International Journal of Computer Mathematics
Vol. xx, No. xx, Xxxxxx 2013, 1–21

RESEARCH ARTICLE

Proper orthogonal decompositions in multifidelity uncertainty quantification of complex simulation models

Oleg Roderick, [(a)] * Mihai Anitescu, [(a)], Yulia Peet [(b)]

[a]*Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439, USA*

[b]*Arizona State University, 501 E. Tyler Mall, Tempe, AZ 85287, USA*

(*January 30, 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 a 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;

AMS Subject Classification:
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 analysis. As a result, in many fields, such as nuclear engineering, fluid dynamics, or climate modeling, we may be approaching the situation in which assessment of models

*Corresponding author. Email: *roderick@mcs.anl.gov*

ISSN: 0020-7160 print/ISSN 1029-0265 online
© 2013 Taylor & Francis
DOI: 0020716YYxxxxxxxx
http://www.informaworld.com

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 modeling, 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 simulations context 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. Such models can appear in multiple guises. Simulation models do not exist alone, as sole examples in their class of codes. In the process of 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.

By necessity, model simplifications and approximations must be 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.

Once such lower-fidelity models exist, Kennedy and O'Hagan [15, 16] suggested tjat a statistical model should be constructed to describe the imperfection in the cheaper data (error-prone computer simulations versus almost perfect physical observational data, in their case). Together with cheap low-fidelity models, one can estimate the uncertainty effects at much lower cost.

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 drthogonal decomposition 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 method (PCA), proper orthogonal secomposition method (POD), and method of snapshots. It has been restated for applied tasks of simulation and optimization in high dimension by multiple authors [12, 13, 23]. 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; we refer to [7, 20]. 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. To our knowledge, however, its use in uncertainty propagation and analysis has been far less explored.

The key conceptual appeal, which we will not explore here in detail, is that the approach can in principle approximate the dynamical system arbitrarily well. Moreover, this error can be estimated in principle based on runs of the POD model followed by estimates of the residual on the full model (but not fully converged high-fidelity simulations), based on the Kenney-Laub approach [17]. Kenney and Laub introduced the small-sample xondition estimate method (SCE) for matrix and vector operations, based on using random vector products uv^T as predictor for the vector norm $\|u\|_2$. Petzold et al. [3] used this approach to develop error analysis for perturbed, POD-reduced ODEs and 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 additonal 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 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 [18, 21]. 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 remained to be answered, however, is whether it is still suitable for uncertainty quantification, that is, whether, at least in some circumstances, it also captures well parametric dependency.

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 it for a nontrival 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 uncertaintty. Moreover, we will carry the analysis using a high-end fluid dynamics code, Nek5000 [11], 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 recently developed for extrapolation over time and recovery of full-dimensional velocity field [10]. 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 [15, 16]. Their characterization methods are based on standard Bayesian reasoning and Gaussian processes for machine learning. We followed some of their framework, with implementation details for Gaussian processes as recommended by Rasmussen [19]. Combined with our recent insights from Gaussian process analysis [18], 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: proper orthogonal decomposition-based dimensionality reduction, Gaussian processes-based calibration, and putting them both together to recover model statistics from calibrated lower-quality data. In Section 3 we describe the Navier-Stokes models we used to test the performance of the method. In Section 4 we present the results of numerical experiments. In Section 5 we review the performed work, and suggest directions for future development.

2. Method description

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

2.1 *Proper orthogonal decomposition-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 $A = \sum_{i=1}^{n} u_i \sigma_i v_i^T$, 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 eigenvalue energy error $\frac{\sum_{i=k+1}^{n} \sigma_i}{\sum_{i=1}^{n} \sigma_i}$ is close to 1.

According to the method of snapshots [23], 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 $t_1, t_2, ..., t_{n'}$ are recorded as matrix columns: $A = (T(t_1), T(t_2), ..., T(t_{n'}))$. 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

$$C\phi = \lambda\phi. \tag{2.1}$$

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

The matrix Φ 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

$$\hat{T}(t) = \sum_{i=1}^{k} \phi_i(t) q_i(t) \tag{2.2}$$

is effective, $\hat{T}(t) \approx T(t)$. Here, $q(t)$ is the approximating trajectory in low-rank subspace coordinates (a shift term μ, 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. Given the system

$$\begin{array}{c} \frac{du}{dt} = f(u, t, P, x) \\ u(t_0) = u_0(P, x) \end{array}, \tag{2.3}$$

we collect the snapshots $A = [u(t_1), u(t_2), ..., u(t_{n'})]$ and solve (2.1). Once Φ is defined, the reduced-order approximation to the model solution, written as $\hat{u}(t) = \Phi q$, is determined by solving

$$\begin{array}{c} \frac{dq}{dt} = \Phi^T f(\Phi q, t, P, x) \\ q(t_0) = \Phi^T u_0(P, x) \end{array}. \tag{2.4}$$

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 $k << n$. If needed, adjoint differentiation of the reduced model is also computionally cheaper.

Note that q and u are functions of both time and parameters, $q \equiv q(t, x)$ and $u \equiv u(t, x)$, though we may choose to expose the one on 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 x defined by the uncertainty analysis can be well approximated by the random variable* $\Phi q(t, x)$. In turn, this will allow for statistics of u to be computed based on the much cheaper statistics of q.

Representation of the reduced model as (2.4) 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 very 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

$$\begin{array}{c} \frac{\partial u}{\partial t} + (u \cdot \nabla)u - \frac{1}{Re}\Delta u + \nabla p = 0 \\ \nabla \cdot u = 0 \end{array}. \tag{2.5}$$

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

$$(\frac{\partial u_i}{\partial t}, \phi_i) + ((u_i \cdot \nabla)u_i, \phi_i) + (\frac{2}{Re}D(u_i), \nabla\phi_i) = 0, i = 1, 2, ..., k, \tag{2.6}$$

and integrated in the form of the reduced-order ODE

$$\frac{dq_k}{dt} = A_{ijk}q_iq_j + B_{ik}q_i + C_k. \tag{2.7}$$

The resulting bilinear nature of (2.7) 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 ϕ_i $\phi_i \cdot (-(u \cdot \nabla)u + \frac{1}{Re}\Delta u)$ with substitution $u = \Phi q$. The results of the substitution can be fairly bulky, for example $C_i =$

$\frac{du_0}{dx} u \cdot \phi_{ui} + \frac{du_0}{dy} v \cdot \phi_{ui} + \frac{dv_0}{dx} u \cdot \phi_{ui} + \frac{dv_0}{dy} v \cdot \phi_{ui} + \nabla u_0 \cdot \phi_{ui} + \nabla v_0, \phi_{vi}$ (where u, v are the components of velocity in axial directions x, y). Nevertheless, the derivation of terms is a straightforward exercise, performed in the same way for any right-side function f.

The parameters x are present in the arrays A, B, C, and in many circumstances this dependence can be recovered analytically (for example, the dependence of Reynolds number enters linearly in B) but a complete practical framework is difficult to derive and present. We thus decided to not represent it explicitly and exploit its structure at this time, though that is definitely a most intriguing endeavor 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.

Now, 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 to it. 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 Φ. The most straightforward way to improve (or "enrich") the reduced model is to enhance the POD basis Φ by combining snapshots from trajectories started from several different points in the uncertainty space. Here, the effective improvement can be limited: a procedure for improvement 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 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 leave 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 Gaussian process approach in §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*

Stochastic process [8] is a standard tool for representing dynamics of models with indeterminant source of variability; it 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 $\mu(x)$, $cov(x, x')$. The role of the mean function can be played (initially, until revision) by any regression model $R(x)$; the variance structure needs to be additionally fitted to the available data. Gaussian processes (GP) is a special case of the stochastic processes approach, 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 spatio-temporal and other multidimensional uncertainty [5, 19, 22].

At the core, GP machine learning is a response surface method. Unlike commonly used regression, we are 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 $X_O = (x_{O1}, x_{O2}, ..., x_{ON}) : x_{Oi} \in \mathbb{R}^n$.

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

$$cov(x, x'; \theta) = \delta^2 \cdot \sum_{i=1}^{m} (1 + \sqrt{3}|\frac{x_i - x'_i}{\theta_i}|) exp(-\sqrt{3}|\frac{x_i - x'_i}{\theta_i}|). \tag{2.8}$$

A small number of other suitable covariance functions are available, the most commonly used being the *squared exponential* $cov(x, x'; \theta) = \delta^2 \cdot \sum_{i=1}^{m} exp(\frac{x_i - x'_i}{\theta_i})^2$. 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 [22]. We refer to the work of Rasmussen and Williams [19], and also our previous work [18] for a discussion of the choice between covariance functions. Part of our choice originates in our experience in [18] that Matern 3/2 is a safe and robust choice; extended studies of the proper classes are certainly necessary in future work.

The parameter δ^2 in the expression can be estimated by $var(Y_O)$. The *hyperparameters* θ_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 $cov(x, x'; \theta)$ in a negative logarithmic form. The task of fitting the hyperparameters amounts to maximizing the log-likelihood

$$-log(Pr(Y_O|X, \theta)) = -\frac{1}{2} Y_O^T K^{-1} Y_O - \frac{1}{2} log|K| - \frac{n}{2} log(2\pi), \tag{2.9}$$

where $K = (cov(x_i, x_j, \theta))$, $x_i, x_j \in X_O$ 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 [18] show that the resulting distribution approximates the uncertainty well. We use standard nonlinear optimization tools (Matlab routines *fminsearch* and *fminunc* are acceptable [24]). 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 adressed 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 testing set to the training set:

$$K_V = (cov(x, x'), \theta), x \in X_O, x' \in X_V. \tag{2.10}$$

The outputs are predicted as

$$Y_V = K_V^T \cdot K^{-1} \cdot Y_0. \tag{2.11}$$

Given the initial (or *prior*) version of the mean function $R(x)$, a posterior version of the mean on testing set is

$$R'(x) = R(X_O) + K_V^T \cdot K^{-1} \cdot (Y_V(x) - R(x)). \tag{2.12}$$

A posterior prediction for the variance is given by

$$V(x) = var(Y(x_i)) = \delta^2 \cdot (cov(x_i, x_i; \theta) - K_V^T \cdot K^{-1} \cdot K_V). \tag{2.13}$$

The estimate (2.13) 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 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. We can define the training data as

$$Y_O = \mathcal{J}(X) - \hat{\mathcal{J}}(X) \tag{2.14}$$

and then output the calibrated data as $\mathcal{J}_C = \hat{\mathcal{J}}(X_V) + Y_V$, using (2.11). Confidence intervals are written as

$$(\mathcal{J}_C(x) - zV(x), \mathcal{J}_C(x) + zV(x)), \tag{2.15}$$

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 θ 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 completentary subspace of dimension $n - k$, as estimated from the nondominant 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 (2.14) 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 indepedently of each other (because feedback and adaptive improvement requires 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 at least a partial sampling of the model trajectory. Thus the basic cost is one full-model evaluation (or less, if it is possible to collect snapshots from an incomplete time interval and still capture the essential dynamics). We choose a single point in the uncertainty space (in the absence of additional information, all uncertainty quantifiers are given nominal values, for example, $x_i = 0$) and evaluate the model. SVD of the empirical correlation matrix $C = AA^T$ needs to be performed only once; only the projection matrix Φ 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 $\frac{dJ(T)}{dA}$. 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 (2.1) to the form

$$\Lambda C\phi = \Lambda\lambda\phi, \tag{2.16}$$

where Λ is the identity matrix with the jth component on the diagonal multiplied by weight $w > 1$. The influence of a particular snapshot $T(t_k)$ is amplified by modifying (2.1) to the form

$$\frac{1}{w}AWA^T\phi = \lambda\phi, \tag{2.17}$$

where W is the identity matrix with the kth component on the diagonal multiplied by weight $w > 1$. The operations can be repeated and combined, resulting in a dual-weighted POD reduction procedure [7]. 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 $\Phi_1, \Phi_2, ...$ They can be aggregated into the form $\Phi = span(\Phi_1, \Phi_2, ...)$; in practice that can be accomplished by Gramm-Schmidt orthogonalization procedure. If the computational budget allows estimation of some of the sensitivites $\frac{d\mathcal{J}}{dx}$, one can choose directions of interest in the uncertainty space and sample collections of snapshots along them.

Increasing the reduced model dimension k slightly may improve the performance of the reduced model over the points in the uncertainty space distinct from the one where the snapshots were generated.

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 on the training set $\hat{\mathcal{J}}(X_O)$ and the testing set $\hat{\mathcal{J}}(X_V)$ known. In practice, there may be differences in organization of the full- and the reduced-model code. For example, the reduced model may have a nonexplicit initial state (using the first snapshot instead) or use 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.

Choice of 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 a pairs on a training set X_O and use $Y_O = \mathcal{J}(X) - \hat{\mathcal{J}}(X)$. If the budget must be less than m, we can construct an intermediate regression model $r(x)$ using the available full-model runs only: in practice, $r(x)$ can be a very low-rank least squares linear approximation for $\mathcal{J}(x)$. Then we can compare full and reduced models with $r(x)$ instead of with each other. Technically, we define $\{Y_O = \mathcal{J}(X_O) - r(X_O); \hat{\mathcal{J}}(X_O) - r(X_O)\}$ with calibration procedure taking the form of interpolation from the nearest known values of $\mathcal{J}$ and values of r. The quantities $r(x)$ and $V(x)$ are assumed to be statistically independent. The full-model outputs in this training set can be weighted, i.e. repeated multiple times 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 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 $\mathcal{J}_C \approx \mathcal{J}(X_V)$ should be performed without loss of quality.

Of course, some model dynamics can be reduced in complexity almost perfectly, preserving the essential response to uncertainty. For example, it is known that POD-based reduction does very well on certain stiff models of diffusive flow and interaction. In this case, calibration is not required: the reduced model replaces full model for all purposes.

GP-based learning of variance structure is treated mostly as a black-box procedure for producing the posterior mean $R''(x)$ and variance $V(x)$ based on training data Y_O however it may be defined. 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.

The scheme of the complete process is shown in Figure (2.1). 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 $\mathcal{J}_C$ that comes with a variance function $V(x)$; 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 [6]. Consider the pth quantile for the distribution of the output of interest $\mathcal{J}$, and choose a confidence level α (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 space and recording the pth quantile for each sample, thus forming a new variable ζ_p. The required double-side confidence interval is estimated from the empirical distribution of ζ_p, that is, from the EDF defined on M points by $Z(t) = \frac{1}{M}\sum_{i=1}^{M} 1\{\zeta_{pi} < t\}$; here $1\{...\}$ is the binary indicator of the event.

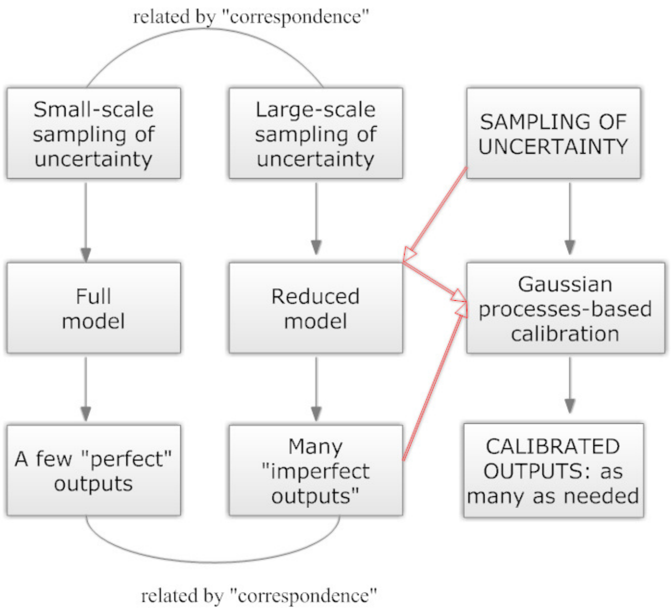
related by "correspondence"
Small-scale sampling of uncertainty
Large-scale sampling of uncertainty
SAMPLING OF UNCERTAINTY
Full model
Reduced model
Gaussian processes-based calibration
A few "perfect" outputs
Many "imperfect outputs"
CALIBRATED OUTPUTS: as many as needed
related by "correspondence"

Figure 2.1. Reduced order model sampling and calibration process

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.

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

$$\begin{array}{c} \frac{\partial u}{\partial t} + (u \cdot \nabla)u - \frac{1}{Re}\Delta u + \nabla p = 0 \\ \nabla \cdot u = 0 \end{array} \tag{3.1}$$

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

$$\frac{dq_k}{dt} = A_{ijk}q_iq_j + B_{ik}q_i + C_k. \tag{3.2}$$

Physically, the nonlinear coefficient A represents higher-order convective transport between modes. The linear term B represents the effect of viscosity and the enrgy 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. For any other point, we would need to rerun the full-dimensional PDE to obtain the updated values. It is technically possible to interpolate the arrays over the uncertainty space. However, this would be another variation on learning the uncertainty effect by extensive sampling, not possible for complex models at high uncertainty dimension. 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 (of dimensions 3×10), we introduced irregularities in the shape of several small bumps on one of the walls. We show an example of a velocity field in Figure 3.1 (image created by using a prototype code, not Nek5000). The deformations are in the shape of $m = 4$ semicircles of uncertain radius, uniformly selected from the interval $-0.25 < \xi_i < 0.25$, with the negative numbers corresponding to concave shapes. The output of interest was the drag

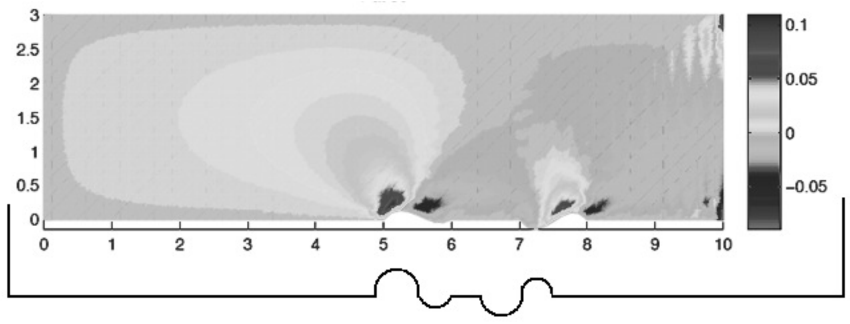

Figure 3.1. Velocity field for channel flow with wall bumps

coefficient, averaged over the time interval $0.25 < t < 1$ 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 mesh elements to area 0 to represent the nonpermeable parts of the domain (bumps). Note that with this representation, all the full and reduced representations of the channel flow will, *technically*, be defined on the same geometric grid; the solutions $\hat{u}(x)$ and $u(x)$ are of the same dimension for all 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 in turn is an important operating characteristic [9].

For our Matlab prototype, with state dimesions $n = 2000$, $k = 50$, the reduced model 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 §2 (i.e. both perfect data and imperfect data are compared against the data from an intermediate regression model). While the results 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 of 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 sections on more notable experiments in uncertainty analysis of a very complex, high-resolution simulation model that Nek5000 embodies.

To demonstrate the performance of the method on a model of industrial complexity, we use Nek5000 to simulate a flow in rectangular channel with geometric uncertainties (bumps) 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 100K 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 undelying fluid flow model. One can define the time-dependent, anisotropic forcing, the initial conditions, and the boundary conditions and can modify Reynolds number and a few other (less significant) constants.

Recall that we are interested in geometric uncertainty. This could present a seroius challenge in a wider class of codes: the effect of a local change geometry may be global, while, at the same time, geometry may not be explicitly present in user-accessible routines. For Nek5000, we can edit a file that maps finite-element vertices to their coordinates, which allows us to flatten or expand area elements to represent the presence of the bump. A different simulation code, with adaptive remeshing, could require extra implementation work to keep the dimension of the model state technically the same while the shape of the domain is uncertain.

In most cases, full-resolution Nek5000 is not practical to run on an average-performance machine. Since we had access to previous work on POD-based reduction of the underlying advection-diffusion equation [10], 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 (2.7). This reduced form of the advection-diffusion equations is very stiff; a forward integration scheme with a very small time step is adequate for the task. Note that this creates a reduced-order submanifold approximations for the velocity but not for pressure. That has to be recovered from the completed velocity solution.

We note that the code we are dealing with is a research tool, and, as such, it is not fully customer friendly nor GUI-oriented. Some familiarity with the code structure is required in order to set up examples, and additional is needed during construction of the reduced-order model.

Inserting specific values of uncertainty-influenced quantities will always be challenging for codes of almost any organization, since it is not conceivable that the developers will allow complete freedom in changing the physical or the geometric properties of the model, and all computational precision thesholds. 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 inaccessbile.

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 Φ and the assembled right-side terms A, B, C. At the same time, the initial state of the model and the boundary conditions are easily defined for the full model. The geometry of the domain is defined expilicitly (both full and reduced models access the same mapping file), but the changes required for recovery of the full model velocity and pressure field, in modified domain, based on the unmodified projection matrix, are not trivial and require additional development effort.

Once the full model state was recovered, we used a standard routine (on a high-performance machine) to evaluate the drag coefficient at a given moment in time.

We point out two of the code features that would be sufficient to enable reduced-order modeling: underlying right-side equations of the ODE are known (perhaps up to some negligible closure term that can be recomputed in full space), and changes introduced by uncertainty do not imply a change in the model dimension.

Codes such as Nek5000 have verification needs in various contexts; there are physical fluid-flow models that need to be validated and engineering designs that 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.

The setup we used with Nek5000 is much the same as before: we simulate flow in a rectangular channel (of proportions 1×10) and introduce a number of semi-circular irregurarities on one of the walls. In our experiments, we used $m = 4$ and $m = 20$. The output of interest was the drag coefficient at $t = 1$, averaged over the model's 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 $k = 40$. The evaluation of the full code took approximately 40 minutes, 15 seconds for the reduced code. For $m = 4$, the calibration procedure used either 5 or 3 full model runs. We have also performed experiments using a large dimension of uncertainty, $m = 20$, the calibration procedure used 20 model runs. Note that, 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 irregurailties, implemented by Nek5000 solver. In Figure 4.1, we compare pointwise predictions for two calibration strategies on the same randomly selected 80 points: using 5 points and training Gaussian model on pointwise difference between full and reduced model outputs, versus using only 3 points, and training Gaussian model on a mixed, weighted set of full and reduced model outputs. A cheaper prediction strategy is worse pointwise, the confidence interval is less conservative; the important message here, however, is that both approaches provide effective results at very limited computational cost (lower that the cost of linear approximation on full model data!) We compare some of the metrics in Table 4.1, we see that our estimates cover the test sets at least at the nominal value, and actually better. The latter is of course 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.

Table 4.1. Predictions using calibrated data for Nek5000 with 4 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%

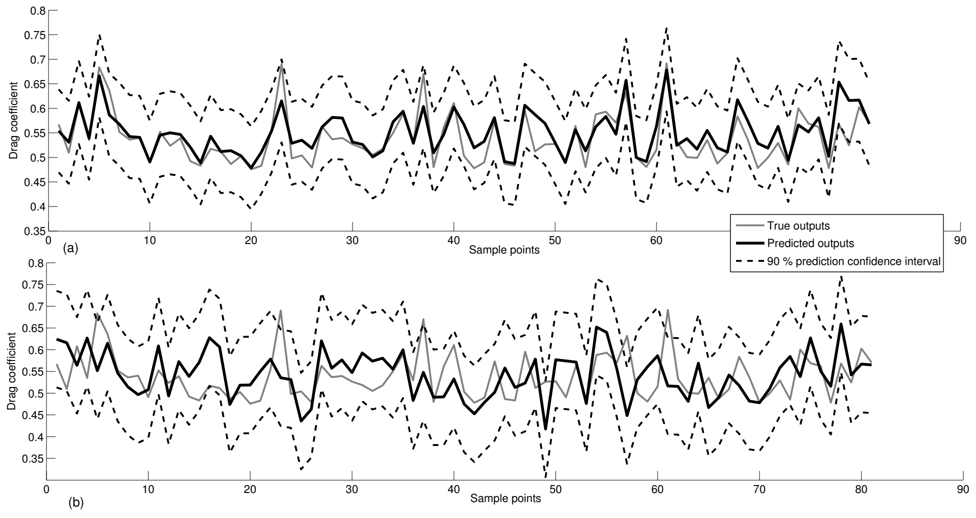

Figure 4.1. Pointwise predictions for Nek5000 output, also showing 90% confidence interval. (a) Dimension of uncertainty $m = 4$, 5 points used. (b) Dimension of uncertainty $m = 4$, 3 points used.

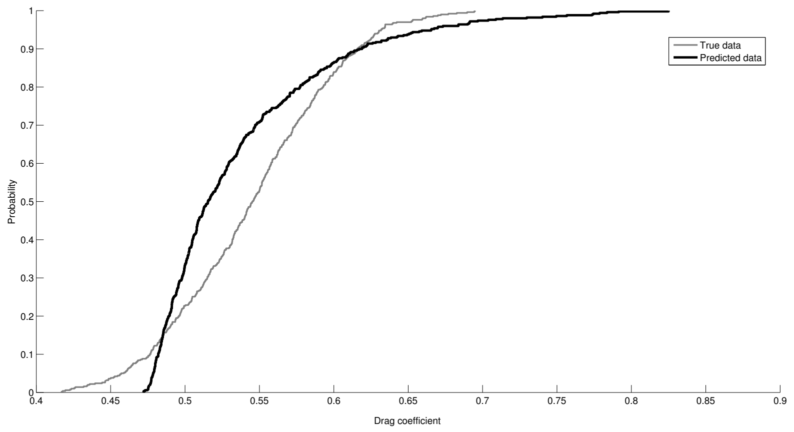

Figure 4.2. Nek5000: 4-dimensional uncertainty, cumulative distribution plot

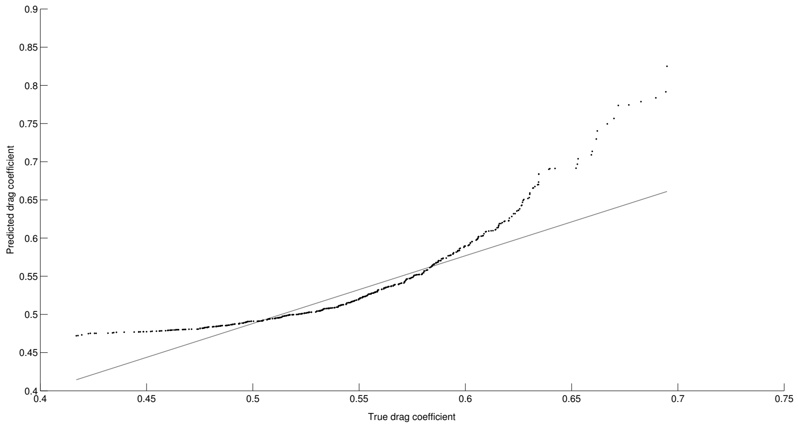

Figure 4.3. Nek5000: prediction for 4-dimensional uncertainty, quantile-quantile plot

We use the better of the two predictions to look at how well our prediction reproduces the statistical distribution of uncertainty influnced output. In Figure 4.2 we compare cumulative distributions of true and predicted data; in Figure 4.3, 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 4 dimensions with 5 samples). However, we should note that the uncertainty is as large that it shows a *factor of 2 variation* of the drag coefficient over the range of the uncertainty space, an enormous variation by engineering standards. Moreover, what is truly relevant is that the tails and 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 4.1 and Figure 4.1.

We use the calibrated data to obtain the popularly 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 4.4. 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 using only 5 full-model evaluations; normally such a task would require sampling hundreds of points.

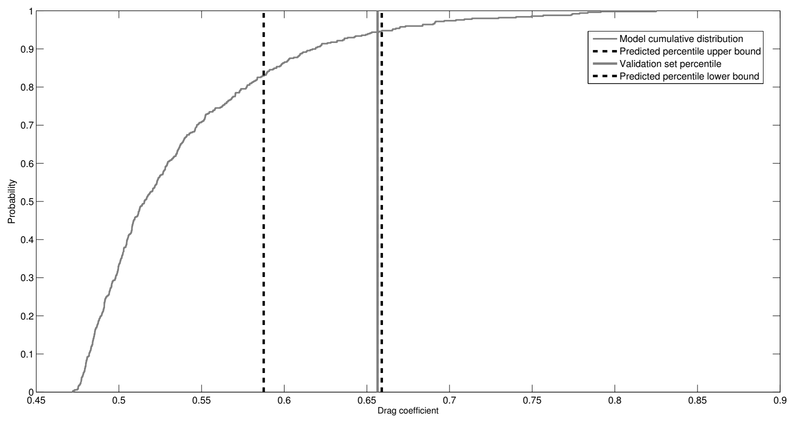

Figure 4.4. Nek5000: "95-95" estimate for the drag coefficient, prediction after calibration process on 5 points

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. However, at this moment 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. This case consists of twenty wall deformations, although of relatively smaller size for the reasons mentioned ($m = 20, -0.3 < \xi_i < 0.3$). The calibration process required 20 full-model evaluations; in addition, at this dimension the optimization problem within Gaussian processes 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 4.2. The quality of pointwise prediction appears to be good, even for $m = 20$; but we are more interested in the global metrics comparing the distributions, which we analyze next.

In Figure 4.6 we show a comparison of cumulative distribution functions. As before, the prediction by calibrated lower-fidelity data is somewhat conservative,

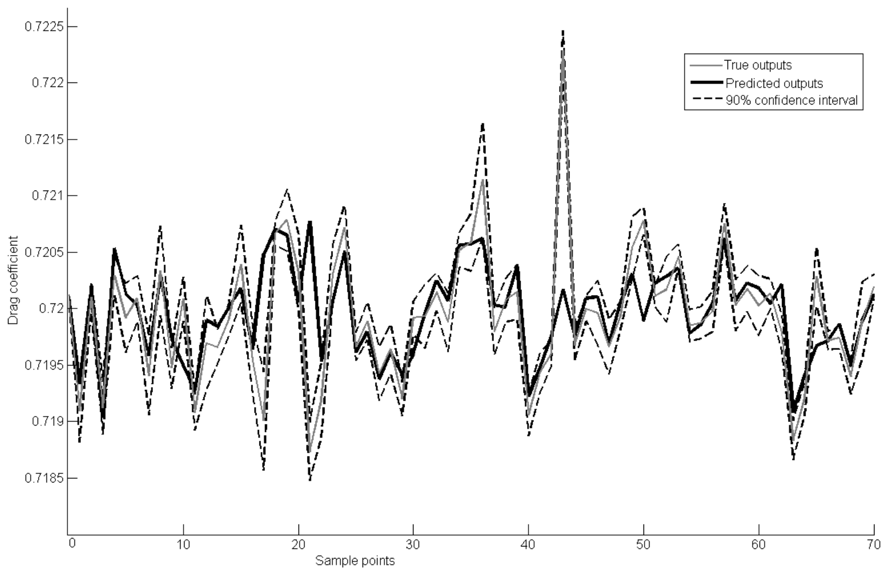

Figure 4.5. Nek5000: prediction for 20-dimensional uncertainty

Table 4.2. 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%

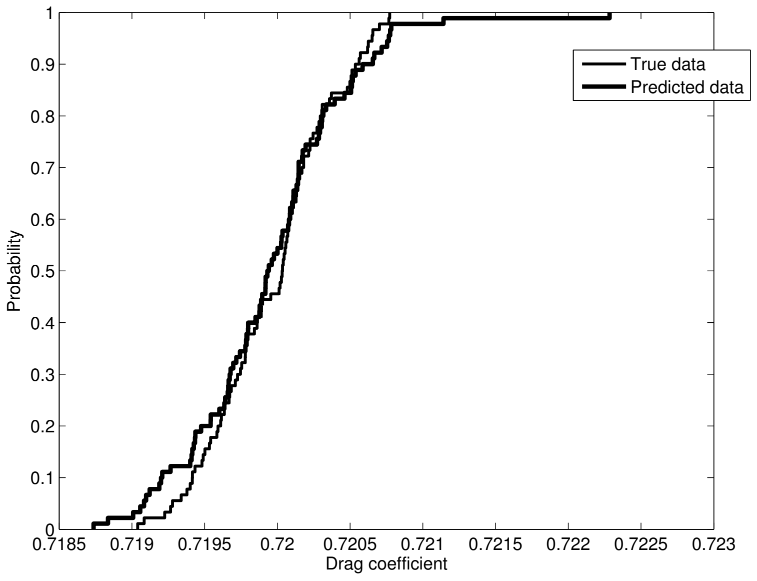

Figure 4.6. Nek5000: 20-dimensional uncertainty, cumulative distribution plot

with heavier tails of the distribution. It is appropriate for engineering tasks focused on not exceeding a certain safety threshold (a common setup in nuclear engineering). For another comparison of the predicted and the true distribution, we are presenting a Q-Q plot in Figure 4.7; again, note the heavier tails of the predicted distribution. From the last two graphs we can informally state that our prediction

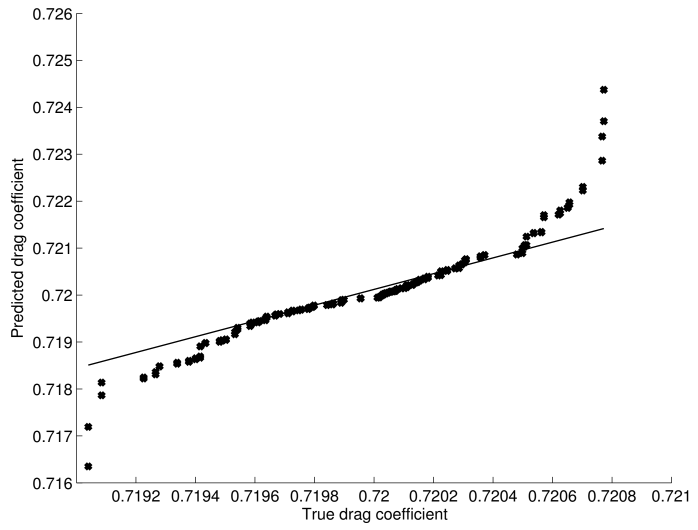

Figure 4.7. Nek5000: prediction for 20-dimensional uncertainty, quantile-quantile plot

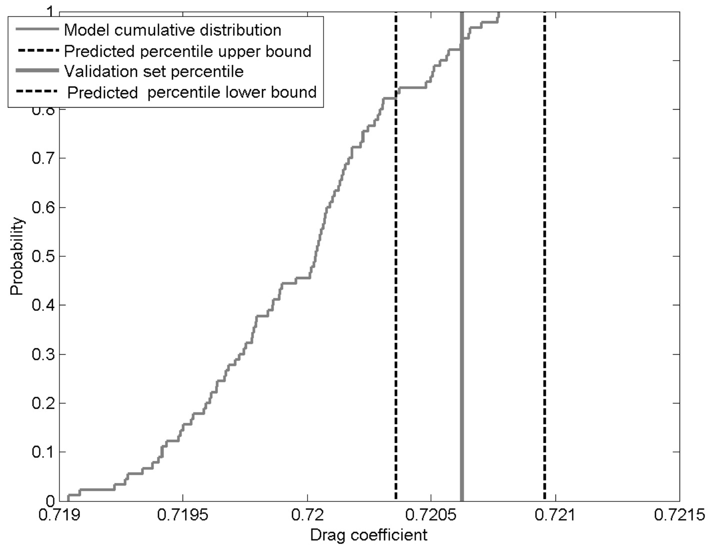

Figure 4.8. Nek5000: "95-95" estimate for the drag coefficient, 20-dimensional uncertainty, prediction after calibration process on 20 points

of global statistical properties of the model is essentially correct and conservative.

In Figure 4.8, we obtain the "95-95" estimate for the output with 20-dimensional uncertainty: our confidence interval is placed very 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. In any case 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 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 the sense that they cover at least the amount of data specified by the nominal coverage, and the excess is in the low percentages. So the methods of this work is succesfully validated on these examples using Nek5000 while using only a small number of calibration runs (comparable to m).

4.1 *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. With additional development, our work may become a first step in removing the constraints on uncertainty analysis related to prohibitive computational cost of sampling over high-fidelity spaces.

Based on our experiments, we give the following (informal) characterization of the method's performance. For a class of models considered, we constructed 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, sometimes even better quality using 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 outright 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 what would be another way to do this. 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.

5. Conclusions and future work

In our work, we have found that a Navier-Stokes models 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).

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 [3], but this would be good to understand a priori on given physical applications such as Navier-Stokes. The tools in [3] 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. Finally, there is 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 [18], and by reducing the uncertainty space by some form of component analysis. In terms of capability, we will aim 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 asuming 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, even if moderately succesful, this technique will help researchers to carry out uncertainty quantification for applications that need extreme-scale computing.

Acknowledgments

This work was supported in part by the U.S. Department of Energy under Contract No. DE-AC02-06CH11357. We thank our colleagues Paul Fischer, Elia Merzari, Alex Obabko and Zhu Wang for their assistance and input.

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*, Transactions of American Nuclear Society 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 Journal on Scientific Computing 34 (2012), pp. A240–A262, URL `http://epubs.siam.org/doi/abs/10.1137/110831143`.

[3] L.P. C. Homescu and R. Serban, *Error estimation for reduced order models of dynamical systems.*, Tech. rep., United States. Department of Energy., 2003.

[4] N. Cressie, *Spatial prediction and ordinary kriging*, Mathematical Geology 20(4) (1988), pp. 405–421.

[5] ———, *Statistics for spatial data*, Wiley series in probability and mathematical statistics. Applied probability and statistics section, John Wiley & Sons. New York. (1993).

[6] A. DasGupta, *Asymptotic theory of statistics and probability.*, Springer (2008).

[7] I.N.I. D.N. Daescu, *A dual-weighted approach to order reduction in 4dvar data assimilation*, Mon. Wea. Rev 136 (2008), p. 10261041.

[8] J.L. Doob, *Stochastic processes*, vol. 101, New York. (1953).

[9] J. Duderstadt and L. Hamilton, *Nuclear reactor analysis*, Wiley, New York (1976).

[10] .P.F. E. Merzari, W.D. Pointer., *A POD-based solver for the advection-diffusion equation*, in *ASME*, 2011.

[11] P. Fischer, *Nek5000-open source spectral element cfd solver*, https://nek5000. mcs. anl. gov/index. php/MainPage (2008).

[12] R.A.H. G. A. Webber and L. Sirovich, *The karhunenloeve decomposition of minimal channel flow*, Phys. Fluids (1997), pp. 1054–1066.

[13] M. Gunzburger, *Reduced-order modeling, data compression and the design of experiments.*, in *Second DOE Workshop of Multiscale Mathematics*, 2004.

[14] M. Hinze and S. Volkwein, *Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: error estimates and suboptimal control*, in *Lecture Notes in Computational Science and Engineering*, vol. 45, Springer, 2005.

[15] 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.

[16] ———, *Bayesian calibration of computer models*, Journal of the Royal Statistical Society. Series B, Statistical Methodology 63 (2001), pp. 425–464.

[17] C. Kenney and A. Laub, *Small-sample statistical condition estimates for general matrix functions*, SIAM Journal of Scientific Computing 15,1 (1994), pp. 36–61.

[18] B.A. Lockwood and M. Anitescu, *Gradient-enhanced universal kriging for uncertainty propagation.*, Nuclear Science and Engineering 170(2) (2012), pp. 168–195.

[19] C. Rasmussen and C. Williams, *Gaussian processes for machine learning*, MIT Press, Cambridge, Massachusets. (2006).

[20] O. Roderick, *Model reduction for simulation, optimization and control*, Ph.D. thesis, Portland State University (2009).

[21] O. Roderick, M. Anitescu, and P. Fischer, *Polynomial regression approaches using derivative information for uncertainty quantification*, Nuclear Science and Engineering (2010), to appear.

[22] M. Stein, *Interpolation of spatial data: Some theory for Kriging*, Springer-Verlag, Berlin (1999).

[23] S. Volkwein, *Proper orthogonal decomposition: applications in optimization and control*, CEA-EDF-INRIA Numerical Analysis Summer School (2007).

[24] .D.C. Y. Lacouture, *How to use matlab to fit the ex-gaussian and other probability functions to a distribution of response times*, Tutorials in Quantitative Methods for Psychology 4(1) (2008), pp. 35–45.