

Analysis of a Polynomial Chaos-Kriging Metamodel for Uncertainty Quantification in Aerospace Applications

Justin Weinmeister*, Nelson Xie†, and Xinfeng Gao‡

Computational Fluid Dynamics and Propulsion Laboratory

Colorado State University, Fort Collins, CO 80523, USA

Aditi Krishna Prasad§ and Sourajeet Roy¶

High Speed System Simulation Laboratory

Colorado State University, Fort Collins, CO 80523, USA

Metamodeling can be effective for uncertainty quantification in computational fluid dynamics simulations. In this research, we introduce modifications to our existing metamodel¹ that combines a reduced polynomial chaos expansion approach and universal Kriging (RPC-K) and evaluate the new metamodel for aerospace applications. Focus is given to determine which metamodel parameters most effect the solution accuracy by measuring the errors. Additionally, a new adaptive refinement algorithm is explored and the methodology is presented. Results show the metamodel’s need for robustness in aerospace engineering applications, including the non-smooth output of separated airflow.

I. Introduction

Design optimization studies which require uncertainty information from computational fluid dynamics (CFD) simulations can be limited by the computational cost of running a large number of simulations. Metamodeling, also known as surrogate modeling, reduces the computational cost of these studies by evaluating only a small subset of the proposed CFD simulations and fitting a computationally cheap model to them. This model can then be evaluated quickly to quantify uncertainty and further guide designs instead of simply running additional CFD simulations.² These metamodeling methods are popular in aerospace applications where optimization problems are very complex. For example, the methods can be used to evaluate nacelle designs for new high-bypass turbofan engines under extreme operating conditions such as takeoff in strong crosswinds. Our recent work¹ created a new metamodeling approach using a reduced polynomial chaos (RPC) expansion as the trend function for universal Kriging (UK)³ to evaluate such problems. Our approach, the reduced polynomial chaos-Kriging (RPC-K) metamodel, combines the global behavior of polynomial chaos methods with the local accuracy of Kriging methods on a reduced variable space to give excellent accuracy for its computational cost. When an unreduced variable space is used, such as in one dimensional problems, a standard polynomial chaos (PC) trend is used in the generic combined metamodel, polynomial chaos-Kriging (PC-K). For convenience, the difference in the behavior of the PC, Kriging, and PC-K methods is illustrated on an example one-dimensional problem in Fig. 1. As one can see, the PC-K method provides superior properties to either the PC or Kriging method.

The trend function of the RPC-K method uses the polynomial basis of the RPC method developed by Gao et al.⁴ The polynomial basis is constructed from polynomials which are orthonormal to the probability density function of the design variables. The number of terms in the polynomial basis is given in Eq. 1

$$p + 1 = \frac{(m + n)!}{m!n!}, \quad (1)$$

*Graduate Research Assistant, Email: jrwein@rams.colostate.edu, AIAA Student Member

†Research Associate, Email: nelson.xie@colostate.edu

‡Associate Professor, Email: xinfeng.gao@colostate.edu, AIAA Member

§Graduate Research Assistant, Email: aditikip@colostate.edu

¶Assistant Professor, Email: sjetroy@engr.colostate.edu

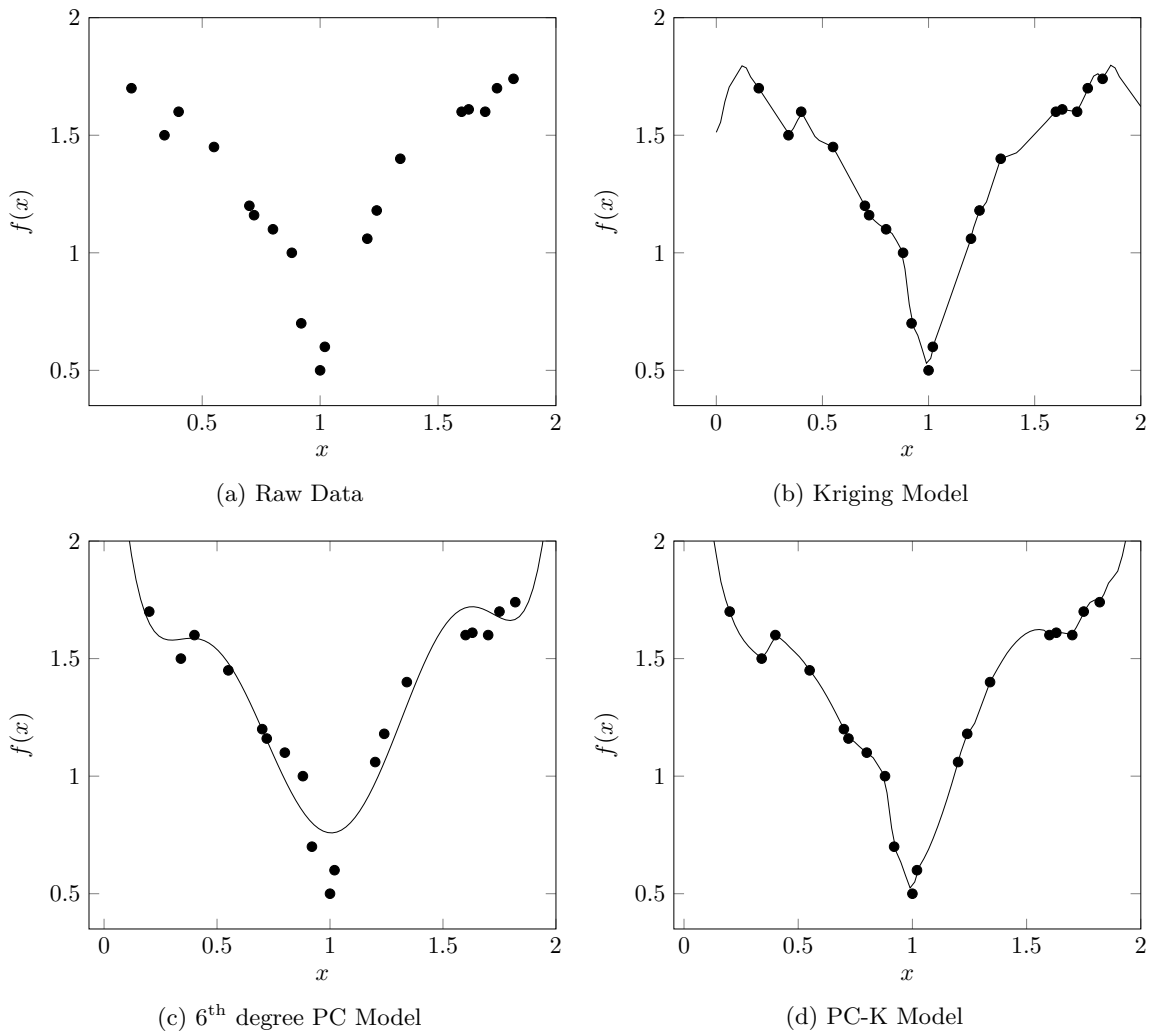


Figure 1: Comparison of PC, Kriging, PC-K methods in 1 Dimension

where n is the number of design variables and m is the maximum degree of the polynomials used. The polynomial basis is fit to the data using a least squares method which requires $p + 1$ simulations to find the coefficients of the polynomial. This expresses that the number of simulations required to use a PC method grows factorially depending on the degree of the polynomials used and the number of design variables. This growth is known as the curse of dimensionality, and it prohibits the analysis of designs with a large number of variables due to computational cost. The RPC method ameliorates this issue by removing the design variables which have little impact on the output variance. The low-impact variables are found by conducting a sensitivity study based on the first-order impact of each design variable. These low impact design variables are then removed from the uncertainty quantification study by setting them to their mean value. In the engine nacelle inlet study used in our previous works,^{1,4} we were able to reduce the number of design variables from 7 to 5. Accordingly, the number of CFD simulations is reduced from 120 to 56. These simulations define our deterministic design points. For larger design spaces and problems, this reduction is expected to be more significant. While the RPC method is promising for efficient uncertainty quantification in CFD, we observed it was insufficient for our non-smooth flow case.⁴ In general, PC-based metamodels fit global trends accurately but they fail to capture local variations. In non-smooth results, such as the onset of separated flow in the engine nacelle, these local variations can be important.

In order to increase the accuracy of the RPC metamodel for non-smooth flows, we combined the RPC metamodel as the trend for UK.³ Kriging methods are best linear unbiased predictors,⁵ and they exactly interpolate values in a design space based on the distance between two variables with known variance

information. Universal Kriging interpolates these values with a defined global trend for the data. In our case, this trend is the RPC metamodel. The RPC-K metamodel estimates off-design outputs by assuming a global trend constructed on the reduced variable space and adjusts estimates based on the deviation of nearby design points from the trend values. This trend data are critical to the metamodel since it reduces the uncertainty of the estimates where the sampling is sparse. This is of particular concern at the edge of the design space. When no trend data are provided, UK reduces to ordinary Kriging (OK) which assumes a constant, global mean. This Kriging behavior is illustrated in Fig. 1. Ordinary Kriging is commonly used today as it is difficult to determine reliable trend estimates. This can reduce the accuracy of Kriging.

In our previous work,¹ we found that the RPC-K metamodel provides more accurate estimates for non-smooth flows than the RPC metamodel. However, a few solutions indicated discontinuities and/or under-predicted values. From this, we have observed that the choice of autocorrelation functions, optimization functions, and other metamodel parameters play an important role in the solution accuracy and robustness. In the present work, we further investigate these parameters to gain a better understanding of the RPC-K method. In addition, we propose an adaptive algorithm for the RPC-K method. This adaptive algorithm can be used to refine the metamodel by running additional CFD simulations at locations of maximum variance or uncertainty. This process can be run iteratively until the global variance of the metamodel falls below a specified value.

We will begin with a description of how the RPC-K metamodel is constructed and how it fits into a full uncertainty quantification methodology. Also, the adaptive algorithm is presented. We then briefly cover the parameters of the RPC-K metamodel that impact the robustness of the algorithm. The method is then verified by a convection-diffusion-reaction equation which resembles the Navier-Stokes equations in terms of the important physics (convection, diffusion, and source). Finally, the results on an engine nacelle inlet design study are reported and conclusions are drawn.

II. Adaptive Reduced-Polynomial-Chaos-Kriging Method

The RPC-K metamodel approximates the solution variable, \mathbf{U} , as

$$\mathbf{U}(t, \vec{\omega}) \approx \mathbf{U}^{(RPC-K)}(t, \vec{\omega}) = \sum_{k=0}^p \phi_k(\vec{\omega}) \mathbf{U}_k(t) + \sigma^2(t) Z(t, \vec{\omega}), \quad (2)$$

where t is time and $\vec{\omega}$ are n mutually uncorrelated random variables, $\vec{\omega} = [\omega_1, \omega_2, \dots, \omega_n]^T$. The first term on the right-hand side of the equation is the trend function which is the sum of $p + 1$ polynomial basis functions, $\phi_k(\vec{\omega})$, multiplied by their coefficients, $\mathbf{U}_k(t)$. The second term on the right-hand side represents the variation at each design point as the product of the process variance, $\sigma^2(t)$, and a random process, $Z(t, \vec{\omega})$. This is a special form of the UK method⁵ where the trend is now defined by the RPC polynomial basis.

The multivariate polynomial basis, $\phi_k(\vec{\omega})$, is an orthonormal basis and constructed as the tensor product of the individual univariate polynomials by

$$\phi_k(\vec{\omega}) = \prod_{i=1}^n \phi_{k_i}^{(i)}(\omega_i). \quad (3)$$

The type of univariate polynomials used is determined by the probability density function (PDF) of the input variable. For a uniform distribution, this is the Legendre polynomials. For a normal distribution, the Hermite polynomials⁶ are the orthonormal basis. To keep this orthonormal basis, the input variables are converted to standard random variables which are centered at 0. For uniform PDFs, they take values from -1 to 1 as the limits of the variation whereas for normal variables each standard deviation is mapped to the integers. The reduced polynomial basis is found by conducting a sensitivity study and identifying which input variables are estimated to contribute little variance to the solution variable.⁴

The random process, $Z(t, \vec{\omega})$, is a zero-mean and unit-variance Gaussian process. It describes an estimated variation from the trend for the operating condition $\vec{\omega}$ at time t based on information from nearby data points. This deviation is scaled by the Kriging process variance, $\sigma^2(t)$. The dependency on time is dropped for the remainder of the paper to keep equations concise. The random process is described by an autocorrelation function $R(\vec{\omega} - \vec{\omega}')$ which is fit to the empirical variance values and is dependent only on the lag, or distance

between two design points.⁵ The autocorrelation function is fit by modifying the RPC-K metamodel's hyperparameters which depend on the specific autocorrelation function used.

Once the autocorrelation function is defined, the polynomial coefficients and process variance for the RPC-K metamodel are determined.⁷ Specifically, the coefficients are solved by

$$\mathbf{U}_k = (\mathbf{F}^T \mathbf{R}^{-1} \mathbf{F})^{-1} \mathbf{F}^T \mathbf{R}^{-1} \mathbf{U}, \quad (4)$$

and the process variance as

$$\sigma^2 = \frac{1}{N} (\mathbf{U} - \mathbf{F} \mathbf{U}_k)^T \mathbf{R}^{-1} (\mathbf{U} - \mathbf{F} \mathbf{U}_k). \quad (5)$$

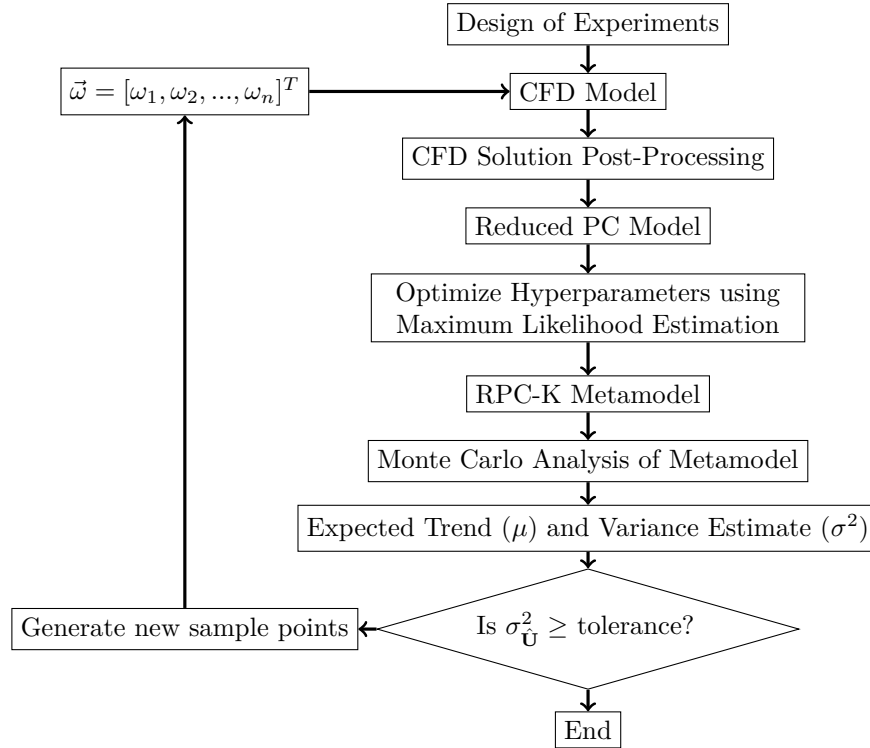
Here, \mathbf{F} is the matrix which contains the polynomial basis and \mathbf{R} is the correlation matrix which describes the correlation of all design points. It should be noted that these values are estimates of the true parameters for the metamodel based on the estimates of the hyperparameters. These are then used to estimate the mean and variance of the solution variable at off-design points where the mean is given as

$$\mu_{\hat{\mathbf{U}}}(\vec{\omega}) = \mathbf{f}(\vec{\omega})^T \mathbf{U}_k + \mathbf{r}(\vec{\omega})^T \mathbf{R}^{-1} (\mathbf{U} - \mathbf{F} \mathbf{U}_k), \quad (6)$$

and the variance is

$$\sigma_{\hat{\mathbf{U}}}^2(\vec{\omega}) = \sigma^2 \left(1 - [\mathbf{f}(\vec{\omega})^T \mathbf{r}(\vec{\omega})^T] \begin{bmatrix} \mathbf{0} & \mathbf{F}^T \\ \mathbf{F} & \mathbf{R} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{f}(\vec{\omega}) \\ \mathbf{r}(\vec{\omega}) \end{bmatrix} \right). \quad (7)$$

A full uncertainty quantification methodology using the RPC-K metamodel is shown in the included flowchart. In this methodology, the RPC-K metamodel can be adaptively refined based on the estimates of the variance in the model.



First, the design of the experiments must be determined in order to provide the best training data for the RPC-K metamodel. This experimental design is applied to the input variables considered and used in the CFD model. The modeled data can be primitive or derived solution variables. Then a sensitivity study is performed and the design variable space reduced. Afterwards, the number of simulations required to construct the RPC basis is known and we can construct a RPC-K metamodel based on the CFD data. The RPC-K metamodel can then be evaluated for a computationally cheap Monte Carlo analysis. This data provides both output mean and variation estimates. For example, 10,000 random trials are run for each

metamodel in this work. This value was determined by running an increasing number of simulations on the convection-diffusion-reaction verification problem. Convergence was found at 1000 simulations. Another order of magnitude of simulations are run for robustness. The estimates of the variance are then compared to our tolerance values and further CFD simulations run if necessary. This process repeats until the entire domain has a variance estimate below the specified tolerance. At this point, the solution is considered converged and the metamodeling process is complete.

II.A. Adaptive Algorithm

An adaptive algorithm was added to the RPC-K metamodel to improve its performance on under-sampled problems. We adapt the criterion of the adaptive method from Kawai and Shimoyama⁸ and apply it to the RPC-K metamodel. When it is evaluated across the design space, the maximum value of the criteria gives the location in the design space that reduces the error the most when added to the experimental design. The criterion is given as

$$\text{Crit}(\vec{\omega}) = \left(\left| \frac{\partial \hat{\mathbf{U}}(\vec{\omega})}{\partial \vec{\omega}} \right| \Delta\omega + D_{\hat{\mathbf{U}}}(\vec{\omega}) \right) \sigma_{\hat{\mathbf{U}}}^2(\vec{\omega}) \text{PDF}(\vec{\omega}). \quad (8)$$

Here $\frac{\partial \hat{\mathbf{U}}(\vec{\omega})}{\partial \vec{\omega}} = \left[\frac{\partial \hat{\mathbf{U}}(\vec{\omega})}{\partial \omega_1}, \frac{\partial \hat{\mathbf{U}}(\vec{\omega})}{\partial \omega_2}, \dots, \frac{\partial \hat{\mathbf{U}}(\vec{\omega})}{\partial \omega_n} \right]^T$ is the vector of gradients of the predicted mean $\hat{\mathbf{U}}(\vec{\omega})$ with respect to the input variables $\vec{\omega}$. It can be derived analytically as

$$\frac{\partial \hat{\mathbf{U}}(\vec{\omega})}{\partial \vec{\omega}} = \left(\frac{\partial \mathbf{r}(\vec{\omega})}{\partial \vec{\omega}} \right)^T \mathbf{R}(\boldsymbol{\theta})^{-1} (\mathbf{U} - \mathbf{F}\mathbf{U}_k) \quad (9)$$

where $\frac{\partial \mathbf{r}(\vec{\omega})}{\partial \vec{\omega}}$ is an $N \times n$ matrix with (i, j) entries

$$\left[\frac{\partial \mathbf{r}(\vec{\omega})}{\partial \vec{\omega}} \right]_{i,j} = R'(\omega_j - \omega_j^i) \times \prod_{\substack{k=1 \\ k \neq j}}^n R(\omega_k - \omega_k^i), \quad (10)$$

and N is the number of design points. It is multiplied by $\Delta\omega = \min_{i=1,2,\dots,n} |\vec{\omega} - \vec{\omega}^i|$ to keep the units consistent with $D_{\hat{\mathbf{U}}}(\vec{\omega})$. The term $D_{\hat{\mathbf{U}}}(\vec{\omega})$ is defined as

$$D_{\hat{\mathbf{U}}}(\vec{\omega}) = |\hat{\mathbf{U}}(\vec{\omega}) - \hat{\mathbf{U}}_{\text{pre}}(\vec{\omega})| \quad (11)$$

which is the difference in the estimate at the design point $\vec{\omega}$ of the current sample set and the previous set of $N - 1$ points. For the first iteration, $D_{\hat{\mathbf{U}}}(\vec{\omega})$ is set to 0. The value $\sigma_{\hat{\mathbf{U}}}^2$ is taken from Eq. (5) and $\text{PDF}(\vec{\omega})$ is taken from the probability density functions of the uncertainties in the input variables.

This adaptive algorithm was found to work well for Kriging⁸ methods as it considers the gradients of the solution and the local uncertainty due to sampling density. This gradient and uncertainty based strategy works well as high gradient regions are generally fit accurately after only a few evaluations and the global error of the metamodel can be more quickly reduced by sampling sparsely evaluated regions. However, it is challenging for the adaptive algorithm to address multiple output variables, times, and locations in the problem domain as they are fit with different metamodels. Running the adaptive algorithm for separate output measures will yield multiple new sampling locations, and consequently, a separate scheme is needed to combine the multiple adaptive measures. For the present study, we simply use the mode of the criteria for all evaluation points and metamodels. The algorithm is evaluated in our verification section and will be used on the engine nacelle case in our upcoming work.

II.B. Metamodel Parameters

The RPC-K metamodel can be constructed differently depending on the experimental design, the autocorrelation function, the optimization function, and the minimization function. These parameters are important to evaluate as they can greatly effect the quality of the metamodel.

The experimental design of the study is perhaps the most important consideration when constructing the metamodel. Specifically, when running an uncertainty quantification study which uses metamodels, the

design must allow the metamodel to fit the data easily. Possible methods investigated include Gaussian quadrature sampling, Latin hypercube sampling (LHS), quasi-random sequences, and K-means sampling.^{3,9} Gaussian quadrature sampling was used in our original study¹ as it provides the optimal evaluation for polynomial chaos expansions; however, the regular spacing of the data points produced difficulty in fitting the autocorrelation function.¹⁰ K-means sampling provides reasonable results, though we found LHS and quasi-random Sobol sequences both provide better results for our cases.

The autocorrelation function used within the Kriging process is important because it ultimately determines the relationship between design points. For in depth information on these functions, please consult a complete resource such as Webster and Oliver’s book.⁵ We tested the bounded linear, cubic, exponential, Gaussian, and Matérn models on our data and found that all provide sufficient, though varying results. This was a result of the hyperparameters for each model being fit with differing correlation lengths.

The hyperparameters of a Kriging metamodel, or the parameters of the autocorrelation function, must be correctly estimated from the data to provide reliable results. Two optimization methods are commonly used, maximum likelihood estimation and leave-one-out cross validation. Both methods are covered in detail by existing literature.^{6,7,9,11,12} We found maximum likelihood estimation to be the most robust method for our cases, though cross validation out-performed it in a few trials.

The maximum likelihood estimation and leave-one-out cross validation methods can be maximized using a variety of algorithms. This is achieved by minimizing the reciprocal of the optimization method. We evaluated multiple methods that already exist as MATLAB[®] built-in functions to evaluate their fitness. There are two general classifications, global and local optimization methods. The local optimization algorithms *fminsearch* and *fmincon* and the global optimization algorithms *patternsearch*, *particleswarm*, *globalsearch*, *simulannealbnd*, and *ga* (genetic algorithm) were tested. It was found in the Nacelle cases that local methods were often sufficient to get the desired accuracy of the metamodel. It was also found that two methods, *fminsearch* and *particleswarm*, often failed due to nearly singular matrices. Kriging frequently produces ill-conditioned matrices, so in these cases, we simply considered other minimization functions.

III. Verification

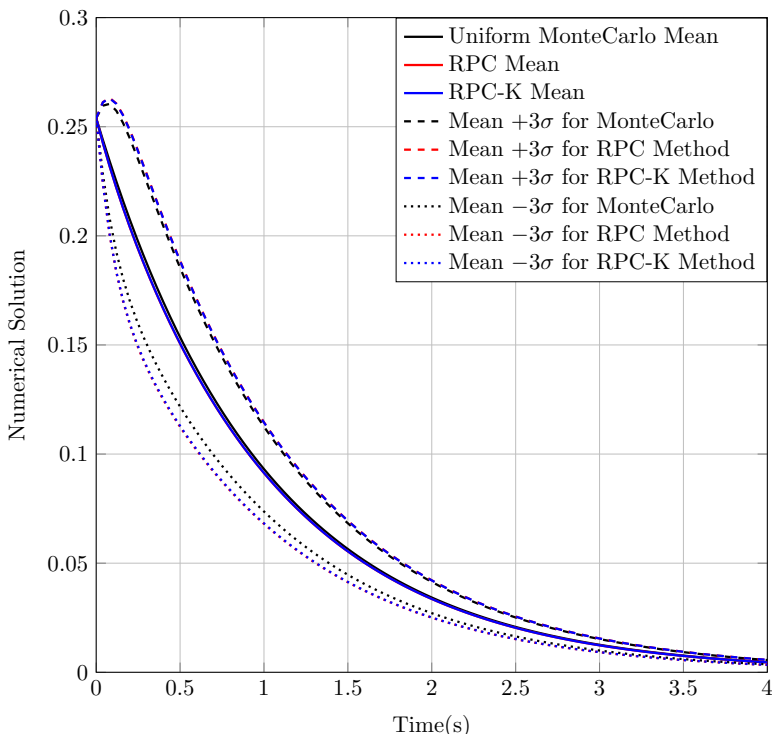


Figure 2: Predicted mean and 3σ standard deviations for RPC and RPC-K methods compared to Monte Carlo data under uniform uncertainty.

The adaptive RPC-K method was verified on a one-dimensional transient convection-diffusion-reaction problem. This same problem was used in our previous studies^{1,4} to verify the RPC and RPC-K methods. A brief description of the problem is provided below for convenience. Full details can be found in the original study.⁴ The problem is represented in a unit domain of $0 \leq x \leq 1$ as

$$\frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial x} - k \frac{\partial^2 \phi}{\partial x^2} + c\phi = f, \quad (12)$$

where ϕ is the solution variable and $u = 1$, $k = 1$, and $c = 1$ are the convection speed, diffusivity, and source coefficient, respectively. The source term is modeled as $f = \alpha(x - 1)e^{-t}$ with $\alpha = 2$. The true mean and uncertainty of the problem were taken as the result of a 10,000 point Monte Carlo sample of CFD simulations. The methods were then evaluated on a small set of simulations to the problem. The CFD model is based on a fourth-order finite-volume method.¹³

We first evaluated this problem using uniform probability density functions with 10% variation for each input variable. In our previous work,⁴ we found that c and α could be removed from the input variable space with only minor losses in accuracy. This was achieved by setting c and α as model parameters at their mean values. The RPC and RPC-K metamodels were constructed using 3rd degree Legendre polynomials on a set of 16 simulations. The RPC-K metamodel also used the cubic autocorrelation function and the hyperparameters were fit using maximum likelihood estimation and the *fmincon* optimization algorithm.

Figure 2 shows that both the RPC and RPC-K metamodel give nearly identical results which have only slightly greater standard deviations than the Monte Carlo data. These good results are expected as the CDR problem has a smooth solution, so we expect the RPC metamodel to accurately predict the data without the added Kriging information. However, adding this information does not degrade the solution.

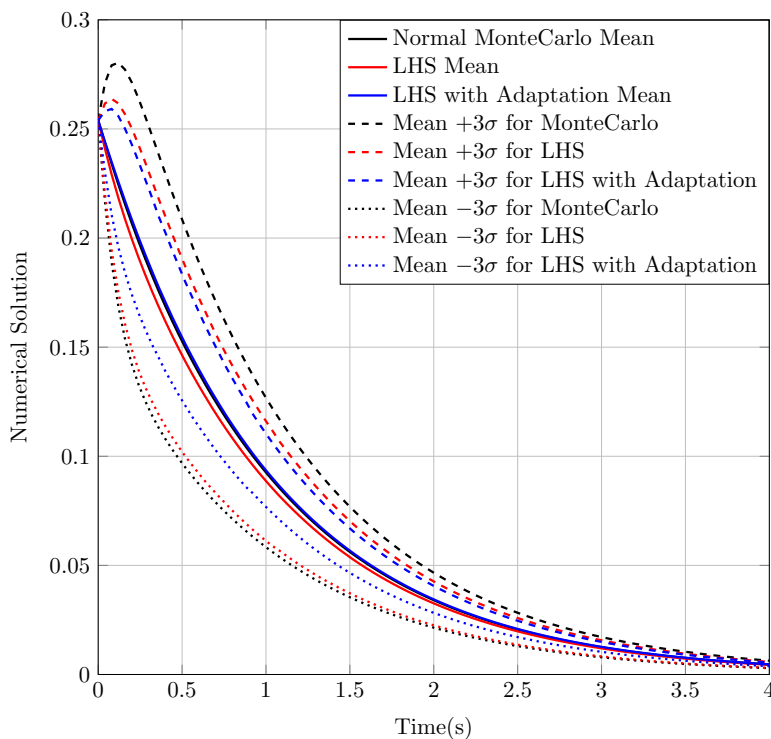


Figure 3: Predicted mean and 3σ standard deviations for RPC-K metamodel and adaptive RPC-K metamodels under normally distributed uncertainty.

We then repeated this experiment using normally distributed variables with standard deviations of 10% to test our adaptive algorithm. These experiments were run with the full four design variables and would require 70 simulations to evaluate the full third order polynomial basis. Instead, first order expansions were used which required only five CFD simulations. Fitting normal distributions presents many challenges that uniform distributions do not and less literature is available for combined PC-K metamodels fitting such data. We believed the adaptive algorithm could prove useful in this use because of the lack of data.

Our baseline measurement uses ten Latin hypercube samples, the exponential autocorrelation function, maximum likelihood estimation, and *fmincon*. Ten design points were used to prevent over-fitting. Our adaptive algorithm started with five Latin hypercube samples to meet the minimum number of sampling points required for the first order PC expansion. The RPC-K metamodel was then constructed and the adaptive criteria evaluated to select the design variables for the next simulation. This process was repeated until ten design points were evaluated.

We see in the results that neither model correctly captures the variance in the problem, though the RPC-K metamodel constructed from ten Latin hypercube samples better predicts the standard deviation. However, this metamodel does not correctly estimate the mean values whereas our adaptive RPC-K metamodel does. Our adaptive RPC-K metamodel under-predicts the variation in the problem as is later observed in Fig. 11. We believe this is the result of a different kind of over-fitting, over-fitting of the mean, which is further discussed in the results. The average percent error in the mean for the RPC-K metamodel is 4.11% and it is 8.35% for the standard deviation. For the adaptive RPC-K metamodel the error in the mean is 1.07% and 12.42% for the standard deviation.

IV. Results

IV.A. Problem Setup

We evaluated the model in the present study on an engine nacelle inlet design case from our previous works.^{1,4} This case shows the performance of the metamodel on a realistic CFD simulation with non-smooth flow behavior. The CFD simulations were solved using the CFD++ solver from Metacomp Technologies.¹⁴ The solver is a second-order, upwind, finite-volume method. The governing equations are the Reynolds-Averaged Navier Stokes (RANS) equations where the gas is assumed calorically perfect. The solution was solved for steady-state conditions using the Gauss-Seidel relaxation method accelerated with an algebraic multigrid. Turbulence was modeled using the Spalart-Allmaras turbulence model as in our previous works.

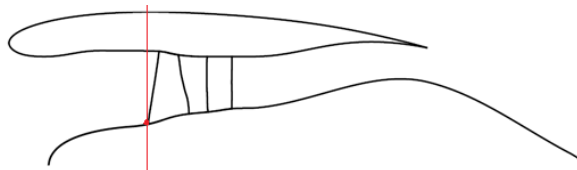


Figure 4: Hub station where data was collected.

The axisymmetric schematic of the engine nacelle inlet modeled for this design problem is shown in Fig. 4 which is the same geometry as in the previous studies. The rotor and stator blades shown in Fig. 4 are not modeled to save computational cost and are solely illustrated to show the region of engineering interest. The geometry has been meshed as an unstructured grid generated using Pointwise®'s T-Rex automated mesh generation tool. The five modeled design variables are the freestream Mach number, the angle of attack, the mass flow rate, the diameter of the nacelle, and the length of the nacelle. These variables were modeled with 15% uniform variation. More detailed information on boundary conditions and other parameters can be found in the original study.⁴

Data is collected at the hub station which is shown by the red line. This line is perpendicular to the nacelle axis at the location where the leading edge of the rotor blade would intercept the hub, represented by the red dot. For our simulated uncertainty quantification study, the total pressure and total temperature along three profiles (-Y,+Y,-Z) at the hub station are evaluated. To be concise, the total temperature is not shown as results are similar to the total pressure. Further, results along the +Y and -Z profiles are not shown as their smooth profiles do not show significant deviations among the metamodels.¹ The uncertainty quantification study aims to accurately identify the effect of flow separation on the engine's operating conditions.

IV.B. Results and Discussion

First, we construct the RPC metamodel based on standard random variables. This conversion is calculated by

$$\vec{\omega}_{\text{Standard}} = (\vec{\omega} - \bar{\vec{\omega}})/\sigma. \quad (13)$$

However, in our previous results,¹ we used a non-optimal polynomial expansion for the RPC-K metamodel. Specifically, the design variables were not converted to standard random variables as in Eq. 13 prior to constructing the RPC model. This results in a discrepancy in the two solutions. To fully examine the discrepancy, we compare the RPC-K metamodel’s predicted mean values for the normalized total pressure on the -Y profile from the previous results (5a) to the present study (5b). The mean of the 56 simulations is given in light grey, the mean of the OK metamodel in black, the mean of the RPC metamodel in blue, and the mean of the RPC-K metamodel is in red. The x -axis is the normalized location where 0 is at the hub wall and 1 is at the nacelle wall. On the y -axis, the total pressure values are normalized with respect to the freestream conditions. The results of the RPC-K metamodel based on the standard random variables are

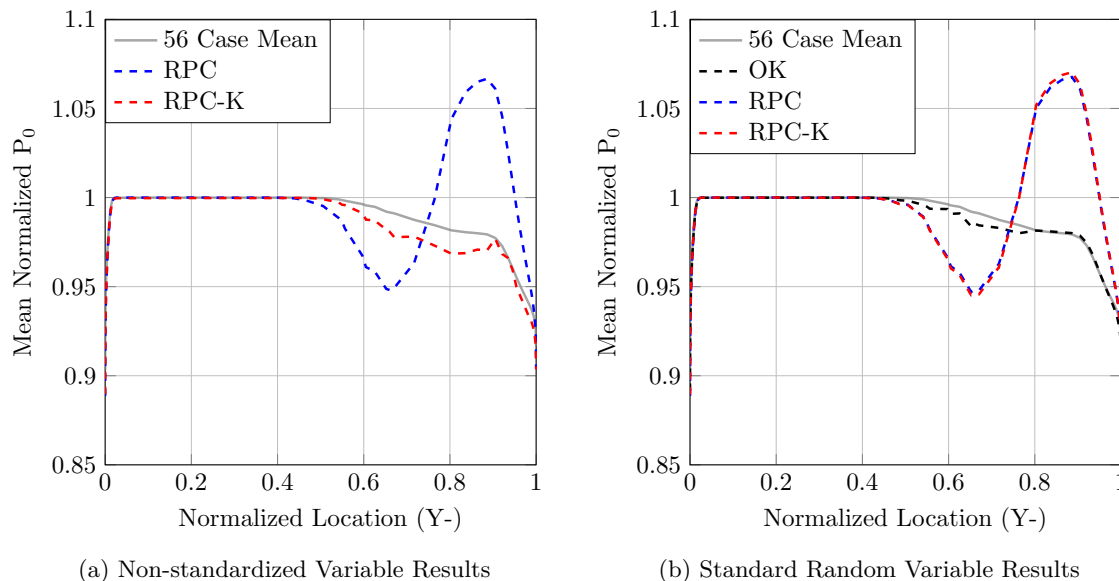


Figure 5: Comparison of non-standardized and standard variable results.

quite poor in comparison to the previous work. Indeed, this occurs because the RPC-K metamodel simply recovers the RPC metamodel which does not accurately describe the data. As mentioned in our previous work,¹ the RPC metamodel produces incorrect pressure values as total pressure can not exceed a value of 1 in this region of our model. At first, it would seem that improving the fit of the RPC metamodel should only improve the solution quality, as the combined PC-K method has previously been found to provide a solution of at least the best of the two component metamodels.^{3,6,15} However, this is not the case when the RPC metamodel over-fits the data. Analyzing Eq. 6, when the data are over-fit, the value of the term $(\mathbf{U} - \mathbf{F}\mathbf{U}_k)$ is nearly 0. Consequently, the entire second term is negligible and the solution value is solely approximated by the RPC metamodel, or the trend component, alone. This over-fitting introduces large errors into the solution since the accuracy of the RPC metamodel diverges rapidly away from design points. When we run the Monte Carlo simulation on the metamodel, the points calculated on the diverged portion ruin the statistical data. This behavior has previously been discussed with regards to PC-K,⁶ though this behavior was analyzed using leave-one-out error estimation, not the physical realization of the data. The over-fitting is believed to result from having the number of simulations equal to the number of polynomial basis terms, leading to perfect regression. Without running more CFD simulations, a method is needed to combat this over-fitting.

The best method for improving the quality of the results is not simply reverting to an OK solution because long-range trend data improves the quality of Kriging solutions for data with significant trends. Identifying and preventing over-fitting is not straight forward, especially given the goal of our approach is to accurately quantify nonlinear data. Several established methods exist,^{16,17} but they are not considered in this work due to notable drawbacks.

1. For example, large oversampling of the problem combined with regression techniques alleviate over-fitting, though this requires either a greater number of simulations or a reduced-order polynomial basis. This negates many of the benefits of such a surrogate model. Previous work has addressed this issue

by iteratively finding the next polynomial basis term which will improve accuracy the most. This least angle regression (LARS) algorithm, originally applied to PC methods,¹⁷ has been used previously in PC-K metamodels.^{6, 15, 18}

2. The problem can be evaluated at the extreme conditions to prevent the tails of the polynomial expansion from lying within the considered domain. This method does not fully work with normal probability density functions and suffers from the curse of dimensionality. Additionally, over-fitting in the interior is still possible.
3. Approaches exist which identify regions of large oscillations in the data that do not contain data points and smooth the solution to linear trends at these locations.¹⁶ This method is not considered as our over-fitting occurs primarily outside of all known data points.

Because the above methods are not compatible with our current research goals, we have devised and experimented with additional methods to combat over-fitting. They are described as follows.

1. Directly modify Eq. 6 by adding perturbations to the predicted mean values so as to increase the influence of the $(\mathbf{U} - \mathbf{F}\mathbf{U}_k)$ term in Eq. 6. This method is implemented by adding random perturbations to the predicted values of up to 5% of the mean. This perturbation method presents issues because it directly modifies the data values without particular criteria on perturbations.
2. Recover the mean of the CFD data if the value predicted from the RPC trend data differs too much from the CFD data's mean. This method slowly reduces the RPC-K metamodel to an OK metamodel when the metamodel is over-fit. This transition is linear and starts with 100% of the RPC trend when the mean of the CFD simulation data and RPC metamodel match. When the means differ by more than 10%, the trend is reduced to the mean of the CFD simulation data. We call this method the trend adjustment method as it should remove the effects of over-fitting when they exist. Two versions are tested, one which applies this adjustment only to the trend value and another which applies it to the trend and predicted value. This method does lose trend data when the correction is applied.
3. Reduce the order of the polynomial chaos expansion at locations where we identify over-fitting. This is implemented by taking the sum of the errors of the metamodel and determining if they are close to zero. This occurs when the metamodel well represents all data points. Then, if the metamodel also predicts a mean value that deviates more than 5% from the average of the data points, the order of the polynomial expansion is reduced under the assumption it suffers from over-fitting. This reduction removes polynomial terms and reduces the likelihood that the data can be over-fit. This method does lose the ability to fit more complicated trends and relies on imperfect identification of over-fitting. This approach is called the multiple fidelity polynomial approach since different locations use different orders of polynomials as the trend function.
4. Bound output values using prior knowledge. In this case, we limit any total pressure or total temperature values to a maximum of 1. This method uses problem-specific information to place bounds on reasonable values for the metamodel and can be generalized to any problem where known minimum and maximum values exist. Disadvantages to the limiting method come from the lack of automation in creating the metamodel and the inability to apply these limits to all cases.
5. Employ a non-optimal polynomial basis. This polynomial basis will retain the higher-order terms which provide the opportunity for more detailed trend functions. However, it is not precise enough to over-fit the data. An example of a non-optimal polynomial basis is to use the true values of the variables instead of standard random variables. As many forms of polynomials exist, alternative implementations of a non-optimal method are nearly infinite. This method can not recover a true polynomial chaos expansions which have greater accuracy than arbitrary polynomial expansions.
6. Run the RPC-K metamodel using the initial polynomials from the evaluation of the sensitivity indices. These polynomials include the mean and main effects up to third order with no mixed effects. This method requires no further evaluations and alleviates over-fitting issues by reducing the accuracy of the polynomial. This method loses accuracy if any mixed effects contribute significantly to the problem.

These methods are all evaluated on the $-Y$ profile of the total pressure as it shows the greatest error in the mean of all the profiles tested. In addition to mean values, the standard deviations of all the methods are also shown. The legend used in Fig. 5 applies to the rest of the figures in this section. Our results show that all methods other than the perturbation method provide improvement, though using non-optimal bases and lower order polynomials seems to be most effective.

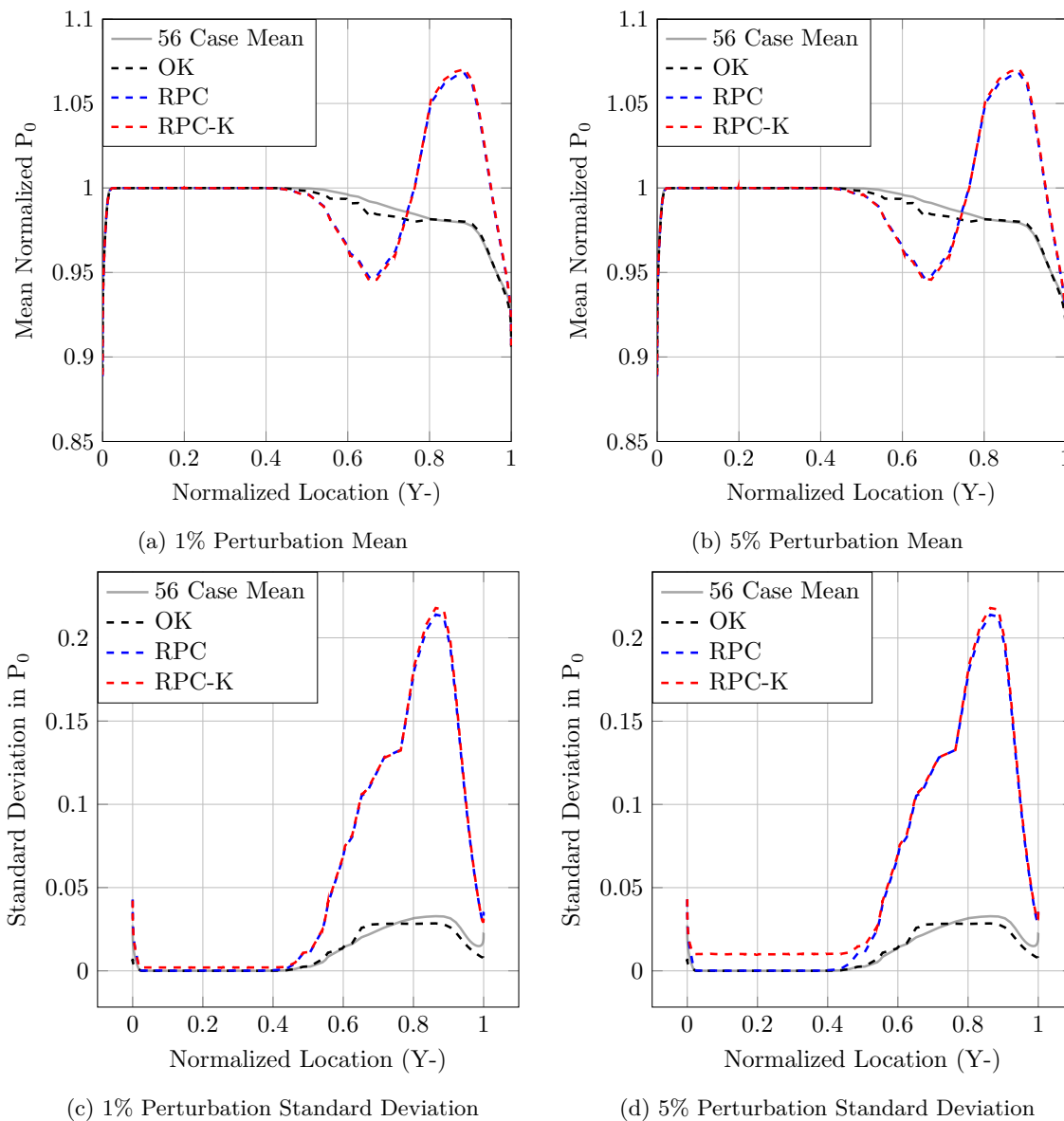


Figure 6: Comparison of over-fitting solutions using perturbations in the prediction.

Figure 6 shows the results of fitting the profile when either 1% or 5% random perturbations are added to the predicted values of the known data points. These perturbations do not increase the accuracy of the solution and they artificially increase the baseline variation in the data. As this method is the most likely of any direct modification method to work, it shows that any direct modifications of the right-hand side of Eq. 6 will not improve the fit of the RPC-K metamodel without losses in accuracy.

Figure 7 shows the results of fitting the profile when either the trend value or the trend and prediction values are slowly adjusted back to mean values. The results show this method is effective for limiting the large peak centered around 0.9, though it does not sufficiently improve the solution accuracy. Both methods provide nearly identical results which still feature a large dip in total pressure values near 0.7 and feature a small, non-physical, double peak in the neighborhood of 0.9. Additionally, both methods predict mean

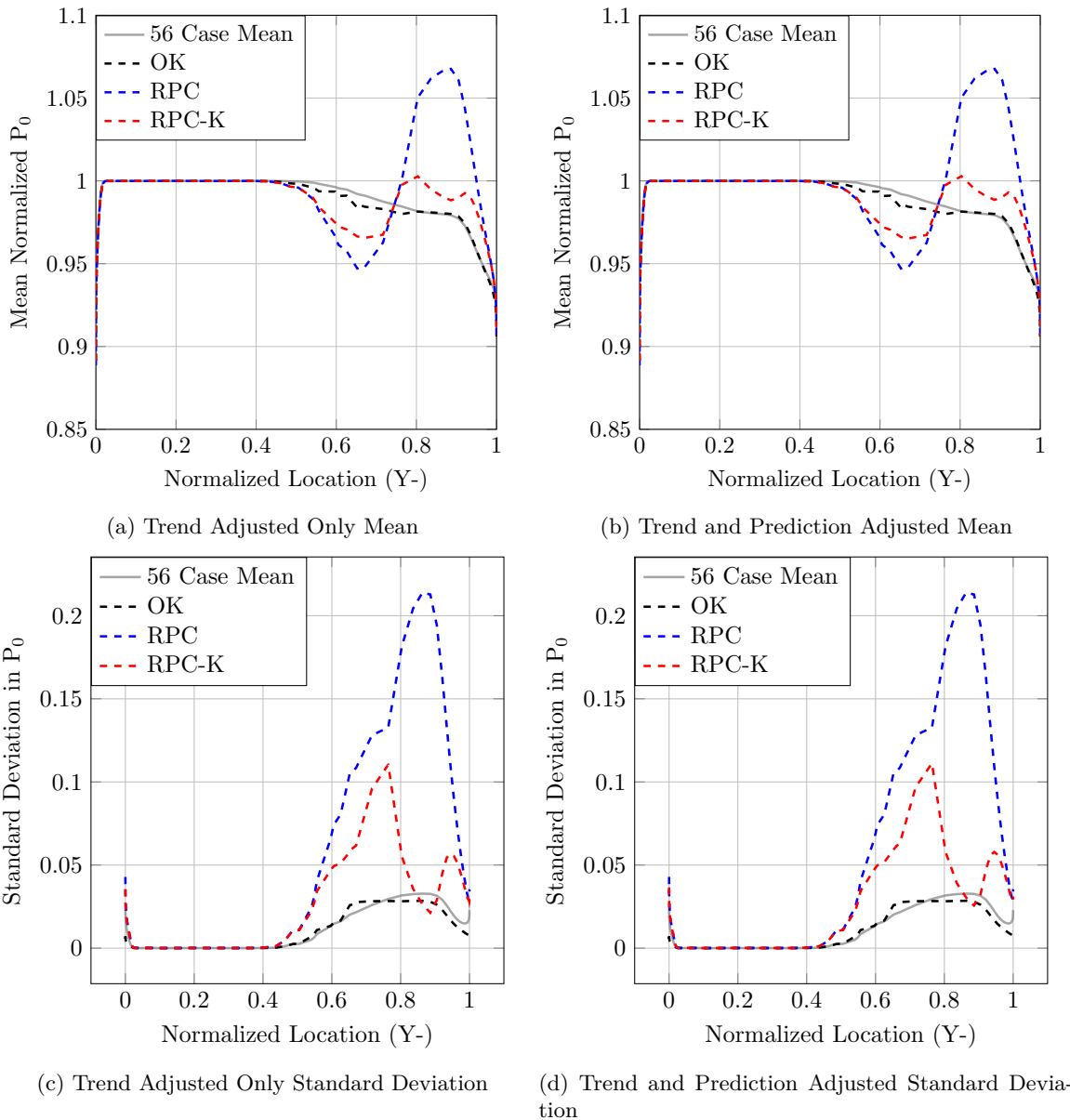


Figure 7: Comparison of over-fitting solutions using adjusted trend values.

operating values slightly greater than 1 at the 0.8 location. The results from these methods, along with the previous perturbation method, indicate that the polynomial basis must be adjusted to satisfactorily improve over-fit results.

Figure 8 shows the results for reducing the order of the polynomials used in the expansion. Both 1st and 2nd order polynomials are used. The 1st order polynomials provide great results with both a relatively smooth profile and values that nearly recover the mean of the 56 cases. Additionally, the standard deviations are similar to both the 56 cases and the OK results. The 2nd order polynomials improve the solution, though additional unsatisfactory features are present. The predicted mean values are lower than the 56 case mean and OK mean. They also feature a few sharp dips which produce less easily interpreted data. Finally, the standard deviation values predicted using 2nd order polynomials are higher than either the 56 cases or the OK solution.

Figure 9 shows the results for using multiple fidelity polynomials. These results closely mirror the results of using global 1st order polynomials since the solution is reduced to 1st order in each region with large deviations. While the method still uses 3rd order polynomials in the smooth region of the solution, any

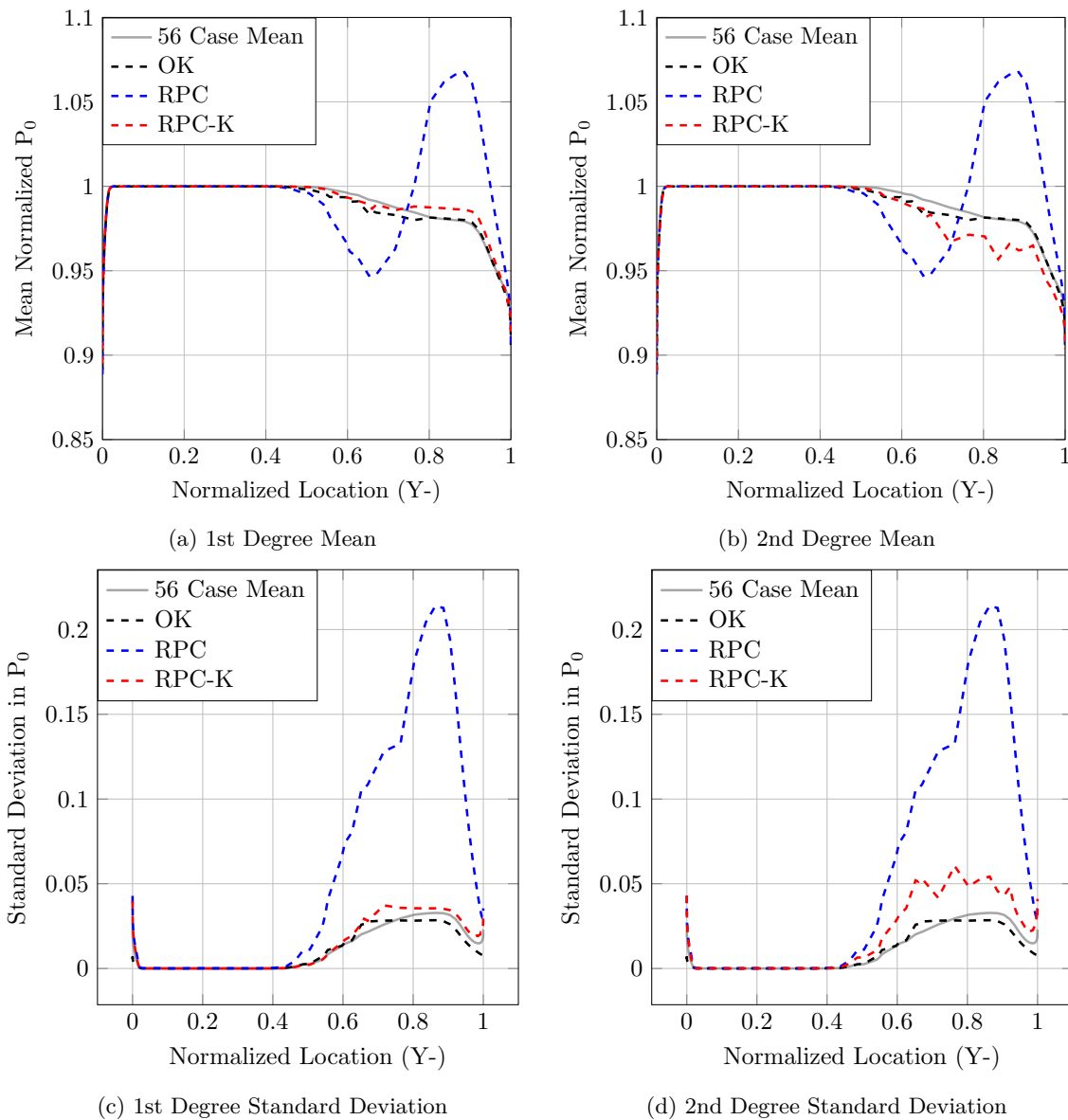
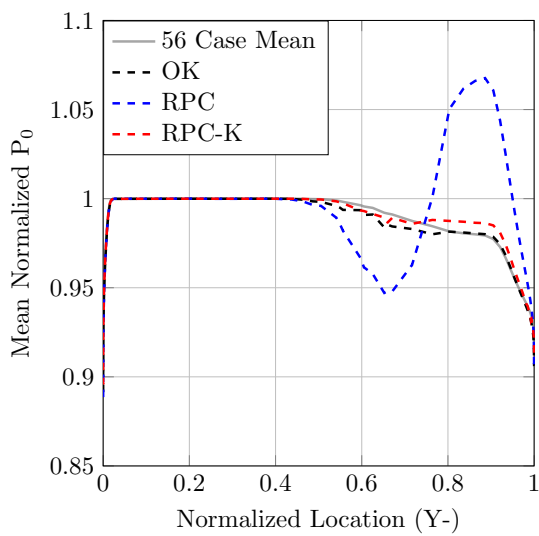


Figure 8: Comparison of over-fitting solutions using lower order polynomials.

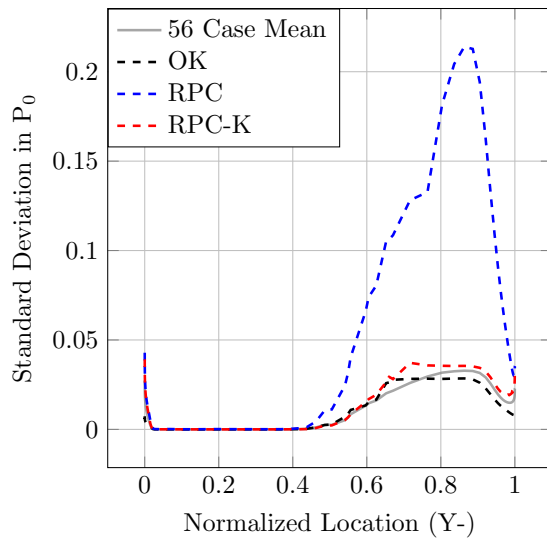
improvement in the quality of the solution in this region is minor. This method is reliable and produces an improvement over using lower order polynomials globally. Nevertheless, it does not benefit from more accurate trend prediction in the region of interest.

Figure 10 shows the results for using the first 21 coefficients calculated during the sensitivity study which do not include mixed effects. The results are an improvement over the full system, though still insufficient. Notably, there are not enough terms to over-fit the solution and cause non-physical values. It is believed that the oscillations are still present in the solution because the 3rd order main effect polynomials can still contribute significant error to the solution. Additionally, the solution under-predicts in the region of interest. This method could benefit from a combination with the LARS algorithm used by Schöbi et al³ to select the correct polynomial terms in this sparse expansion.

Figure 11 plots the results of using a non-optimal polynomial basis. In this case, using true variable values. The predicted mean from the solution nearly matches that of the 56 cases, but the standard deviations are under-predicted in the region of interest. This method is very promising as the polynomial trend still retains all possible terms to allow for fitting of more complex relations among the variables and the method better recovers the mean than OK. This method avoids over-fitting due to its non-optimality. As discussed in our

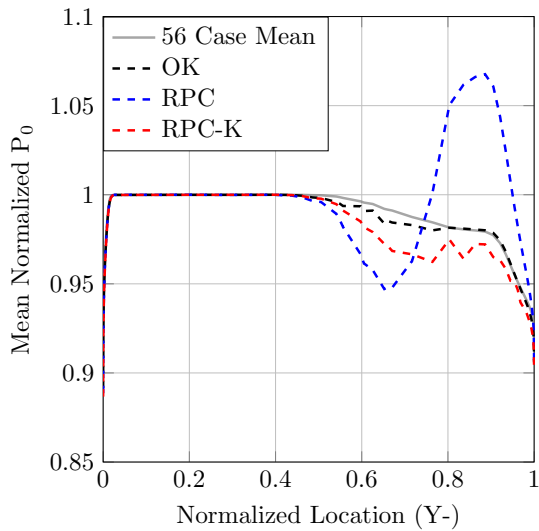


(a) Multiple Fidelity Polynomial Mean

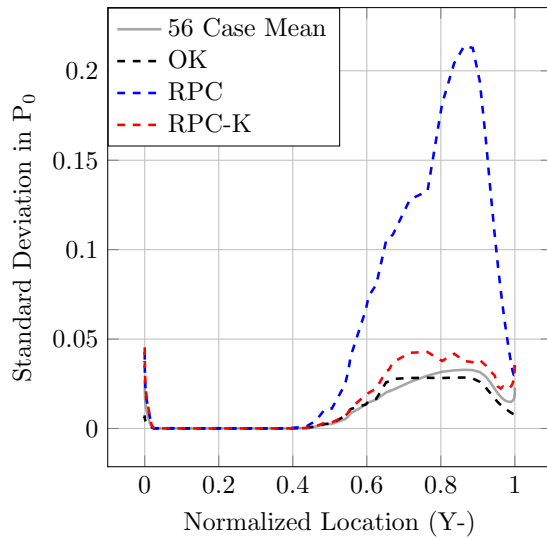


(b) Multiple Fidelity Polynomial Standard Deviation

Figure 9: Comparison of over-fitting solutions using variable order polynomials.



(a) First 21 Coefficients Mean



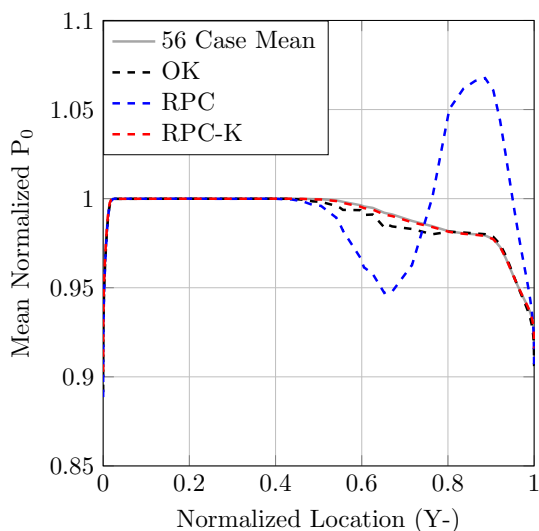
(b) First 21 Coefficients Standard Deviation

Figure 10: Comparison of over-fitting solutions using first 21 coefficients.

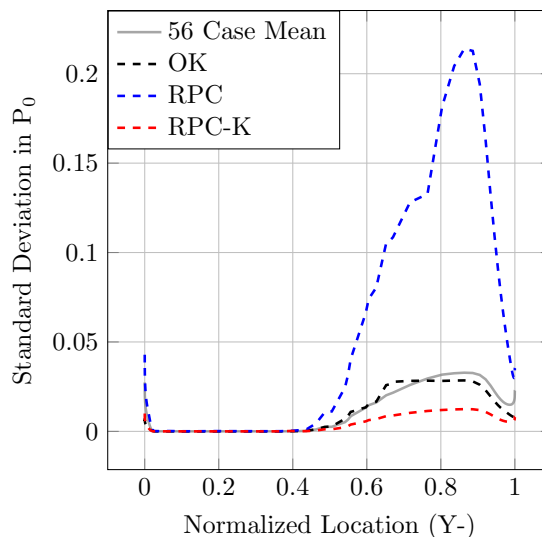
verification work, we believe the low predicted standard deviation is the result of over-fitting of the mean. This would result from a smooth polynomial function which masks the Kriging data as the true polynomial chaos expansion masked the mean values. It may be possible to recover some of the standard deviation data from the Kriging components of the equation.

Figure 12 shows the results of limiting output values to a maximum of 1. This method slightly improves results. The mean predicted in the region of interest is low and the standard deviations greater than expected. Additionally, the trends have the same features as the standard RPC-K method including a large dip near 0.7 and recovery to a “peak” near 0.9. These drawbacks are similar to previous methods which did not directly modify the polynomial trend. Regardless, placing upper and lower bounds on problems may provide more robust solutions in general.

All of the above methods were evaluated using the initial Gaussian quadrature sampling with various autocorrelation functions, optimization functions, and minimization functions. Their behavior followed that

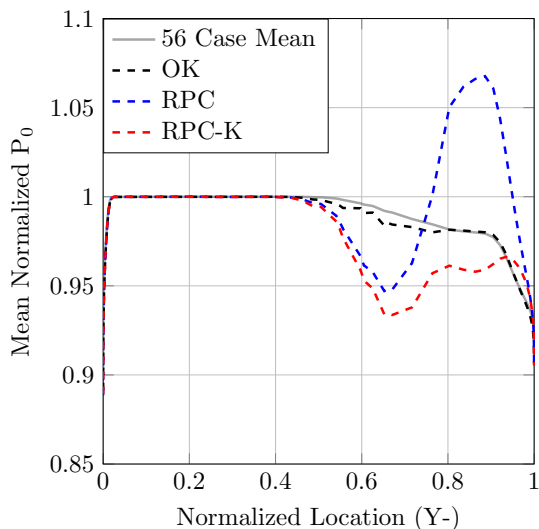


(a) True Variable Values Mean

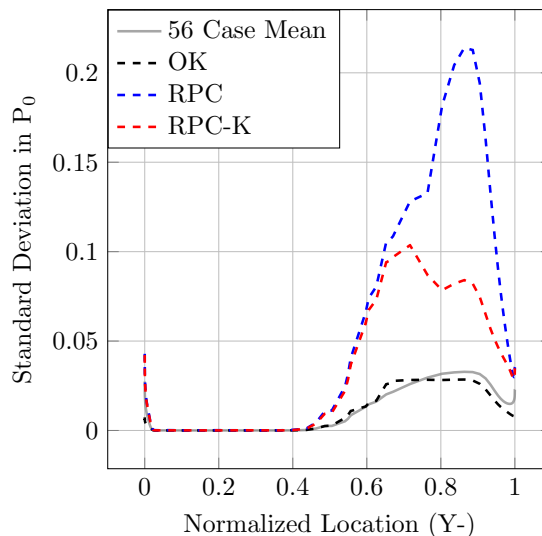


(b) True Variable Values Standard Deviation

Figure 11: Comparison of over-fitting solutions using true variable values.



(a) Maximum Limited Mean



(b) Maximum Limited Standard Deviation

Figure 12: Comparison of over-fitting solutions using maximum value of 1.

of the convection-diffusion-reaction problem covered in the verification section where their minor differences were insignificant to the over-fitting issues.

V. Concluding Remarks and Future Work

We have identified the main source of fragility in the RPC-K model as over-fitting of the polynomial trend function. When this occurs, combined PC-K methods can provide erroneous results because Kriging has negligible effect in the optimization process. Of the several methods we have proposed and experimented on, only those which directly modified the RPC expansion produced satisfactory results. In general, the experimental design of the data is critical to producing an accurate metamodel, especially under normally distributed uncertainty. In addition, the RPC-K metamodel is very sensitive to the choice of the autocorrelation function and minimization function. Nevertheless, the RPC-K metamodel can still provide significantly

better results than the RPC metamodel and can improve upon OK in this aerospace engineering application given careful selection of the above factors. Additionally, we introduced an adaptive algorithm to the RPC-K metamodel which can select more optimal design points to evaluate than standard experimental design approaches. Immediately, a follow-up work will focus on the application of the adaptive algorithm and its impact in combating over-fitting the polynomial basis and over-fitting of the mean.

References

- ¹Weinmeister, J., Xie, N., Gao, X., Prasad, A. K., and Roy, S., “Combining a Reduced Polynomial Chaos Expansion Approach with Universal Kriging for Uncertainty Quantification,” AIAA, Denver, CO, June 2017.
- ²Wang, G. G. and Shan, S., “Review of Metamodeling Techniques in Support of Engineering Design Optimization,” *ASME Transactions, Journal of Mechanical Design*, Vol. 129, No. 4, April 2007, pp. 370–380.
- ³Schöbi, R., Kersaudy, P., Sudret, B., and Wiart, J., “Combining Polynomial Chaos Expansions and Kriging,” Tech. Rep. RSUQ-2014-001, ETH Zurich, 2014.
- ⁴Gao, X., Wang, Y., Xie, N., Spotts, N., Roy, S., and Prasad, A., “Fast Uncertainty Quantification in Engine Nacelle Inlet Design Using a Reduced Dimensional Polynomial Chaos Approach,” AIAA 2016-0298, 54th AIAA Aerospace Sciences Meeting, 2016.
- ⁵Webster, R. and Oliver, M. A., *Geostatistics for Environmental Scientists*, John Wiley and Sons Ltd., West Sussex, England, 2007.
- ⁶Schöbi, R., Sudret, B., and Wiart, J., “Polynomial-Chaos-Based Kriging,” *International Journal for Uncertainty Quantification*, Vol. 5, No. 2, 2015, pp. 171–193.
- ⁷Dubourg, V., *Adaptive surrogate models for reliability analysis and reliability-based design optimization*, Ph.D. thesis, Blaise Pascal University, Clermont-Ferrand, France, 2011.
- ⁸Kawai, S. and Shimoyama, K., “Kriging-modl-based uncertainty quantification in computational fluid dynamics,” *32nd AIAA Applied Aerodynamics Conference*, No. 2014-2737, AIAA, Atlanta, GA, June 2014.
- ⁹Santer, T. J., Williams, B. J., and Notz, W. I., *The Design and Analysis of Computer Experiments*, Springer, 2003.
- ¹⁰Wackernagel, H., *Multivariate Geostatistics: An Introduction with Applications*, Springer, Berlin, Germany, 3rd ed., 2003.
- ¹¹Dubrule, O., “Cross Validation of Kriging in a Unique Neighborhood,” *Mathematical Geology*, Vol. 15, No. 6, 1983, pp. 687–699.
- ¹²Bachoc, F., “Cross Validation and Maximum Likelihood estimations of hyper-parameters of Guassian processes with model misspecification,” *Computational Statistics and Data Analysis*, Vol. 66, March 2013, pp. 55–69.
- ¹³Gao, X., Guzik, S. M. J., and Colella, P., “Fourth Order Boundary Treatment for Viscous Fluxes on Cartesian Grid Finite-Volume Methods,” AIAA 2014-1277, 52nd AIAA Aerospace Sciences Meeting, 2014.
- ¹⁴“<http://www.metacomptech.com>,” 2015.
- ¹⁵Schöbi, R., Sudret, B., and S., M., “Rare Event Estimation Using Polynomial-Chaos Kriging,” *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, Vol. 3, No. 2, 2017.
- ¹⁶Nguyen, H. M., Couckuyt, I., Knockaert, L., Dhaene, T., Gorissen, D., and Saeys, Y., “An Alternative Approach to Avoid Overfitting for Surrogate Models,” *Proceedings of the Winter Simulation Conference*, Winter Simulation Conference, 2011, pp. 2765–2776.
- ¹⁷Blatman, G. and Sudret, B., “Adaptive sparse polynomial chaos expansion based on least angle regression,” *J. Comput. Physics*, Vol. 230, No. 6, March 2011, pp. 2345–2367.
- ¹⁸Schöbi, R. and Sudret, B., “PC-Kriging: A New Metamodeling Method Combining Polynomial Chaos Expansions and Kriging,” *Proceedings of the 2nd International Symposium on uncertainty Quantification and Stochastic Modeling*, Rouen, France, June 2014.