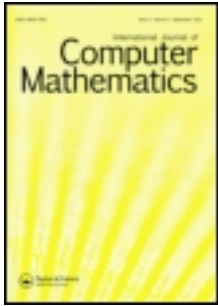


This article was downloaded by: [Yulia Peet]

On: 25 January 2014, At: 16:13

Publisher: Taylor & Francis

Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



## International Journal of Computer Mathematics

Publication details, including instructions for authors and subscription information:

<http://www.tandfonline.com/loi/gcom20>

### Proper orthogonal decompositions in multifidelity uncertainty quantification of complex simulation models

Oleg Roderick<sup>a</sup>, Mihai Anitescu<sup>a</sup> & Yulia Peet<sup>b</sup>

<sup>a</sup> MCS, Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439, USA

<sup>b</sup> School for Engineering of Matter, Transport and Energy, Arizona State University, 501 E. Tyler Mall, Tempe, AZ 85287, USA

Accepted author version posted online: 14 Oct 2013. Published online: 20 Jan 2014.

To cite this article: Oleg Roderick, Mihai Anitescu & Yulia Peet , International Journal of Computer Mathematics (2014): Proper orthogonal decompositions in multifidelity uncertainty quantification of complex simulation models, International Journal of Computer Mathematics, DOI: [10.1080/00207160.2013.844431](https://doi.org/10.1080/00207160.2013.844431)

To link to this article: <http://dx.doi.org/10.1080/00207160.2013.844431>

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis makes every effort to ensure the accuracy of all the information (the "Content") contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims, proceedings, demands, costs, expenses, damages, and other liabilities whatsoever or howsoever caused arising directly or indirectly in connection with, in relation to or arising out of the use of the Content.

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden. Terms &

Conditions of access and use can be found at <http://www.tandfonline.com/page/terms-and-conditions>

# Proper orthogonal decompositions in multifidelity uncertainty quantification of complex simulation models

Oleg Roderick<sup>a\*</sup>, Mihai Anitescu<sup>a</sup> and Yulia Peet<sup>b</sup>

<sup>a</sup>MCS, Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439, USA; <sup>b</sup>School for Engineering of Matter, Transport and Energy, Arizona State University, 501 E. Tyler Mall, Tempe, AZ 85287, USA

(Received 31 January 2013; revised version received 13 August 2013; accepted 29 August 2013)

We investigate uncertainty propagation in the context of high-end complex simulation codes, whose runtime on one configuration is on the order of the total limit of computational resources. To this end, we study the use of lower-fidelity data generated by proper orthogonal decomposition-based model reduction. A Gaussian process approach is used to model the difference between the higher-fidelity and the lower-fidelity data. The approach circumvents the extensive sampling of model outputs – impossible in our context – by substituting abundant, lower-fidelity data in place of high-fidelity data. This enables uncertainty analysis while accounting for the reduction in information caused by the model reduction. We test the approach on Navier–Stokes flow models: first on a simplified code and then using the scalable high-fidelity fluid mechanics solver Nek5000. We demonstrate that the approach can give reasonably accurate while conservative error estimates of important statistics including high quantiles of the drag coefficient.

**Keywords:** uncertainty quantification; model reduction; proper orthogonal decomposition; Gaussian processes; kriging

2010 AMS Subject Classifications: 60G15; 62P30; 34C20

## 1. Introduction

Major tasks of modern science and technology require the use of numerical simulations of physical systems to augment the rare and expensive experimental data. The current state of development of hardware and computing techniques enables models of high geometric resolution and physical fidelity, with a large number of parameters taken into account by simulation. The analysis tasks for such models require increasingly large amounts of computational and development resources. In turn, this requirement makes black-box-type, sampling-based uncertainty propagation a computationally demanding task.

Moreover, the situation in a *discovery* mode or for simulation of *prototypes* is even more restrictive. In such cases, past data has less relevance while the aim is to push the limit of resolution. In turn, this will typically result in computational requirements for one analysis being at the limit of availability, leaving an even smaller percentage of computational time available for uncertainty

\*Corresponding author. Email: [roderick@mcs.anl.gov](mailto:roderick@mcs.anl.gov)

analysis. As a result, in many fields, such as nuclear engineering, fluid dynamics, or climate modelling, we may be approaching the situation in which assessment of models of meaningful complexity using traditional sampling-based methods is no longer feasible.

Our main focus is propagation of uncertainties (resulting from measurement errors, simplifications, and discretization in the model setup and inherently unmeasurable quantities) from the inputs and parameters of the model to an output of interest. The existing work on the subject mainly follows the same scheme, called, variously, surrogate modelling, response surface learning, or multivariate regression.

An uncertainty-influenced output is fitted to an a priori chosen algebraic structure on the inputs, based on extensive sampling, or exploration of the uncertainty space by multiple evaluations of the code. Considerable development has gone into advanced sampling methods, while relatively little is suggested for the situation when sampling is very limited. And yet, if a simulation code takes *hours to days* to complete on a high-performance machine, as is the case for some of the domain areas and complex simulation contexts described above, the number of times a simulation can reasonably be repeated for the same geometric and physical setup will be easily exceeded by the size of sampling required to explore the uncertainty space.

### 1.1 *Using lower fidelity models in uncertainty assessments*

Given these restrictions, we posit that perhaps the only possible way to carry out such an assessment is by using a lower-fidelity model that takes less time to run.

We note the distinction between a regression-based model and a lower-fidelity model obtained by simplification of the full model's structure. A regression-based model (a 'data-driven' emulator) is a result of fitting a pre-determined algebraic structure to a set of known full model outputs. A simplified model is a result of accessing the full model's internal structure, and revising it to reduce complexity and computation time, while preserving the essential properties. The computational cost of evaluating lower-fidelity models is expected to be very small, although higher than that of a regression-based model.

Opportunities for simplification appear in multiple guises. For a start, simulation models do not exist alone, as sole examples in their class of codes. In the process of development, simplified versions of the code will exist 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 such simpler versions are no longer accessible, it is possible to simplify the full model's structure manually or in a semi-automatic manner. In either case, model simplifications and approximations are used as a rough tool to explore the model, either as direct replacements for the source of the more expensive data, or as a source of first guess at the model's properties. Empirically justified 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.

Kennedy and O'Hagan [16,17] suggested that a statistical model should be constructed to describe the imperfection in the cheaper data. Using this statistical model together with cheap outputs of lower-fidelity models, one can estimate the uncertainty effects at much lower cost. The premise of the work presented here is somewhat different. While Kennedy and O'Hagan discuss the use of error-prone computer simulations together with perfect physical observational data, we will be combining outputs of effectively the same computer code at multiple levels of fidelity. We aim to demonstrate that the approach is effective even for very complex simulation codes. While practical demonstration requires overcoming technical difficulties that are not encountered in study of simpler examples, the main mathematical approach is sound, and can be adjusted to new situations – in particular, to a very limited computational budget.

## 1.2 Our work

In our high-performance context, however, we may be in the regime where such low-fidelity models are scarce in a discovery context. One thus runs into the interesting question of whether such a model can be created directly from the high-fidelity code, in some holistic manner. To this end, and as we are interested in dynamical systems, we investigate whether proper orthogonal decomposition (POD)-based model reduction [14] is a suitable choice and we shall use it in our work.

The approach is known, variously, as the principal component analysis (PCA) method, POD method, and method of snapshots. It has been restated for applied tasks of simulation and optimization in high dimension by several authors [27,28]. Additional attention was 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 [6,24]. In engineering fields, multiple researchers use POD-reduced models in a straightforward manner to replace the full model and cheaply propagate model dynamics forward in time. The key conceptual appeal, which we will not explore here in detail, is that our approach can in principle approximate the dynamical system arbitrarily well.

Multiple research efforts have encountered the situation where, due to problem size, complexity, and dimension of the parameter space, large-scale sampling of the simulation model is not feasible. In response, a lot of recent work is done on the use of lower-fidelity models, in particular, reduced models, for uncertainty propagation [3,11–13,20]. There are many advantages to using a reduced model: it is often easier to manipulate to introduce uncertainty, faster to evaluate, the reduction of state dimension leads to a natural way to reduction in the dimension of the parameter space. However, most if not all of the existing work is set in the situation where full model data is abundant, and a very good reduced representation can be created. We are operating in conditions where there may not be enough data to create a POD model that is valid anywhere except a very small region in the parameter space. We cannot trust the model outputs to be effectively equivalent to full model outputs outside of that region, nor can we cheaply estimate its boundaries, therefore we are using a procedure that accounts for the error introduced by reduction.

In principle, the error introduced by reduction can be estimated using several runs of the POD-reduced model followed by estimates of the residual on the full model (but not fully converged high-fidelity simulations), based on the Kenney–Laub approach [18]. Kenney and Laub introduced the small-sample condition estimate method for matrix and vector operations, based on using random vector products  $uv^T$  as predictor for the vector norm  $\|u\|_2$ . Homescu and Serban [15] used this approach to develop error analysis for perturbed, POD-reduced ordinary differential equations (ODEs) and partial differential equations (PDEs); the technique requires a small, dimension-independent number of adjoint evaluations of the model. The (a posteriori) error estimation is, arguably, the most effective available tool for assessment and control of POD-reduced models performance.

Such estimates can in principle be exploited in uncertainty analysis and would quite possibly result in an additional way to reduce the number of expensive full-model runs. However, as we aim primarily to produce a proof of concept for a code of engineering relevance and as the development efforts are not trivial, we focus our efforts on the simpler approach of using a limited number of calibration experiments to estimate the POD model error.

Our choice of POD is based on two facts. First, POD has proved to be an excellent model reduction technique for many dynamical systems. Second, among intrusive techniques, it seems to be one of the most practical techniques to implement. We have had some conceptual success with derivative-enhanced surrogates that, in principle, can be used to similar ends [21,25]. However, implementation of gradient and adjoint methods, while greatly assisted by automatic differentiation [1], requires considerable effort and expert knowledge. The situation is similar with intrusive Galerkin methods.

Given the fact that POD methods are composed of fairly standard forward steps (one model run plus PCA followed by an evolution of the projected equation) we believe that it is the simpler of the approaches mentioned and thus is more likely to achieve engineering acceptance. The question to be answered, however, is whether it is still suitable for uncertainty quantification, that is, whether, it also captures parametric dependency well.

Given the novelty of this question and the complexity of the problem tackled, we will not aim to characterize this situation theoretically or comprehensively. Rather, we will try to identify a model problem where this can be answered positively. On the other hand, to enter the area of engineering relevance and address issues of complexity of implementation of this intrusive approach, we will do so for a non-trivial problem class and code.

As benchmark problems to validate our approach, we will concern ourselves with Navier–Stokes channel flow models in the presence of geometric uncertainty. Moreover, we will carry out the analysis using a high-end fluid dynamics code, Nek5000 [10], a highly complex scalable computational fluid dynamics solver that cannot be sampled extensively. A reduced-order solver that uses Nek5000 intermediate model states (snapshots) has been developed recently for extrapolation over time and recovery of the full-dimensional velocity field [22]. We used it, with some modifications, as our low-fidelity model.

The idea of using lower-fidelity data, with additional statistical models characterizing the imperfection, properly belongs to Kennedy and O’Hagan [16,17]. Their characterization methods are based on standard Bayesian reasoning and Gaussian processes (GPs) for machine learning. We followed their essential framework, with implementation details for GPs as recommended by Rasmussen and Williams [23]. Combined with our recent insights from GPs analysis [21], this approach creates an uncertainty analysis tool, as described in Section 2.

The paper is organized as follows. In Section 2 we provide technical details for the components of the approach: POD-based dimensionality reduction, Gaussian processes-based calibration, and their combination used to recover model statistics from calibrated lower-quality data. In Section 3 we describe the Navier–Stokes models and the methodology we used to test the performance of the method. In Section 4 we present the results of numerical experiments. We discuss the results in Section 5. In Section 6 we review the performed work and suggest directions for future development.

## 2. Method description

We now provide technical details about the main components of the proposed approach. We then summarize the algorithm used for generation and calibration of lower-fidelity data.

### 2.1 POD-based model reduction

Given a static data set of  $n'$  observations, represented as a matrix  $A \in \mathbb{R}^{n \times n'}$ , an optimal approximation  $\hat{A}$  of fixed rank  $k < n$  can be expressed by using singular value decomposition. If  $A = \sum_{i=1}^n \mathbf{u}_i \sigma_i \mathbf{v}_i^T$ , with singular values  $\sigma_i$  in descending order, then  $\hat{A} = \sum_{i=1}^k \mathbf{u}_i \sigma_i \mathbf{v}_i^T$ ; 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 eigenvalue energy error  $\sum_{i=k+1}^n \sigma_i / \sum_{i=1}^n \sigma_i$  is close to 1.

According to the method of snapshots [27], a similar technique can be used to reduce the dimensionality of dynamically evolving data; the projection is applied to the underlying model equations. Suppose that a number of observations (or snapshots) from a single 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  $C = A \cdot A^T$ . In practice, instead of a singular value decomposition

of  $A$  we solve an eigenvalue problem

$$C\phi = \lambda\phi. \quad (1)$$

Then the  $k$  dominant eigenvectors are recorded as  $\Phi = (\phi_1, \phi_2, \dots, \phi_k)$ . Often,  $n' \ll n$ , and it is more convenient to solve an equivalent problem  $A^T \cdot A\phi' = \lambda'\phi'$ ; then  $\lambda_i = \lambda'_i$  and  $\phi_i = (1/\sqrt{\lambda'_i})U\phi'_i$ .

The matrix  $\Phi$  is used to project the model dynamics into a dominant eigenspace of the correlation matrix. This 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 it is expected that the POD approximation of the model state trajectory  $u(t)$

$$\hat{u}(t) = \sum_{i=1}^k \phi_i(t)q_i(t) \quad (2)$$

is effective,  $\hat{u}(t) \approx u(t)$ . Here,  $q(t)$  is the approximating trajectory in low-rank subspace coordinates (a shift term  $\mu$ , corresponding to the average of snapshot values, is sometimes added here; we set  $\mu = 0$  to simplify the explanation). POD-based projection can be used for many formats of the problem.

For practical purposes it is sufficient to describe a model discretized to an ODE. Consider the system

$$\begin{aligned} \frac{du}{dt} &= f(u, t, P, \mathbf{x}), \\ u(t_0) &= u_0(P, x), \end{aligned} \quad (3)$$

where  $u(t)$  is the model state at time  $t$ ,  $P$  are the modelling parameters, and  $\mathbf{x}$  are generic quantifiers of the uncertainty. We accept any forms in which uncertainty can be introduced into the model. The independent quantities  $\mathbf{x}$  can be dimensionless (for example, multiplicative errors in measurement of parameters), have physical meaning (additive errors), represent programmers' decisions in discretization of the model, or be a mixed set of all the above. We also allow hierarchical relationships that include uncertainty, for example, in the form  $P := P(u, t, \mathbf{x})$  [25].

We collect the snapshots  $A = [u(t_1), u(t_2), \dots, u(t_{n'})]$  and solve (1). Once  $\Phi$  is defined, the reduced-order approximation to the model solution, written as  $\hat{u}(t) = \Phi q$ , is determined by solving

$$\begin{aligned} \frac{dq}{dt} &= \Phi^T f(\Phi q, t, P, x), \\ q(t_0) &= \Phi^T u_0(P, \mathbf{x}). \end{aligned} \quad (4)$$

The main advantage to using the reduction is the reduced integration time in comparison with that of the full model equations, especially for well-reducible problems with  $k \ll n$ . If needed, adjoint differentiation of the reduced model is also computationally cheaper.

Note that  $q$  and  $u$  are functions of both time and parameters,  $q \equiv q(t, \mathbf{x})$  and  $u \equiv u(t, \mathbf{x})$ , although we may choose to expose the one on  $\mathbf{x}$  only implicitly at times. In this framework, we can state one of the main aims of our study: *determine in our case whether the random variable  $u(t, x)$  induced by the probability density on  $\mathbf{x}$  defined by the uncertainty analysis can be well approximated by the random variable  $\Phi q(t, \mathbf{x})$* . This will allow for statistics of  $u$  to be computed based on the much cheaper statistics of  $q$ .

Representation of the reduced model as (4) may be misleading: as written, it appears that the right-side function  $f$  is evaluated in the full-dimensional space  $\mathbb{R}^n$ . This could be ineffective.

In practice, the expression  $\Phi^T f(\Phi q, t, P, x)$  is explicitly known and can be simplified so that all computations are performed in  $\mathbb{R}^k$ .

Consider the Navier–Stokes PDE

$$\begin{aligned} \frac{\partial u}{\partial t} + (u \cdot \nabla)u - \frac{1}{Re} \Delta u + \nabla p &= 0 \\ \nabla \cdot u &= 0. \end{aligned} \quad (5)$$

The reduced-model solution is an expansion  $\hat{u}(x, t) = u_0(x) + \sum_{i=1}^k q_i(t) \phi_i(x)$  with an additional inner-product assumption  $(\nabla p, \phi_i) = 0$ . The POD-Galerkin representation of the reduced model is written as

$$\left( \frac{\partial u_i}{\partial t}, \phi_i \right) + ((u_i \cdot \nabla)u_i, \phi_i) + \left( \frac{2}{Re} D(u_i), \nabla \phi_i \right) = 0, \quad i = 1, 2, \dots, k, \quad (6)$$

and integrated in the form of the reduced-order ODE

$$\frac{dq_k}{dt} = A_{ijk} q_i q_j + B_{ik} q_i + C_k. \quad (7)$$

The resulting bilinear nature of (7) is an immediate consequence of the bilinearity of the Navier–Stokes equations. The arrays  $A, B, C$  are assembled from terms of the Navier–Stokes operator multiplied by  $\phi_i \phi_i \cdot (-(u \cdot \nabla)u + (1/Re)\Delta u)$  with substitution  $u = \Phi q$ . The results of the substitution can be bulky, for example  $C_i = (du_0/dx)u \cdot \phi_{ii} + (du_0/dy)v \cdot \phi_{ii} + (dv_0/dx)u \cdot \phi_{ii} + (dv_0/dy)v \cdot \phi_{ii} + \nabla u_0 \cdot \phi_{ii} + \nabla v_0 \cdot \phi_{ii}$  (where  $u, v$  are the components of velocity in axial directions  $x, y$ ,  $\phi_u, \phi_v$  are corresponding subvectors of a complete eigenvector  $\phi = (\phi_u \phi_v)^T$ ). Nevertheless, the derivation of terms is a straightforward exercise, performed in the same way for any right-side function  $f$ .

The parameters  $\mathbf{x}$  are present in the arrays  $A, B, C$ , and in many circumstances this dependence can be recovered analytically; but a complete practical framework is difficult to derive and present. We thus decided not to represent it explicitly or exploit its structure at this time, although that is definitely a most intriguing endeavour in the long run. Nevertheless, all changes that can be made to the parameters of the full model (to the initial state of the model, to intermediate parameters on the right side of the ODE, to the geometric grid on which the original PDE is approximated by an ODE) will also influence the obtained reduced model in some form, and that is sufficient to allow us to investigate the aim stated above.

If the model is not reducible (i.e. its essential dynamics cannot be described in just a few evolving variables), the method suggested here is not applicable. The situation is more ambiguous when the reduced model can be defined but does not approximate the full model well. This is usually attributed to an incorrect choice of snapshots or an ineffective POD basis  $\Phi$ . The most straightforward way to improve the reduced model is to enhance the POD basis  $\Phi$  by combining snapshots from trajectories started from several different points in the uncertainty space. Here, the effective improvement can be limited: a procedure for an improved POD basis will run into problems of high computational cost and generic circular logic (we cannot emphasize the response to uncertainty in the reduced model without knowledge about the full model's response to uncertainty). Other methods of improvement of the reduced model (such as snapshot weighting) run into similar limitations: too many diagnostics and revisions are expensive, and no diagnostics at all leaves us with no information on what features to emphasize in the reduced model dynamics. For most of our work, we accept the reduced-model data as is and expect that the reduced model is of sufficient quality;  $u \approx \hat{u}$ , correspondingly,  $\mathcal{J} \approx \mathcal{J}(\hat{u})$ . We have an opportunity to test the discrepancy of this model on the target functional a limited number of times, and we use this in



the GPs approach in Section 2.2. If the fit is bad, we expect that the posterior confidence interval will be very large, indicating a lack of quality.

We pass the reduced-model data to the next stage of the process: automatic learning of the difference between full- and reduced-model outputs over the uncertainty space.

## 2.2 Gaussian-processes based machine learning

A stochastic process [8] is a commonly used tool for representing the dynamics of models with an indeterminant source of variability. The approach is naturally applied to quantification of uncertainty. In the language of stochastic processes, an output of interest  $\mathcal{J}$  follows a statistical distribution with certain mean and covariance functions  $\mu(\mathbf{x})$ ,  $\text{cov}(\mathbf{x}, \mathbf{x}')$ . The role of the mean function can be played (initially, until revision) by any regression model  $R(\mathbf{x})$ ; the variance structure needs to be additionally fitted to the available data. GPs are a special case of the stochastic processes, restricted to considering only normal multivariate distributions. This restriction, which us allows to describe the statistics derived from observations explicitly, is not very limiting for practical applications, as can be seen from its large use in expressing spatiotemporal and other multidimensional uncertainty [5,23,26].

At the core, GP machine learning is a response surface method. Unlike commonly used regression, it involves estimating the shape of the covariance function on the data rather than the shape of the data itself. Consider a basic task of constructing the GP representation on the training set of outputs  $Y_O$  corresponding to a set of inputs  $X_O = (\mathbf{x}_{O1}, \mathbf{x}_{O2}, \dots, \mathbf{x}_{ON}) : \mathbf{x}_{Oi} \in \mathbb{R}^n$ .

We denote the covariance function on the inputs by  $\text{cov}(\mathbf{x}, \mathbf{x}'; \theta) : \mathbb{R}^n \rightarrow \mathbb{R}$ . For the algebraic form of covariance, we use a *Matern 3/2 function*:

$$\text{cov}(\mathbf{x}, \mathbf{x}'; \theta) = \delta^2 \cdot \sum_{i=1}^m \left( 1 + \sqrt{3} \left| \frac{\mathbf{x}_i - \mathbf{x}'_i}{\theta_i} \right| \right) \exp \left( -\sqrt{3} \left| \frac{\mathbf{x}_i - \mathbf{x}'_i}{\theta_i} \right| \right). \quad (8)$$

A small number of other suitable covariance functions are available, the most commonly used being the *squared exponential*  $\text{cov}(\mathbf{x}, \mathbf{x}'; \theta) = \delta^2 \cdot \sum_{i=1}^m \exp((\mathbf{x}_i - \mathbf{x}'_i)/\theta_i)^2$ . We point out, however, that its use raises some issues about assumptions on the data, and the Matern class is a more conservative choice that requires less restrictive assumptions about the smoothness of the data process [26]. We refer to the work of Rasmussen and Williams [23] and also our previous work [21] for a discussion of the choice between covariance functions. Part of our choice originates in our experience in [21] that Matern  $\frac{3}{2}$  is safe and robust; extended studies of the proper classes are certainly necessary in future work.

The parameter  $\delta^2$  in the expression can be estimated by  $\text{var}(Y_O)$ . The *hyperparameters*  $\theta_i$  are essential in determining the shape of the covariance function. They are chosen to best fit the training data in the maximal marginal likelihood sense. It is convenient to write out the probability that  $Y_O$  follows Gaussian distribution with covariance  $\text{cov}(x, x'; \theta)$  in a negative logarithmic form. The task of fitting the hyperparameters amounts to maximizing the log-likelihood

$$-\log(\Pr(Y_O | X, \theta)) = -\frac{1}{2} Y_O^T K^{-1} Y_O - \frac{1}{2} \log |K| - \frac{n}{2} \log(2\pi), \quad (9)$$

where  $K = (\text{cov}(\mathbf{x}_i, \mathbf{x}_j, \theta))$ ,  $\mathbf{x}_i, \mathbf{x}_j \in X_O$  is the covariance matrix on the training data. The nonlinear optimization problem itself presents a computational challenge and is not guaranteed to yield an ideal solution, yet most of our tests and prior work [21] show that the resulting distribution approximates the uncertainty well. We use standard nonlinear optimization tools (Matlab routines *fminsearch* and *fminunc* are acceptable [19]). Calculating the log-likelihood is straightforward once the Cholesky factorization of the matrix  $K$  is obtained, which is a fairly simple calculation

unless the dimension  $n$  is very large. Given the scarce data nature of our problem, this is not an issue here. We point out, however, that we have addressed the issue of scalable algorithms for Gaussian processes elsewhere [2]. The numerical conditioning of the matrix inverse procedure can be used as a diagnostic for the solution quality; we restart the search with a different initial guess if  $K$  is poorly conditioned.

Once the covariance function is specified, we can perform pointwise predictions on any testing set  $X_V$  by Kriging. A cross-covariance matrix relates the testing set to the training set:

$$K_V = (\text{cov}(\mathbf{x}, \mathbf{x}'), \theta), \quad \mathbf{x} \in X_O, \mathbf{x}' \in X_V. \quad (10)$$

The outputs are predicted as

$$Y_V = K_V^T \cdot K^{-1} \cdot Y_O. \quad (11)$$

Given the initial (or *prior*) version of the mean function  $\mu(\mathbf{x})$ , a posterior version of the mean on the testing set is

$$\mu'(\mathbf{x}) = R(X_O) + K_V^T \cdot K^{-1} \cdot (Y_V(\mathbf{x}) - \mu(\mathbf{x})). \quad (12)$$

A posterior prediction for the variance is given by

$$V(x) = \text{var}(Y(\mathbf{x}_i)) = \delta^2 \cdot (\text{cov}(\mathbf{x}_i, \mathbf{x}_i; \theta) - K_V^T \cdot K^{-1} \cdot K_V). \quad (13)$$

The estimate (13) provides confidence information for pointwise predictions. The effectiveness of this estimate depends on how well the underlying assumptions for Kriging are satisfied by the data (or, the training data is of sufficient quality to estimate the variogram correctly [4]).

We now return to the task of uncertainty quantification. We have just formulated a tool that allows unbiased, locally smooth prediction on training data and 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. We can define the training data as

$$Y_O = \mathcal{J}(u) - \mathcal{J}(\hat{u}), \quad (14)$$

with  $u = u(t, P, \mathbf{x})$ ,  $\mathbf{x} \in X_O$ . and then output the calibrated data as  $\mathcal{J}_C(X_V) = \hat{\mathcal{J}}(X_V) + Y_V$  for the training set of inputs  $X_V$ , using Equation (11). Confidence intervals are written as

$$(\mathcal{J}_C(\mathbf{x}) - zV(\mathbf{x}), \mathcal{J}_C(\mathbf{x}) + zV(\mathbf{x})), \quad (15)$$

where  $z$  is the appropriate coefficient corresponding to the measure of confidence; for example,  $z = 1.645$  for 90% confidence using an empirical rule for the Gaussian distribution.

Note that the hyperparameters  $\theta$  can be interpreted as measures of anisotropy in the data or rates at which the data correlation decreases with distance in each direction; in principle, they can all be non-negligible and independent. For some outputs (in particular, for choices of  $\mathcal{J}(T)$  that are linear with respect to  $T$ ), POD reduction will guarantee a low degree of freedom on the complementary subspace of dimension  $n - k$ , as estimated from the non-dominant eigenvalue distribution. However, we must also allow for the outputs of interest that are not smooth with respect to the model state or, in fact, have an unknown effect because they are not explicitly extracted from the code. Thus, in general, the computational budget required for the prediction process of the training data (14) will require at least  $m$  training points. If such a budget is not available, we will have to use a different strategy in defining the training data and, correspondingly, a different expression for the calibrated data. One such alternative, using a weighted training set of full- and reduced-model outputs without pairwise correspondence between them, is mentioned in the following section.

We now review the method as it applies to the specific task of uncertainty quantification.

### 2.3 Summary of the method

In the most straightforward form, 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 would require additional model evaluations): POD-based model reduction, assembly of training data for calibration, and GP-based calibration. We consider the components step by step.

*Model reduction* requires collecting snapshots from the model trajectory. Thus the essential cost of model reduction is at least one full model evaluation. We choose a single point in the uncertainty space: in the absence of additional information, it may be done randomly, or with all uncertainty quantifiers given nominal values, e.g.  $\mathbf{x}_i = 0$ , evaluate the model, and record model state over moderately many time instances.

Singular value decomposition of the empirical correlation matrix  $C = AA^T$  needs to be performed only once; only the projection matrix  $\Phi$  needs to be stored.

To improve the performance of the reduced model in reproducing the output of interest  $\mathcal{J}(T)$ , we can use data-weighting techniques. Where possible, we compute or estimate from external information the magnitudes of sensitivities  $d\mathcal{J}(T)/dA$ . This array measures the importance of the individual model state components (compared along the rows) and of the individual time moments (compared across the columns). Schematically, the influence of particular state component  $T_j$  is amplified by modifying the eigenvalue problem (1) to the form

$$\Lambda C\phi = \Lambda\lambda\phi, \quad (16)$$

where  $\Lambda$  is the identity matrix with the  $j$ th component on the diagonal multiplied by weight  $w > 1$ . The influence of a particular snapshot  $T(t_k)$  is amplified by modifying (1) to the form

$$\frac{1}{w}AWA^T\phi = \lambda\phi, \quad (17)$$

where  $W$  is the identity matrix with the  $k$ th component on the diagonal multiplied by weight  $w > 1$ . The operations can be repeated and combined, resulting in a dual-weighted POD reduction procedure [6]. Note that the choice of weights may influence the numerical stability of the resulting reduced-order model.

Suppose that several full-model runs are available, with projection matrices  $\Phi_1, \Phi_2, \dots$ . They can be aggregated into the form  $\Phi = \text{span}(\Phi_1, \Phi_2, \dots)$ ; in practice that can be accomplished by a Gram-Schmidt orthogonalization procedure. If the computational budget allows estimation of some of the sensitivities  $d\mathcal{J}/dx$ , one can choose directions of interest in the uncertainty space and sample collections of snapshots along them. Increasing the reduced model dimension  $k$  may also improve the performance of the reduced model over the points in the uncertainty space distinct from the one where the snapshots were generated.

Overall, while in many cases a small improvement of quality of the reduced model is possible, we do not expect to be able to do it cheaply and with confidence. Instead, we accept that the reduced model may be adequate only at a point in the uncertainty space on which the set of snapshots was generated. Elsewhere, the outputs of the reduced model will be imperfect, but we cannot afford much analysis or revision of the reduction. Instead, the lower-fidelity outputs are passed, as is, to Bayesian correction procedure.

By design, the reduced model is cheap to evaluate at any point in the uncertainty space; we can have samples of almost arbitrary size. Consider the outputs  $Y_O$  on the training set  $X_O$  and corresponding testing outputs  $Y_V$  based on  $X_V$ . In practice, the organization of the full- and the reduced-model code may differ. For example, the reduced model may have a non-explicit initial state (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. It becomes a development task to modify the code or to interpolate. Notably, the proposed calibration procedure accounts for the differences between the full and the reduced models without distinguishing their source. The calibration procedure attempts to make up for interpolation errors as well; the overall quality of prediction may decrease.

In the scope of this work, we do not provide guidance on sampling the parameter space to obtain training or testing data. In practice, our computational budget is so limited that only very few high-quality model outputs will be considered, not sufficient to make conclusions about the difference between sampling strategies. On the other hand, the lower-fidelity model is so cheap it can be sampled with almost arbitrary density. Thus, we used uniform random sampling everywhere, with bounds of the region as wide as an applied example allowed.

*Use of sampled data to construct the training set* begins with interpretation of the ‘difference’ between full and reduced data. Besides the arithmetic difference between 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  $Y_O = \mathcal{J}(X) \leftrightarrow \hat{\mathcal{J}}(X)$  is that it must be reversible, that is, at least locally bijective and smooth.

For the computational budget of full-model runs approximately equal to the uncertainty space dimension  $m$ , we sample full and reduced models in pairs on a training set  $X_O$  and use  $Y_O = \mathcal{J}(u) - \mathcal{J}(\hat{u})$ . If the budget must be less than  $m$ , we can construct an intermediate regression model  $r(\mathbf{x})$  using the available full-model runs only: in practice,  $r(\mathbf{x})$  can be a very low-rank least-squares linear approximation for  $\mathcal{J}(u) = \mathcal{J}(u(t, P, \mathbf{x}))$ . Then we can compare full and reduced models with  $r(\mathbf{x})$  instead of with each other. Technically, we define  $\{Y_O = \mathcal{J}(u(t, P, \mathbf{x})) - r(\mathbf{x}); \mathbf{x} \in X_O$ . The Bayesian calibration procedure will then be a form of interpolation from the nearest known values of  $\mathcal{J}(u(t, P, \mathbf{x}))$  and  $r(\mathbf{x})$ . The quantities  $r(\mathbf{x})$  and  $V(\mathbf{x})$  are assumed to be statistically independent. The full-model outputs in this training set can be weighted, that is, repeated multiple times in the training data for increased influence on the resulting variance structure. This approach will have a lower quality in pointwise prediction, but it 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 training set  $Y_O$  should require only a few full model evaluations. On the other hand, the correspondence must be reversible; deciphering of the predicted values  $Y_V$  into calibrated  $\mathcal{J}_C$  should be performed without significant loss of quality.

*GP-based learning of variance structure* is treated mostly as a black-box procedure for producing the posterior mean  $\mu'(\mathbf{x})$  and variance  $V(\mathbf{x})$  based on training data  $Y_O$ . For improved performance, we recommend that the values in  $Y_O$  be of the same order of magnitude; the rest of the process depends on the choice of nonlinear optimization algorithm.

Note that our recovered values  $\mathcal{J}_C$  are technically random variables: we are using the posterior mean of the GPs, combined with output from the reduced model, as pointwise prediction, and the variance of the GPs as the measure of confidence in the pointwise prediction.

The scheme of the complete process is shown in Figure 1. We use a general term ‘correspondence’ to describe the many possible relationships between the full and reduced data that could be used in the process of calibration.

Once the calibration process is complete, we in effect have a cheap source of training data that comes with an explicit variance function; the computational effort for producing more data amounts to additional reduced model runs and Kriging. The statistics of the model under uncertainty can now be assessed by sampling calibrated outputs.

*Estimation of confidence intervals for order statistics* follows a straightforward approach by DasGupta [7]. Consider the  $p$ th quantile for the distribution of the output of interest  $\mathcal{J}$ , and choose a confidence level  $\alpha$  (a typical example is the 95/95 estimate:  $p = 0.95$ ,  $\alpha = 0.05$ ). A conditional quantile estimate approach amounts to uniformly sampling  $\mathcal{J}_C$  over the uncertainty

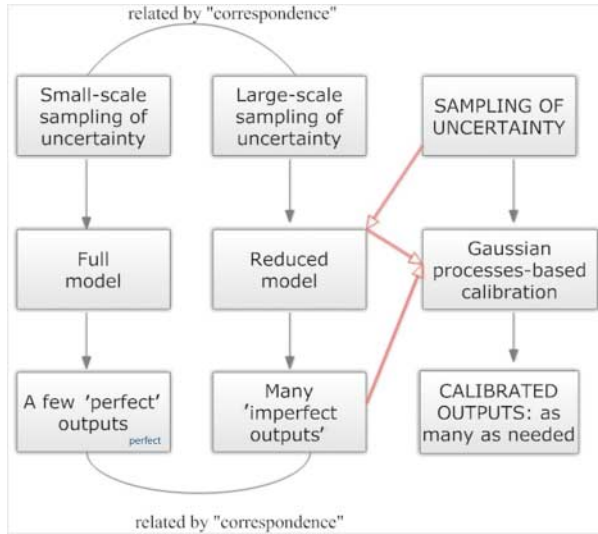


Figure 1. Reduced-order model sampling and calibration process.

space and recording the  $p$ th quantile for each sample, thus forming a new variable  $\zeta_p$ . The required double-side confidence interval is estimated from the empirical distribution of  $\zeta_p$ , that is, from the empirical distribution function defined on  $M$  points by  $Z(t) = (1/M) \sum_{i=1}^M 1\{\zeta_{pi} < t\}$ ; here  $1\{\dots\}$  is the binary indicator of the event.

### 3. Demonstration on Navier–Stokes channel flow models

In the development of our approach, we started with a simplified, prototype code for Navier–Stokes flow in a rectangular channel. With the code that we developed ourselves, it was easier to avoid the main implementation difficulty: inconsistency between input formats for the reduced and the full models. Moreover, the reduction and calibration approach we use is largely model-independent; it can be assessed on simpler models. We shall now explain how the prototype model was constructed, overview the technical tasks required to apply the approach to a high-fidelity code Nek5000, and summarize the methodology used in numerical experiments.

#### 3.1 Prototype model

For a prototype code, we used the previously described Galerkin-POD approach. The original PDE

$$\frac{\partial u}{\partial t} + (u \cdot \nabla)u - \frac{1}{\text{Re}} \Delta u + \nabla p = 0 \tag{18}$$

$$\nabla \cdot u = 0$$

is integrated to obtain a set of snapshots. Once the reduction basis  $\Phi$  is defined, the reduced-order ODE is written in the form

$$\frac{dq_k}{dt} = A_{ijk}q_iq_j + B_{ik}q_i + C_k. \tag{19}$$

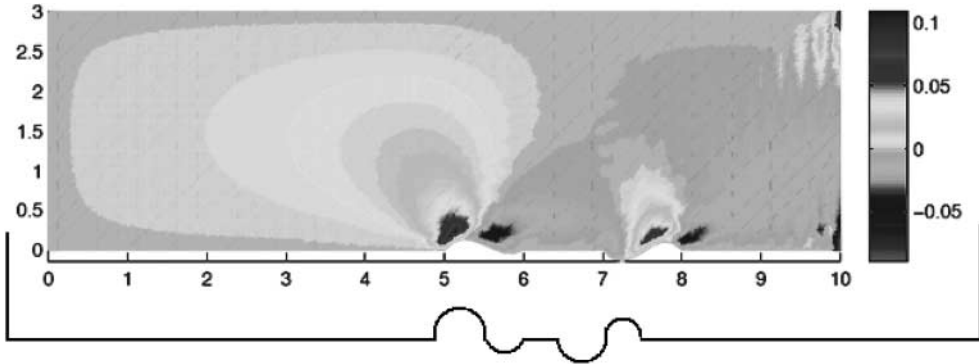


Figure 2. Velocity field for channel flow with wall bumps.

Physically, the nonlinear coefficient  $A$  represents higher-order convective transport between modes. The linear term  $B$  represents the effect of viscosity and the energy transfer with the mean flow field. The forcing term  $C$  represents the coupling between modes and the Reynolds stresses.

Of course, the specific values of  $A, B, C$  are completely valid only at a single point in the uncertainty space, where the snapshots were obtained. To calculate them for any other point, we would need to rerun the full-dimensional model, or to interpolate the values in the arrays over the uncertainty space. However, this would be just another variation on uncertainty quantification by extensive sampling, not feasible for complex models at a high dimension of uncertainty. Instead, to start our inquiry and assess the potential of our method without introducing additional complicating factors at this time, we use a single reduction as a source of imperfect training data.

We set up an uncertainty quantification problem based on geometric uncertainty. Given a rectangular channel, we introduced irregularities (convex or concave semi-circles) on one of the walls.

We show an example of a velocity field in Figure 2 (image created by using a prototype code, not Nek5000). The radius of each semi-circle is uncertain. The output of interest was the drag coefficient, averaged over time and over the geometric domain of the model. Formally, a change in geometric shape of the domain would lead to a change in the dimension of the model state vector;  $n$  and  $k$  would no longer be constant. We got around this difficulty by flattening the appropriate mesh elements to area 0 to represent the non-permeable parts of the domain (bumps). In this representation, the solutions  $\hat{u}(\mathbf{x})$  and  $u(\mathbf{x})$  are of the same dimension for all  $\mathbf{x}$ .

The configuration, while stylized, is relevant for nuclear engineering. The issue of determining drag coefficients in this configuration is connected to the pressure drop in a nuclear reactor channel, which is an important operating characteristic [9].

While the results on the prototype model were very good, we do not give full details here, reserving most of the discussion for the Nek5000 case; but this was our first confirmation that one can perform uncertainty analysis using fewer points than the uncertainty dimension  $m$ . Such experiments on simple models are not difficult to reproduce, and we spend most of the following section on more notable experiments in uncertainty analysis of a very complex, high-resolution simulation model that Nek5000 embodies.

### 3.2 Nek5000 models

To demonstrate the performance of the method on a model of industrial complexity, we use Nek5000 to simulate the same two-dimensional flow in the rectangular channel with geometric irregularities on one of the walls.



Nek5000 is an open-source Fortran77/C code based on the spectral-elements method, following MPI parallelization standards. It implements a high order spectral element method for the Navier–Stokes equation in about 100 K lines of code and was developed over more than 20 years. It has won the 1999 Gordon Bell prize for sustained quality of performance and scalability, and it is one of the typical test codes for new high-end architectures. Approximately 100 specialists in two dozen institutions actively use Nek5000 for simulation and design projects in and outside the nuclear engineering field. We are fortunate to work in close cooperation with its development team.

The code has a user-defined component that allows access to the internal variables and allows a range of customizations of the underlying fluid flow model; geometric mesh can also be edited directly.

In most cases, full-resolution Nek5000 is not practical to run on an average-performance machine. Using previous work on POD-based reduction of the underlying advection–diffusion equation [22], we could treat the full code as a black box that would output the high-dimensional vectors (velocity and pressure) required to assemble the terms of (7). The lower-fidelity reduced-order state vectors could be raised to full dimension and submitted to Nek5000 to calculate aggregate quantities of interest, such as averaged drag coefficient.

Inserting specific values of uncertainty-influenced quantities will always be challenging for codes of almost any organization, since the developers will not allow complete freedom in changing the physical or the geometric properties of the model and all computational precision thresholds. The correct effect of uncertainty on the variables that are defined early in the computational flow of the model may be hard to define explicitly; the variables that are declared deep inside the code and late in the computational flow may be completely inaccessible. The reduced model also adds development challenges. For example, in the POD-reduced version of Nek5000, the initial conditions and the boundary conditions cannot be edited directly: they are defined in the reduced-order subspace implicitly within the projection matrix  $\Phi$  and the assembled right-side terms  $A, B, C$ .

Note that the two main technical prerequisites for multi-fidelity analysis on a complex code such as Nek5000 are to establish format consistency for all values of uncertain quantities, and implement communication between lower-fidelity and higher-fidelity versions of the code. Some quantities are easier to manipulate in the reduced version of the model; the others are better represented in the full version. The trade-off between convenience and quality that appears in development of the method is not trivial: for the method to be effective, one has to decide which calculations to leave on a high-performance machine, and which to reproduce locally, with lower precision and higher relative cost.

Codes such as Nek5000 have verification needs in various contexts; physical fluid-flow models that need to be validated, and engineering designs need to be assessed for safety and productivity margins. At extreme computational scale, such tasks are very challenging; they have to be simplified by using external information or physical intuition, at least to make the scale of sampling over the parameter space tolerable. A lot of dedicated work therefore has gone into intelligent sampling. However, the results of analysis are limited by computational constraints. Some types of output (in particular, confidence bars for performance statistics) will not be produced at all, except for the trivial cases. Now that a multifidelity approach is also available through this work, we may begin to resolve such difficulties and provide verification procedures for such codes.

### 3.3 Experimental methodology

Our numerical experiments on the prototype code and on Nek5000 are set up in a similar manner. In both cases, we simulate two-dimensional flow in a rectangular channel with geometric irregularities in the shape of semicircular bumps. We used the default (computationally cheap) options for POD-based model reduction, and for GP-based calibration, as described in Section 2.

For the prototype code, we used a channel of dimensions  $3 \times 10$ , and  $m = 4$  semicircles of uncertain radius. POD-based reduction was based on a uniform set of snapshots along the solution trajectory based on a single point in the uncertainty space with fixed values  $\mathbf{x}_i = 0.1$  (the point could also be chosen randomly, leading to very similar results). The runtime of the models was  $0 < t < 1$ ; the drag coefficient was averaged over the time interval  $0.25 < t < 1$  and over the geometric domain. For testing purposes, we used random sampling of the uncertainty space, uniformly over the domain  $-0.25 < \mathbf{x}_i < 0.25$ , with the negative numbers corresponding to concave shapes.

The full model dimension was  $n = 2000$ , a reduced model of dimension  $k = 50$  could be evaluated approximately 20 times faster. For data calibration, we used 2 full-model outputs and 10 reduced-model outputs, assembled into the training set as described in the end of Section 2 (i.e. both perfect data and imperfect data are compared with the data from an intermediate regression model).

For Nek5000, we used a rectangular channel of proportions  $1 \times 10$ , and uncertainty dimensions  $m = 4, m = 20$ . The runtime of the models was  $0 < t < 1$ ; the drag coefficient was measured at  $t = 1$  and averaged over the geometric domain (averaging over time would have required a lot of additional model data storage). The full model dimension was  $n = 20,000$ ; we used reduction to a fixed dimension  $k = 40$ . POD-based reduction was based on a uniform set of snapshots along the solution trajectory based on a single point in the parameter space, the point was chosen randomly. The evaluation of the full code took approximately 40 min, 15 s for the reduced code. For  $m = 4$ , the calibration procedure used either 5 or 3 full model runs, and the testing data was sampled uniformly over the domain  $-0.5 < \mathbf{x}_i < 0.5$ . We have also performed experiments using a larger dimension of uncertainty,  $m = 20$  and the smaller range,  $-0.3 < \mathbf{x}_i < 0.3$ . In this case, the calibration procedure used 20 model runs. As appropriate for preliminary investigation of the higher-dimensional case, we used a somewhat simplified physical setup for a high number of geometric deformation; it is better to think of such experiments as performed on an independent problem. Nevertheless, we find it reassuring to see that with a computational budget of only 20 full-model runs, we are able to recover the global statistics of the model correctly.

#### 4. Numerical results

We now report on some of the tests on our main testing problem: Navier–Stokes flow in a channel with geometric irregularities, implemented by the Nek5000 solver. In Figure 3, we compare pointwise predictions for two calibration strategies on the same randomly selected 80 points: using 5 points and a Gaussian model trained on pointwise difference between full- and reduced-model outputs, versus using only 3 points and a Gaussian model trained on a mixed, weighted set of full- and reduced-model outputs. A cheaper prediction strategy is worse pointwise, and the confidence interval is less conservative; the important message here, however, is that both approaches provide effective results at limited computational cost (lower than the cost of linear approximation on full model data!) Comparing some of the metrics in Table 1, we see that our estimates cover the test sets at least at the nominal value, and actually better. The latter is a sign that the method is a bit conservative; but since it is close, and given the application area, we believe this to be acceptable.

We use the better of the two predictions to look at how well our prediction reproduces the statistical distribution of uncertainty influenced output. In Figure 4 we compare cumulative distributions of true and predicted data; in Figure 5, we also present a Q–Q (quantile-to-quantile) comparison. We see that the distribution is off, showing a nonlinear discrepancy that we do not capture (and we cannot be expected to capture over four dimensions with five samples). However, we note that the uncertainty shows a *factor of two variation* of the drag coefficient over the range of the uncertainty space, an enormous variation by engineering standards. Moreover, the tails and



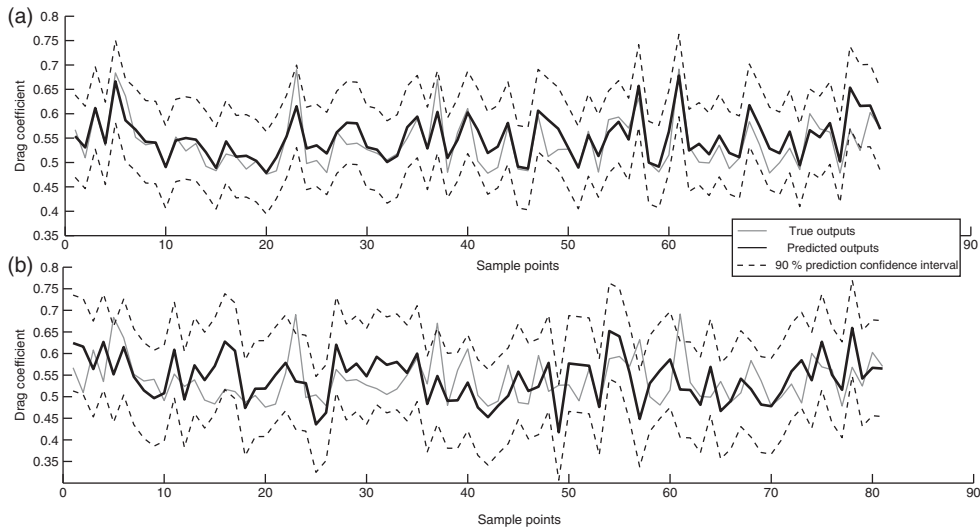


Figure 3. Pointwise predictions for Nek5000 output, also showing 90% confidence interval. (a) Dimension of uncertainty  $m = 4$ , 5 points used. (b) Dimension of uncertainty  $m = 4$ , 3 points used.

Table 1. Predictions using calibrated data for Nek5000 with four bumps; two calibration strategies.

Metric	Predic w. 5 points	Predict w. 3 points	Validating set
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
Data within 90% interval	98.5%	95.5%	90%

variability are comparable, and hence the confidence intervals are correct at the nominal value (in the sense that they cover at least as much percentage of data as stated), as seen from Table 1 and Figure 3.

We use the calibrated data to obtain the commonly used 95–95 estimate for the output, that is, a 95% double-sided confidence interval for the 95th percentile. The estimate is shown in Figure 6. The true percentile is within the predicted interval; cumulative distribution of the outputs is also shown. Once again, we stress that this estimate was obtained with only 5 full-model evaluations; normally such a task would require sampling hundreds of points.

The limiting factor of the method from our perspective at this time is the difficulty of nonlinear optimization in high-dimensional space, an issue that can be removed by using a Bayesian framework where the likelihood needs to be computed, but not really optimized, or a linear superposition Gaussian model. At this moment, however, the feature is not implemented, and for the case  $m = 20$  this constrained us to taking smaller uncertainties in the geometry in terms of amplitude. At this dimension, the optimization problem within GPs needs to be restarted multiple times to find an acceptable local minimum. Once this is resolved, the results are consistent with what we have shown before. Metrics from this experiment are given in Table 2. The quality of pointwise prediction appears to be good, even for  $m = 20$  (see Figure 7); but we are more interested in the global metrics comparing the distributions, which we analyse next.

In Figure 8 we show a comparison of cumulative distribution functions. As before, the prediction by calibrated lower-fidelity data is somewhat conservative, with heavier tails of the distribution. It is appropriate for engineering tasks focused on not exceeding a certain safety threshold (a common

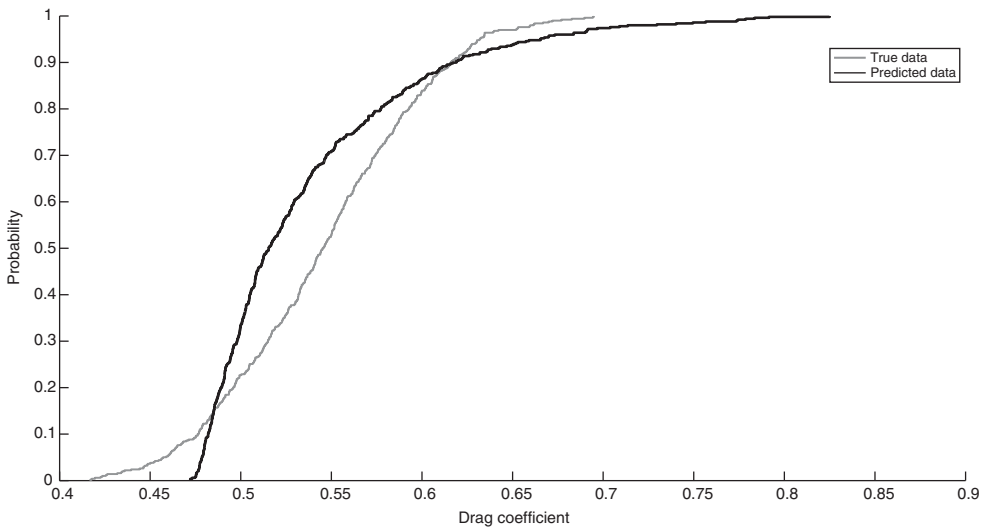


Figure 4. Nek5000: four-dimensional uncertainty, cumulative distribution plot.

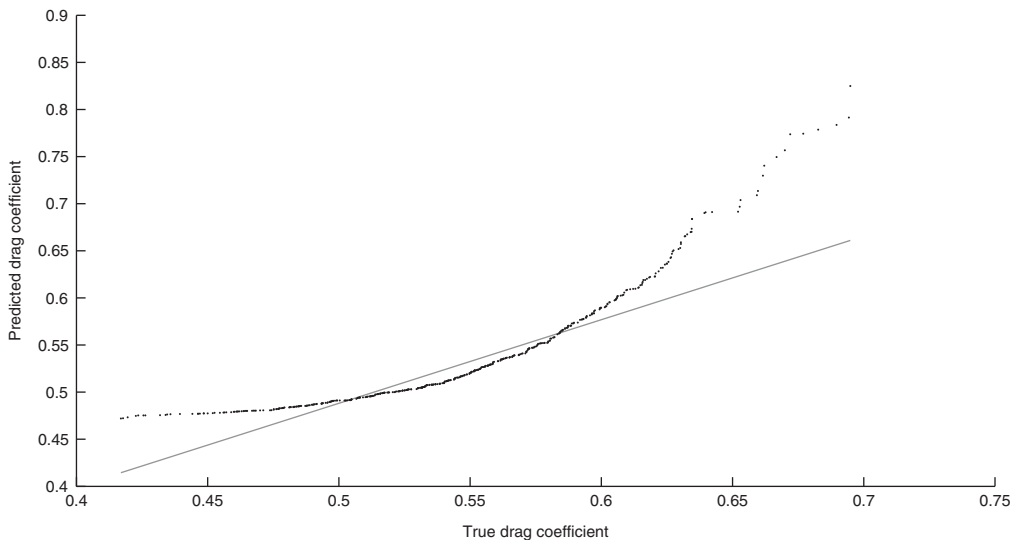


Figure 5. Nek5000: prediction for four-dimensional uncertainty, quantile-quantile plot.

setup in nuclear engineering). For another comparison of the predicted and the true distribution, we present a Q-Q plot in Figure 9; again, note the heavier tails of the predicted distribution. From the last two graphs we can informally state that our prediction of global statistical properties of the model is essentially correct and conservative.

In Figure 10, we obtain the 95–95 estimate for the output with 20-dimensional uncertainty: our confidence interval is placed symmetrically around the true percentile. The results are better than the  $m = 4$  case; however, we recall that the size of the uncertainty is smaller because of the difficulties with optimization. This  $m = 20$  case also shows that quantile estimates and confidence intervals are again correct if a bit conservative. Of course, getting the distribution of the discrepancy

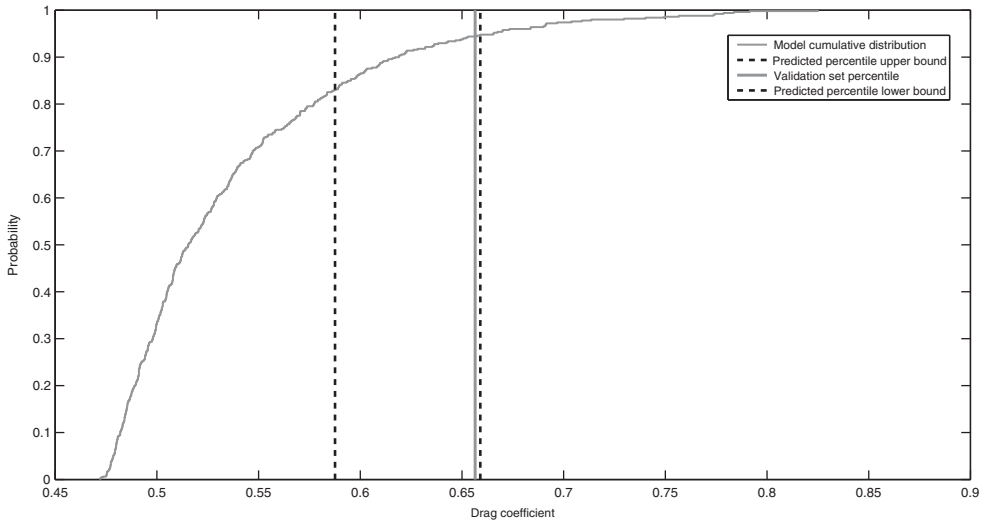


Figure 6. Nek5000: 95–95 estimate for the drag coefficient, prediction after calibration process on 5 points.

Table 2. Predictions using calibrated data for Nek5000 with 20 bumps.

Metric	Prediction	Validating set
Mean	0.7299	0.7200
Range	0.7187:0.7223	0.7190:0.7208
St. deviation	0.0054	0.0041
Data within 90% confidence	92%	90%

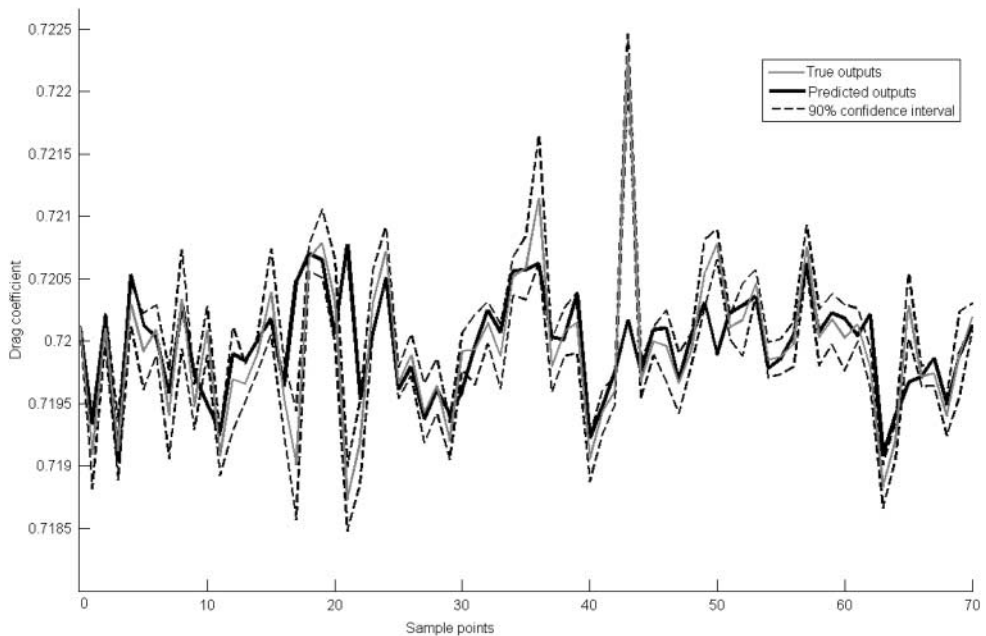


Figure 7. Nek5000: prediction for 20-dimensional uncertainty.

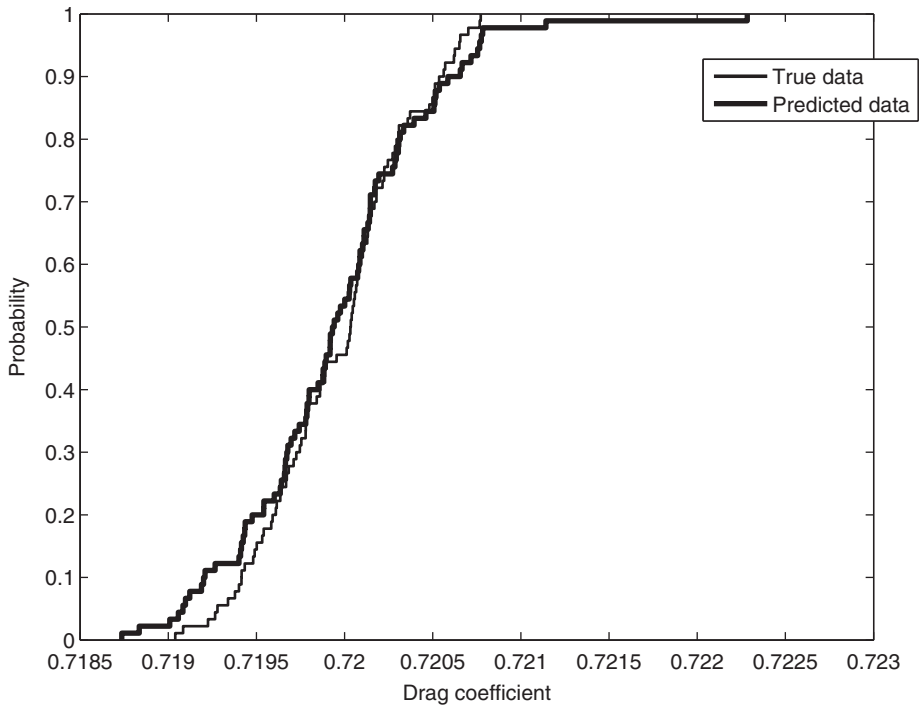


Figure 8. Nek5000: 20-dimensional uncertainty, cumulative distribution plot.

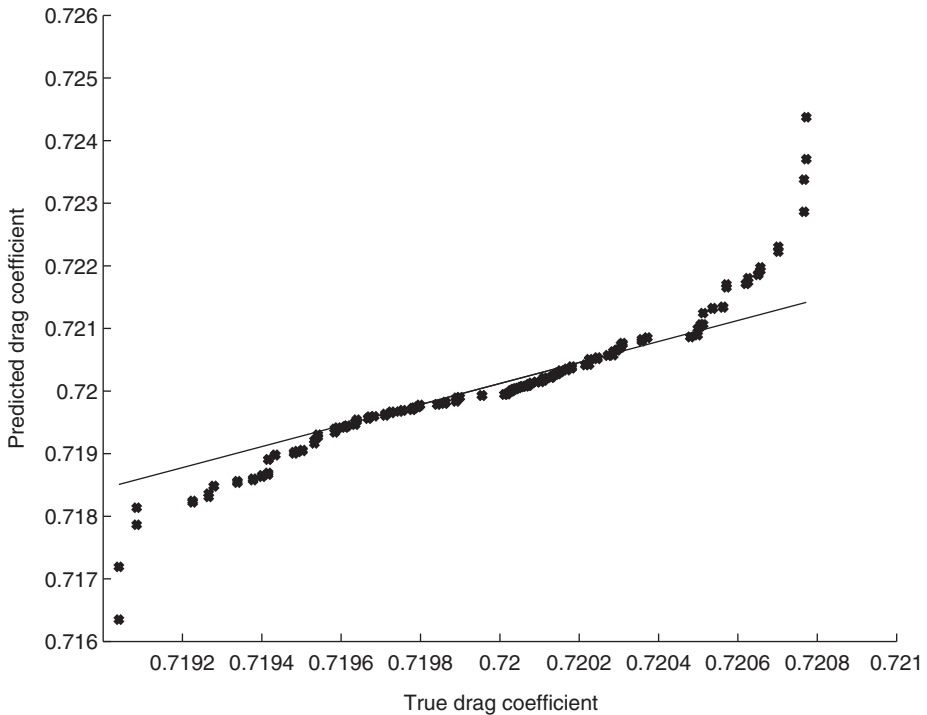


Figure 9. Nek5000: prediction for 20-dimensional uncertainty, quantile-quantile plot.

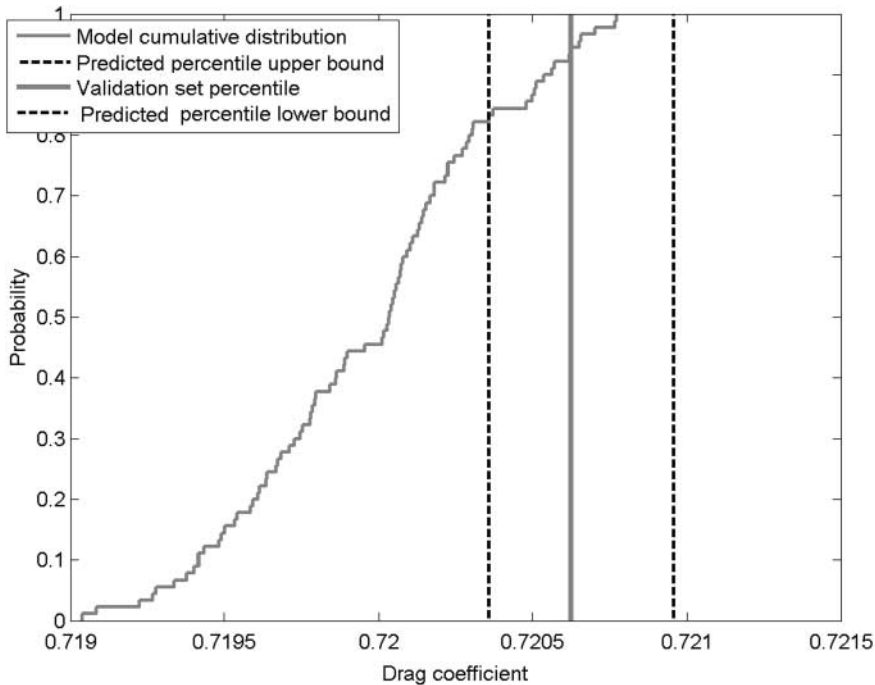


Figure 10. Nek5000: 95–95 estimate for the drag coefficient, 20-dimensional uncertainty, prediction after calibration process on 20 points.

to be exactly normal would be unlikely in any case, so this is not a flaw of the method; and absent additional information at the time of calibration, no other option seems feasible.

We conclude that for both  $m = 4$  and  $m = 20$  our uncertainty estimates are efficient and correct in that they cover at least the amount of data specified by the nominal coverage, and the excess is in the low percentages. So the method of this work is successfully validated on these examples using Nek5000 while using only a small number of calibration runs (comparable to  $m$ ).

## 5. Discussion

Our numerical experiments show that the introduced approach to uncertainty quantification based on lower-fidelity data can be effective for a class of flow models, even for such high-resolution codes as Nek5000. This demonstration addresses a challenging task in uncertainty analysis and does so at an unprecedented low cost, as measured in computationally expensive evaluations of the full code. This has implications for the previously inaccessible tasks of code analysis, and ultimately for engineering tasks of confidence assessment and improvement of performance under uncertainty.

Based on our experiments, we give the following (informal) characterization of the method's performance. For a class of models considered, we constructed 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. The pointwise quality of prediction is not remarkable; one could get almost the same, and sometimes even better,

quality by linear interpolation or just sampling of the reduced model alone. The motivation behind the method is in the capability for providing confidence intervals for data. Such intervals were observed to be essentially correct for high quantiles (the ones of interest) even if they were a bit larger than optimal, of course a better situation than the alternative. We could even output a correct confidence interval for high-order statistics (percentile), a task that would normally require large-scale sampling of the outputs and would be impossible for Nek5000 when it needs high-end resources for just one run, as it is often the case for some of its configurations.

At the moment we can make such positive statements only about the configuration we tested: flow in a channel subject to geometric uncertainty. But should such observations hold for other configurations and much more expensive calculations we could provide uncertainty results in cases where it is difficult to see another way to do so. We note that for the case  $m = 20$  we believe we could have probably done considerably better by doing some form of component analysis among the perturbations to reduce the dimension of the uncertainty space first and create three levels of fidelity, as opposed to two; however, at the moment this is only an unproven hypothesis.

Our suggested scheme for lowering the cost of prediction even further through the use of an intermediate regression function and/or component analysis 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.

Overall, uncertainty analysis on extremely small training sets augmented with large, lower-quality calibrated data sets based on model reduction approaches in the context of industrial codes 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.

## 6. Conclusions and future work

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

Our suggested approach is computationally inexpensive and estimates the variability of the uncertainty induced output well. In particular, high quantiles of the drag coefficient are correctly, though sometimes conservatively, estimated. 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).

We note that components of our approach are not new, and have received additional attention in the recent decade. At the core of our work is a realization that it is possible to use reduced-order modelling in analysis of uncertainty (without assuming a particular quality of the reduced-order approximation) together with Bayesian characterization of lower-quality data. We operated in a realistic situation of scarcity of the computational budget and lack of external (perfect) sources of data. In effect, we based our analysis only on examination of the same simulation code on multiple levels of fidelity. For a simulation model that is very complex and expensive to evaluate, this raised additional technical difficulties of access to the model’s structure for reduction, and of consistency of introducing uncertainty into lower-fidelity and higher-fidelity codes.

We demonstrated that the approach is feasible on an advanced applied example, even when using the most basic choices for model reduction and for Bayesian calibration. In future work, we believe that it will become important to use lower-fidelity models that are not arbitrary, but, rather, are constructed in a goal-oriented manner. With additional research, we should be able

to adapt the adjoint-based procedure of controlling the error introduced by reduction. That will allow us to achieve tighter confidence intervals.

In the long term, we plan to implement our approach as a part of uncertainty analysis for the exascale nuclear engineering codesign project CESAR. The ultimate result will be improved understanding of safety and operational margins of nuclear reactors and a streamlined licensing process. However, the generalization power of the approach is unknown to us and needs more work and answering questions along several directions. Theoretically, one would like to elucidate under what circumstances and for what prediction functionals one can expect that the parametric response itself can be well approximated by POD. Some tools to this end are provided in [15], but this would be good to understand a priori on given physical applications such as Navier–Stokes. The tools in [15] can be used, however, to produce a posteriori error estimates based on adjoint calculations with the POD model only. Mixing such estimates with fewer high-fidelity runs is a promising approach, in our opinion. There is also the issue we identified in the preceding sections of using more approximation levels for a true multifidelity approach that would result from using mean functions, as we did in [21], and by reducing the uncertainty space by some form of component analysis. In terms of capability, we plan to run some very large three-dimensional cases that need the highest-end resources; but our expectation, given the physics involved, is that the same quality of the confidence intervals will be observed if geometric effects on drag are what is quantified.

While such comprehensive validations and extensions are necessary, it is worth contemplating what the technique will contribute assuming that its performance holds along these directions. The distinguishing features of our work are reliance on a POD-reduced form of the model, one of the least-difficult (in terms of development) intrusive techniques to implement, and a very cheap and in principle convenient process of assigning a Gaussian-process-based characterization to the imperfection of reduced data. Comparing this with the impossibility of non-intrusive sampling assessments of codes that need computational resources on the limit of what is available for point prediction. This technique, even if moderately successful, will help researchers carry out uncertainty quantification for applications that need extreme-scale computing.

## Acknowledgements

We thank our colleagues Paul Fischer, Elia Merzari, Alex Obabko and Zhu Wang for their assistance and input. The submitted manuscript has been created by the University of Chicago as Operator of Argonne National Laboratory ('Argonne') under Contract No. DE-AC02-06CH11357 with the US Department of Energy. The US Government retains for itself, and others acting on its behalf, a paid-up, non-exclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.

## References

- [1] M. Alexe, O. Roderick, M. Anitescu, J. Utke, T. Fanning, and P. Hovland, *Using automatic differentiation in sensitivity analysis of nuclear simulation models*, Trans. Amer. Nucl. Soc. 102 (2010), pp. 235–237.
- [2] M. Anitescu, J. Chen, and L. Wang, *A matrix-free approach for solving the parametric Gaussian process maximum likelihood problem*, SIAM J. Sci. Comput. 34 (2012), pp. A240–A262. Available at <http://epubs.siam.org/doi/abs/10.1137/110831143>.
- [3] O.G.T. Bui-Thanh and K. Willcox, *Model reduction for large-scale systems with high-dimensional parametric input space*, SIAM J. Sci. Comput. 30(6) (2008), pp. 3270–3288.
- [4] N. Cressie, *Spatial prediction and ordinary kriging*, Math. Geol. 20(4) (1988), pp. 405–421.
- [5] N. Cressie, *Statistics for Spatial Data*, Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics Section, John Wiley & Sons, New York, 1993.
- [6] D.N. Daescu and I.M. Navon, *A dual-weighted approach to order reduction in 4D-var data assimilation*, Mon. Weather Rev. 136(3) (2008), pp. 1026–1041.
- [7] A. DasGupta, *Asymptotic Theory of Statistics and Probability*, Springer, New York, 2008.
- [8] J.L. Doob, *Stochastic Processes*, Vol. 101, John Wiley & Sons, New York, 1953.

- [9] J. Duderstadt and L. Hamilton, *Nuclear Reactor Analysis*, Wiley, New York, 1976.
- [10] P. Fischer, *Nek5000-open source spectral element cfd solver*, 2008. Available at <https://nek5000.mcs.anl.gov/index.php/MainPage>.
- [11] M. Frangos, B.v.B. Waanders, and Y. Marzouk, *Surrogate and Reduced-order Modeling: A Comparison of Approaches for Large-scale Statistical Inverse Problems*, John Wiley & Sons, New York, 2011, pp. 123–150.
- [12] K.W.O.G.D. Galbally and K. Fidkowski, *Non-linear model reduction for uncertainty quantification in large-scale inverse problems*, *Int. J. Numer. Methods Eng.* 81 (2010), pp. 1581–1608.
- [13] M. Gunzburger, *Reduced-order modeling, data compression and the design of experiments*, Second DOE Workshop of Multiscale Mathematics, Bromfield, CO, 2004.
- [14] M. Hinze and S. Volkwein, *Proper Orthogonal Decomposition Surrogate Models for Nonlinear Dynamical Systems: Error Estimates and Suboptimal Control*, *Lecture Notes in Computational Science and Engineering*, Vol. 45, Springer, New York, 2005.
- [15] L.P.C. Homescu and R. Serban, *Error estimation for reduced order models of dynamical systems*, Tech. Rep., United States, Department of Energy, 2003.
- [16] M. Kennedy and A. O’Hagan, *Predicting the output from a complex computer code when fast approximations are available*, *Biometrika* 87 (2000), pp. 1–13.
- [17] M. Kennedy and A. O’Hagan, *Bayesian calibration of computer models*, *J. R. Stat. Soc. Ser. B Stat. Methodol.* 63 (2001), pp. 425–464.
- [18] C. Kenney and A. Laub, *Small-sample statistical condition estimates for general matrix functions*, *SIAM J. Sci. Comput.* 15(1) (1994), pp. 36–61.
- [19] D.C.Y. Lacouture, *How to use matlab to fit the ex-gaussian and other probability functions to a distribution of response times*, *Tutorials Quant. Methods Psychol.* 4(1) (2008), pp. 35–45.
- [20] O.G.C. Lieberman and K. Willcox, *Parameter and state model reduction for large-scale statistical inverse problems*, *SIAM J. Sci. Comput.* 32 (2010), pp. 2523–2542.
- [21] B.A. Lockwood and M. Anitescu, *Gradient-enhanced universal kriging for uncertainty propagation*, *Nucl. Sci. Eng.* 170(2) (2012), pp. 168–195.
- [22] E. Merzari, P. Fischer, and D. Pointer, *A POD-based Solver for the Advection-Diffusion Equation*, *Proceeding of ASME-JSME, KSME Joint Fluids Engineering Conference*, Hamamatsu, Shizuoka, Japan, 2011.
- [23] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, MA, 2006.
- [24] O. Roderick, *Model reduction for simulation, optimization and control*, Ph.D. thesis, Portland State University, 2009.
- [25] O. Roderick, M. Anitescu, and P. Fischer, *Polynomial regression approaches using derivative information for uncertainty quantification*, *Nucl. Sci. Eng.* 164 (2) (2010), pp. 122–139.
- [26] M. Stein, *Interpolation of Spatial Data: Some Theory for Kriging*, Springer-Verlag, Berlin, 1999.
- [27] S. Volkwein, *Proper orthogonal decomposition: Applications in optimization and control*, CEA-EDF-INRIA Numerical Analysis Summer School, Cadarache, France, 2007.
- [28] L.S.G. Webber and R. Handler, *The Karhunen-Loeve decomposition of minimal channel flow*, *Phys. Fluids* 9 (1997), pp. 1054–1066.