from sampling. But that approach is hardly computationally cheaper. For example, the Kolmogorov-Smirnov technique for comparing an empirical distribution against

a template will require sampling in the hundreds even for the one-dimensional case. This choice also limits the ability for interpolation.

Given our tight limitations on computational budget (as measured in evaluations of the simulation model at distinct points in the uncertainty space), we aim to use the model capability to provide additional data via simplifying approximations. We have to use a process that can correct this additional data at the cost of few full-model evaluations. Informally, we can say that we seek imperfect data that is different from perfect data by only a few degrees of freedom.

Once the pathway to generating and calibrating lower-quality data is established, we propose to use this data to create both regression-based response surfaces and to find statistical parameters by sampling of calibrated data.

The mathematical content behind work has received increased attention in the past decade. The dimensionality reduction component is based, initially, on a statistical technique known as the Karhounen-Loeve expansion, and also variously as the principal component analysis method, POD, and method of snapshots [8]. It has been restated for applied tasks of simulation and optimization in high dimension by many authors [9,10]. Moreover, additional attention has been given to developing goal-oriented reduced models; the techniques range from simple data weighting to iterative procedures resulting in a POD-reduced model that optimally reproduces a known feature in the full-model dynamics. Multiple researchers use POD-reduced models in a straightforward manner: to replace the full model and cheaply propagate model dynamics forward in time. In particular, E. Merzari published a paper on using the reduced-order ODE solver to recover the Nek5000 velocity field from high-dimensional snapshots and simplified, reduced Navier-Stokes equations [6].

The central idea of using statistical models to characterize the imperfection in training data and correct it while recovering the covariance properly belongs to Kennedy and O'Hagan [5]. For the background of Gaussian-processes machine learning aspects of our work, we refer to the reader to a publicly available manual by Rasmussen [11].

We are also aware of several other directions of development in uncertainty analysis of fluid mechanics models, based on Kennedy and O'Hagan's work (for example, [12]). We point out a line of development that is perspective in general but is not effective on small training data, an approach currently called multifidelity kriging [13]. In this approach, high-fidelity and low-fidelity data responses to uncertainty are considered to be statistically independent, so two separate response models are constructed and then aggregated. This approach uses the fewest assumptions on the relationship between perfect data and its approximations; it may, however, require separate sampling procedures.

The suggested approach for the use of lower-fidelity data fits well with our previous work on using derivatives to accelerate surrogate model construction. In fact, we can reformulate our older work in a new language: having a gradient is equivalent to having access to lower-quality (linear) approximations of the model. It appears that for uncertainty quantification of advanced simulation models *at least some* intrusive analysis capacity must be made available: either algorithmic differentiation or POD-based reduction (or some other as-yet-undeveloped approach).

2.2 POD-Based Dimensionality Reduction of Model Dynamics

Given a static data set of n' observations, stored as a matrix $A \in R^{n \times n'}$, an optimal approximation \hat{A} of fixed rank $k < n$ is obtained using Singular Value Decomposition (SVD). If $A = \sum_{i=1}^n u_i \sigma_i v_i^T$, with singular values σ_i in descending order, then $\hat{A} = \sum_{i=1}^k u_i \sigma_i v_i^T$;

the error $\|A - \hat{A}\|_2 = \sum_{i=k+1}^n \sigma_i$ is minimal for the given k . In practice, the dimension k is chosen

so that the relative error $\sum_{i=k+1}^n \sigma_i / \sum_{i=1}^n \sigma_i$ is close to 1.

A similar approach for reducing evolving data is variously known as Proper Orthogonal Decomposition (POD), Principal Component Analysis (PCA), or method of snapshots. According to the method of snapshots (Sirovich), reducing projection can also be applied to underlying equations of a mathematical model. The available observations (or snapshots) from a solution trajectory at times t_1, t_2, \dots, t_n are recorded as matrix columns: $A = (T(t_1), T(t_2), \dots, T(t_n))$. An empirical correlation matrix is defined as $C = AA^T$. In practice, instead of a SVD of A , we may solve the related eigenvalue problem $C\phi = \lambda\phi$, and record the k dominant eigenvectors as $\Phi = (\phi_1, \phi_2, \dots, \phi_k)$. The matrix Φ is used to project the model state-vector into a dominant eigenspace of the correlation matrix. The projection is optimal for the training data. If training data is representative of the model dynamics (and if true model dynamics are essentially low-dimensional), then the POD approximation

$$\hat{T}(t) = \sum_{i=1}^k q_i(t) \phi_i \quad (3)$$

is effective. Here, $q(t)$ is the approximating trajectory in low-rank subspace coordinates. POD-based projection can be used for many formats of the problem. We can write a reduced form of the general DAE (2) as $F(\Phi q, t, P, x) = 0$. In particular, for a model discretized to an ODE we have:

$$\begin{aligned} \frac{du}{dt} &= f(u, t, P, x) \\ u(t_0) &= u_0(P, x) \end{aligned} \quad (4)$$

The reduced-order approximation is determined by solving

$$\begin{aligned} \frac{dq}{dt} &= \Phi^T f(\Phi q, t, P, x) \\ q(t_0) &= \Phi^T u_0(P, x) \end{aligned} \quad (5)$$

The main advantage here is the reduced integration time in comparison to the full model equations. The difficulty of implementing reduction and, furthermore, the difficulty of intrusive model analysis depend on access to the right side operator $f(u, t, P, x)$. This operator has to be identified in the code and evaluated, furthermore, evaluated in the lower-dimensional space, without explicitly calculating the ‘‘lifted’’ term Φq .

This problem of operator identification and evaluation will appear in any representation of the reduced model equations, in particular, in the POD-Galerkin setup for the reduced-order Navier-Stokes equations.

Suppose the model dynamics are reducible. We expect that if $u \approx \hat{u}$, then the reduced model will also provide an effective approximation for an output of interest: $\mathfrak{I}(u) \approx \mathfrak{I}(\hat{u})$. In general, we cannot make similar empirical statements concerning the effects of uncertainty. The reduced model is based on a projection matrix, based on a set of snapshots. There is no guarantee that a set of snapshots obtained at a specific point in the uncertainty space will be adequate for estimating model's response to other values of x .

It is possible to the POD basis Φ to some extent, by weighting of snapshot data, or by combining snapshots from trajectories started from several different points in the uncertainty space-15]. In practice, this improvement can be very limited, and any procedure for enrichment will run into problems of computational cost (we will not have the additional budget of full model runs) and circular logic (we cannot "emphasize" the response to uncertainty in the reduced model snapshots without knowledge about full model's response to uncertainty).

We can, however, take the reduced-model output data as is, and pass it to the next stage of the process: machine learning of the imperfection in the reduced model performance over the uncertainty space.

2.3 Gaussian-Processes-Based Inference of Covariance

At the core, Gaussian-processes (GP) machine learning is a response surface fitting technique; unlike standard regression, it estimates the shape of the covariance function on the training data, rather than the shape of the data itself [11].

Consider a basic task of constructing the GP representation on the training set of outputs Y_o corresponding to a set of inputs $X_o = (x_1, x_2, \dots, x_N): x_i \in R^n$. We denote the covariance function on the inputs by $\text{cov}(x, x'; \theta): R^n \rightarrow R$. For the algebraic form of covariance, we shall use an empirically effective "Matern 3/2" function

$$\text{cov}(x, x'; \theta) = \sigma^2 \sum_{l=1}^n \left(1 + \sqrt{3} \left| \frac{x_l^i - x_l^j}{\theta_i} \right| \right) \cdot \exp \left(-\sqrt{3} \left| \frac{x_l^i - x_l^j}{\theta_i} \right| \right) \quad (6)$$

(Note: a small number of other suitable functions are also available). The parameter σ^2 here is estimated by $\text{var}(Y_o)$. The *hyperparameters* θ are modified to adjust the shape of the covariance function to best fit the training data using the maximal marginal likelihood approach. It is convenient to write out the probability that training data follows Gaussian distribution with covariance $\text{cov}(x, x'; \theta)$ in negative logarithmic form, then fitting of the hyperparameters amounts to maximizing

$$-\log(\Pr(Y_o | X_o; \theta)) = -\frac{1}{2} Y^T K^{-1} Y - \frac{1}{2} \log |K| - \frac{m}{2} \log(2\pi) \quad (7)$$

where $K = \text{cov}(X_o, X_o; \theta)$ is the covariance matrix on the training data. The nonlinear optimization problem (7) may present a computational challenge, and does not, in practice, yield an ideal solution: we expect to find an acceptable local minimum using standard tools based on a BFGS algorithm (Matlab routines *fminsearch* and *fminunc* are acceptable here).

Now that the covariance function is specified, we can perform pointwise predictions on any testing set by kriging. A cross-covariance matrix $K_v = \text{cov}(X_o, X_v; \theta)$ relates testing set X_v to the training set X_o . The outputs are predicted as

$$Y_v = K_v^T \cdot K^{-1} \cdot Y_o \quad (8)$$

For the initial (or *prior*) version of mean function, we use a regression model $R(x)$ (it can be linear, if higher-order approximation cannot be constructed). A posterior version of the mean is written as

$$R'(x) = R(X_o) + K_v^T \cdot K^{-1} \cdot (Y_v(x) - R(x)) \quad (9)$$

A more important posterior prediction for the variance is given by

$$V(X) = \sigma^2 \cdot (\text{cov}(x, x; \theta) - K_v^T \cdot K^{-1} \cdot K_v) \quad (10)$$

The estimate (10) provides confidence information for pointwise predictions such as (9) by using Chebyshev's rule, or the less conservative empirical rule. The effectiveness of the estimate depends on how well the underlying assumptions for Kriging are satisfied by the data: essentially, the training data is of sufficient quality to estimate the variogram correctly [17].

Now we return to the context of uncertainty quantification. GP provides us with a tool that performs unbiased, locally smooth prediction on training data, and also provides confidence intervals for the prediction, we seek to apply it to the situation where training data is some measure of the difference between outputs of the perfect and imperfect versions of the simulation model.

A straightforward application of GP prediction is to define the training data as $Y_o = \mathfrak{S}(x) - \hat{\mathfrak{S}}(x)$, and then output the calibrated data as $J_c = \hat{\mathfrak{S}}(X_v) + Y_v$. The confidence intervals would be written as $(J_c(x) - zV(x), J_c(x) + zV(x))$ where z is the appropriate coefficient corresponding to the measure of confidence: for example, $z = 1.645$ for 90% confidence using the empirical rule. Note that this approach will require n full model evaluations to form the training data (a bit fewer if we allow the parameters θ to have dependencies on each other; usually they are assumed to be independent). If the computational budget does not allow this, we will need to use a different strategy of relating reduced model data to full model data: one alternative is suggested in the following section.

2.4 Method Algorithm and Tuning Options

The proposed method consists of three components that are executed in sequence, and can be tuned for improved performance independently of each other (because feedback and adaptive improvement will, counter-productively, require additional model evaluations). The components are:

- POD-based model reduction
- Assembly of training data for calibration
- GP-based calibration.

Model reduction requires sampling of the model trajectory; thus the basic cost is one complete (or incomplete!) model evaluation. We choose a single point in the uncertainty space (by default, uncertainty quantifiers are given nominal values, for example, $x_i = 0$) and evaluate the model. From this process, only the projection matrix Φ needs to be stored.

In principle, POD-based model reduction has a lot of tuning options available. It is reasonable to use the snapshot selection and weighting schemes if they are suggested by the model developer, but in general such schemes require more knowledge about the model (in particular, about its sensitivities) than would be available in complex cases under the influence of uncertainty.

If several full-model runs are available, with projection matrices Φ_1, Φ_2, \dots , they can be aggregated into the form $\Phi' = \text{span} \langle \Phi_1, \Phi_2, \dots \rangle$; in practice that can be accomplished by the Gram-Schmidt orthogonalization procedure. Moreover, increasing the reduced model dimension k may improve the performance of the reduced model.

By design, the reduced model will be inexpensive to evaluate at any point in the uncertainty space. As opposed to costly high-fidelity data, we can have low-fidelity samples of almost arbitrary size and consider the reduced model outputs on both the training set and the testing set known, almost for free.

Note, however, that the organization of the full- and the reduced-model code may differ; in fact, most of the development effort for implementing the overall method may be spent on overcoming such difficulties. For example, the reduced model may have a nonexplicit initial state of the model (using the first snapshot instead) or may use an adaptive geometrical mesh that ends up being different from that of the full model. Initial-state uncertainties and geometric uncertainties are, correspondingly, of different format; and one needs to modify the code and/or to interpolate.

There is no question as to what side the code modifications should be performed on: we advise always modifying the reduced code. Notably, the calibration procedure accounts for all the differences between the models, without distinguishing their source. Thus it will also partially make up for interpolation errors and even human coding errors. Of course, the overall quality of prediction depends on the situation.

Assembly of the training set depends on which interpretation of the “difference” between the full and the reduced data better fits the available budget. Besides the arithmetic difference between the outputs, we can use any correspondence between the perfect and approximate outputs on the same input that could be used in the calibration process. The only restriction on this generic relationship $Y_o = \mathfrak{S}(X_o) \leftrightarrow \hat{\mathfrak{S}}(X_o)$ is that it must be reversible, that is, at least locally bijective and smooth.

With the computational budget of the full-model runs approximately equal to the dimension of the uncertainty space, we sample the full and reduced models in pairs and use $Y_o = \mathfrak{S}(X_o) - \hat{\mathfrak{S}}(X_o)$.

If the number of runs must be less than m , we suggest a scheme based on comparing *both* the full- and reduced-model outputs *with an intermediate response function*, instead of *with each other*. We construct an intermediate regression model $r(x)$ using the available full-model runs only: in practice, it will be a low-rank least-squares linear approximation for $\mathfrak{S}(x)$. Then we define the training set as $Y_o = \{\mathfrak{S}(X_o) - r(X_o), \hat{\mathfrak{S}}(X_o) - r(X_o)\}$. The calibration procedure will then add an error estimate to an interpolation from the nearest known values of $\mathfrak{S}(x)$ and $r(x)$. The full-model outputs in this training set can be weighted, in other words, repeated multiple times for increased influence on the resulting variance structure. This approach will have a lower quality in pointwise prediction but is dramatically (for its computational cost) successful in recovering the variance.

Ultimately, the choice of the correspondence relationship “ \leftrightarrow ” involves a tradeoff between computational cost and quality. On the one hand, the assembly of the training set should require only a few full-model evaluations; on the other hand, the correspondence must be reversible, allowing one to decipher the predicted values Y_v into calibrated data J_c without significant loss of quality. Clearly, more options will be discovered with additional research effort.

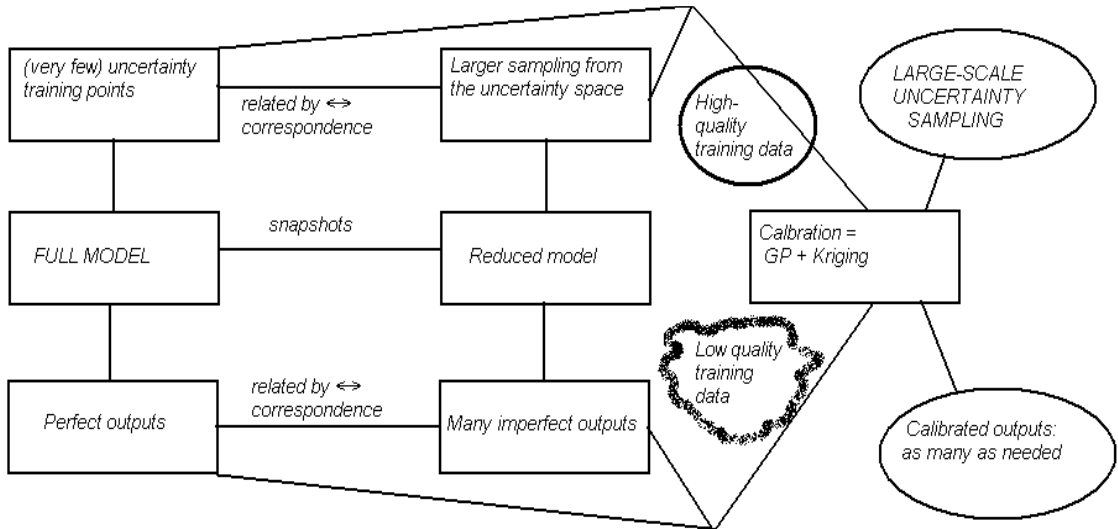


Figure 1: Reduced-order model sampling and calibration process.

Gaussian-processes learning of covariance structure is treated mostly as a black-box procedure for producing posterior mean and variance from a training data set, however that is defined. The performance of the process largely depends on the nonlinear optimization algorithm chosen to solve the marginal likelihood problem.

The scheme of the complete process, from construction of the reduced model to calibration of lower-quality data, is depicted in Figure 1.

Once the calibration process is complete and we have an inexpensive source of training data J_C with a corresponding estimation of variance, the statistics of the model under uncertainty can be assessed by sampling the calibrated data. In particular, we now have enough data for a straightforward method for estimating confidence intervals for order statistics (DasGupta, 2008). Consider the p th quantile for the distribution of the output of interest \mathfrak{S} , and choose a confidence level α . A commonly used “95/95 estimate” has $p = 0.95, \alpha = 0.05$. A conditional quantile estimate approach involves uniformly sampling J_C over the uncertainty space and recording the p th quantile for each sample, thus forming variable c_p . The sought-for double-sided confidence interval is estimated from the empirical distribution of c_p , that is, from an empirical distribution function (EDF) defined on M points by $Z(t) = \frac{1}{M} \sum_{i=1}^M \mathbb{1}[c_p < t]$; here $\mathbb{1}[\dots]$ is the binary indicator of the event.

We note that insight into the behavior of the model has traditionally been used in tuning model reduction and choosing the most appropriate transform for representation of training data for subsequent statistical analysis. At this stage, we cannot provide a full range of options even for the field of fluid mechanics (although this is an interesting direction for future work). We emphasize, however, that including lower-fidelity data and relating it to the high-fidelity data by the means of an additional statistical model enables many forms of additional analysis, previously not possible because of computational budget constraints.

3 Results

In this section we present and analyze our numerical results with prototype models and models using Nek5000.

3.1 Prototype Models and Nek5000

In preparation for the advanced uncertainty analysis on the high-performance fluid dynamics simulation code Nek5000, we tested our approach on several prototype models of Navier-Stokes flow. Since we developed the models, it was straightforward to implement dimensionality reduction and to ensure that the format of uncertainty is consistent between the versions of the model. We briefly describe two of the codes used in the method early development stages.

Model 1 was a much-simplified 2-D flow, described by a PDE

$$\begin{aligned} \rho \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) &= \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \\ \rho \left(\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right) &= \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right), \end{aligned} \tag{11}$$

discretized on a rectangular grid, that is, in a rectangular channel with proportions 1×10 . The ODE was integrated over a time interval of length 1 with a time step $\Delta t = 0.01$, with in-flow conditions

$$\begin{aligned} u(x, y = 0, t) &= (u_1, u_2, \dots, u_n)^T \\ v(x = 0, y, t) &= 0 \end{aligned} \quad (12)$$

Uncertainty was introduced into values u_i and allowed to vary +/- 30% from unity value.

The output of interest was an averaged quantity $J = \frac{2\Delta t}{20n^2} \sum_{x,y} \sum_{t=0.25}^{0.75} \frac{1}{v^2(x, y, t)}$ (this quantity is

approximately proportional to the drag coefficient, is smooth as long as the velocity is non-zero, and is sensitive to small changes). The usual dimension of uncertainty m was 7 to 12.

The corresponding state vector dimension $10n^2$ ranged from 490 to 1440. A straightforward, unweighted POD-based reduction to dimensions 40 to 150 was implemented; a reduced model could be integrated 15–20 times faster than the full model (this computation time does not include the one-time construction of reducing projection based on a given set of snapshots). The reduced model produced pointwise relative errors of 3–5%.

Model 2 was a better approximation to realistic Navier-Stokes flow with uncertainty. We implemented a finite-element method (FEM) model of Navier-Stokes flow in a rectangular channel of proportions 3×10 , with uniform inflow and isolated walls.

$$\begin{aligned} \frac{\partial u}{\partial t} + (u \cdot \nabla)u - \frac{1}{\text{Re}} \Delta u + \nabla p &= 0 \\ \nabla \cdot u &= 0 \end{aligned} \quad (13)$$

For the reduced model $u(x, t) \approx u_c(x) + \sum_{i=1}^k q_i(t)\phi_i(x)$, with an assumption for the inner product $(\nabla p, \phi_n) = 0$, we have a POD-Galerkin representation

$$\left(\frac{\partial u_i}{\partial t}, \phi_i \right) + ((u_i \cdot \nabla)u_i, \phi_i) + \left(\frac{2}{\text{Re}} D(u_i), \nabla \phi_i \right) = 0 \Leftrightarrow \frac{dq}{dt} = A + Bq + q^T Cq \quad (14)$$

$i = 1, 2, \dots, k$

This setup follows the work of E. Merzari et al. [6] in implementing a reduced-order solver for Nek5000. Note that we only need to obtain the terms A , B , C from the full, high-fidelity model. In terms of the ODE reduction described previously, this is equivalent to obtaining the expression for the reduced right-side operator of a reduced ODE. Of course, the stored arrays A , B , C are completely valid only on a single point of the uncertainty space.

We introduced uncertainty in two ways. In setup **2a**, the only uncertainty quantity was the Reynolds number Re . In setup **2b**, we kept the Reynolds number constant (at $\text{Re} = 200$) but introduced geometric irregularities into the shape of the channel. In Figure 2 we show the configuration with the dimension of uncertainty equal to 4. The uncertain quantities describe the shape of the two small bumps (each made of two semi-circles of uncertain radius) on one of the sides of the channel; gradient coloring shows one of the solutions for the velocity field). In both cases, the output of interest was the average drag coefficient over the deformed wall. (Note: formally, a change in the geometric shape would lead to a change in the dimension of model state vector; we used FEM area elements with area 0 to represent the deformations.)

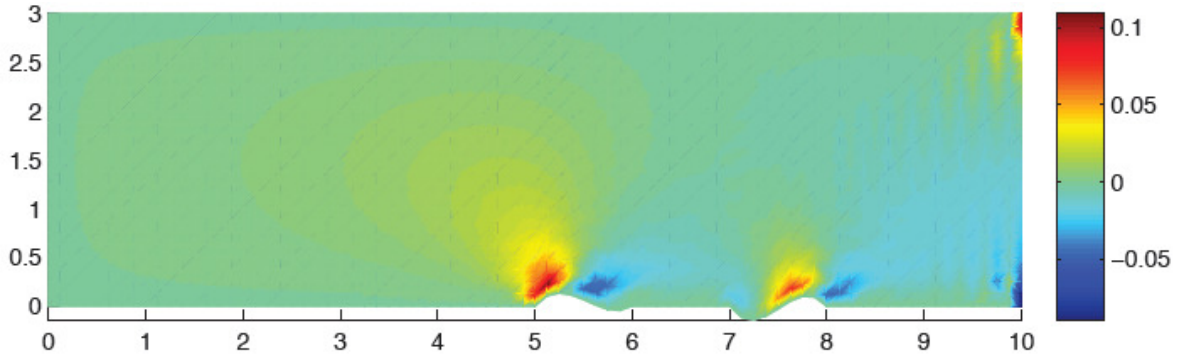


Figure 2: Navier-Stokes flow, uncertainty in geometry.

Motivated by the good performance of the method on the prototype models described above (and many others), we reproduced tests **2a** and **2b** with the Navier-Stokes flow implemented by Nek5000, with the reduced-model solver developed by Merzari and appropriate modifications (i.e., insertion of the uncertainty structure into the code) developed with the help of Yulia Peet. This time, the eigenvalue problem and the resulting projection and the evaluation of the output of interest were essentially a black box to us: all we could do with confidence was perform statistical analysis on the reduced-model and the full-model outputs. In this case, evaluation of the full code took approximately 40 minutes on a high-performance PC. Evaluation of the reduced model took 5–15 seconds.

3.2 Numerical Results

In our preliminary work on combined use of the reduced- and full-model data in uncertainty quantification, we examined **Model 1** with uncertainty dimension 12. When training the Gaussian-processes covariance model on the difference between full and reduced models, we used a training set of size 15. We compared a sample of 1,000 surrogate model outputs with full-model evaluations. A selection of metrics for full-model outputs and the corresponding predicted outputs is reported in Table 1. (Note: here, chosen output of interest is fairly arbitrary; we refer to the model description.)

Table 1: Model 1: prediction on reduced model data

Metric	Reduced Model + GP Prediction	Validation Set Measurement
Mean	9998	10065
Range	8562 - 11259	7900 - 11259
St. deviation	1012	1081
Validation data within predicted 90% confidence interval	99.0%	N/A

We calculated the 90% confidence interval, as shown in Figure 2, for a randomly selected sample of 85 points in the uncertainty space. The first 15 points shown are outputs of the surrogate model on its own training set. In this experiment, the confidence interval is fairly tight: only one full model output lies outside of it, while many points are placed closely to upper and lower bounds.

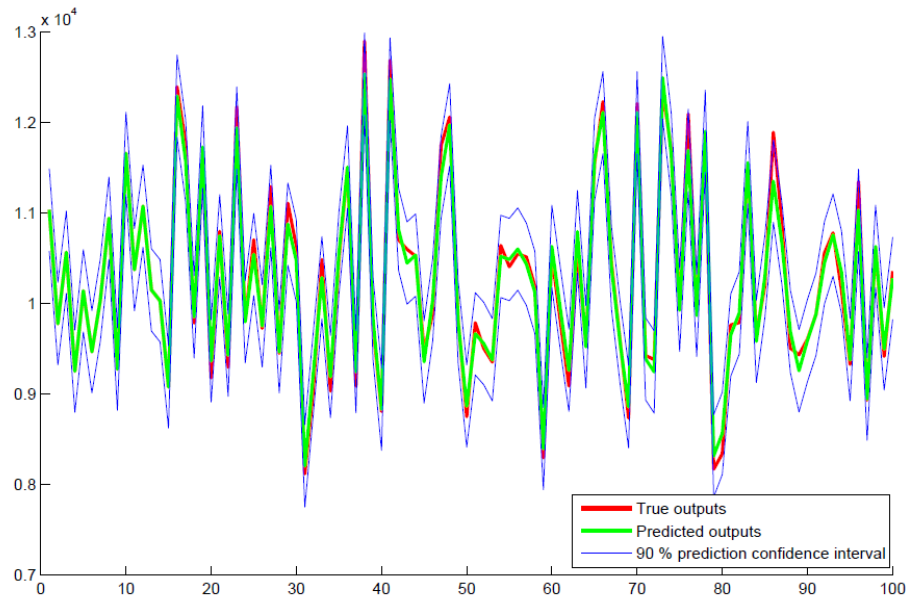


Figure 2: Model 1: prediction on reduced model outputs; 90% confidence interval.

For **Model 2b**, we used 10 reduced model outputs and 2 full model outputs and a scheme that involved *comparison against an intermediate regression model*. Figure 3 shows the 90% confidence interval for a randomly selected sample of 30 points. The first 10 points correspond to a training set assembled, hence the low quality of prediction and incorrect confidence interval. The quality of the prediction becomes noticeably better outside of the training region.

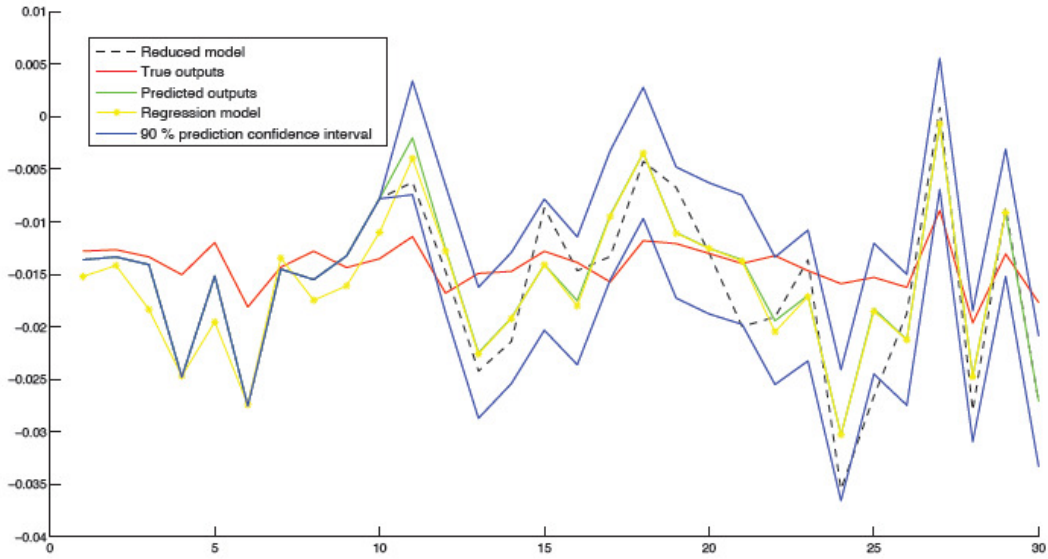


Figure 3: Model 2b: prediction on reduced model outputs; uncertainty dimension 4.

We now present a number of results for Nek5000 with setup **2b**. In Table 2, we provide some of the prediction metrics: notice that the results obtained using >4 model evaluations (comparison of full model with reduced model) are almost equivalent to the results obtained using <4 model evaluations (comparison of full and reduced model with an intermediate regression model). The difference in performance is somewhat better shown in Figures 4 and 5, where we show the pointwise predictions for randomly selected 80 points in the uncertainty space (training data not shown).

Table 2: Nek5000: prediction on reduced model data

Metric	GP Prediction Using Training Set of 5 Points	GP Prediction Using Training Set of 3 Points	Validation Set Measurement
Mean	0.5426	0.5345	0.5365
Range	0.4774 – 0.6733	0.4774 – 0.6422	0.4721 – 0.8250
St. deviation	0.0481	0.0543	0.0613
Validation data within predicted 90% confidence interval	98.5%	95.5%	N/A

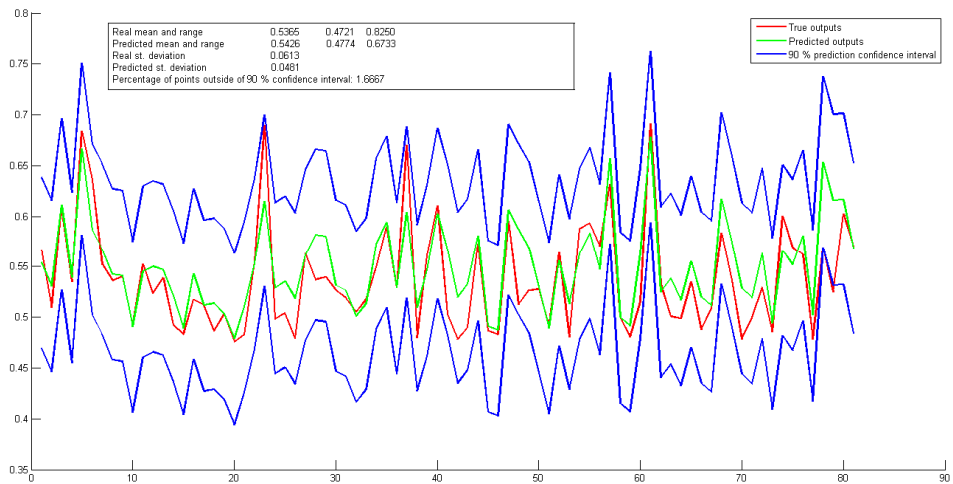


Figure 4: Nek5000: prediction and confidence interval, training set of 5 points.

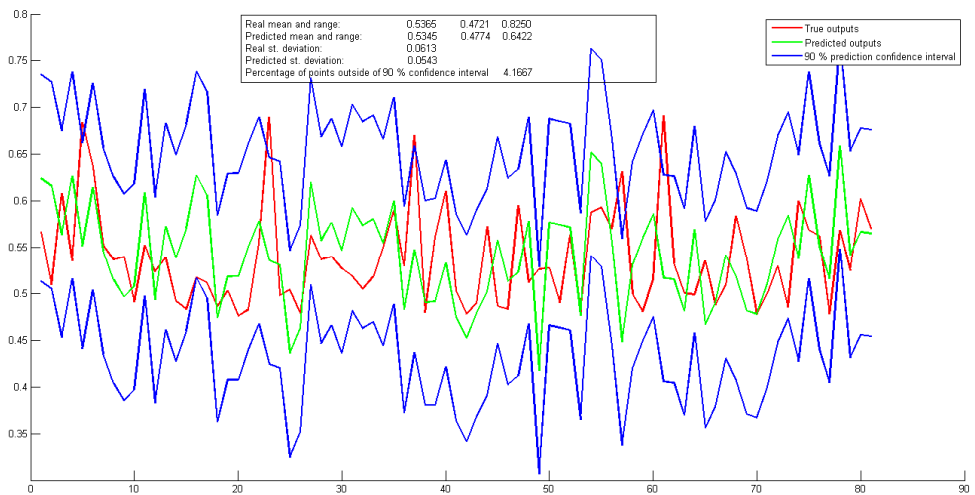


Figure 5: Nek5000: prediction and confidence interval, training set of 3 points.

In Figure 6, we show the “95-95” estimate for Nek5000 output, that is, 95% double-sided confidence interval for the 95th quantile. Again, note that this estimate was obtained using 5 evaluations of the full model; normally, this task requires sampling in the hundreds.

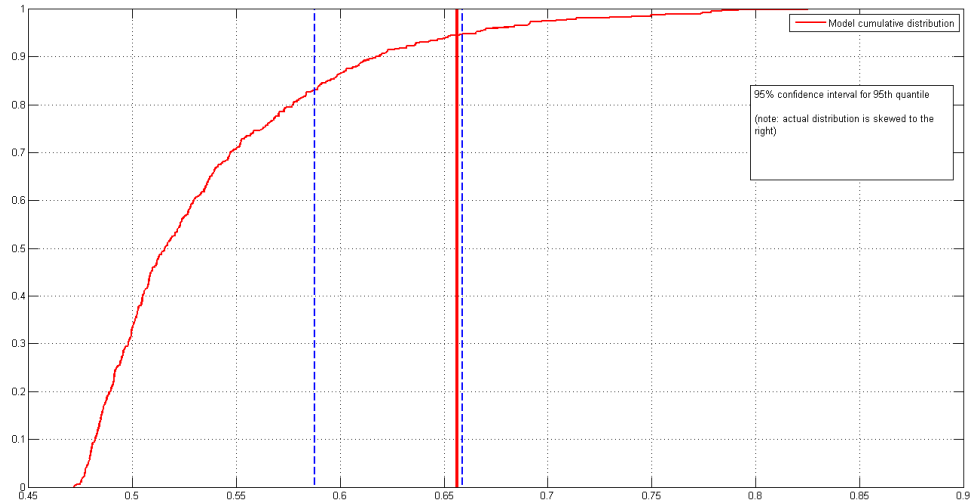


Figure 6: Nek5000: 95-95 estimate.

3.3 Discussion

Our numerical experiments show that the introduced approach to uncertainty quantification based on lower-fidelity data can be effective even for high-resolution codes of industrial complexity such as Nek5000. While it would be helpful to also test the method on a wide range of other codes, this demonstration addressed one of the most challenging themes in uncertainty analysis for nuclear engineering applications: response of a high-fidelity thermohydraulics model. This is a significant result, with implications for previously inaccessible tasks of code verification and validation. With additional development, our work may become a first step to removing the constraints on uncertainty analysis related to prohibitive computational cost of sampling over high-fidelity spaces.

Based on our experiments with prototype codes and Nek5000, we give the following (informal) characterization of the method performance. For a class of models considered, one can construct an effective representation of the response to uncertainty at a cost comparable to that of a linear interpolation. That is, the number of full-model runs required to fit the response can be equal to the dimension of the uncertainty space. (In comparison with the usual costs of running a high-performance model, the other tasks, such as multiple evaluations of the reduced model and nonlinear optimization required for Gaussian processes fitting of covariance, are negligibly cheap). This result is significantly different from traditional sampling-based methods, where the required number of samples can grow exponentially with dimension.

The pointwise quality of prediction is not exceptional: we observe errors of 10–50% in many places. The more important feature of the method, however, is the ability to provide confidence assessment of data. Our confidence intervals are observed to be essentially correct, unless wrong assumptions on data were forced on the model; see comments for Figure 3). We can even output a correct confidence interval for order statistics (Figure 6), a task that would normally require large-scale sampling of the outputs.

An interesting result is that the quality of the prediction does not grow significantly when using larger training sets. It appears that incremental increase of computational budget without the knowledge of the uncertainty response has a good chance of not benefiting the uncertainty analysis (because of emphasis on the “wrong” regions in the uncertainty space, because of detrimental effect on the nonlinear optimization search, etc.). If available, a *slightly* larger computational budget should be used to construct a better reduced model (again, with the important remark that in general one cannot know a priori the consequences of adding a point to the training data for reduction).

Our suggested scheme for lowering the cost of prediction even further through the use of an intermediate regression function is fairly complex and will need additional investigation. It is nevertheless a remarkable demonstration of advanced uncertainty analysis performed with a sample size *smaller than the dimension of space*.

The method has a prerequisite implementation of POD-based dimensionality reduction on the model equations. However, we can see from some of the numerical experiments (Figure 3) that the reduced model can be of fairly low quality. This means that partial access, namely, access to a simplified form of model equations, may be sufficient. Other forms of lower-quality data also may be used, as long as their computational cost per point is significantly lower and the data is of sufficient quality to fit the covariance model.

Overall, uncertainty analysis on extremely small training sets augmented with large, lower-quality calibrated data sets is an ambitious new direction of development. Even now it is capable of producing model assessments and diagnostics previously not available because of the prohibitive computational cost. With additional research, we intend to more fully characterize the class of models for which the suggested approach is effective.

4 Summary

In our work, we have found that the model’s response to a moderate number of uncertain parameters can be effectively predicted by using a novel approach based on sampling and calibrating lower-fidelity data, with only a few high-fidelity data points used in the calibration process. We have demonstrated the effectiveness of the approach on a number of simplified prototype models, as well as on a channel flow simulation performed using the high-resolution code Nek5000.

Our suggested approach is computationally inexpensive and estimates the variability of the uncertainty induced output well. This type of result was not previously available except at a significantly higher cost (and in effect not available at all for the more expensive simulations).

In the long term, we plan to implement our approach as a part of an automatic analysis suite of tools within NEAMS. Achieving this goal will result in a transformation of the status of uncertainty analysis in the field of nuclear engineering and will provide an overall stronger position of NEAMS in both development and analysis of advanced, high-fidelity nuclear engineering codes. The ultimate result will be improved understanding of safety and operational margins of nuclear reactors and a streamlined licensing process.

Acknowledgments

We thank Elia Merzari, Yulia Peet, Zhu Wang, and Xu Zhang for their contributions to POD-based reduction of models of fluid mechanics and their assistance with our experiments on Nek5000.

References

1. P. Fischer et al., “Nek5000 Overview: 2010 Users Meeting,” Argonne, 2010.
2. D. Downing, R. Gardner, and F. Hoffman, “An Examination of Response-Surface Methodologies for Uncertainty Analysis in Assessment Models,” *Technometrics*, 163(11), pp. 27–151, 1985.
3. O. Roderick, M. Anitescu, and P. Fischer, “Polynomial Regression Approaches Using Derivative Information for Uncertainty Quantification”, *Nuclear Science and Engineering*, 164(2), pp. 122–139, 2010.
4. B. Lockwood and M. Anitescu, “Gradient-Enhanced Universal Kriging for Uncertainty Propagation,” *Nuclear Science and Engineering*, 170(2), pp. 168–195, 2011.
5. M. Kennedy and A. O’Hagan, “Bayesian Calibration of Computer Models,” *Journal of the Royal Statistical Society*, B 63(3), pp. 425–464, 2001.
6. E. Merzari, D. Pointer, and P. Fischer, “A POD-Based Solver for the Advection-Diffusion Equation,” in *Proceedings of ASME-JSME,KSME Joint Fluids Engineering Conference*, Hamamatsu, Shizouka, Japan, 2011.
7. N. Draper and H. Smith, eds., *Applied Regression Analysis*, Wiley InterScience, pp. 307–312, 1998.
8. G. Webber, R. Handler, and L. Sirovich, “The Karkunen-Loève Decomposition of Minimal Channel Flow,” *Phys. Fluids*, 1054(9), 1997.
9. E. Sachs and S. Volkwein, “POD-Galerkin Approximations in PDE-constrained Optimization,” *GAMM-Mitteilungen*, pp. 194–208, 2010.
10. J. Burkardt, M. Gunzburger, and H.C. Lee, “POD and CVT-Based Reduced Order Modeling of Navier-Stokes Flows,” *Comp. Methods in Applied Mechanics and Engineering*, 196(1-3), pp. 337–355, 2006.
11. C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.
12. M. Arnst, R. Ghanem, E. Phipps, and J. Red-Horse, “Dimension Reduction in Stochastic Modeling of Coupled Problems,” *Int. J. Numerical Methods in Engineering*, 2012.
13. L. Gratiet, “Bayesian Analysis of Hierarchical Multi-Fidelity Codes,” *Methodes d’Analyse stochastique des Codes et Traitements numeriques*, to be published in 2012.
14. W. H. Schilders, H. A. van der Vorst, and J. Rommes, eds., *Model Order Reduction: Theory, Research Aspects and Applications*, Springer, Berlin, 2008.
15. D. Daescu and I. Navon, “A Dual-Weighted Approach to Order Reduction in 4D-var Data Assimilation,” *Monthly Weather Review*, 136, pp. 1026–1041, 2008.

16. N. Cressie, *Statistics for Spatial Data*, Wiley, New York, 1993.
17. F. Stein, F. Van der Meer, and B. Gorte, *Spatial Statistics for Remote Sensing*, Springer, New York, 2002.
18. A. DasGupta, *Asymptotic Theory of Statistics and Probability*, Springer, New York, 2008.



Mathematics and Computer Science Division

Argonne National Laboratory
9700 South Cass Avenue, Bldg. 240
Argonne, IL 60439

www.anl.gov



Argonne National Laboratory is a U.S. Department of Energy
laboratory managed by UChicago Argonne, LLC