AIAA JOURNAL
Vol. 57, No. 6, June 2019

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

Justin Weinmeister,* Xinfeng Gao,† and Sourajeet Roy‡
Colorado State University, Fort Collins, Colorado 80523

DOI: 10.2514/1.J057527

New aerospace aerodynamic bodies increasingly require robust design methods that demand data of the key problem variables in the presence of uncertainty. When these bodies are subject to complex physics, such as turbulence, separation, or secondary flows, the uncertainty data become more difficult to produce economically. Metamodels (surrogate models) can be used to produce data in the presence of uncertainty more efficiently by propagating the uncertainty from the model parameters to the outputs. However, the chief difficulty of metamodels is in consistently producing statistical data of the full system from a sparse number of evaluations. Recently, the polynomial chaos and kriging metamodeling approaches have been combined to take advantage of both their benefits. This research explores the combined method's effectiveness on airfoil and aircraft engine nacelle examples. It demonstrates that, although the combined method can produce more accurate results than either method alone, there is always a compromise between accuracy and efficiency.

Nomenclature

A	=	covariance term for least-angle regression (LARS)
AoA	=	angle of attack, deg
$\boldsymbol{B}$	=	leave-one-out RPC-K term
C	=	maximum correlation
C_d	=	coefficient of drag
C_l	=	coefficient of lift
$\boldsymbol{c}$	=	correlation vector for LARS
c	=	airfoil chord length, m
D	=	diameter of engine nacelle, m
$\boldsymbol{F}$	=	Vandermonde matrix
$f(\boldsymbol{\omega})$	=	polynomial basis evaluated at point $\boldsymbol{\omega}$
$\boldsymbol{H}$	=	leave-one-out error correction term
$\boldsymbol{h}$	=	vector pf lag distances for autocorrelation function
I	=	free-stream turbulence intensity, %
L	=	length of engine nacelle, m
M	=	molecular weight, kg/kmol
Ma	=	Mach number
m	=	maximum degree of polynomials
$\dot{m}$	=	mass flow rate, kg/s
N	=	number of design points
n	=	number of design parameters
n_r	=	$n_r \subseteq n$, where the number of terms is determined by a sensitivity study
P_0	=	total pressure
$p+1$	=	number of terms in polynomial expansion
$R(\boldsymbol{\omega}-\boldsymbol{\omega}')$	=	kriging autocorrelation function between points $\boldsymbol{\omega}$ and $\boldsymbol{\omega}'$
Re	=	Reynolds number
$\boldsymbol{r}(\boldsymbol{\omega})$	=	autocorrelation vector
S_j	=	sensitivity index for parameter j
T	=	temperature, deg
U	=	solution variable
$\boldsymbol{U}$	=	solution vector
U_k	=	coefficient for basis term
$\boldsymbol{u}$	=	equiangular vector
$Z(\boldsymbol{\omega})$	=	kriging Gaussian random process
ϵ	=	irreducible model error
γ	=	step size for LARS
$\mu_{\hat{U}}$	=	mean of estimate U
μ_t/μ	=	turbulent to laminar viscosity ratio
ψ	=	CDR solution variable
ω	=	design parameter
$\boldsymbol{\omega}$	=	design parameter vector
$\boldsymbol{\Phi}$	=	matrix of polynomial basis terms
ρ	=	air density, $\mathrm{kg/m^3}$
$\phi(\boldsymbol{\omega})$	=	multivariate orthogonal polynomial basis term for design point $\boldsymbol{\omega}$
σ	=	standard deviation
σ^2	=	kriging process variance
σ_U^2	=	variance for solution variable
$\sigma_{\hat{U}}^2$	=	variance of estimate U

Subscripts

$\mathcal{A}$	=	active set
$-i$	=	leave-one-out for th design point

Modifier

$\wedge$	=	estimate of variable

Received 29 May 2018; revision received 3 January 2019; accepted for publication 11 January 2019; published online 28 February 2019.

*Graduate Research Assistant, Computational Fluid Dynamics and Propulsion Laboratory; jrwein@rams.colostate.edu. Student Member AIAA.

†Associate Professor, Computational Fluid Dynamics and Propulsion Laboratory; xinfeng.gao@colostate.edu. Member AIAA.

‡Assistant Professor, High Speed System Simulation Laboratory; sjeetroy@engr.colostate.edu.

I. Introduction

AS AEROSPACE design methods develop, more information (off-design operating conditions and uncertainties) is needed for engineers to meet requirements. This is especially true of new robust design methods that require statistical data along with mean operating conditions in order to create more reliable and efficient products. In certain cases, such as aerodynamic flows, the cost of each new test or simulation is significant enough that generating the statistical data can be prohibitively costly. This prevents the use of Monte Carlo methods that require a very large number of model evaluations. Metamodels (surrogate models) allow engineers to extract the Monte Carlo data of a problem with far cheaper computational time and memory costs. These metamodels are a closed-form functional mapping of the uncertain model parameters to the output responses. These models can give engineers data in the presence of uncertainty and the ability to estimate performance at off-design points at low computational cost. This allows engineers to introduce advanced design techniques that have been previously too costly and reduce turn-around time for new iterations. Further, these methods

are not intrusive and can be applied to a wide range of problems and models. Specifically in aerodynamics, physical testing can be prohibitively costly when geometric configurations and fluid properties are part of the problem parameters. Computational fluid dynamics (CFD) simulations remove most of the cost in varying problem parameters, but they require far too many runs in order to generate stochastic data. Metamodels reduce the cost of these methods by requiring fewer CFD evaluations. The difficulty in constructing them is accurately re-creating the nonlinear physics of fluid motion governed by the Navier–Stokes equations.

Metamodeling techniques [1–7] have received a lot of attention as demand for more design optimization data has not been outpaced by advances in computational power. The first step in creating a metamodel is determining the experimental design through the identification of key parameters and the best sampling strategy. Common forms are Latin hypercube sampling (LHS), quasi-random sequences, Gaussian quadrature, and orthogonal arrays, among others [8,9]. Then, a metamodel is chosen. Common forms are radial basis functions, polynomial chaos (PC) expansions, and kriging or Gaussian process modeling [5,8]. Polynomial chaos expansions are a specific form of polynomial trend function that use polynomials that are orthogonal with respect to the probability density function of the model parameters [10–12]. It is an effective and convergent tool to approximate a problem, though it suffers from the curse of dimensionality as it scales factorially with the number of problem parameters and the degree of expansion. This can be overcome by either reducing the number of parameters studied through variance analysis [12,13] or using sparse expansions [11,14,15] to reduce the total number of basis terms and model evaluations. One of the most promising ways to achieve a sparse expansion is to apply least-angle regression (LARS), a variable selection algorithm related to forward stagewise and Lasso methods [16]. This tool has been used to find sparse PC expansions applied to linear and nonlinear problems. Nevertheless, the application of sparse PC expansions to complicated nonlinear problems is not yet well explored [11,17]. Kriging is another common metamodel, which is a spatial interpolation process developed by Georges Matheron that estimates the deviation of design points from a mean term [18]. Two forms are commonly used: ordinary kriging, where the deviations are measured from a mean value, and universal kriging, where the deviations are measured relative to a trend function. Kriging has been used in metamodeling activities for quite some time in a variety of forms and applications [5]. Recently, PC expansions were used as the trend function for universal kriging by Schöbi et al. [17,19] to get improved estimates. This combined metamodel is known as polynomial chaos-kriging (PC-K). Our previous works [20,21] have focused on using reduced PC (RPC) expansions in PC-K metamodels to form a reduced PC-K (RPC-K) metamodel that requires fewer model evaluations. In this work we align the RPC-K metamodeling method with the work of Schöbi et al. [17,19] by introducing the LARS algorithm to further reduce the PC expansion size by combining parameter reduction and sparse expansion strategies. The interacting effects of reducing both the size of the PC expansion and the parameter space of the problem are evaluated for their effectiveness in the metamodeling of sparse computer experiments. This work further improves the methods ability to accurately fit very small data sets.

In this work we present comprehensive work on the development and use of PC-K metamodels to perform uncertainty quantification studies on CFD simulations of aerospace aerodynamic bodies. In our previous studies we found that while these new metamodeling technologies could provide theoretically better results, they were hamstrung by a lack of robustness. In our original work [12], we developed an RPC metamodel, which used a standard PC expansion on a reduced set of simulation parameters determined from a sensitivity analysis of the first-order effects of these parameters. In that work we found that the method suffered from overfitting issues stemming from the few simulations that could be run in order to keep computational costs reasonable. In our following work [20], we combined the reduced polynomial chaos metamodel with kriging based on similar work conducted on standard polynomial chaos expansions [17,19]. The results of that work proved promising, although more extensive research [21] was required to identify the key strengths and weaknesses of the combined metamodel. In the end, we found that overfitting of the trend of the data was the biggest cause of errors in RPC-K methods just as with RPC methods. The present research was conducted to show that when the risk of overfitting is properly managed with sufficient oversampling and generalization error estimation, sparse PC-K-based metamodels can provide excellent accuracy and computational efficiency for uncertainty quantification studies of aerospace aerodynamic bodies.

The National Advisory Committee for Aeronautics (NACA) 4412 airfoil is studied in this work as it provides an excellent demonstration platform for PC-K-type metamodels because it has experimental data (without uncertainties) and it is computationally efficient to solve. The NACA4412 airfoil can be run under the influence of eight different parameters and includes separated flow that is difficult to predict due to its dependence on nonlinear terms in its mathematical model. An aircraft engine nacelle problem is also studied in this work for the complex physics that occur when the airflow separates from the nacelle wall. Separation occurs in only part of the design space, further complicating the problem and increasing the difficulty of fitting an accurate metamodel.

In this paper we first briefly review the structure of PC-K metamodels in Sec. II, including their mathematical structure and parameters along with the weaknesses and strengths of our RPC-K method. Then the LARS algorithm implemented in this work is detailed, and the error measures used for the CFD studies are presented. In Sec. IV our new LARS-improved RPC-K metamodel is then validated on a convection-diffusion-reaction (CDR) case. This case is a finite-volume CFD solution to a representative fluids equation. This solution can be solved quickly enough that 10,000 cases are run to generate a reliable Monte Carlo data set for validation, though it does not show the complex physics the RPC-K metamodel has been constructed for. Then in Sec. V.A, the problem configurations for the NACA 4412 airfoil and engine nacelle cases are presented in detail. The NACA 4412 airfoil case is used to compare the metamodel's performance on physics of medium difficulty in a problem where a few hundred simulations worth of data can be gotten economically. The engine nacelle case is used as a problem with more complex physics, though a data set of only a few 10s of simulations is available. Finally, the results for both aerodynamic bodies are presented and discussed in Sec. V.B before our conclusions are drawn in Sec. VI.

II. Review of Reduced Polynomial Chaos-Kriging Method

The PC-K method is a combination of generalized polynomial chaos expansion methods and universal kriging elucidated in the works by Schöbi and Sudret [17,19] and Kersaudy et al. [22] in 2015. Polynomial chaos methods create a metamodel for a system by determining a polynomial basis set and fitting coefficients to those basis terms. For example, a random output variable, $U(\boldsymbol{\omega})$, can be represented as

$$U(\boldsymbol{\omega}) \approx U^{(\mathrm{PC})}(\boldsymbol{\omega}) = \sum_{k=0}^{p} \phi_k(\boldsymbol{\omega}) U_k \tag{1}$$

where $\phi_k(\boldsymbol{\omega})$ is the kth multivariate orthogonal polynomial basis term and U_k is the corresponding coefficient. Here $\boldsymbol{\omega} = [\omega_1, \omega_2, \ldots \omega_n]$ denotes the vector of parameter values that the random output variable is evaluated at. The number of basis terms can be calculated as

$$p + 1 = \frac{(m+n)!}{m!n!} \tag{2}$$

where n is the number of design parameters and m is the maximum degree of the polynomials. The RPC method uses Eq. (1); however, it eliminates the unimportant input parameters from the problem using Sobol sensitivity indices in order to reduce the number of basis terms needed [12]. These Sobol sensitivity indices are calculated after a small initial set of runs where the variance that each parameter contributes is estimated by its first-order effects using a high-dimensional model

representation [23–26]. The parameters that contribute negligible variance can be set to their mean value for the remainder of the runs, and so the new number of basis terms required for a given maximum degree can now be represented as

$$p+1=\frac{(m+n_r)!}{m!n_r!} \tag{3}$$

where n_r is the number of parameters retained after the unimportant dimensions are removed. For a simulation with seven parameters reduced to five and approximated with third-order polynomials, the number of basis terms will be reduced from 120 to 56 with a marginal loss in accuracy. When dealing with problems that can be well approximated with reduced parameter sets, the computational savings benefits are more important than the corresponding loss in accuracy.

A PC method can then be combined with universal kriging to get a PC-K method. Universal kriging is a type of kriging where the mean term is given as trend function throughout the domain [18]. When the trend function is estimated using PC, a PC-K method is given as

$$U(\boldsymbol{\omega})\approx U^{(\mathrm{PC}-K)}(\boldsymbol{\omega})=\sum_{k=0}^{p}\phi_k(\boldsymbol{\omega})U_k+\sigma^2 Z(\boldsymbol{\omega}) \tag{4}$$

where $\sum_{k=0}^{p}\phi_k(\boldsymbol{\omega})U_k$ is the polynomial trend function and $\sigma^2 Z(\boldsymbol{\omega})$ is the deviation from the trend. The kriging term σ^2 is the process variance and $Z(\boldsymbol{\omega})$ is a Gaussian random process term. Because of this, kriging is sometimes known as Gaussian process modeling, although it can be applied to non-Gaussian variables when handled appropriately. The coefficients are solved for using linear regression as

$$\boldsymbol{U}_k=(\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{F})^{-1}\boldsymbol{F}^T\boldsymbol{R}^{-1}\boldsymbol{U} \tag{5}$$

and the process variance is solved for as

$$\sigma^2=\frac{1}{N}(\boldsymbol{U}-\boldsymbol{F}\boldsymbol{U}_k)^T\boldsymbol{R}^{-1}(\boldsymbol{U}-\boldsymbol{F}\boldsymbol{U}_k) \tag{6}$$

where $\boldsymbol{R}$ is the autocorrelation matrix, $\boldsymbol{F}$ is the Vandermonde matrix, $\boldsymbol{U}_k$ is the vector of coefficients, $\boldsymbol{U}$ is the output response of the evaluated design points, and N is the number of evaluated design points [17].

In kriging, the relationship between any two design points is described by an autocorrelation function:

$$R(\boldsymbol{\omega}-\boldsymbol{\omega}')=R(\boldsymbol{h}) \tag{7}$$

which is fit to the empirical values. The autocorrelation matrix, $\boldsymbol{R}$, is an $N\times N$ matrix of the autocorrelation values between the evaluated design points. The independent variables of the autocorrelation function are the lags, $\boldsymbol{h}$, which is a vector of the distances between any two design points in the problem's parameters space. Among the many functions available, the Matérn autocorrelation function is used in this research as it is quite commonly used for computer experiments [18]. The multivariate forms of these functions are often taken as a multiplication of the univariate terms, though other, nonseparable autocorrelation functions exist. These autocorrelation function can then be used to give the best estimate of a new design point's correlation with existing design points. Bounded autocorrelation functions have a finite distance at which no correlation between design points exists, and unbounded autocorrelation functions, like the Matérn model, asymptotically approach zero. In this research we used the Matérn autocorrelation function with the shape parameter set to $3/2$. It is given in Eq. (8)

$$R(\boldsymbol{\omega}-\boldsymbol{\omega}')=\prod_{i=1}^{n}\left(1+\frac{\sqrt{3}|\omega_i-\omega_i'|}{l_i}\right)\exp\left(\frac{-\sqrt{3}|\omega_i-\omega_i'|}{l_i}\right) \tag{8}$$

where l_i is the range parameter for each corresponding design parameter, $i=(1\ldots n)$, and a hyperparameter of the metamodel. The hyperparameters of the kriging model are commonly fit using either maximum likelihood estimation or cross-validation methods. In our cases, cross-validation is used because the exact correlation structure is unknown, although both methods were tested and neither was found to be significantly more effective.

The variation from the trend at a given design point is the sum of the nearby design point's variation multiplied by its autocorrelation. For example, a response prediction will be weighted so that the predicted response is below the trend value if the simulation data near the prediction point are also below the trend value. The weighting is assumed to be the likelihood of the new design point and is described by the random process, $Z(\boldsymbol{\omega})$. The likelihood is then scaled to the design variables ($\boldsymbol{U}$) by the process variance, σ^2, which is the dimensional form of the process correlation.

In the RPC-K metamodel, the polynomial trend's basis terms, $\phi_k(\boldsymbol{\omega})$, are from the RPC metamodel. The specific form of these basis terms is given from the PC theory that uses orthonormal polynomials for each of the problem parameters. For example, if the problem parameters are uniformly distributed, Legendre polynomials are used. If they are normally distributed, the Hermite polynomials are used [17]. Again, the multivariate forms are constructed as a product of the univariate terms. The coefficients, $\boldsymbol{U}_k$, are then fit using least squares regression with the Vandermonde and autocorrelation matrices. The Vandermonde matrix $\boldsymbol{F}$ is an $N\times(p+1)$ matrix of the basis terms evaluated at each of the known design points.

Once the kriging metamodel has been constructed, the estimated mean at a new design point can be given as

$$\mu_{\hat{U}}(\boldsymbol{\omega})=\boldsymbol{f}(\boldsymbol{\omega})^T\boldsymbol{U}_k+\boldsymbol{r}(\boldsymbol{\omega})^T\boldsymbol{R}^{-1}(\boldsymbol{U}-\boldsymbol{F}\boldsymbol{U}_k) \tag{9}$$

and the variance at the design point as

$$\sigma^2_{\hat{U}}(\boldsymbol{\omega})=\sigma^2\left(1-[\boldsymbol{f}(\boldsymbol{\omega})^T\boldsymbol{r}(\boldsymbol{\omega})^T]\begin{bmatrix}\boldsymbol{0} & \boldsymbol{F}^T\\ \boldsymbol{F} & \boldsymbol{R}\end{bmatrix}^{-1}\begin{bmatrix}\boldsymbol{f}(\boldsymbol{\omega})\\ \boldsymbol{r}(\boldsymbol{\omega})\end{bmatrix}\right) \tag{10}$$

where $\boldsymbol{f}(\boldsymbol{\omega})$ is the polynomial basis evaluated at the new design point and $\boldsymbol{r}(\boldsymbol{\omega})$ is the autocorrelation vector for the estimate point evaluated against all design points.

Although the value $\sigma^2_{\hat{U}}$ is an accurate estimation of the variance in the design point estimation, it is not an estimate of the variance of an uncertain design point. This can be shown where $\sigma^2_{\hat{U}}=0$ at a design point, but the design point is simply a single deterministic solution to a stochastic problem and itself carries some uncertainty. To get this design variance value, a Monte Carlo simulation of the kriging metamodel can be run and the statistical moments calculated from the resulting predicted values. Polynomial chaos does not require this additional computational cost, as the statistical moments can be obtained directly from the coefficients of the basis terms where variance is the sum of squares of the coefficients. Polynomial chaos-kriging metamodels can have their statistical moments calculated either using just the coefficients of the trend function like PC or using a Monte Carlo simulation like ordinary kriging (OK), but we demonstrate later that using a Monte Carlo simulation of the metamodel provides the best estimates.

Kriging is a best linear unbiased predictor, and so it will give the best estimator based on the estimates' relative variance from the "true" values under the unbiased constraint. While it is possible to have biased models with less variance, the unbiased constraint is generally considered an important characteristic for models [18]. Additionally, the assumption of an underlying trend in the data for universal kriging breaks down some of the strong assumptions of kriging such as stationarity of the data and its variance throughout the problem domain. However, universal kriging still provides excellent results when these weaknesses are considered properly.

Polynomial chaos-kriging can be an advantageous metamodel in certain circumstances as it combines both general trend information with local deviation values to provide the most accurate estimates of unknown problem responses. If the underlying problem is easily approximated by a basic polynomial trend function, the PC trend function within the PC-K metamodel will accurately estimate off-design

points as the result of the first right-hand term of Eq. (4). The calculated deviation term, $Z(\boldsymbol{\omega})$, will then be approximately zero and contribute nothing to the estimate as the experimental design is well-approximated. When overfitting occurs, the number of terms in the PC basis and the number of design points are approximately equal and the trend function will go through all design points exactly. Unfortunately, the trend function will not be accurate as it will introduce additional variance into the data by having spurious oscillations between design points. When the PC trend function does not approximate the design points well, the second right-hand term in Eq. (4) will contribute to the estimate of the off-design response. Nearby design points will not be well-described by the trend function and will contribute significant values to $Z(\boldsymbol{\omega})$. This deviation term will be scaled to the problem by σ^2, and adjust the estimate from the trend function. If the nearby design points are underpredicted by the trend function, the deviation term will adjust the estimate of the off-design point up and vice versa. In the case of no trend information, the problem reverts to ordinary kriging where the trend is simply estimated as the mean of all design points in the experiment. Because of these characteristics, PC-K metamodels can accurately model both global trend functions and local deviations from these trends accurately.

However, one possible source of error is in accurately modeling the problem's covariance from the computational experimental design. The most important parameters in the fitting process are the correlation structure and its range. They can be accurately modeled by a good initial design of experiments, a representative correlation function, and using robust optimization methods. In our previous work [21], we analyzed a variety of autocorrelation functions, experimental designs, minimization functions, and minimization algorithms with an adaptive algorithm to improve fitting. In addition, the problem can be simplified by fitting fewer parameters. Our previous work [12] investigated using Sobol sensitivity indices to find and eliminate problem parameters that did not contribute much variance to the studied output variables. The smaller PC expansions required for these reduced problems are frequently superior to the full PC expansion [11].

Despite these improvements, overfitting is still the chief issue for producing robust results on aerodynamic problems. The complex computer simulations required for these models limit the size of the experimental designs and thus reduce the number of basis terms that can be used. However, when these problems have nonlinear effects, a large number of basis terms can often be required to accurately approximate their behavior. As a result, the uncertainty studies frequently require reducing the oversampling of the problem to as low as possible and subjecting the problem to more risk of overfitting. In this work, we apply a hierarchical LARS method [16] to the polynomial basis to combat the overfitting issue [11]. This aligns our RPC-K method with the foundational PC-K work by Schöbi and Sudret [19].

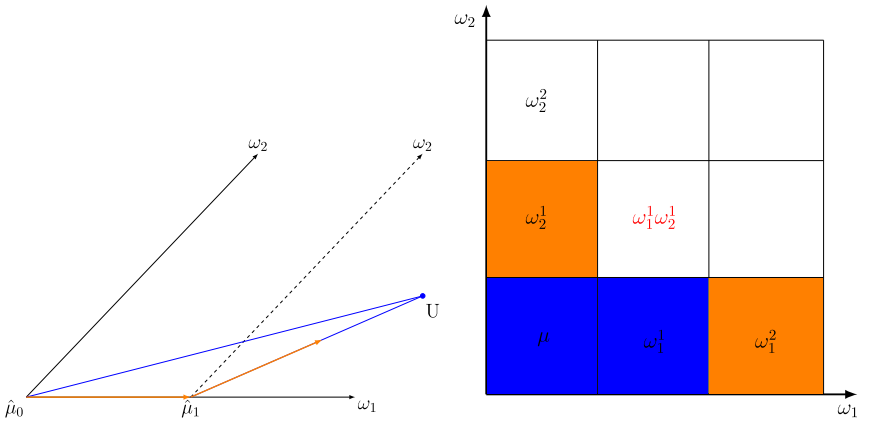
III. Least-Angle Regression for RPC-K

LARS is a model selection algorithm that is related to forward stagewise and Lasso selection methods [16] for choosing predictors for a model. The method works by choosing the predictor, in this case a basis term, most correlated with the output response and increasing its coefficient until another basis term is equally correlated. From there, the method progresses in a direction equiangular between these basis terms [16]. One by one, the method chooses the basis terms most correlated with the response until all basis terms are chosen. LARS has already been used to improve metamodels before [19,27], though in a variety of forms. Here we propose a method that is suited for use on the smallest data sets.

Because using a sparse basis set is seen as one of the most effective ways to reduce overfitting for the PC-K method, we employed LARS to choose the basis terms that have distinctively high correlations with the response for every possible size of the basis set. Then the metamodel's cross-validation error is calculated for every basis size and the final sparse set chosen is the one with minimum error. This process is covered in detail by Blatman and Sudret [11]. It should be noted that the calculated coefficients in LARS are replaced with those calculated through the RPC-K metamodel.

The LARS process can be explained graphically in a two-parameter space using Fig. 1a. Initially, the polynomial basis will include only a constant that defines the origin, $\hat{\mu}_0$. The first basis term added will be the basis term for parameter ω_1 because the solution, U, is most correlated with it. The coefficient for ω_1 will then be increased until ω_1 and ω_2 are equally correlated. At this point, the basis term for ω_2 will be added.

When applied to PC expansions, the parameter space is composed of every possible basis term for the expansion, including higher-order terms and interaction effects. Because of this, the model can conceivably add high-order interaction terms before the first-order main-effect terms of parameters that are known to have a large impact on the solution. However, the sparsity of effects principle states that main effects and low-order terms dominate the problem. Considering this principle, it is important to construct metamodels so that low-order terms are added before their respective higher-order terms [16,28]. Doing so creates models that are more realistic and easier to comprehend. One possible method originally proposed by Efron et al. is to run the LARS algorithm on only first-order main effects first, then run the LARS algorithm on second-order effects, and so on until a desired model is reached [16]. Efron et al.'s method builds the model in a logical order, but it either restricts a model to having all of the parameters represented equally at lower-order basis terms or requires manually restricting the terms at each order of the model. To get around these constraints, the heredity of the terms can be exploited so that higher-order terms are added to the model only if the average of it and its dependent terms is higher than all other terms [29]. This method penalizes higher-order terms that have many



a) Basis term selection for two variables **b) Hierarchical LARS algorithm**

Fig. 1 Graphical interpretation of the LARS method.

dependent terms from joining the model. Unfortunately, if a higher-order term is chosen, all terms are added simultaneously. This property may cause difficulties in very small data sets when adding multiple terms could increase the error significantly. Instead, in this research we implement a hierarchical LARS method that adds only one term at a time and will chose a term only for which its dependent terms already have been included.

The hierarchical LARS method will at first consider only the first-order basis terms of the problem. This is accomplished by adding them to a candidate set that is the subset of all available basis terms based on the expansion truncation scheme. The terms in the candidate set are chosen by heredity, and so the lower-order versions of a polynomial must be chosen first. Once a first-order term has been chosen, its respective second-order term is added to the candidate set. This process is repeated so that all higher-order terms are not included in the candidate set before their lower-order ones. Interaction terms are added to the candidate set only if each component term has been added by itself up to the same order, similar to the strong heredity principle of Yuan et al. [29] or the important dimension of Jakeman et al. [27]. These works have demonstrated that improved metamodels can be achieved when unimportant interaction terms are not added to the basis before more impactful terms. If any of the basis terms added to the candidate set have a correlation higher than the current max correlation, those terms are added without adjusting the estimated coefficients. The process is repeated until all terms are chosen.

A graphical example is shown in Fig. 1b. If the mean term (μ) and the first-order term for the first parameter (ω_1^1) are in the current basis set (blue), the candidate set for the next step (orange) has the terms ω_1^2 and ω_2^1. Even if the interaction effect $\omega_1^1\omega_2^1$ is more correlated with the solution at this stage, it is not in the candidate set, and it will not be added because its component term ω_2^1 has not been added yet. If in this hypothetical case ω_2^1 is chosen next, the interaction term $\omega_1^1\omega_2^1$ will be added immediately after, because its correlation exceeds the current correlation of the basis set. The LARS algorithm is used to find the order of importance for the basis terms so that the sparse PC expansions are constructed in accordance with the sparsity of effects principle, and this way logical basis terms are chosen and the model is more robust to overfitting than if any basis term could be added.

Because the order of terms to include is known, a separate basis can be constructed for each possible basis size from 1 to $p+1$. As such, the basis of size 10 has the first 10 basis terms and the basis of size 11 has the first 11 terms added by LARS. From these separate bases, $p+1$ PC-K or RPC-K metamodels can be constructed using each expansion size. Then, their accuracy's can be compared using their respective LOOCV errors. The metamodel used in the uncertainty quantification study is chosen as the metamodel with the basis that minimizes the LOOCV error. This methodology follows the optimal PC-K method described by Schöbi et al. [17] with the modification of using a hierarchical LARS algorithm for the RPC-K method.

When constructing a metamodel, the generalization error is more critical to analyze than the training error as it measures the expected error of the metamodel on new design points. The generalization error is usually calculated as a mean squared error (MSE) term, and so it shows quadratic behavior with decreasing error initially as additional terms are added to the model to eliminate underfitting. Then, the error will increase as too many terms are added and overfitting of the solution occurs. The most efficient way to approximate the generalization error of metamodels is to calculate the cross-validation error of the metamodel, as the method does not require separate testing and training data points. Cross-validation works by calculating the metamodel with a subset of the design points and then estimating the remaining design points. The process is repeated for all of the data and a resulting MSE calculated. In this work, we used leave-one-out cross-validation (LOOCV) as it efficiently estimates the true generalization error and has an analytical form that saves computational cost [30,31]. It is the special case when only one point is left out of the training set at a time. The leave-one-out error for a PC-K metamodel is expressed as

$$\mathrm{Err}_{\mathrm{LOO}} = \frac{1}{N}\sum_{i=1}^{N}\left(U_i - U_{-i}^{(\mathrm{PC}-K)}(\boldsymbol{\omega}_i)\right)^2 \tag{11}$$

where $U_{-i}^{(\mathrm{PC}-K)}(\boldsymbol{\omega}_i)$ is the PC-K metamodel constructed from the simulation data set minus point i evaluated at point i.

Compact calculations for the LOOCV error exist for both PC and PC-K methods, allowing the error to be calculated quickly and efficiently by not computing N separate metamodels for error estimation. The LOOCV error for a PC metamodel can be calculated as

$$\mathrm{Err}_{\mathrm{LOO}}^{(\mathrm{PC})} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{U_i - U^{(\mathrm{PC})}(\boldsymbol{\omega}_i)}{1-H_i}\right)^2 \tag{12}$$

where H_i is the ith diagonal term in $\boldsymbol{H} = \boldsymbol{F}(\boldsymbol{F}^T\boldsymbol{F})^{-1}\boldsymbol{F}^T$. The compact form requires the construction of only the full PC metamodel. Polynomial chaos-kriging methods cannot use this compact form for PC metamodels, but Olivier Dubrule [32] found a way to calculate $U_{-i}^{(\mathrm{PC}-K)}(\boldsymbol{\omega}_i)$ for every i without computing the N individual metamodels. The mean estimate can be given as

$$U_{-i}^{(\mathrm{PC}-K)}(\boldsymbol{\omega}_i) = -\sum_{j=1,j\neq i}^{N}\frac{\boldsymbol{B}_{ij}}{\boldsymbol{B}_{ii}}U_j = -\sum_{j=1}^{N}\frac{\boldsymbol{B}_{ij}}{\boldsymbol{B}_{ii}}U_j + U_i \tag{13}$$

where the new matrix $\boldsymbol{B}$ is given in Eq. (14).

$$\boldsymbol{B} = \begin{bmatrix}\sigma^2\boldsymbol{R} & \boldsymbol{F}\\ \boldsymbol{F}^T & \boldsymbol{0}\end{bmatrix}^{-1} \tag{14}$$

In Eq. (14), σ^2 is the kriging process variance, $\boldsymbol{R}$ is the correlation matrix, and $\boldsymbol{F}$ is the Vandermonde matrix from the PC-K metamodel. These analytical forms of the LOOCV error can quickly provide accurate estimates of the accuracy of the corresponding metamodel. These computations then allow for efficient determination of the optimal metamodel from all those considered. For optimal PC-K metamodels, the set of metamodels with varying basis sizes is evaluated for their LOOCV error, and the metamodel with minimum LOOCV error is deemed optimal. In this work all errors are calculated from data normalized to the mean so that data of varying magnitude can be compared directly.

IV. Validation

The PC-K method was first tested on analytical functions [17,19] and was further expanded into structural reliability and geotechnical problems [33]. In our previous works [20,21], we found that the applied RPC-K method provided excellent results on the convection-diffusion-reaction verification problem that was not subject to overfitting issues. For the engine nacelle geometry, the additional modifications of the RPC expansion did not eliminate the overfitting issue [21]. Because of this, we first validated the new hierarchical LARS procedure with RPC-K on the convection-diffusion-reaction (CDR) problem to match our previous results. For comparison, our cases compare the performance of the RPC-K method to the original optimal PC-K metamodel [19], a regression-based PC metamodel, an RPC metamodel, and ordinary kriging. The PC-K and RPC-K metamodels both use Matérn autocorrelation functions, cross-validation minimization measures for the hyperparameters, and the *fmincon* MATLAB minimization routine to solve the cross-validation measure. They follow the construction of Schöbi and Sudret [19] other than the hierarchical LARS algorithm for RPC-K. Our regression-based PC model is representative of response surface metamodeling methods, and it shows how they might be expected to perform versus a combined method for nonlinear aerodynamic flows. The ordinary kriging method is representative of distance-weighted averages or kernel methods.

It is also important to mention the relative computational costs of these methods. All metamodeling methods were written as an in-house code using MATLAB and were run in serial on an Intel Xeon E5-2620 v4 CPU run at 2.1 GHz. The in-house code was not

optimized for efficiency, and so the given computation times should be considered relative to each other and not as the method's absolute efficiency. Polynomial chaos methods are by far the fastest methods with it taking approximately 300 microseconds to compute the coefficients for a 70-term basis. Kriging methods require significantly more time because of the process of solving for the hyperparameters. For 70 data points, it takes around 0.5 s to construct the metamodel. Polynomial chaos-kriging methods require even greater computational time as the additional basis terms in the correlation matrix increase the computational burden. For 70 data points and a 70-term basis, it is approximately 1 s. To run a full evaluation takes much longer because of the overhead functions required to setup and run the problems, the multiple points evaluated, and the multiple basis sizes considered in the hierarchical LARS process. For complex problems and many evaluation points, this can take up to tens of minutes. However, these computational costs are minor compared with the aerodynamics simulations they are being applied to, which generally take several hours even when run in parallel.

The one-dimensional transient CDR problem is a finite-volume CFD solution to the equation

$$\frac{\partial \psi}{\partial t} + u\frac{\partial \psi}{\partial x} - k\frac{\partial^2 \psi}{\partial x^2} + c\psi = f \tag{15}$$

in a unit domain from $0 \le x \le 1$ with $f = \alpha(x-1)e^{-t}$. Here ψ is the solution variable and $u = 1$, $k = 1$, $c = 1$, $\alpha = 2$ are the input parameters. The CDR verification problem was covered in our previous works [12,20,21] and full details can be found in the original study [12]. The solution response is a asymptotic decrease in ψ over time where the stochastic equation is modeled as an ensemble of deterministic solutions with uncertain input parameters. The exact solution to the nominal case and two uncertain realizations of the problem are shown in Fig. 2 as reference. Our evaluation case took 70 samples using Sobol sequence sampling of normally distributed input parameters with standard deviations of 10% of the mean and compared the results to a Monte Carlo study that used a separate 10,000-simulation data set. All of the metamodels were constructed using the same data set for an equitable comparison. A sensitivity study was conducted and found the parameters u and k to be the most impactful, and so RPC and RPC-K metamodels were constructed based on these two parameters alone for each time step. The solutions used in the sensitivity study for parameter reduction can be re-used for fitting the metamodels, which is true for all cases, though not always wise. When projecting higher-dimensional data onto lower-dimensional models, the uncertainty of the unmodeled parameters can introduce additional errors into the model that would not be present if the samples were re-taken with these parameters set to their respective mean values. However, this work focuses on applications to problems where simulations are too costly to re-run and where the unmodeled parameters have been determined to contribute relatively little variance compared with the modeled parameters. Our hierarchical LARS algorithm was applied to the PC, RPC, PC-K, and RPC-K metamodels with a maximum degree of four and the leave-one-out errors were plotted against the number of basis terms in the polynomial function.

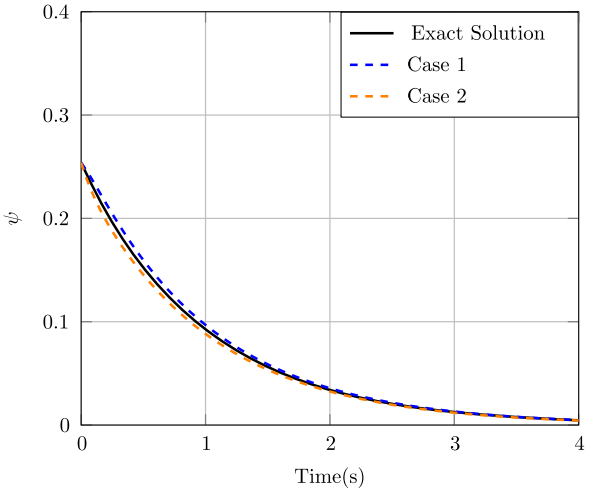
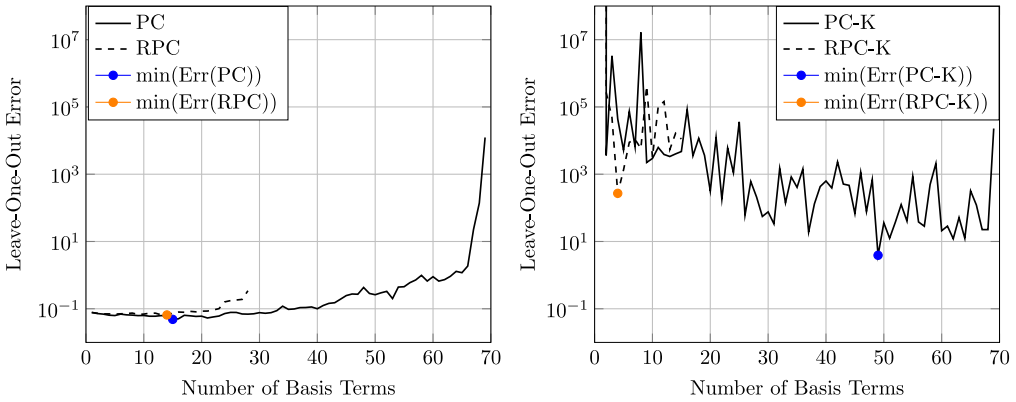


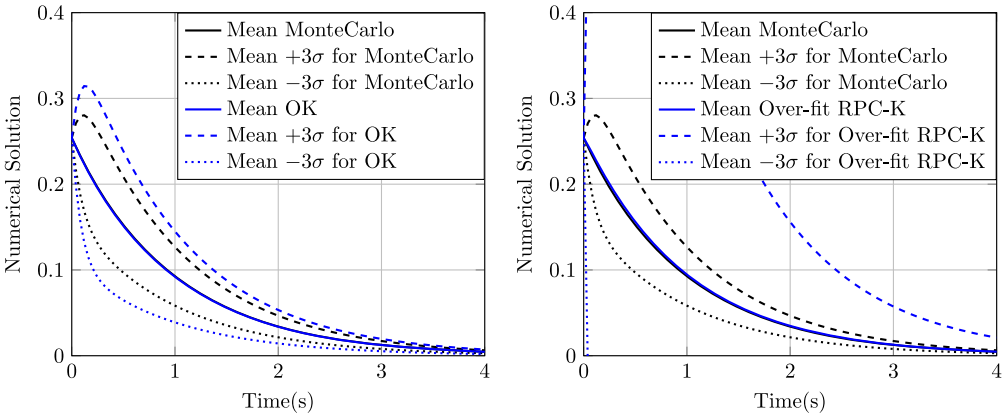
Fig. 2 Exact solution to CDR problem.

Figure 3 plots the LOOCV error for the metamodels versus the number of terms chosen by the hierarchical LARS algorithm. Figure 3a contains the error progression for the PC and RPC metamodels, and Fig. 3b contains the error progression for the PC-K and RPC-K metamodels. It can be seen that the PC and RPC metamodels have strong trends in the LOOCV error progression and do not have much oscillation, whereas the PC-K and RPC-K metamodels do. To determine the cause of the oscillating behavior, multiple investigations were conducted. First, multiple types of polynomial bases were used in the models instead of the standard probabilists' version of Hermite polynomials, including the physicists' version, normalized versions, and a simple basis $(x, x^2, x^3, \ldots)$. It was found that better results could be found with varied basis types. Unfortunately, no apparent trend was found for the various basis types. The models had only a narrow range of accuracy and mostly under- or overpredicted the variation of the solution. The fluctuations in the LOOCV error for the PC-K metamodels persisted through all forms. To rule out the addition of the correlation information in the least squares regression for the PC-K metamodels as the cause of the oscillations, the cases were also run using the coefficients chosen from plain least squares regression, but the oscillations persisted. Additionally, the basis size of the PC-K metamodels was always chosen to be near 50 terms for



a) Polynomial chaos metamodels **b) Polynomial chaos-kriging metamodels**

Fig. 3 Leave-one-out cross-validation error progression of ψ for varying basis sizes.



a) The OK metamodel vs. Monte Carlo results **b) Results of over-fit RPC-K metamodel**

Fig. 4 Results of mean, $+3\sigma$, and -3σ for OK metamodel and overfit RPC-K metamodel on CDR problem.

all polynomial forms and the results seemed to indicate that these 50 basis term models were suffering from overfitting that the LOOCV error did not show. Ultimately, the cause of the oscillations in the LOOCV error for the PC-K and RPC-K metamodels is the inconsistent correlation structure at varying solution times. The correlation structure is inconsistent because the CDR problem shows very smooth behavior with little variation in ψ at later solution times versus the significant variation before $t = 2$. As a result, the LOOCV error is amplifying noise in the metamodel solution. The oscillating behavior is not present in the PC solution because the PC metamodel does not try to fit the ill-conditioned correlation structure, but it just performs a regression on the data points. Regardless, the results show that the methods are still effective enough to validate them against the CDR Monte Carlo data.

First, the results of the OK metamodel built with the reduced parameters are plotted in Fig. 4a as a reference solution for a reduced metamodel with no trend function. The results show fairly close alignment with the Monte Carlo solution for the mean, but greater errors for the $+3\sigma$ and -3σ values when compared from time $t = 0$ to 4. The average MSE of the mean of the OK metamodel relative to the Monte Carlo data is 4.39×10^{-5}, and it is 4.00×10^{-2} for the standard deviation. These errors are compared with the RPC and RPC-K metamodels in Table 1. Because the Monte Carlo simulation had 10,000 data points, the error of the first statistical moment is less than the standard error of the Monte Carlo experiment, whereas the error for the standard deviation is not. These error values make sense as the plot shown in Fig. 4a shows that the metamodel overpredicts the standard deviation.

The results of an overfit solution for the RPC-K metamodel are also shown in Fig. 4b. The overfit solution approximates the mean quite well, but the standard deviations are illogically high. Most values are outside the plotted range, and the results actually give negative values for the lower bound. This behavior is typical of an overfit solution and is the reason the basis set must be reduced to prevent the spurious results. Next, we analyze the results of RPC and RPC-K metamodels to validate our RPC and RPC-K methods using the reduced parameter set and hierarchical LARS to reduce the basis set.

The mean, $+3\sigma$, and -3σ values of the RPC metamodel are compared with the Monte Carlo data in Fig. 5a. The minimum error RPC metamodel selected by hierarchical LARS used 14 basis terms to fit the 70 data points. The average MSE of the mean of the RPC model compared with the mean of the Monte Carlo simulation over all time steps is 2.52×10^{-5}. For standard deviation, it is 2.76×10^{-5}.

Table 1 Average mean square error for CDR metamodels compared with Monte Carlo solution

Model	OK	RPC	RPC-K
Mean	4.39×10^{-5}	2.52×10^{-5}	2.13×10^{-7}
σ	4.00×10^{-2}	2.76×10^{-5}	4.86×10^{-4}

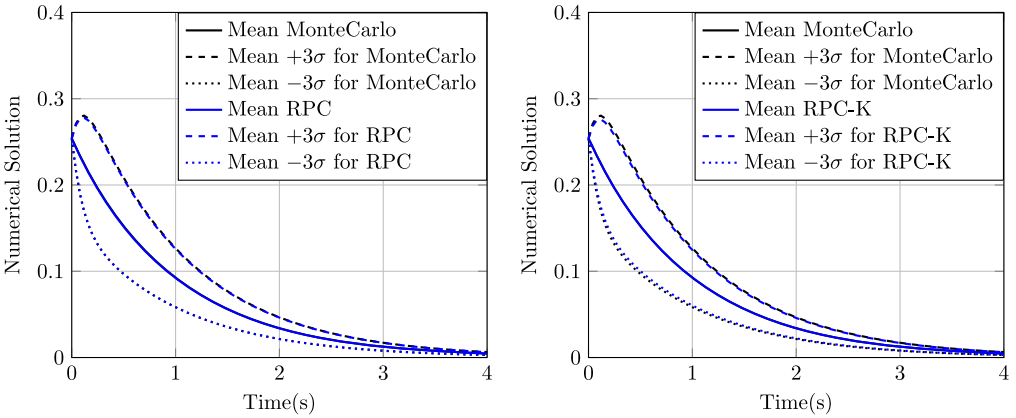
Again, these errors are compared in Table 1. The error for the mean is again within the standard error of the Monte Carlo simulation while the standard deviation is also under the Monte Carlo simulation's standard error for the RPC metamodel. Because of this, we can say that the LARS algorithm selected a PC expansion that with only 70 samples modeled the problem just as effectively as the Monte Carlo simulation. This CDR case helps validate the effectiveness of the LARS algorithm when it is used by the RPC metamodel, the component of the RPC-K metamodel prone to overfitting. The mean, $+3\sigma$, and -3σ values of the RPC-K metamodel are compared with the Monte Carlo data in Fig. 5b. The minimum error RPC-K metamodel chosen by hierarchical LARS used four basis terms in addition to the correlation data to fit the 70 data points. Table 1 compiles the average MSEs that are 2.13×10^{-7} for the mean and 4.86×10^{-4} for the standard deviation. The mean and standard deviation estimates are again within the standard error of the Monte Carlo simulation. Despite the oscillating behavior of the LOOCV error and the RPC-K metamodels' inappropriateness for this smooth solution problem, the hierarchical LARS algorithm still chose a suitable metamodel. Critically, hierarchical LARS improves the RPC-K metamodel as the sparse polynomial basis outperforms the full basis set. We also want to note that the RPC-K errors relative to the Monte Carlo simulation are much lower than the LOOCV error predicted. Generalization error estimates are useful for informing the construction of metamodels, but they are not reliable for determining whether metamodels with different constructions are more accurate. In the CDR validation case, the LOOCV error predicted the RPC metamodel to be three orders of magnitude more accurate when the difference was negligible when compared with the Monte Carlo solution.

We have validated that the RPC- and RPC-K-type metamodels can accurately model an example physics problem and that using a sensitivity study to reduce the parameter size of a problem can effectively reduce the computational size of the problem and still maintain sufficient accuracy. Additionally, we have shown that the construction of RPC-K metamodels is not straightforward due to the interacting effects of fitting data with both a least squares solution and correlation function. No metamodel fits all problems best, and metamodels are most effectively used when the appropriate metamodel can be chosen based on the underlying behavior of the problem. Now we investigate strongly nonlinear aerodynamic flows where the additional correlation information allows the RPC-K method to outperform the other methods.

V. Results

A. Problem Configuration

The RPC-K metamodel is applied to two aerodynamics problems. The first problem is a two-dimensional Reynolds-averaged Navier-Stokes (RANS) solution to the NACA 4412 airfoil at low speed and a high angle of attack (AoA). The second problem is a RANS solution to



a) Results of RPC metamodel for CDR **b) Results of RPC-K metamodel for CDR**

Fig. 5 Mean, $+3\sigma$, and -3σ values of the RPC and RPC-K metamodels compared with Monte Carlo results for the CDR problem.

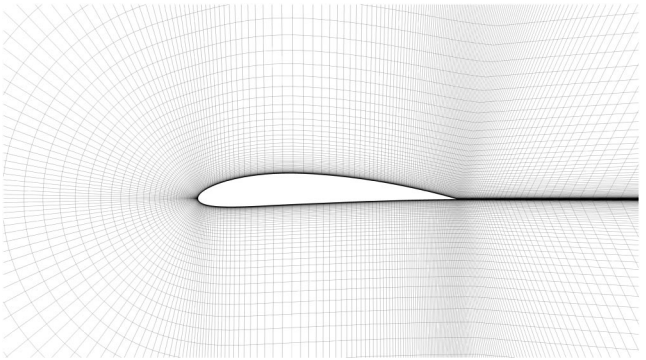


Fig. 6 Image of the mesh detail around the airfoil body for the NACA 4412 airfoil.

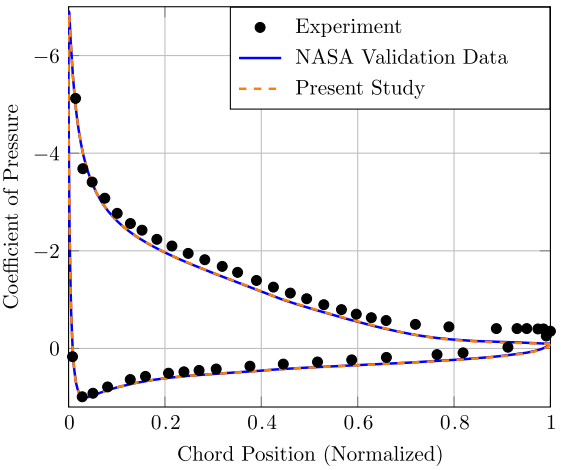


Fig. 7 Comparison of experimental and computational results for coefficient of pressure.

study the inlet air of an aircraft engine at takeoff in a strong crosswind. Both of these cases have separated flow regions, which we attempt to capture in our metamodel with only a sparse evaluation of the problem. ICFD++ [34] is used for all of our CFD simulations.

1. *Airfoil Case*

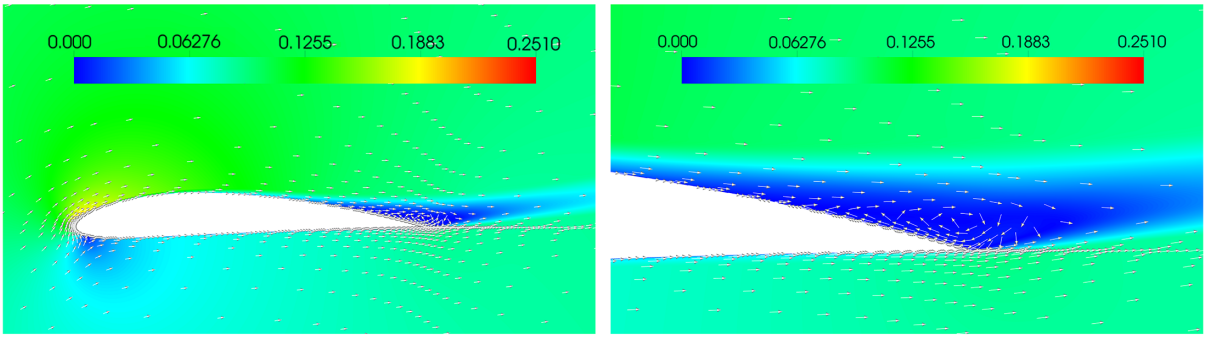
The NACA 4412 CFD cases were run as two-dimensional fully turbulent simulations using the RANS shear stress transport (SST) $k-\omega$ turbulence model on the 897×257 structured C grid shown in Fig. 6 from NASA Langley Research Center's Turbulence Modeling Resource (TMR) [35]. The case is based on experimental wind tunnel data from Coles and Wadcock; however, the CFD case differs from the experiment in that it is two-dimensional [36]. As such, it is a weak validation case for CFD [35]. The nominal validation case was run following the TMR setup with Riemann boundary conditions, an adiabatic solid wall for the airfoil, a Mach number (Ma) of 0.09 at an AoA of 13.87 deg, a nondimensional chord length (c) of 1, an operating temperature (T) of 536 deg R (297.8 deg K), and a Reynolds number (Re) based on chord length of 1.52 million. The initial conditions were specified to these nominal operating conditions, and the velocity inlet was set using AoA and Mach number with a zero-pressure outlet. The validation cases were run on the NASA codes CFL3D, FUN3D, and OVERFLOW, whereas our cases were run using ICFD++ from Metacomp Technologies [34]. The simulations took approximately 15 min to run on 20 cores. The comparisons of the coefficients of lift (C_l) and drag (C_d) for these codes are compiled in Table 2 and show that ICFD++ compares favorably to the TMR results [35]. The experiment, NASA validation data, and ICFD++ simulation coefficient of pressure around the airfoil are compared in Fig. 7 also.

Table 2 Coefficients of lift and drag from the validation cases of the NACA 4412 airfoil

Case	C_l	C_d
CFL3D	1.616	0.0311
FUN3D	1.615	0.0320
OVEFLOW	1.621	0.0321
ICFD++	1.620	0.0320

Next, the flow-field around the airfoil is visualized to ensure a proper solution and describe the region of interest in the simulation. Figure 8a is a psuedocolor plot of the velocity magnitude around the airfoil with velocity vectors superimposed throughout the domain. The separation bubble on top of the airfoil can be seen near the trailing edge as the large blue area of low velocity. This separation bubble responds nonlinearly to the flow parameters. As the separation bubble grows and shrinks under uncertain operating parameters, it effects the C_l of the airfoil greatly and the C_d to a lesser extent. Typically, the problem is made tractable by modeling the C_l and C_d responses to one parameter only, such as the AoA. Traditional metamodels can model only the multiparameter responses with a large number of simulations. Here, our goal is to model the bubble's multiparameter response accurately using fewer simulations.

For our RPC-K metamodeling study, 200 simulations were run of the problem geometry under uncertain operating conditions. We assigned normally distributed uncertainty to eight design parameters: the Mach number (Ma), the AoA, the temperature in kelvin (T), the Reynolds number (Re), the chord length (c), the molecular mass of air (M), the free-stream turbulence intensity (I), and the turbulent



a) Full airfoil **b) Separation region**

Fig. 8 Psuedocolor plots of velocity magnitude with vectors around the NACA 4412 airfoil.

viscosity ratio (μ_t/μ). The percent of uncertainty for each parameter varies based on realistic values, and the standard deviations for each parameter are compiled in Table 3. The Sobol quasi-random sequence was used to sample the parameter values for the 200 simulations based on the parameters' statistical moments.

The airfoil case was too computationally costly to conduct a full Monte Carlo study on, but the convergence of the C_l and C_d can be plotted for the 200 cases. Figure 9 plots these values normalized to their 200 simulation values. It can be seen that the coefficients converge quickly and the true values of these parameters for the stochastic problem are unlikely to vary beyond the significance of the given values.

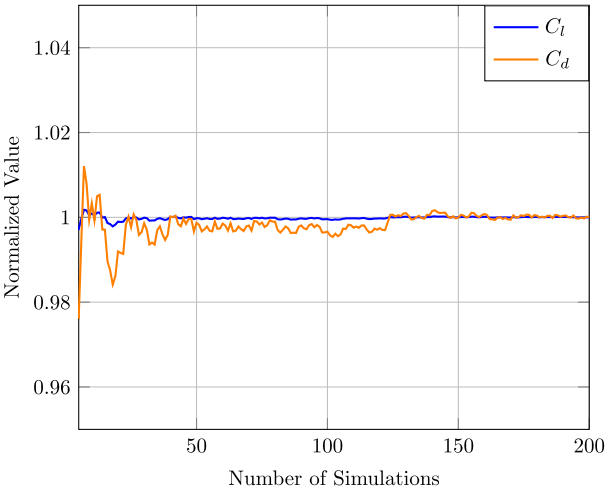


Fig. 9 Convergence of C_l and C_d for the NACA 4412 airfoil case.

2. *Engine Nacelle Case*

The engine nacelle case used in our previous studies [12,20,21] and based on research conducted by Nathan Spotts [37] was also analyzed using the hierarchical LARS algorithm to select the optimal RPC-K metamodel. Basic information is covered here, and details of the case can be found in the original work [12]. The problem analyzes the airflow conditions of a new nacelle design for a representative aircraft engine under an extreme operating condition, takeoff under a strong crosswind. The low-speed, high-AoA condition can cause separation of the airflow if the nacelle is sufficiently wide and short as is common in modern high-bypass turbofan engines. For computational simplicity, only half of the geometry was modeled, and the geometry included only the nacelle and engine hub as engine components. Figure 10 shows the surface mesh, including the nacelle walls, the engine hub, and the mass flow boundary towards the end of the nacelle. The cases were run as steady-state RANS solutions with the Spalart-Allmaras turbulence model solved on a 4.5-million-cell tetrahedral mesh. The far-field Riemann boundary condition was specified using the Mach number and AoA, the nacelle and hub walls were set to no-slip walls, and the pressure outlet was iterated to achieve a desired mass flow rate through the engine. The cases took approximately 3.5 h to run on 24 cores.

Figure 11 shows an axisymmetric schematic of the geometry, with a red line indicating where profiles of the airflow where taken inside the engine. The profile lines are located at the fan station of the engine. The fan and stator blades were not modeled for simplicity and are shown only in Fig. 11 for clarity on the profile location's importance.

The output variable of interest in this engine nacelle study is the total pressure P_0. The uncertainty quantification study varied seven input parameters: free-stream Mach number (Ma), AoA, mass flow rate ($\dot{m}$), turbulent-to-laminar viscosity ratio (μ_t/μ), nacelle diameter (D), nacelle length (L), and air density (ρ). These parameters were subject to uniform uncertainty of 15% of their mean. After a sensitivity study, the problem was reduced to five parameters ($n_r = 5$) by setting the viscosity ratio and air density to their mean values because they contributed very little variance to the problem. The parameter's means, minimums, and maximums are shown in Table 4. The total pressure profiles were taken along three axes at the fan station, $Y-$, $Z-$, and $Y+$ shown in Fig. 12. Results are shown for the flow along the $Y-$ profile because the flow separated here under a small range operating conditions and was thus the most difficult flow to construct a metamodel for due to the nonlinear physics governing its behavior. The flow separation is shown in Fig. 13, which shows total pressure on at the rotor station, flow streamlines colored by total pressure on the left, and streamline vectors on the right.

To compare results from multiple nacelle sizes, results in Figs. 14–16 are plotted along a nondimensional length, x, which starts at zero at the engine hub and ends at one at the nacelle's inner wall. The total pressure values are also nondimensionalized against their free-stream values.

B. Results and Discussion

1. *Airfoil Case*

First, the lift and drag on the airfoil were analyzed. The LOOCV error is plotted against the number of basis terms chosen by the hierarchical LARS algorithm for a 200-point data set obtained from

Table 3 Parameter mean and standard deviation values for the NACA 4412 case

Parameter	Ma	AoA	T	Re	c	M	I	μ_t/μ
Mean	0.09	13.87	297.7778	1,520,000	0.9012	28.95	0.086%	0.009
Standard deviation	0.003	0.4623	9.9259	25,333	0.006	0.0096	0.0287%	0.003

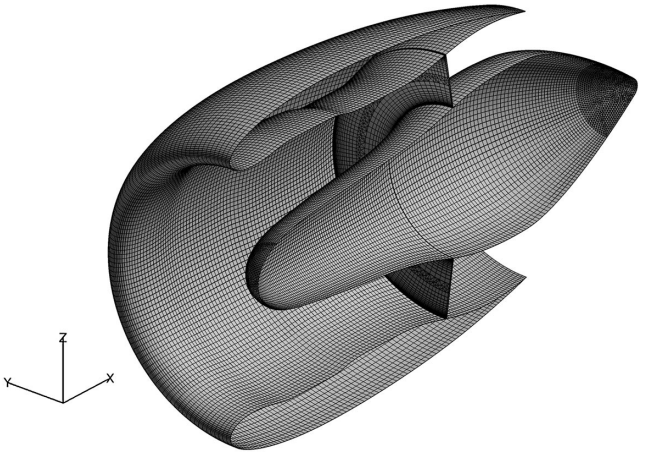


Fig. 10 Isometric view of the nacelle case surface mesh.

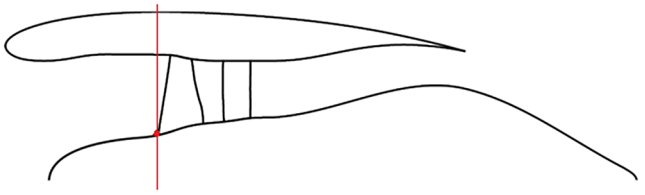


Fig. 11 Schematic, axisymmetric view of the nacelle geometry with the profile location denoted by the red line.

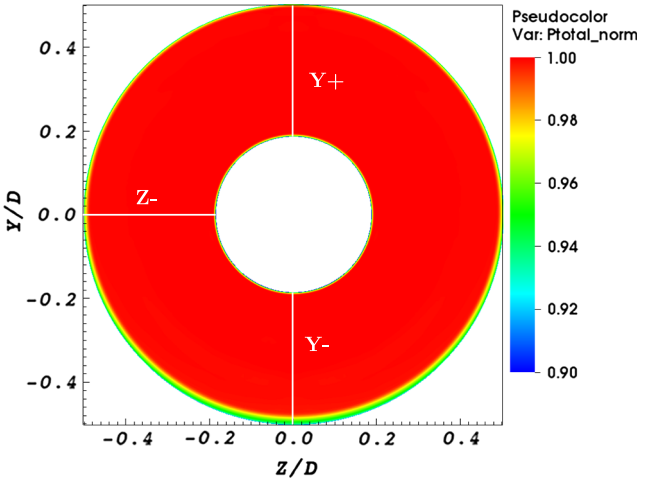


Fig. 12 Examined profiles for engine nacelle case on a pseudocolor plot of total pressure.

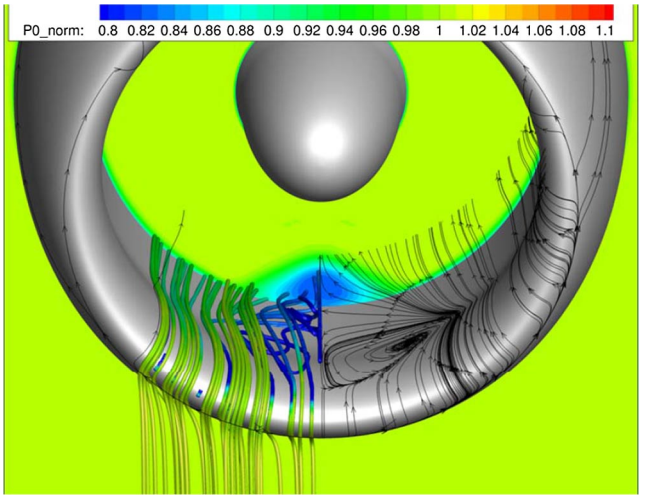


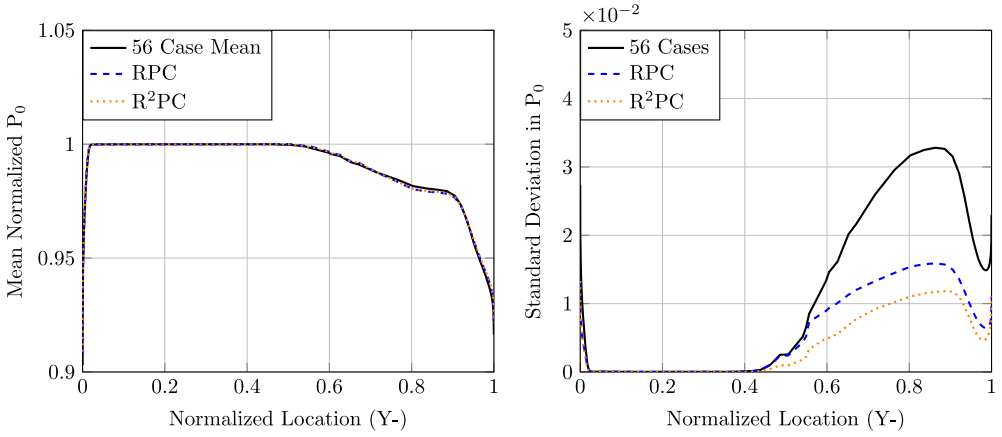
Fig. 13 Flow separation on bottom of nacelle under uncertain conditions.

ICFD++ simulations in Fig. 17, where Fig. 17a shows the PC results and Fig. 17b shows the PC-K results. It can be seen that only a few basis terms are needed to reduce the LOOCV error several orders of magnitude before it steadily rises. The PC metamodel with the smallest generalization error for C_d had 73 basis terms, whereas the PC metamodel for C_l had the least generalization error with seven basis terms. Both PC-K metamodels showed minimum generalization error with seven basis terms as well. These minimum error metamodels were selected as the optimal metamodels. Their statistics for the airfoil case are compiled in Table 5 and show good agreement for both the mean, $+3\sigma$, and -3σ values for the PC and PC-K metamodels. The OK metamodel, with no trend function, poorly predicts the standard deviation of C_l and C_d. The poor prediction of the variation for the OK metamodel is due to the strong trend in the data, which the metamodel does not fit because its only "basis" term is the mean of the problem. The data show that both the PC and PC-K methods can fit the uncertainty of the data accurately in the airfoil problem using 200 CFD simulations. However, the rather large 200-point data set is unrealistic for normal engineering CFD simulations, which typically model more complex three-dimensional geometries, requiring much greater computational cost per simulation. Next, we analyzed the metamodels' performance on a more reasonably sized subset of the first 50 simulations for this case.

It can be seen in Fig. 18 that the LOOCV error predicts increasing error for each method from the very beginning, and it does not give the quadratic behavior we are expecting for a generalization error measure. In fact, only the PC-K metamodel for C_l has a basis of more than the mean term, and it selects only three terms. The data in Table 6 show that the resulting metamodels are not effective as they model the mean values accurately, but they do not give accurate estimates of the standard deviation. The PC metamodels and the PC-K metamodel for C_d showed zero standard deviation because the metamodels have only the mean term in their basis. As such, they return the mean value for all input realizations with zero variation. The PC-K metamodel for C_l showed some variance because it included three basis terms, but these terms do not provide enough variance to match the observed values. Additionally, as in the 200-simulation case, OK fails to capture the variance of the problem because of the method's lack of any trend function to model global behavior. However, the standard deviation estimates for the PC-K metamodels can be improved by running a Monte Carlo simulation on the PC-K metamodel and taking the statistics from the Monte Carlo simulation instead of the estimated statistics from the PC-K metamodel's basis coefficients. These results are shown in the last column of Table 6, and they are better than the predicted statistics for coefficient data. The improved results are seen in the PC-K metamodel for C_d where the predicted standard deviation is 0, but the Monte Carlo simulation of the metamodel returns a modest standard deviation. The prediction for the PC-K metamodel for C_l is better because the trend function has three basis terms, but the Monte Carlo data improve the standard deviation estimate even further. The OK metamodel does not provide as accurate of estimates of the problem variance from the Monte Carlo simulation conducted on it because it has no trend function. Regardless, even the PC-K Monte Carlo does not provide accurate enough estimates. While the LOOCV error indicates that these metamodels have the lowest generalization error, they are clearly not

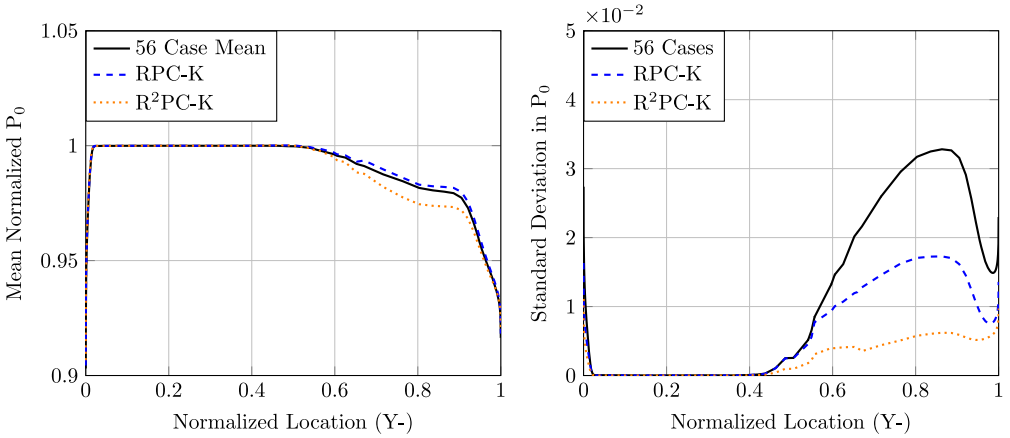
Table 4 Mean values for the 7 parameters in the nacelle study

	Mach	AOA, °	$\dot{m}$, kg/s	μ_t/μ	D/D_{ref}	L/L_{ref}	ρ, kg/m^3
Minimum	0.2125	22.1	20.4	4	0.85	0.85	1.225
Mean	0.25	26	24	4	1	1	1.225
Maximum	0.2875	29.9	27.6	4	1.15	1.15	1.225



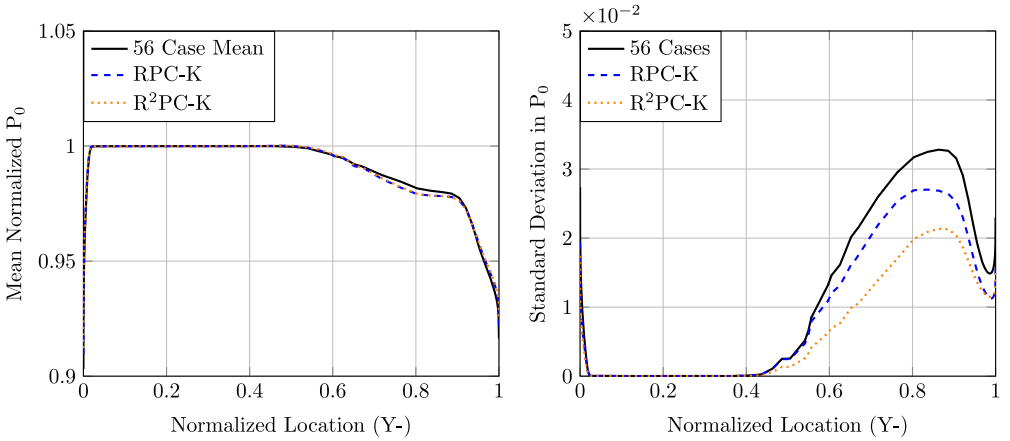
a) Mean P_0 for RPC and R^2PC b) Standard deviation in P_0 for RPC and R^2PC

Fig. 14 Mean and standard deviation of P_0 on the $Y-$ profile of the nacelle problem for PC metamodels.



a) Mean P_0 for RPC-K and R^2PC-K b) Standard deviation in P_0 for RPC-K and R^2PC-K

Fig. 15 Mean and standard deviation of P_0 on the $Y-$ profile of the nacelle problem for PC-K metamodels.

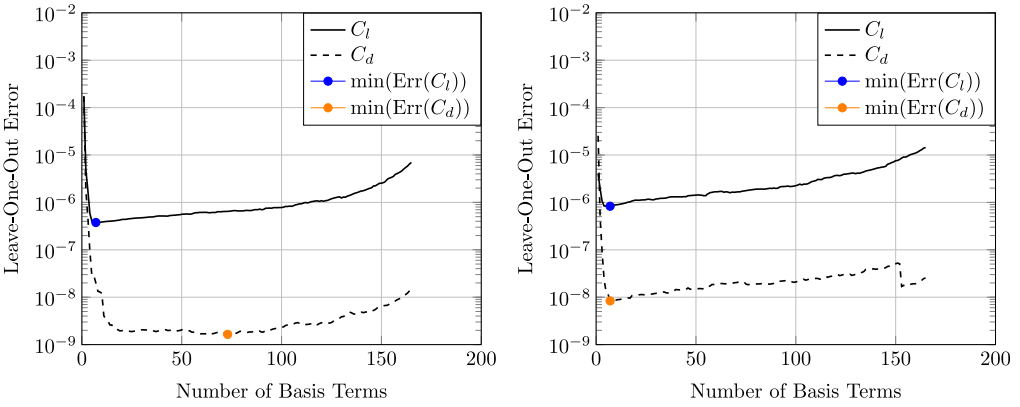


a) Mean P_0 for RPC-K and R^2PC-K b) Standard deviation in P_0 for RPC-K and R^2PC-K

Fig. 16 Mean and standard deviation of P_0 on the $Y-$ profile of the nacelle problem calculated from a Monte Carlo study of the PC-K metamodels.

the best metamodel in general as they do not accurately represent the problem variability.

The apparent failure of the PC- and PC-K-type methods to select the metamodel with minimal error can be explained by the curse of dimensionality. When working with very large parameter sets, even space-filling sampling methods like LHS and Sobol sequences will in fact only sample the boundary values of the problem due to the exponential increase in volume of the parameter space. This sampling issue is exacerbated with small data sets, and so the sampling is not sufficient to cover the entire design space when only a few design points are used per parameter. As a result, regression methods that rely on dense sampling for accurate fits experience fitting errors. Least squares



a) Polynomial chaos metamodel **b) Polynomial chaos-kriging metamodel**

Fig. 17 Leave-one-out cross-validation error progression of C_l and C_d with 200 data points.

regression fits the data based on minimizing the MSE of the problem. The MSE can be decomposed using the bias-variance decomposition to gain insight into its behavior. The decomposition gives

$$\text{Err}(x) = \epsilon^2 + \text{Bias}^2(\hat{f}(x)) + \text{Var}(\hat{f}(x)) \tag{16}$$

so the MSE is composed of bias error, variance error, and irreducible model error (ϵ) [30]. Underfitted models will have a high bias error and overfitted models will have a high variance error. Normally, least squares regression will select a low error solution with minimal bias and variance error. In problems with few data points per model parameter, the bias error is not easily reduced by incorporating additional basis terms, but the variance error grows rapidly. Thus, the least error solution is the one with just a single basis term. The underfitting problem cannot be fixed by using a different error measure such as the Akaike information criterion (AIC), the Bayesian information criterion (BIC), or the coefficient of determination (R^2), as all of the methods are based off of the MSE. However, the problem can be combated by manually selecting a metamodel from inspection, increasing the number of samples, or decreasing the number of parameters. Generally, manually fitting the model is not only time-consuming but also requires some knowledge of the solution and the approximate statistical moments. As such, manually fitting metamodels are inadvisable in many cases. Further, with the computational cost and additional time required to run additional CFD simulations, increasing the number of samples is generally impossible in practical application. This leaves reducing the number of parameters as a viable solution.

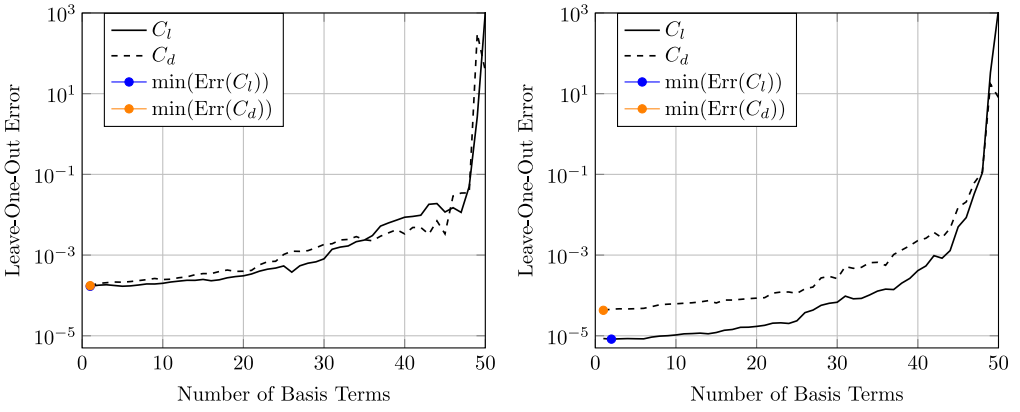
Table 5 Mean and standard deviations for C_l and C_d using metamodels on a 200-point data set

Model	200 simulations	PC	OK	PC-K
$C_l - 3\sigma$	1.5675	1.5671	1.6120	1.5670
Mean C_l	1.6168	1.6168	1.6170	1.6167
$C_l + 3\sigma$	1.6662	1.6665	1.6210	1.6664
$C_d - 3\sigma$	0.025144	0.025172	0.031986	0.025170
Mean C_d	0.032335	0.032312	0.032333	0.032322
$C_d + 3\sigma$	0.039527	0.039452	0.032681	0.039474

The number of parameters in the airfoil problem was reduced using a sensitivity study. Sobol sensitivity indices were constructed from the coefficients of a PC model constructed with only main parameter effects up to third order. The sensitivity indices are given as

$$S_j = \frac{\sigma_j^2}{\sigma_U^2} = \frac{\sum_{i=1}^{m}(\hat{U}_{ki}^j)^2}{\sum_{j=1}^{n}\sum_{i=1}^{m}(\hat{U}_{ki}^j)^2} \tag{17}$$

where $\sigma_U^2 = \sigma_1^2 + \sigma_2^2 + \cdots + \sigma_n^2$ [12,13,38,39], m is the order of the polynomials used in the PC expansion, and U_{ki}^j is the fitted coefficient for the ith-order polynomial basis of parameter j. Because of their form, the sensitivity indices add up to 1, where S_j is the estimated variance in the output due to parameter j. For the airfoil case, the coefficients of the PC metamodel with a basis of main parameter effects only are fit to the 50 airfoil cases, and they are used to calculate the sensitivity indices. From these indices, it was found that the AoA dominated the problem, and the problem could easily be reduced to the same four parameters for both C_l and C_d. All of the indices are compiled in Table 7. The four retained parameters are AoA, Ma, c, and Re, which show an order of magnitude more effect than the four removed parameters.



a) Polynomial chaos metamodel **b) Polynomial chaos-kriging metamodel**

Fig. 18 Leave-one-out cross-validation error progression of C_l and C_d with 50 data points.

Table 6 Mean and standard deviations for C_l and C_d using metamodels on a 50-point data set

Model	200 simulations	PC	OK	PC-K	PC-K Monte Carlo
$C_l - 3\sigma$	1.5675	1.6164	1.6077	1.6152	1.5732
Mean C_l	1.6168	1.6164	1.6174	1.6167	1.6174
$C_l + 3\sigma$	1.6662	1.6164	1.6271	1.6181	1.6615
$C_d - 3\sigma$	0.025144	0.032279	0.031010	0.032383	0.027503
Mean C_d	0.032335	0.032279	0.032390	0.032383	0.032383
$C_d + 3\sigma$	0.039527	0.032279	0.033774	0.032383	0.037263

Table 8 Mean and standard deviations for C_l and C_d using metamodels on a 50-point data set and reduced parameter set

Model	200 simulations	RPC	OK	RPC-K	RPC-K Monte Carlo
$C_l - 3\sigma$	1.5675	1.5669	1.6146	1.5655	1.5676
Mean C_l	1.6168	1.6169	1.6170	1.6168	1.6169
$C_l + 3\sigma$	1.6662	1.6668	1.6193	1.6682	1.6661
$C_d - 3\sigma$	0.025144	0.025202	0.032003	0.025132	0.025179
Mean C_d	0.032335	0.032323	0.032332	0.032287	0.032328
$C_d + 3\sigma$	0.039527	0.039444	0.032662	0.039442	0.039477

The new RPC and RPC-K metamodels were then fit to the 50-dataset airfoil problem. The resulting LOOCV errors are shown in Fig. 19. Additionally, the OK metamodel was also re-fit using these four parameters. It can be seen that now the LOOCV error is initially high and decreases with an increasing number of basis terms, as desired in generalization error measures. The models generally are optimal at 12 basis terms, but the RPC metamodel for C_d is optimal at 17 basis terms. The first basis term chosen in these models was the first-order term for the AoA, the second term was the second-order term for AoA, and then LARS selected the first-order terms for Reynolds number, chord length, and Mach number, respectively. This order matches the physics of the problem as the coefficients are heavily dependent on the AoA. Therefore, the basis set required two terms to model the effect of the AoA before including the first-order effects of the other problem parameters.

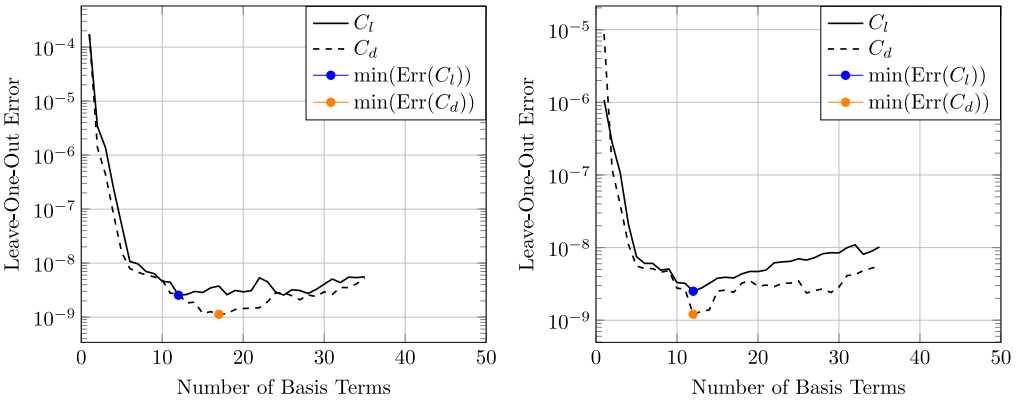
The statistics of these improved models is shown in Table 8, which shows the dramatic improvement in the metamodels selected by the hierarchical LARS algorithm through their increased accuracy for predicting the standard deviation of the problem. The OK metamodel performs poorly again as it is unable to capture the variance of the problem. The RPC and RPC-K metamodels both accurately predict the variance in C_d, but they overpredict the variance in C_l, with the RPC metamodel showing less error. These metamodels are less accurate for the C_l as the data show stronger trends. However, the accuracy of the RPC-K metamodel can be improved by conducting a Monte Carlo simulation of the metamodel to obtain the standard deviation estimate instead of using the basis coefficients. The Monte Carlo simulation on the RPC-K metamodel produces the most accurate uncertainty data of all of the metamodels fit to the reduced data set. While a Monte Carlo simulation of the RPC metamodel will just return the statistics already obtained from the metamodel's coefficients, the Monte Carlo simulation of the RPC-K metamodel includes additional correlation information that increases its accuracy on the problem. The Monte Carlo results do take additional computations, but it is the best way to recover the statistics of a kriging-based metamodel as the estimates from the trend function can be unreliable, as demonstrated. This behavior makes the RPC-K metamodel superior to the other models for the case because the RPC-K metamodel can capture the basic global trend information of the nonlinear effects, but it can give improved estimates of the problem statistics through a Monte Carlo simulation of the metamodel. The relative percent errors for the metamodels built on the reduced parameter set compared with the 200 simulations are shown in Table 9. This relative percent error value is calculated as the mean of the 200 simulations minus the mean from the metamodel, divided by the mean from the 200 simulations and multiplied by 100. The given values are the percentage difference between the metamodel values and values calculated from the 200 simulations. The OK metamodel produces errors of almost 100% of the true value for the standard deviation as it cannot capture the variability of the problem. The RPC metamodel is quite accurate, though it shows approximately 1% error in the

Table 9 Relative percent errors of the studied metamodels

Model	RPC	OK	RPC-K	RPC-K Monte Carlo
Error in mean C_l	0.0062	0.0120	0.0025	0.0061
Error in σ C_l	1.0468	95.341	4.1255	0.3981
Error in mean C_d	0.0371	0.0093	0.1484	0.0216
Error in σ C_d	0.9746	95.425	0.5031	0.5834

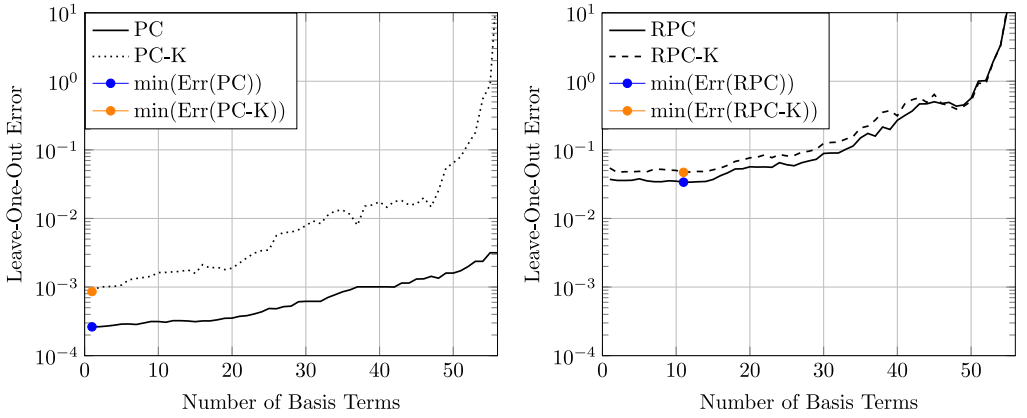
Table 7 Sensitivity indices for the airfoil parameters

Parameter	*Ma*	AoA	*T*	*Re*	*c*	*M*	*I*	μ_t/μ
S_j for C_l	2.25e-04	0.9924	1.80e-05	0.0063	0.0011	1.45e-06	1.79e-06	7.31e-06
S_j for C_d	3.94e-05	0.9978	1.07e-05	0.0018	3.17e-04	1.44e-06	9.97e-07	5.44e-06



a) Reduced polynomial chaos metamodel **b) Reduced polynomial chaos-kriging metamodel**

Fig. 19 Leave-one-out cross-validation error progression of C_l and C_d with 50 data points on a reduced parameter set.



a) LOOCV error for PC and PC-K **b) LOOCV error for RPC and RPC-K**

Fig. 20 Mean and standard deviation of P_0 on the $Y-$ profile of the nacelle problem for the PC and OK/PC-K metamodels.

prediction of the standard deviation for both C_l and C_d. The RPC-K metamodel produces good results except for the prediction of the standard deviation in C_l, which shows 4% error. The Monte Carlo study of the RPC-K metamodel produces the most uniformly accurate statistics as it gives errors below 1% for every variable of interest in this study.

The airfoil case demonstrates how the parameter reduction technique can be applied to an aerodynamics case to improve the fit of both PC- and PC-K-type metamodels. Importantly, we show that using a hierarchical LARS algorithm on the RPC-K metamodel, along with a Monte Carlo simulation of the metamodel, can provide better statistics than existing methods. These improved results are achieved by leveraging both the correlation of the data and a polynomial trend function built with logical bases from a hierarchical method.

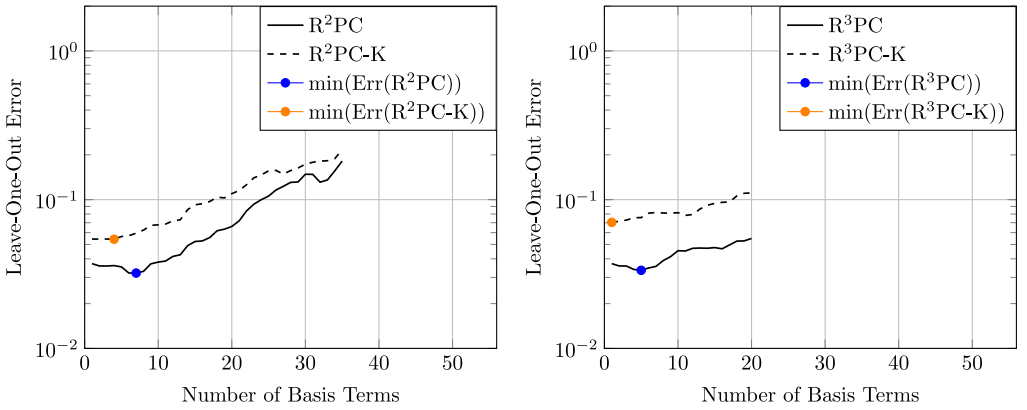
2. *Engine Nacelle Case*

Our previous work [12,20,21] found that the PC and PC-K metamodels were prone to overfitting the engine nacelle case, leading to the development of the RPC and RPC-K metamodels. As a reference, the LOOCV error for varying basis sizes of the PC and PC-K metamodels are plotted in Fig. 20a and the LOOCV errors of the RPC and RPC-K metamodels in Fig. 20b. The RPC and RPC-K models were fit to five parameters, Ma, AoA, $\dot{m}$, D/D_{ref}, and L/L_{ref}, with a maximum exponent of degree 3 for a total of 56 basis terms using 56 data points. The PC and PC-K metamodels used the same five parameters plus the turbulent viscosity ratio (μ_t/μ) and the air density (ρ) parameters from the full investigation with polynomials with a maximum exponent of three as well. As expected, the PC and PC-K metamodels underfit the problem with the minimum LOOCV error metamodels having one basis term. Their LOOCV error then increases steadily until the problem is overfit. It can be seen that the RPC and RPC-K metamodels both suffer less from underfitting as they have 11 basis terms in their trend function. Though, with no large decrease in the LOOCV error for the RPC and RPC-K metamodel initially, the metamodels are likely still suffering from some underfitting issues. However, it does show that the metamodels eventually overfit the data once enough basis terms are added, as the error increases exponentially. At 56 data points, the LOOCV error for the RPC metamodel is 1857 and it is 22,986 for the RPC-K metamodel. As in the validation case, the LOOCV errors suggest that the PC metamodel will be the most accurate. Again, it is important to not assume that LOOCV errors can be directly compared between metamodel types as their relative errors are not accurate. Before comparing the results of the PC, RPC, PC-K, and RPC-K metamodels, the RPC and RPC-K metamodels are further analyzed for additional parameter reductions.

Table 10 Sensitivity indices for the nacelle parameters

Parameter	Ma	AoA	$\dot{m}$	D/D_{ref}	L/L_{ref}
S_j total	0.0742	0.1784	0.2689	0.2984	0.1802
S_j separation	0.0426	0.2550	0.1642	0.2837	0.2544

To combat the initial model underfitting, another sensitivity analysis was run on the reduced nacelle case to remove further nonessential parameters. The sensitivity indices are compiled in Table 10 for both the whole domain and the region of interest, the separated flow at the bottom of the nacelle. It can be seen that the relative impact of the parameters varies from the smooth flow regions to the



a) RPC and RPC-K metamodels using four parameters **b) RPC and RPC-K metamodels using three parameters**

Fig. 21 Leave-one-out cross-validation error progression for further reduced nacelle problems.

Table 11 LOOCV error for the nacelle metamodels

Model	RPC	RPC-K	R^2PC	R^2PC-K	R^3PC	R^3PC-K
min(Err)	3.38×10^{-2}	4.70×10^{-2}	3.21×10^{-2}	5.41×10^{-2}	3.35×10^{-2}	7.04×10^{-2}
Terms	11	11	7	4	5	1

separated flow region, but the Mach number is always the least impactful parameter. As such, the twice-reduced problem removed Ma from the parameter list, and the resulting metamodels fit to the four remaining parameters are indicated as R^2PC and R^2PC-K because they have had the parameter space reduced twice. Then, the problem was further reduced by removing $\dot{m}$ from the parameter space as it is the next least impactful parameter in the region of interest. The metamodels constructed from the remaining three parameter set are indicated as R^3PC and R^3PC-K, and they are used to measure the efficacy of a very small parameter sets.

The LOOCV error progression for the R^2PC and R^2PC-K metamodels is shown in Fig. 21a and the LOOCV error progression for the R^3PC and R^3PC-K metamodels is shown in Fig. 21b. The R^2PC and R^3PC metamodels do show some of the characteristic drop in the LOOCV error when the initial basis terms are added, whereas the R^2PC-K and R^3PC-K metamodels do not. Altogether, reducing the number of parameters does not necessarily prevent underfitting. Additionally, all of the models have minimum LOOCV errors with progressively fewer basis terms as the parameter space is reduced. The R^2PC metamodel has a minimum error with seven terms, the R^3PC with five terms, the R^2PC-K metamodel with four terms, and the R^3PC-K with one. The minimum errors for the metamodels are compiled in Table 11 and show that all of the RPC metamodels have similar errors, whereas the errors for the RPC-K metamodels steadily increase when additional parameters are removed.

The training errors for a metamodel can also be analyzed to determine how well the data are being fit. An underfit model will have large training errors and an overfit model will have almost zero error. The training errors for the RPC, R^2PC, and R^3PC metamodels are shown in Fig. 22. It shows that the training error stays quite large for the RPC metamodels until the model ultimately becomes overfitted when the number of terms is close to the number of data points and the training error exponentially decreases to zero. For the R^2PC and R^3PC metamodels, the error steadily decreases and shows no sign of overfitting. The training error data suggest that the RPC metamodels can have close to 30 basis terms before suffering from overfitting issues, and the current metamodels are all underfitting the data due to the small number of data points. Additionally, trying to model the data with only three parameters does not work as while the number of samples per parameter is much higher, there is not enough fidelity in the R^3PC metamodel to capture the underlying physics of the problem. Meanwhile, the training errors for the RPC-K metamodels are zero as kriging metamodels interpolate data points.

The resulting data of these metamodels can then be plotted against the data from the 56 full evaluations. The total pressure is plotted along the normalized $Y-$ profile where zero is the hub wall and one is the nacelle wall. The mean and standard deviations are plotted to provide an estimate of the problem mean and realistic deviations. Critically, because only 56 CFD simulations were available due to computational cost, the data shown may differ from the model truth values, though it should provide an accurate estimate. First, the results of the PC and PC-K metamodels are plotted in Fig. 23. Because the PC-K metamodel has only one basis term, it reverts back to an OK metamodel and the model can be referred to as either the OK or the PC-K for this particular case because they have the same form. Both the PC and the OK/PC-K metamodel show good results for the mean P_0 in Fig. 23a, with the PC model performing best as it simply recovers the mean of the 56 cases it is built upon. However, when the standard deviation in P_0 is plotted in Fig. 23b, the PC model fails as the single-basis metamodel shows no variance. The OK/PC-K metamodel preforms much better as it recovers a little more than half of the variance. The results of the RPC and R^2PC metamodels are shown in Fig. 14, and the results of the RPC-K and RPC-K

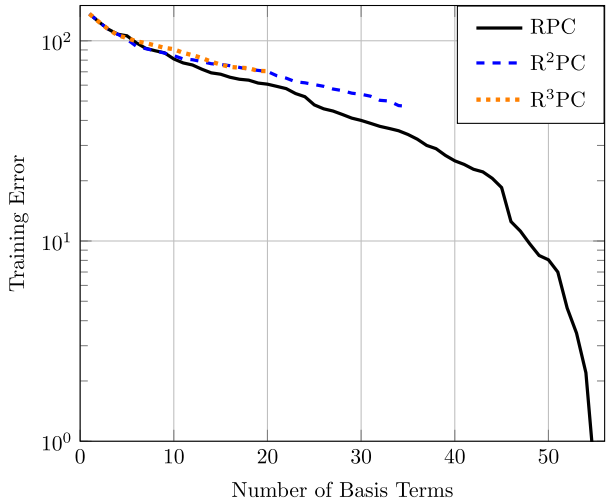
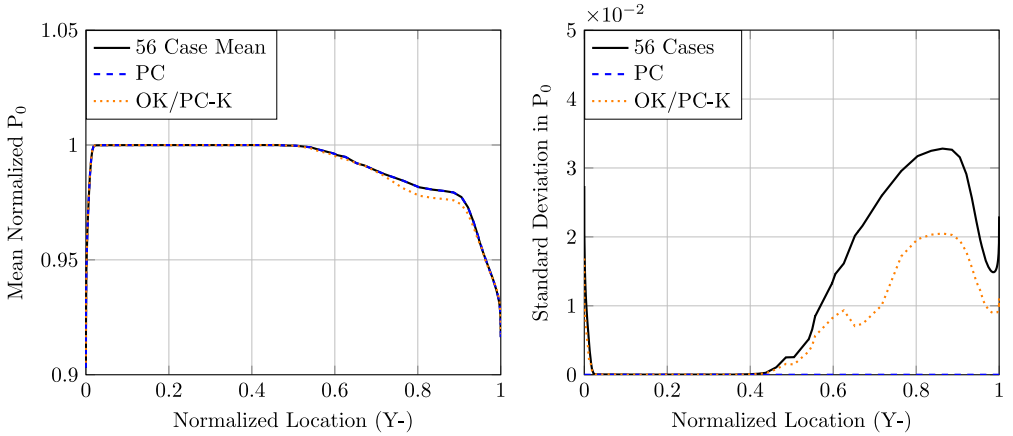


Fig. 22 Training error for the RPC, R^2PC, and R^3PC metamodels.



a) Mean P_0 for PC and OK/PC-K **b) Standard deviation in P_0 for PC and OK/PC-K**

Fig. 23 Mean and standard deviation of P_0 on the $Y-$ profile of the nacelle problem for the PC and OK/PC-K metamodels.

Table 12 Error for the nacelle metamodels

Model	PC	RPC	R^2PC	OK/PC-K	RPC-K	R^2PC-K	RPC-K MC	R^2PC-K MC
Mean MSE	0	5.79×10^{-6}	2.16×10^{-5}	1.57×10^{-6}	1.58×10^{-6}	7.37×10^{-5}	9.25×10^{-6}	1.89×10^{-5}
σ MSE	1	2.39×10^{-1}	2.64×10^{-1}	1.45×10^{-1}	1.80×10^{-1}	3.80×10^{-1}	6.70×10^{-2}	1.40×10^{-1}

metamodels are shown in Fig. 15. The R^3PC and R^3PC-K metamodels are not compared due to their inferior error measures. It can be seen that the mean of the data is well approximated by all of the models; however, the standard deviations of the data are greatly underpredicted. At best, the RPC and RPC-K metamodels predict about half of the standard deviation seen in the data. This error in the standard deviation estimate is significant and means that the variance of the metamodels chosen by the LOOCV error is too small because the problem is underfitted. However, knowing that the standard deviation values given by the polynomial trend of the RPC-K metamodels may not match its Monte Carlo results based on the airfoil results, a Monte Carlo study was conducted on both the RPC-K and R^2PC-K metamodels. These results are shown in Fig. 16. Again, the standard deviation values of the Monte Carlo study of the RPC-K metamodel outperform those of the RPC metamodel and the statistical moments given by the polynomial coefficients of the RPC-K metamodel alone.

The average MSEs of the PC, RPC, OK/PC-K, RPC-K, and Monte Carlo study of the RPC-K metamodel relative to the 56 cases are compiled in Table 12. It can be seen that the error difference of the mean values for all of the metamodels is very small and within the standard error of the simulations. All metamodels showed greater error in the standard deviation estimates, though the once-reduced metamodels fared better than the twice-reduced metamodels. Uniquely, the OK/PC-K metamodel produced excellent results with an MSE of 1.45×10^{-1}, which was only improved upon by the RPC-K and R^2PC-K Monte Carlo data. This performance contrasts the OK metamodel's performance in the airfoil case where it performed the worst in all cases. The Monte Carlo data of the RPC-K metamodel was superior to all other metamodels with an MSE of 6.70×10^{-2} for the standard deviation of P_0.

For the RPC-K metamodel, the hierarchical LARS algorithm selected the basis term for the AoA first, followed by the mass flow rate, Mach number, nacelle diameter, and finally the nacelle length. Selecting the AoA first follows from an understanding of aerodynamic flow separation, which occurs most readily when the aerodynamic object is not aligned with the flow. However, the selection of the mass flow rate basis term before the Mach number is not clearly intuitive as the speed of a flow is one of the main parameters used to determine flow separation. This is an example of where the automated basis selection of LARS can outperform engineering judgment because it uses the correlation information of the input parameters directly to compute the basis order.

Between the airfoil case and the engine nacelle case, the RPC-K metamodel using LARS outperformed every other metamodel studied. The other metamodels included a representative regression-based surface response method in PC, a representative distance-weighted average method in OK, and the optimal PC-K method of Schöbi and Sudret [19].

VI. Conclusions

Our work shows that the RPC-K method can be effectively used to help analyze aerodynamics problems as an accurate metamodel. The RPC-K metamodel can then be used to efficiently generate data in the presence of uncertainty or estimate the response of off-design points. The research has also extended earlier research into how PC-K methods can be adapted for very expensive nonlinear fluid dynamics problems through reducing parameter spaces and efficiently using LOOCV error along with least-angle regression (LARS) to select sparse polynomial expansions.

The creation of a robust metamodeling process is not straightforward for PC-K methods as there are many parameters that can be changed. Most critically, a good sampling method and sufficient sampling of a problem are needed to produce the best results. When constrained to small sample sets, PC-K can provide good results by using a model selection algorithm such as LARS to select only the most important terms for the trend function. Our work shows that using a hierarchical LARS algorithm provides good results and produces models that can be easily understood due to the heredity of the terms included in the model. These models must be selected based on their relative accuracy, which is difficult to accurately assess on sparse problems with standard methods. It is possible that better metamodels can be constructed when manually fit, but the additional time cost negates some of the benefits of the metamodeling process. To assist the process, sensitivity studies can be performed and used to reduce the problem to fewer parameters to increase the robustness of the methods and still provide sufficiently accurate results. Polynomial chaos-kriging metamodels constructed from these reduced parameter spaces are superior to PC metamodels as they provide not only trend information and quick statistical moments, but also correlation information and improved Monte Carlo results. Finally, PC-K methods may still need fine-tuning of the polynomial form, autocorrelation function, and optimization algorithms used to get the best results.

Future work will need to be conducted on the robustness of estimating the polynomial coefficients of PC-K metamodels as they show great sensitivity to the output data form. Our work on the CDR problem indicates that standard least squares regression techniques may be too unreliable to fully automate the process. Additionally, our work indicates that new error measures for these metamodels, which would more reliably work in high-dimensional, sparse problems, would alleviate much of the manual inspection needed in their construction. More complex validation cases will always be needed to help the technology mature and become more well-validated. Ultimately, the RPC-K metamodel will help facilitate practical engineering designs by providing computational fluid dynamics practitioners with statistical data for computational design efficiency.

Acknowledgments

The authors would like to thank Metacomp Technologies Inc. for providing their ICFD++ software for conducting the simulations in this study. The authors would also like to thank Nathan Spotts for his help in preparing figures for the engine nacelle case.

References

[1] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, 1989, pp. 409–423. doi:10.1214/ss/1177012413

[2] Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, No. 4, 1998, pp. 455–492. doi:10.1023/A:1008306431147

[3] Martin, J. D., and Simpson, T. W., "Use of Kriging Models to Approximate Deterministic Computer Models," *AIAA Journal*, Vol. 43, No. 4, 2005, pp. 853–863. doi:10.2514/1.8650

[4] Rasmussen, S. E., and Williams, C. K. I., *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, MA, 2006, Chaps. 2, 4.

[5] Wang, 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, 2006, pp. 370–380. doi:10.1115/1.2429697

[6] Najm, H. N., "Uncertainty Quantification and Polynomial Chaos Techniques in Computational Fluid Dynamics," *Annual Review of Fluid*

Mechanics, Vol. 41, No. 1, 2009, pp. 35–52.
doi:10.1146/annurev.fluid.010908.165248

[7] Sudret, B., "Meta-Models for Structural Reliability and Uncertainty Quantification," *Asian-Pacific Symposium on Structural Reliability and its Applications*, Research Publishing, Voillans, France, 2012, pp. 1–24.

[8] Simpson, T. W., Peplinski, J. D., Koch, P. N., and Allen, J. K., "Metamodels for Computer-Based Engineering Design: Survey and Recommendations," *Engineering with Computers*, Vol. 17, No. 2, 2001, pp. 129–150.
doi:10.1007/PL00007198

[9] Fajraoui, N., Marelli, S., and Sudret, B., "Sequential Design of Experiment for Sparse Polynomial Chaos Expansions," *SIAM/ASA Journal on Uncertainty Quantification*, Vol. 5, No. 1, 2017, pp. 1061–1085.
doi:10.1137/16M1103488

[10] Xiu, D., and Karniadakis, G. E., "The Wiener-Askey Polynomial Chaos for Stochastic Differential Equations," *SIAM Journal Science Computing*, Vol. 24, No. 2, 2002, pp. 619–644.
doi:10.1137/S1064827501387826

[11] Blatman, G., and Sudret, B., "Adaptive Sparse Polynomial Chaos Expansion Based on Least Angle Regression," *Journal of Computational Physics*, Vol. 230, No. 6, 2011, pp. 2345–2367.
doi:10.1016/j.jcp.2010.12.021

[12] Gao, 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," *54th AIAA Aerospace Sciences Meeting*, AIAA Paper 2016-5057, 2016.
doi:10.2514/6.2016-5057

[13] Sobol, I., "Global Sensitivity Indices for Nonlinear Mathematical Models and Their Monte Carlo Estimates," *Mathematics and Computers in Simulation*, Vol. 55, No. 1, 2001, pp. 271–280.
doi:10.1016/S0378-4754(00)00270-6

[14] Blatman, G., and Sudret, B., "An Adaptive Algorithm to Build up Sparse Polynomial Chaos Expansions for Stochastic Finite Element Analysis," *Probabilistic Engineering Mechanics*, Vol. 25, No. 2, 2010, pp. 183–197.
doi:10.1016/j.probengmech.2009.10.003

[15] Blatman, G., and Sudret, B., "Efficient Computation of Global Sensitivity Indices Using Sparse Polynomial Chaos Expansions," *Reliability Engineering and System Safety*, Vol. 95, No. 11, 2010, pp. 1216–1229.
doi:10.1016/j.ress.2010.06.015

[16] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R., "Least Angle Regression," *The Annuals of Statistics*, Vol. 32, No. 2, 2004, pp. 407–499.
doi:10.1214/009053604000000067

[17] Schöbi, R., Sudret, B., and Wiart, J., "Polynomial-Chaos-Based Kriging," *International Journal for Uncertainty Quantification*, Vol. 5, No. 2, 2015, pp. 171–193.
doi:10.1615/Int.J.UncertaintyQuantification.2015012467

[18] Webster, R., and Oliver, M. A., *Geostatistics for Environmental Scientists*, Wiley, West Sussex, England, 2007, Chaps. 4, 5, 8, 9.

[19] Schöbi, R., and Sudret, B., "PC-Kriging: a New Metamodelling Method Combining Polynomial Chaos Expansions and Kriging," *Proceedings of the 2nd International Symposium on uncertainty Quantification and Stochastic Modeling*, Eduardo Souza de Cursi, INSA-Rouen, Rouen, France, 2014,
doi:10.3929/ethz-a-010238689

[20] Weinmeister, J., Xie, N., Gao, X., Prasad, A. K., and Roy, S., "Combining a Reduced Polynomial Chaos Expansion Approach with Universal Kriging for Uncertainty Quantification," *8th Theoretical Fluid Mechanics Conference*, AIAA Paper 2017-3481, 2017.
doi:10.2514/6.2017-3481

[21] Weinmeister, J., Xie, N., Gao, X., Prasad, A. K., and Roy, S., "Analysis of a Polynomial Chaos-Kriging Metamodel for Uncertainty Quantification in Aerospace Applications," *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA Paper 2018-0911, 2018.
doi:10.2514/6.2018-0911

[22] Kersaudy, P., Sudret, B., Varsier, N., Picon, O., and Wiart, J., "A New Surrogate Modeling Technique Combining Kriging and Polynomial Chaos Expansions—Application to Uncertainty Analysis in Computational Dosimetry," *Journal of Computational Physics*, Vol. 286, April 2015, pp. 103–117.
doi:10.1016/j.jcp.2015.01.034

[23] Saltelli, A., Tarantola, S., and Campolongo, F., *Sensitivity Analysis in Practice: A Guide to Assesing Scientific Models*, Wiley, Sussex, England, 2004, Chap. 5.

[24] Prasad, A. K., and Roy, S., "Global Sensitivity Based Dimension Reduction for Fast Variability Analysis of Nonlinear Circuits," *24th IEEE Conference on Electrical Performance of Electronic Packages*, Curran Associates, Inc., Red Hook, NY, 2015, pp. 97–99.
doi:10.1109/EPEPS.2015.7347138

[25] Gratiet, L. L., Marelli, S., and Sudret, B., *Metamodel-Based Sensitivity Analysis: Polynomial Chaos Expansions and Gaussian Processes*, Springer International Publishing, New York, 2016, pp. 1–37.
doi:10.1007/978-3-319-11259-6_38-1

[26] Prasad, A. K., and Roy, S., "Accurate Reduced Dimensional Polynomial Chaos for Efficient Uncertainty Quantification of Microwave/RF Networks," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 65, No. 10, 2017, pp. 3697–3708.
doi:10.1109/TMTT.2017.2689742

[27] Jakeman, J. D., Eldred, M. S., and Sargsyan, K., "Enhancing l1-Minimization Estimates of Polynomial Chaos Expansions Using Basis Selection," *Journal of Computational Physics*, Vol. 289, May 2015, pp. 18–34.
doi:10.1016/j.jcp.2015.02.025

[28] Hesterberg, T., Choi, N. H., Meier, L., and Fraley, C., "Least Angle and l_1 Penalized Regression: A Review," *Statistics Surveys*, Vol. 2, May 2008, pp. 61–93.
doi:10.1214/08-SS035

[29] Yuan, M., Joseph, V. R., and Lin, Y., "An Efficient Variable Selection Approach for Analyzing Designed Experiments," *Technometrics*, Vol. 49, No. 4, 2007, pp. 430–439.
doi:10.1198/004017007000000173

[30] Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning*, Springer, New York, 2017, Chap. 7.

[31] Santer, T. J., Williams, B. J., and Notz, W. I., *The Design and Analysis of Computer Experiments*, Springer, New York, 2003, Chap. 3.

[32] Dubrule, O., "Cross Validation of Kriging in a Unique Neighborhood," *Mathematical Geology*, Vol. 15, No. 6, 1983, pp. 687–699.
doi:10.1007/BF01033232

[33] Schöbi, R., Sudret, B., and Marelli, S., "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, Paper D4016002.
doi:10.1061/AJRUA6.0000870

[34] "ICFD++," Ver. 15.6, Computer Software, Metacomp Technologies Inc., Agoura Hills, CA, 2015, http://www.metacomptech.com.

[35] Rumsey, C., "2DN44: 2D NACA 4412 Airfoil Trailing Edge Separation," March 2018, https://turbmodels.larc.nasa.gov/naca4412sep_val.html.

[36] Coles, D., and Wadcock, A. J., "Flying-Hot-Wire Study of Flow Past an NACA 4412 Airfoil at Maximum Lift," *AIAA Journal*, Vol. 17, No. 4, 1979, pp. 321–329.
doi:10.2514/3.61127

[37] Spotts, N., "Unsteady Reynolds-Averaged Navier-Stokes Simulations of Inlet Flow Distortion in the Fan System of a Gas-Turbine Engine," Master's Thesis, Colorado State Univ., Fort Collins, CO, 2015, http://hdl.handle.net/10217/170387.

[38] Blatman, G., and Sudret, B., "Sparse Polynomial Chaos Expansions and Adaptive Stochastic Finite Elements Using a Regression Approach," *Comptes Rendus Mécanique*, Vol. 336, No. 6, 2008, pp. 518–523.
doi:10.1016/j.crme.2008.02.013

[39] Satelli, A., Tarantola, S., and Campolongo, F., "Sensitivity Analysis as an Ingredient of Modeling," *Statistical Science*, Vol. 15, No. 4, 2000, pp. 377–395, http://www.jstor.org/stable/2676831.

K. E. Willcox
Associate Editor