AIAA JOURNAL
Vol. 40, No. 10, October 2002

Computationally Inexpensive Metamodel Assessment Strategies

Martin Meckesheimer* and Andrew J. Booker†
The Boeing Company, Bellevue, Washington 98124
and
Russell R. Barton‡ and Timothy W. Simpson§
Pennsylvania State University, University Park, Pennsylvania 16802

In many scientific and engineering domains, it is common to analyze and simulate complex physical systems using mathematical models. Although computing resources continue to increase in power and speed, computer simulation and analysis codes continue to grow in complexity and remain computationally expensive, limiting their use in design and optimization. Consequently, many researchers have developed different metamodeling strategies to create inexpensive approximations of computationally expensive computer simulations. These approximations introduce a new element of uncertainty during design optimization, and there is a need to develop efficient methods to assess metamodel validity. We investigate computationally inexpensive assessment methods for metamodel validation based on leave-*k*-out cross validation and develop guidelines for selecting *k* for different types of metamodels. Based on the results from two sets of test problems, $k = 1$ is recommended for leave-*k*-out cross validation of low-order polynomial and radial basis function metamodels, whereas $k = 0.1N$ or $\sqrt{N}$ is recommended for kriging metamodels, where *N* is the number of sample points used to construct the metamodel.

Nomenclature

N	=	number of sample points
$\boldsymbol{x}$	=	design (input) variable
$\boldsymbol{y}$	=	actual output (response) value
$\hat{y}_i$	=	predicted output (response) value from metamodel

I. Introduction

MATHEMATICAL models are widely used to analyze and simulate complex real world systems in many scientific and engineering domains. In many cases, the mathematical representation of the physical system is used to develop several computer modules that interact with each other and capture the input–output relationship in scientific and engineering problems. Because individual modules of such a computer structure, for example, a finite element model or simulation, are often computationally expensive, researchers have investigated the use of different approximation strategies, for example, response surface methods, as inexpensive metamodels of the discipline-specific simulation models. Recent reviews of studies on this subject can be found in the work of Meckesheimer et al.,[1] Simpson et al.,[2] Haftka et al.,[3] and Sobieszczanski-Sobieski and Haftka.[4]

Although metamodels enable faster analyses than the original, complex disciplinary models permit, the metamodel approximation introduces a new element of uncertainty that must be managed. Consequently, there is a need to develop efficient methods to evaluate metamodel fit. The objective in this paper is to investigate inexpensive validation strategies for assessing metamodel fidelity of deterministic computer simulation codes without the use of additional expensive computer simulations. In the next section, the term metamodel is formally defined. In the subsequent sections, we provide a review of metamodel assessment strategies and discuss a leave-*k*-out cross-validation strategy. An experimental study to assess the leave-*k*-out cross-validation strategy based on two sets of response functions is presented in Sec. IV. We conclude with a discussion of the results and suggestions for future work.

Received 16 July 2001; revision received 3 February 2002; accepted for publication 10 May 2002.

*Member, Applied Statistics Group, Mathematics and Computing Technology, Phantom Works, P.O. Box 3707, MS 7L-21; martin.meckesheimer@boeing.com.
†Associate Technical Fellow, Mathematics and Computing Technology Organization.
‡Professor, Management Science and Information Systems.
§Assistant Professor, Departments of Mechanical and Nuclear and Industrial and Manufacturing Engineering, 329 Leonhard Building; tws8@psu.edu. Member AIAA.

II. Approximations of Deterministic Computer Simulation and Analysis Codes

Mathematical and statistical tools have been used for many years to approximate the true input/output relationship of deterministic computer simulation and analysis codes. The basic concept is to construct a simplified model of the disciplinary computer simulation or analysis with a moderate number of computer experiments and then use the approximate relationship to make predictions at additional untried inputs. In the literature, these approximations have been used as surrogates for the original models and are sometimes referred to as metamodels. We emphasize the distinction between surrogate model, which might be a low-fidelity physics code, and a metamodel, which is a mathematical approximation of the disciplinary computer simulation or analysis.

Let $\boldsymbol{X}$ be a matrix of N experiment runs, with each row vector $\boldsymbol{x}_i$, $i = 1, \ldots, N$, specifying a design location based on v input variables. Furthermore, let $\boldsymbol{Y}$ be a matrix of output responses, with each row vector $\boldsymbol{y}_i$, $i = 1, \ldots, N$, containing the performance measures of w output responses. For disciplinary models with several performance measures, a separate metamodel is fit to each response. Then, given a design specified by $\boldsymbol{x}_i$, the mathematical approximation can be written as $\phi(\boldsymbol{x}_i) \cong f(\boldsymbol{x}_i) = y_i$, where $f(\boldsymbol{x}_i)$ represents the ith analysis performed by the original disciplinary model function and $\phi(\boldsymbol{x}_i) = \hat{y}_i$ is the ith response using the metamodel approximation to $f(\boldsymbol{x}_i)$.

We define metamodeling as the process of building a model of a model.[5] This process involves the choice of an experimental design, a metamodel type and its functional form for fitting, and a validation strategy to assess the accuracy of a metamodel. Recent surveys of choices for each of these metamodeling aspects can be found in the work of Simpson et al.[2] and Barton.[6]

The use of metamodels during design overcomes the computational expense and thereby permits designers to identify relevant variables and provides insight into the complex disciplinary model through inexpensive evaluations and interactive design. However, the use of approximations introduces additional concerns and requires particular attention to the following:

1) Inadequate approximations may lead to bad designs or inefficient search for an optimum.

2) There exists uncertainty in the error/inadequacy of the approximation.

Although the use of metamodels offers many benefits, more insight into the behavior of metamodeling strategies is necessary to increase the usability of a metamodel-based design tool. The appropriate selection of experimental runs and the choice of type and form of approximating function to provide adequate model fits remain important research issues.

The use of metamodels as computationally inexpensive surrogates for deterministic computer simulations also requires that the validation strategies be computationally inexpensive as well. In regression modeling, residual analysis provides inexpensive quantitative measures for assessing how well a model fits a set of data by measuring the deviations of the sample points from the fitted curve. However, when metamodels of deterministic computer simulations are based on interpolating functions, there are no deviations between the sample points and the approximating function, and alternative error measures for assessing a metamodel are needed.[2] This is typically achieved by comparing the true response values with the values predicted using the metamodel over an additional set of (expensive) new data. The objective in this paper is to investigate and evaluate computationally inexpensive methods to assess the fidelity of a metamodel of deterministic computer experiments.

III. Metamodel Assessment

We define metamodel assessment as the judgment of the quality or fidelity of a metamodel, where the quality or fidelity must be evaluated quantitatively. The assessment of metamodels may also provide valuable information for metamodel improvement. In the following subsections we discuss the importance of defining measures of performance and methods for collecting data for metamodel assessment.

A. Measures of Metamodel Performance

An approximation is a value that is close to the true value but is not necessarily exact. When approximate models, that is, metamodels, are used as a tool for design, there are three important features that we would like to attain, namely, we would like to 1) build a good approximation, 2) generate measures of performance to assess the goodness of the approximation, and 3) provide an indicator of confidence for the estimated measures of performance.

The first feature allows us to study a complex (or computationally expensive) phenomenon more rapidly and involves the choice of an appropriate experiment design and metamodel type and form for constructing a simplified model of the (unknown) truth. This raises the issue of defining the meaning of good when using metamodels for engineering design.

The second feature provides measures of performance to assess the loss of information in the metamodel that is traded off for the increase in speed of analysis. The goodness of a metamodel may not be dictated by a single performance measure but could depend on several different measures, depending on its intended use.[7] Initially, designers may be looking for an indication of useful domains of the design variables and the identification of key variables. During optimization, designers may be interested in obtaining measures of performance for assessing the impact of design constraints on the optimal objective, refining optimization formulations, and determining globally optimal designs. Finally, designers may desire measures of performance for evaluating tradeoffs in the presence of competing objectives to determine the adequacy of their solutions. Therefore, it is important that these measures of performance be relevant and informative to the user for judging whether the metamodel is acceptable for its intended purpose.

Finally, the third feature provides an indicator of confidence for the measures of performance when they are estimated. In this paper, we focus on methods for generating estimates of performance measures for assessing metamodels inexpensively. We compare the estimated measures of performance to the true measures of performance by generating additional data. In practice, this would generally not be possible because additional disciplinary simulation or analysis runs to obtain true performance measures are expensive. The issue of providing an indicator of confidence for the estimated performance measures is an area for potential research.

B. Metamodel Validation

Validation is necessary whenever a metamodel is meant to answer questions about a disciplinary model. Kleijnen and Sargent[8] define validation as the "... verification that a model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model." Validation relates to both the metamodel and the full disciplinary model and requires knowledge about the problem and the specified accuracy required of the metamodel. In the validation stage, we are interested in knowing how to characterize and compute the fidelity of the approximate model. In this paper, we are concerned with determining whether a metamodel is an adequate representation of the disciplinary simulation model. During validation, we use measures of performance that allow us to determine whether the approximate model is good enough for modeling the complex full disciplinary model. With an accurate assessment of metamodel fidelity, the user can decide whether to make additional model runs to increase metamodel accuracy.

To assess the fidelity of a metamodel, an efficient validation strategy must be identified. A validation strategy specifies how data are generated and used to evaluate and improve an approximate model. Two important criteria that need to be accounted for are 1) the computational expense involved, for instance, making additional runs of the full disciplinary model for validation and 2) the validity of the assessment, that is, how reliable the assessment is and what can be inferred from the measures of performance.

Based on these criteria, validation alternatives may either require additional data or use existing data. When using additional data, the predicted responses using the metamodel are compared to the true responses at untried locations, which are obtained using additional runs of the full disciplinary model. Here, the choice of points at which to run the full disciplinary models for validating the metamodel is critical. Yesilyurt and Patera[9] and Yesilyurt et al.[10] discuss a Bayesian-validated approach for surrogate models that attempts to minimize the use of expensive additional runs and provides a probabilistic quantification of the approximation error.

Although using the full disciplinary model is the most reliable way of validating the metamodel, the computational expense of this approach is high, even when used efficiently, and one would prefer using the costly validation samples to build the metamodel, instead of holding them back for validation purposes. Although the validation samples may be used for updating the metamodel in subsequent analyses,[8,11] the idea of using a subsample of the data for estimating the prediction error suggests other approaches that are applicable in certain situations.[12]

When using existing data, prediction accuracy of the metamodel is determined without requiring additional (presumably computationally expensive) simulations or analyses. In this approach, we take advantage of the data used to fit the metamodel for validation purposes. For example, Laslett[13] proposes sampling strategies using sparse samples to study the properties of approximation methods. In the next section, we describe cross-validation methods that can be used to gain insight into how much a statistic observed from a random sample differs from the population parameter that it estimates.

C. Cross-Validation Methods

Cross validation is a family of methods that can be used to estimate the error of a given metamodel. It can also be used for metamodel selection by choosing the metamodel having the smallest cross-validation error, or selecting the most relevant subset of input variables. In this study, we are interested in estimating a statistical measure of performance, an estimated cross-validation error measure, to provide an assessment of the fidelity of a metamodel.

In the basic cross-validation approach, one starts with a data set, $S\{\boldsymbol{X}, \boldsymbol{Y}\}$, consisting of N input output pairs $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, for $i = 1, \ldots, N$, where $\boldsymbol{y}_i$ is the model (output) response at the design (input) sample point $\boldsymbol{x}_i$ and N is the total number of disciplinary model samples. In the first step, the data set is randomized and split into two parts. The first part, for example, $S^1\{\boldsymbol{X}^1, \boldsymbol{Y}^1\}$ of size n^1, is used for fitting the metamodel, whereas the second part, $S^2\{\boldsymbol{X}^2, \boldsymbol{Y}^2\}$ of size n^2, is used for computing the estimated cross-validation error measure. This is the difference between the metamodel predictions $\hat{\boldsymbol{y}}^2$ and the actual

values $\boldsymbol{y}^2$ at the omitted design points $\boldsymbol{x}^2$. The cross-validation step consists of switching the data sets for metamodel fit and prediction to obtain an additional estimation of how well the metamodel predicts a new set of data. However, the data need not necessarily be split into two parts, and other schemes for randomizing and partitioning a data set may be used as discussed by Laslett.[13]

In p-fold cross validation, the initial data set is split into p mutually exclusive and exhaustive subsets, that is, $S\{\boldsymbol{X}, \boldsymbol{Y}\} = S^1\{\boldsymbol{X}^1, \boldsymbol{Y}^1\}$, $S^2\{\boldsymbol{X}^2, \boldsymbol{Y}^2\}, \ldots, S^p\{\boldsymbol{X}^p, \boldsymbol{Y}^p\}$. Then, the metamodel is fit p times, each time leaving out one of the subsets from training and using the omitted subset to compute the cross-validation error measure.

A variation of p-fold cross validation is the leave-k-out approach, in which all possible

$$\binom{N}{k}$$

subsets of size k are left out, and the metamodel is fit to each remaining set. Each time, the cross-validation error measure is computed at the omitted points. This approach is a computationally more expensive version of p-fold cross validation; however, Mitchell and Morris[14] describe how the cross-validation error measure may be computed inexpensively for the special case when $k = 1$, which is called leave-one-out cross validation. For linear regression metamodels, the cross validation with $k = 1$ is known as the prediction error sum of squares, which is obtained inexpensively from a single least-squares fit to all N sample points using the hat matrix. (For a detailed discussion, see Myers and Montgomery.[15])

IV. Experimental Study to Assess a Leave-k-Out Cross-Validation Strategy

Based on the discussion on cross-validation methods in the preceding section, a leave-k-out cross-validation strategy is proposed to overcome the computational expense for metamodel assessment. For the purpose of estimating the metamodel prediction error, the leave-k-out cross-validation strategy provides k $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ input output data pairs that serve as validation samples for each metamodel, fit to N–k input output data pairs.

The experimental study for assessing the leave-k-out cross-validation strategy consists of two stages. During the first stage of the experimental study, the leave-k-out cross-validation root mean squared error is computed for different values of k to provide an inexpensive cross-validation estimate of the metamodel prediction error. During the second stage of the experimental study, the discrepancy between the cross-validation root mean squared error estimate from the first stage and the true error is assessed. The experimental study for assessing the leave-k-out cross-validation strategy is illustrated in Fig. 1, and the two stages of the experimental study are explained as follows.

The first stage involves an estimate of the cross-validation error measure, for instance, the cross-validation rms error ($\mathrm{RMSE}_{\mathrm{CV}}$). Consider a number of expensive disciplinary model runs that generate a set of data S of size N. Randomly draw samples of size N–k (without replacement) from S and fit approximation models using only those samples. Then compute the estimated cross-validation error measure at the k omitted sample points. This is equivalent to the leave-k-out cross-validation strategy, except that only N, rather than

$$\binom{N}{k}$$

samples are drawn.

The second stage involves assessing the under- or overestimation of the cross-validation error measure. The under- or overestimation of the cross-validation error measure with respect to the true error measure can be assessed using the information obtained from the first stage of the experimental study. During the first stage of the experimental study, each of the N metamodel fits provides k cross-validation estimates of the prediction error. These data can be used to compute the deviations of the leave-k-out cross-validation error estimates from the true error. In this experimental study, the true rms error is approximated using a validation data set, consisting of 1000 additional (presumably expensive) disciplinary model runs.

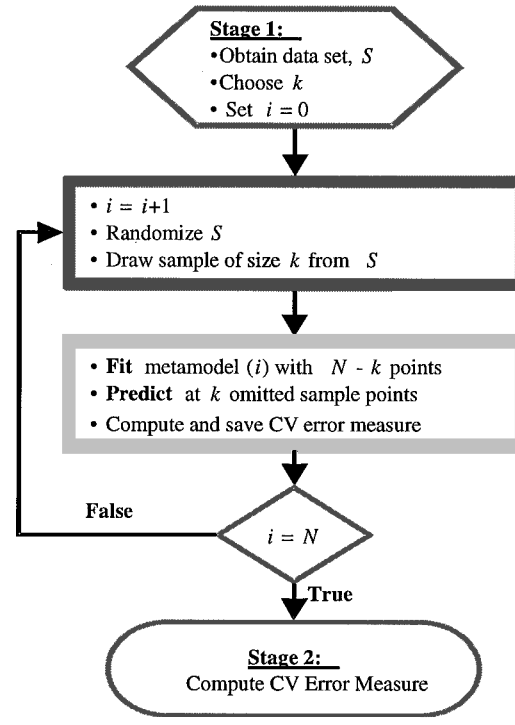

Fig. 1 Experimental study to assess the leave-k-out cross-validation strategy.

The difference between the true output responses obtained from the disciplinary model and the predicted output responses obtained with the metamodel approximates the true metamodel prediction error [true rms error ($\mathrm{TRUE}_{\mathrm{RMSE}}$)]. This expensive validation is required only for this experiment to assess how well the leave-k-out strategy performs; it would not be computed in a practical application.

During the experimental study, the number of data pairs omitted during cross validation k is varied during the first stage of the experimental study to look for an ideal subsample size for estimating the metamodel prediction error. The leave-k-out cross-validation strategy for metamodel assessment is applied to two test problems. Kriging, low-order polynomial, and radial basis function metamodels are fit using different methods for selecting design points.

The objective in the experimental study is not to optimize a response, but to test the leave-k-out cross-validation strategy for practicality and numerical efficiency and evaluate its effectiveness in terms of accuracy and precision for estimating an error measure. The potential factors affecting the results are the type of response function, the type of metamodel, the experimental design strategy, that is, the size and type of the fitting design, and the number of points omitted in the leave-k-out cross-validation strategy. The following subsections describe the response functions, approximation model types, and fitting designs used to evaluate the leave-k-out cross-validation strategy for metamodel assessment.

A. Simulation Response Functions

Two test problems are employed in this study, namely, the flight manual simulation from The Boeing Company and Problem 100 from Hock–Schittkowski.[16] The flight manual (FM) is a deterministic simulation that models the characteristics of aircraft performance under different operating and environmental conditions. The Boeing Company developed the software to improve airline operations by allowing users to increase revenue payload, improve operational economies, and extend engine life under simulated airport conditions. For this study, we examine seven input variables (fm_i, $i = 1, \ldots, 7$) that characterize the flight conditions and three aircraft performance measures: 1) M_1, climb limit; 2) M_2, field length limit; and 3) M_3, obstacle clearance limit.

Because of the proprietary nature of code, further details about the problem variables have been omitted, and the true responses have been scaled. The functional form of the responses is not known explicitly because the simulation is a black box; therefore, selecting

a good metamodeling strategy a priori, based on knowledge of the responses, is not possible.

The Hock–Schittkowski Problem 100 is a test problem involving seven variables, one objective, and four constraints.[16] For our analysis, however, we consider only the objective function and one of the constraints, z and c_1, respectively:

$$z = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 \quad (1)$$

$$c_1{:}127 - 2x_1^2 - 3x_2^4 - x_3 - 4x_4^2 - 5x_5 \geq 0 \quad (2)$$

B. Approximation Models

We consider three types of metamodels in our experiment: kriging metamodels, radial basis functions, and low-order polynomials. Kriging metamodels are a class of approximation techniques that show good promise for building accurate global approximations of a design space.[17–19] A kriging metamodel is a combination of a polynomial model plus departures of the form

$$f(\boldsymbol{x}) \cong \phi(\boldsymbol{x}) = g(\boldsymbol{x}) + Z(\boldsymbol{x}) \quad (3)$$

where $f(\boldsymbol{x})$ is the unknown function of interest, $g(\boldsymbol{x})$ is any regression model of $\boldsymbol{x}$, and $Z(\boldsymbol{x})$ is the realization of a normally distributed Gaussian random process with mean zero, variance σ^2, and nonzero covariance. In our experiment, $g(\boldsymbol{x})$ is taken as a constant, and the correlation parameters of the original kriging metamodel are reused to avoid the optimization process when applying the leave-k-out cross-validation strategy.

Radial basis function metamodels are constructed using a linear combination of a radially symmetric function based on Euclidean distance of the form[20]

$$f(\boldsymbol{x}) \cong \phi(\boldsymbol{x}) = \beta_0 + \sum_i^N \beta_i \|\boldsymbol{x} - \boldsymbol{x}_i\| \quad (4)$$

where $\|\|$ represents the Euclidean norm and the sum is over an observed set of system responses, $\{[\boldsymbol{x}_i, f(\boldsymbol{x}_i)]\}, i = 1, \ldots, N$. Given the set of system responses and design points at which the responses are sampled, the coefficients β_i can be obtained by solving the resulting linear system. Radial basis function approximations interpolate observed performance measure values and have produced good fits to arbitrary contours of both deterministic and stochastic response functions.[21]

Finally, metamodels based on low-order polynomials discussed by de Boor and Ron[22] are used. Note that the low-order polynomials used in our experiments are different from the polynomial regression metamodels used in response surface methodology.[15] The low-order polynomials used here interpolate responses in keeping with the philosophy of approximating deterministic responses as discussed by Sacks et al.[23] They are a class of interpolators based on polynomial basis functions determined by the locations of the independent variable values. Basis functions are successively added to the multidimensional polynomial model until all of the data points are interpolated. The method favors low-degree terms over higher-degree terms, that is, the method chooses a polynomial interpolant of minimal degree in the sense described by de Boor and Ron[24] and, hence, are called least interpolants. Nevertheless, the method can choose higher-degree terms depending on the distribution of independent variable values and the number of levels. Thus, the fit is not limited to linear or quadratic polynomials, and a number of good properties of low-order polynomials among polynomial interpolants are given in Refs. 22 and 24. The computational method for fitting a low-order polynomial is described in Ref. 25, and the code is available online at URL: http://netlib.bell-labs.com/netlib/a/mvp.tgz.

C. Fitting Designs

The experimental designs that we use to evaluate the leave-k-out cross-validation strategy include central composite designs, orthogonal arrays, Latin hypercubes, Hammersley sampling sequences, and uniform designs. A central composite design is a two level ($2^{(v-r)}$ or 2^v) factorial design, augmented by n_0 center points and two star points positioned at $\pm\alpha$ for each design variable v. A face-centered design locates the star points on the centers of the faces of the central composite design cube. This design consists of $2^{(v-r)} + 2v + n_0$ total design points to estimate $2v + v(v-1)/2 + 1$ model coefficients; however, a central composite design need not be restricted to fitting quadratic polynomials. More information about factorial and central composite designs can be found in the work of Myers and Montgomery.[15]

An orthogonal array of strength $t \leq v$ is essentially a matrix A in which in each $N \times t$ submatrices of A, all q^t possible distinct rows occur the same number of times,[26] where q is the number of levels of a design parameter and N is the number of rows, that is, design runs. The strength t of an orthogonal array means that a projection into any t-dimensional subspace is a grid. In a sense, the strength of the orthogonal array indicates to what degree the design models the interactions between factors. For instance, if we could neglect v–t variables, the effects of t variables and all (up to t-way) of their interactions could be estimated. Orthogonal arrays of strength two or higher require at least q^2 runs and can be generated in different ways; however, they do not exist for all combinations of N and q and become expensive for q-level experiments when $q > 2$ and all interactions need to be estimated.

Latin hypercube sampling designs are designed for uniformity along a single dimension where subsequent columns are randomly paired for placement on a v-dimensional cube.[27] These designs are useful when relatively few runs are needed based on a sample of design points located randomly over the design factor space.[18] An orthogonal array based Latin hypercube design is obtained from an orthogonal array by replacing, in a specific way, the entries of each column of the orthogonal array with permutations of $1, \ldots, N$, where N is the number of rows.[28]

Hammersley sampling sequences result in a low-discrepancy experimental design for placing N points in a v-dimensional hypercube.[29] Discrepancy is a measure of uniformity that minimizes the difference between the percentage of points falling in a particular region on a unit cube and the percentage of volume occupied by this region. Hammersley sampling sequence design points provide better uniformity properties over the v-dimensional hyperspace than Latin hypercubes.

Uniform designs are another type of optimal designs that minimize discrepancy and were originally developed with the objective of generating uniformly distributed points in the design space.[30] Uniform designs are based on the number theoretic method that was originally applied in the field of numerical integration.[31] Several uniform designs are available online at URL: http://www.math.hkbu.edu.hk/UniformDesign, and a recent comparison of uniform designs to other space filling experimental designs can be found in Ref. 32.

For evaluating the assessment strategies, we generate designs of different sizes for each design type. The minimum number of computer simulation model runs is determined by the minimum number of runs required to estimate the parameters of a quadratic polynomial regression model in v variables, that is, $2v + v(v-1)/2 + 1$. Both the FM and the Hock–Schittkowski 100 test problems involve seven design variables; therefore, the minimum number of runs is 36, but only Latin hypercube, Hammersley sampling sequences, and uniform designs can be generated for 36 runs. A face-centered central composite experiment design requires 79 (i.e., $2^{v-1} + 2v + 1$) runs, and orthogonal array designs of strength 2 require at least 49 runs for 7 variables. Therefore, we generate orthogonal array designs of strength two consisting of 49, 50, and 81 runs using the code available online at URL: http://lib.stat.cmu.edu/designs/owen.html. In addition, we generate Latin hypercube designs of 49, 50, and 81 runs using our own code and include permuted versions of some orthogonal array and Latin hypercube designs using an exchange method. Finally, we generate Hammersley sampling sequences and uniform designs of 36, 49, and 81 runs. Table 1 lists the resulting 25 experimental designs used in this study.

D. Leave-k-Out Cross-Validation Strategy Applied to the Simulation Response Functions

The $\mathrm{RMSE_{CV}}$ is used as the measure of performance for the experiments. The number of omitted points k is varied from 1 to 10. In

Table 1 Fitting design sizes for leave-*k*-out cross validation

Design	Sizes, *N*	Description
FCD	79	Face-centered central composite
LHS	49, 50, 81	Latin hypercube
LHS_EX	49, 50, 81	Latin hypercube with exchange
OA	49, 50, 81	Orthogonal array
OA_EX	49, 50, 81	Orthogonal array with exchange
OALHS	49, 50, 81	OA-based Latin hypercube
OA_EXLEX	49, 50, 81	OA-based Latin hypercube with exchange
HSS	36, 49, 81	Hammersley sampling sequence design
UNI	36, 49, 81	Uniform design

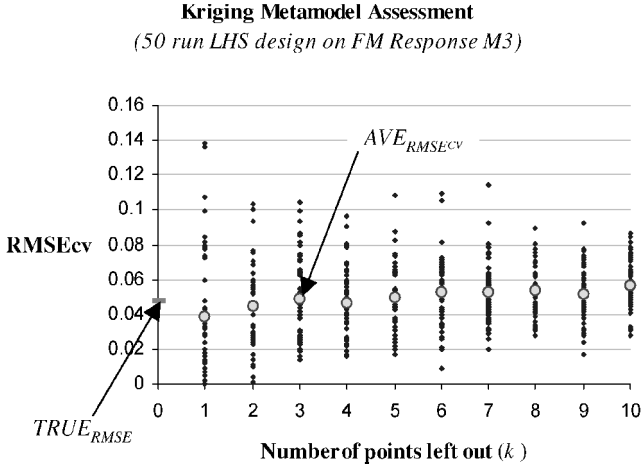

Fig. 2 Example of the leave-*k*-out cross-validation strategy.

addition, $k = 0.1N$ and $\sqrt{N}$ are included in the experiment, which is motivated by a discussion on jackknife estimates of error by Efron and Tibshirani[33] (p. 149) and provides a rule of thumb for choosing k based on the fitting design size. The leave-k-out cross-validation strategy for metamodel assessment is then applied as follows.

1. Stage 1

1) Select a design of experiment of N runs (from Table 1) and evaluate the expensive disciplinary model to obtain an input–output data set S.

2) Specify the number of points k to be left out for cross-validation, that is, 1–10, $0.1N$, and $\sqrt{N}$.

3) Perform the following for $i = 1$ to N:

a) Fit kriging, radial basis function, and low-order polynomial metamodels with N–k data points. When $k = 1$, the ith point is omitted.

b) Obtain kriging, radial basis function, and low-order metamodel predictions ($\hat{y}_i$) at the k omitted data points.

c) For each metamodel prediction, compute $\mathrm{RMSE}_{\mathrm{CV}}$:

$$\mathrm{RMSE}_{\mathrm{CV}}(i) = \sqrt{\frac{1}{k}\sum_{j=1}^{k}(\hat{y}_j - y_j)^2} \tag{5}$$

d) Save the estimated cross-validation root mean squared errors.

Sample results from stage 1 are shown in Fig. 2 for a kriging metamodel fit to the FM simulation response M_3 with a 50-run Latin hypercube design. The average $\mathrm{RMSE}_{\mathrm{CV}}$ ($\mathrm{AVE}_{\mathrm{RMSE_{CV}}}$) is compared with the $\mathrm{TRUE}_{\mathrm{RMSE}}$, which is estimated from 1000 additional runs that generated randomly with Latin hypercube sampling over the fitted region. Note that the $\mathrm{TRUE}_{\mathrm{RMSE}}$ is shown at zero on the horizontal axis.

2. Stage 2

Assess the under- or overestimation of $\mathrm{RMSE}_{\mathrm{CV}}$ with respect to the $\mathrm{TRUE}_{\mathrm{RMSE}}$ as follows:

1) Compute the $\mathrm{AVE}_{\mathrm{RMSE_{CV}}}$: measures:

$$\mathrm{AVE}_{\mathrm{RMSE_{CV}}} = \frac{\sum_{i=1}^{N}\mathrm{RMSE}_{\mathrm{CV}}(i)}{N} \tag{6}$$

2) Compute the $\mathrm{AVE}_{\mathrm{RMSE_{CV}}}$ relative to the estimated $\mathrm{TRUE}_{\mathrm{RMSE}}$ ($\mathrm{REL}_{\mathrm{RMSE_{CV}}}$):

$$\mathrm{REL}_{\mathrm{RMSE_{CV}}} = \frac{\mathrm{AVE}_{\mathrm{RMSE_{CV}}} - \mathrm{TRUE}_{\mathrm{RMSE}}}{\mathrm{TRUE}_{\mathrm{RMSE}}} \tag{7}$$

This procedure was repeated for all 25 experimental design types and sizes listed in Table 1. The next section summarizes the results from this experiment; the complete set of results can be found in Ref. 34.

E. Experimental Results

Figures 3a–3f summarize the results for the FM and the Hock-Schittkowski 100 (HS100) response functions with the leave-k-out cross-validation strategy, using kriging (KRG), radial basis functions (RBF), and low-order polynomial (LOP) metamodels. In the box plots shown in Figs. 3a–3f, each point represents the $\mathrm{REL}_{\mathrm{RMSE_{CV}}}$, defined in Eq. (7), for a metamodel fitted with a particular design type. The $\mathrm{REL}_{\mathrm{RMSE_{CV}}}$ (on the vertical axis) is plotted vs the number of omitted points k during the leave-k-out cross-validation strategy. The dotted lines in the Figs. 3a–3f delimit the region in which the $\mathrm{REL}_{\mathrm{RMSE_{CV}}}$ is within $\pm 25\%$. The horizontal bar in each box plot is the median $\mathrm{REL}_{\mathrm{RMSE_{CV}}}$, and the vertical bars extending from each box indicate the spread of the $\mathrm{REL}_{\mathrm{RMSE_{CV}}}$ for a particular number of omitted points k across all fitting designs for all responses of a test problem. Asterisks indicate values that fall outside this spread. The $\mathrm{REL}_{\mathrm{RMSE_{CV}}}$ indicates how close the $\mathrm{AVE}_{\mathrm{RMSE_{CV}}}$ value is to the $\mathrm{TRUE}_{\mathrm{RMSE}}$; therefore, when the $\mathrm{REL}_{\mathrm{RMSE_{CV}}}$ is zero, the $\mathrm{AVE}_{\mathrm{RMSE_{CV}}}$ value is the same as the $\mathrm{TRUE}_{\mathrm{RMSE}}$ estimate. Furthermore, when the horizontal bar, that is, the median, of a particular box plot is below 0, this means that more than half of the $\mathrm{AVE}_{\mathrm{RMSE_{CV}}}$ values underestimate the $\mathrm{TRUE}_{\mathrm{RMSE}}$. Similarly, when the median of a particular box plot is above 0, this indicates that more than half of the $\mathrm{AVE}_{\mathrm{RMSE_{CV}}}$ values overestimate the $\mathrm{TRUE}_{\mathrm{RMSE}}$.

For Figs. 3a–3c, each box plot represents 75 experiment runs: 25 different fitting designs for each of the three FM response functions. Figure 3a shows the resulting KRG metamodels for the fitting design sizes and types. We observe that the median $\mathrm{REL}_{\mathrm{RMSE_{CV}}}$ approaches zero as the number of omitted points during the leave-k-out cross-validation strategy increases, and choosing $k = \sqrt{N}$ provides a good error estimate when using KRG metamodels. The simple $0.1N$ rule also performs well for KRG metamodels, but the quality of the fit deteriorates rapidly as the number of points is increased beyond one for LOP and RBF metamodels as shown in Figs. 3b and 3c, respectively. For LOP and RBF metamodels, leaving one point out produced more accurate estimates than leaving $\sqrt{N}$ points out: When omitting one point, the $\mathrm{REL}_{\mathrm{RMSE_{CV}}}$ is close to zero for most fitting designs, which is contrary to the results for the KRG metamodels.

For Figs. 3d–3f, each box plot represents 50 experiment runs, 25 different fitting designs for each of HS100 responses, z and c_1. Figure 3d shows the results for the leave-k-out cross-validation assessment of the KRG metamodels based on the fitting design sizes and types. Here, the $\mathrm{AVE}_{\mathrm{RMSE_{CV}}}$ values obtained with the leave-k-out cross-validation strategy overestimates the $\mathrm{TRUE}_{\mathrm{RMSE}}$ for more than half of the fitting designs when more than four points are omitted.

Figures 3e and 3f show the results for the leave-k-out cross-validation strategy assessment of LOP and RBF metamodels for the two HS100 response functions, respectively. The observations are similar to those in Figs. 3b and 3c: Except for $k = 1$, the $\mathrm{AVE}_{\mathrm{RMSE_{CV}}}$ values overestimate the $\mathrm{TRUE}_{\mathrm{RMSE}}$ for the majority of fitting designs used in our experiment.

For the KRG metamodel, it is not possible to identify one good number of sample points to be omitted to obtain an $\mathrm{AVE}_{\mathrm{RMSE_{CV}}}$ estimate from Eq. (6). Instead, there are several values for which k yields an $\mathrm{AVE}_{\mathrm{RMSE_{CV}}}$ that is close to the $\mathrm{TRUE}_{\mathrm{RMSE}}$. When using the leave-$k$-out cross-validation strategy with LOP and RBF metamodels, $k = 1$ generally produces good $\mathrm{RMSE}_{\mathrm{CV}}$ estimates. When

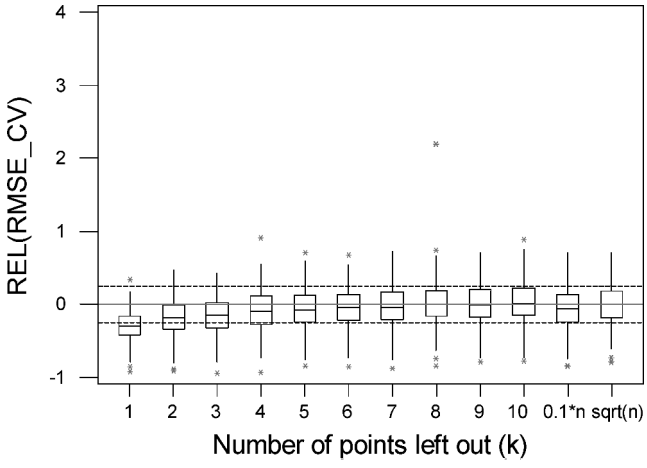

Fig. 3a Assessment evaluation for KRG-FM.

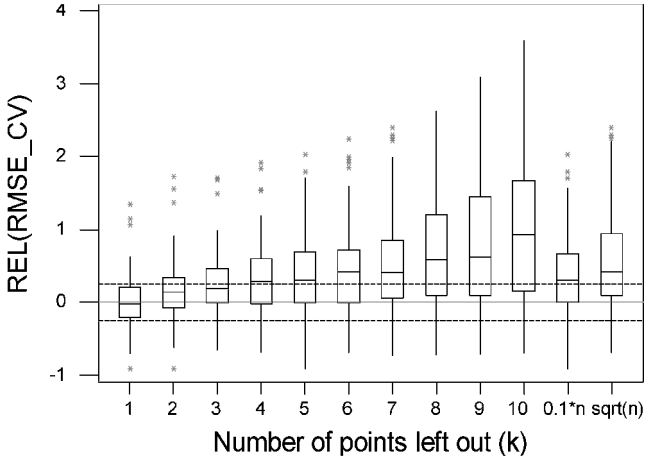

Fig. 3b Assessment evaluation for LOP-FM.

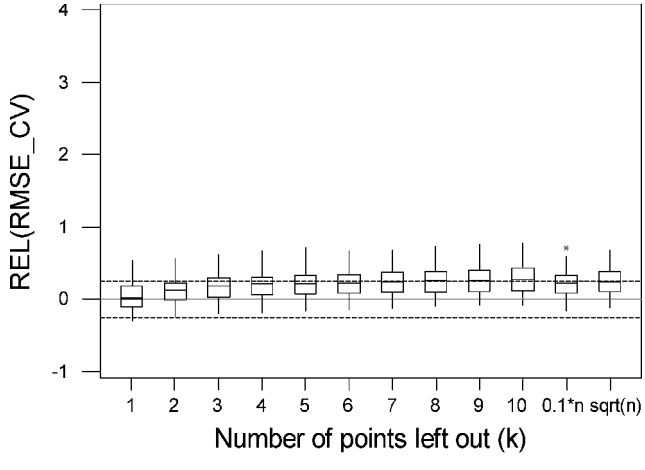

Fig. 3c Assessment evaluation for RBF-FM.

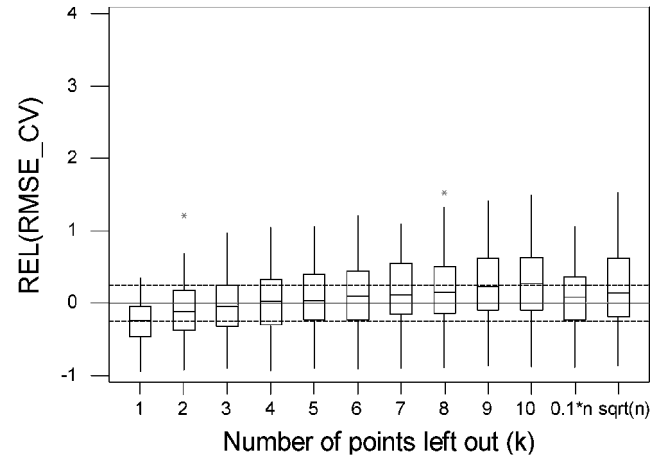

Fig. 3d Assessment evaluation for KRG-HS100.

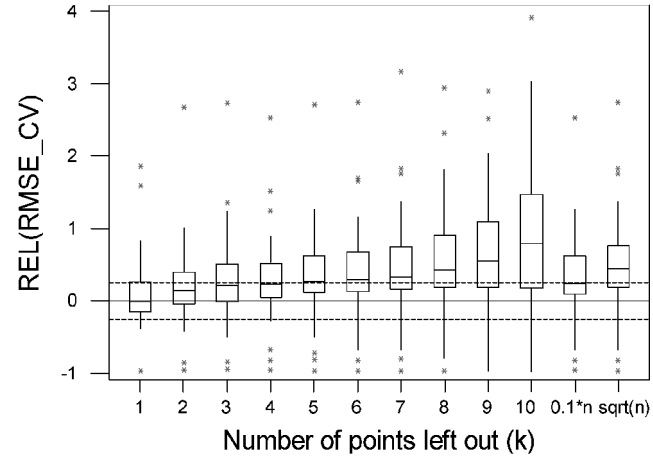

Fig. 3e Assessment evaluation for LOP-HS100.

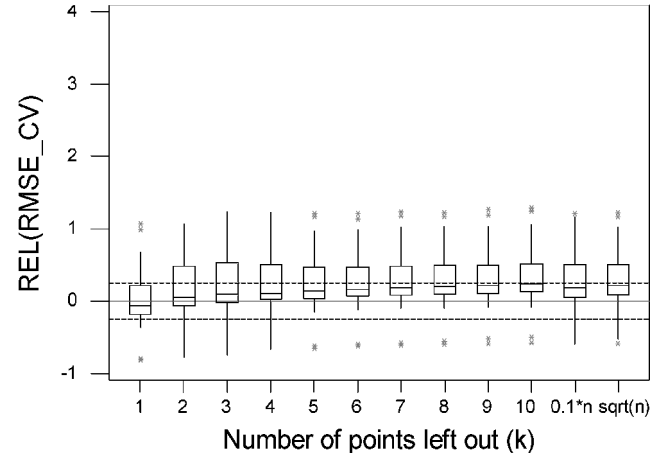

Fig. 3f Assessment evaluation for RBF-HS100.

more than one sample point is left out in the leave-k-out cross-validation strategy, the $\text{TRUE}_{\text{RMSE}}$ for LOP and RBF metamodels is overestimated for most fitting designs. In addition, the spread observed in $\text{REL}_{\text{RMSE}_{\text{CV}}}$, for KRG and RBF metamodels, is typically ±25%. (This is indicated by the dotted lines in Figs. 3a–3f.) The $\text{REL}_{\text{RMSE}_{\text{CV}}}$ is larger for LOP metamodels. Furthermore, in all cases (Figs. 3a–3f) the spread in the $\text{REL}_{\text{RMSE}_{\text{CV}}}$ generally increases as more points are left out. This may be explained by that as the number of points left out for fitting increases, a metamodel fit with a small-sized design, for example, 36 runs, is more likely to be inaccurate than the same metamodel fit with a large design, for example, 81 runs. Note that in all cases, when the number of points omitted depends on the design size (that is, $k = 0.1N$ and $\sqrt{N}$), the $\text{REL}_{\text{RMSE}_{\text{CV}}}$ does not uniformly produce the smallest spread.

Finally, the spread across fitting designs is larger when estimating RMSE_{CV} values for LOP than for KRG and RBF metamodels. The set of basis functions in a LOP model is determined by the location of the points in the experimental design used to fit the model, and leaving k points out of an experimental design can drastically change the set of basis functions. This is in contrast to KRG or RBF metamodels, where leaving k points out nominally only affects the metamodels near the omitted point(s). If the change does not allow

for basis functions important to the fit, then cross-validation errors will be large; otherwise, the cross-validation errors will be relatively small. The large experimental designs cover the design (input) space very well, and, thus, it is unlikely that omission of k points will change the basis enough to remove basis functions important to the fit. Consequently, we expect to see more variability in $\mathrm{REL}_{\mathrm{RMSE_{CV}}}$ for LOP metamodels for small experimental designs, which is, in fact, what is observed.

V. Conclusions

In terms of practicality, the leave-k-out cross-validation strategy provides a reasonable indicator of metamodel fidelity without the use of additional computationally expensive analyses. Furthermore, we may not only benefit from inexpensive metamodel assessment, but also be able to select the best metamodel for further analysis, given different types of metamodels. Because the metamodels are fit N times during each cross-validation cycle, the numerical efficiency of the leave-k-out cross-validation strategy depends on the type of metamodel. For instance, for LOP metamodels the model coefficients are computed relatively easily, whereas in a KRG model, the correlation parameters are obtained through an optimization process, resulting in a more time-consuming validation process. In this study, the correlation parameters of the original KRG metamodel are reused to avoid the optimization process when applying the leave-k-out cross-validation strategy. In general, provided that additional disciplinary model runs are computationally much more expensive than the metamodel fitting process, the validation approach is still justified.

In terms of accuracy and precision of estimating an error measure, our results show that as more points are omitted during the metamodel fit, the $\mathrm{AVE}_{\mathrm{RMSE_{CV}}}$ estimate increases when using LOP and RBF metamodels. In our experimental study, the leave-one-out ($k = 1$) cross-validation strategy is effective for RBF and LOP metamodels, but not for KRG metamodels. Although our results include only three types of metamodels, the type of the metamodel affects the cross-validation error estimates. Past experience with comparative studies of metamodels[7] show that KRG, as well as RBF metamodels, perform better on highly nonlinear responses than polynomial models and may, therefore, also result in better error estimates. In addition, the type and size of the fitting design as well as the form of the simulation response also seem to have an effect on the cross-validation error estimate.

Based on the observations from the experimental study conducted to assess the leave-k-out cross-validation strategy, a value of $k = 1$ is recommended for providing a prediction error estimate for RBF and LOP metamodels but not for KRG metamodels. Several values of k provided a KRG metamodel prediction error estimate that was within $\pm 25\%$ of the true prediction error. Choosing k as a function of the fitting design size (that is, $k = 0.1\,N$ or $k = \sqrt{N}$) is recommended for estimating the prediction error for KRG metamodels. The choice of k based on the size of the fitting design provides a rule of thumb that may be applied (with caution) until further experiments support this observation.

Future work entails computing nonparametric confidence intervals about the cross-validation error estimates to provide a probabilistic assessment of this error measure and using this information to improve metamodel validity. In addition, experiments on different test problems and larger data sets varying the dimensionality of the problem will provide more insight into the practicality and efficiency of the leave-k-out cross-validation strategy.

Acknowledgments

This work is supported by the National Science Foundation (NSF) under NSF Grant DMI-9700040 and NSF Grant DMI-0084918. Both the Mathematics and Computing Technology Organization of The Boeing Company and the Intelligent Design and Diagnostics Research Laboratory at the Pennsylvania State University supplied the computing resources for this work. The authors thank Evin J. Cramer and Paul D. Frank for their assistance with the test problems and the anonymous reviewers for their suggestions to improve the quality of the paper.

References

[1]Meckesheimer, M., Barton, R. R., Simpson, T. W., Limayem, F., and Yannou, B., "Metamodeling of Combined Discrete/Continuous Responses," *AIAA Journal,* Vol. 39, No. 10, 2001, pp. 1955–1959.

[2]Simpson, T. W., Peplinski, J., 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.

[3]Haftka, R., Scott, E. P., and Cruz, J. R., "Optimization and Experiments: A Survey," *Applied Mechanics Review,* Vol. 51, No. 7, 1998, pp. 435–448.

[4]Sobieszczanski-Sobieski, J., and Haftka, R. T., "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments," *Structural Optimization,* Vol. 14, No. 1, 1997, pp. 1–23.

[5]Kleijnen, J. P. C., "A Comment on Blanning's Metamodel for Sensitivity Analysis: The Regression Metamodel in Simulation," *Interfaces,* Vol. 5, No. 1, 1975, pp. 21–23.

[6]Barton, R. R., "Metamodeling: A State of the Art Review," *Proceedings of the 1994 Winter Simulation Conference*, edited by J. D. Tew, S. Manivannan, D. A. Sadowski, A. F. Seila, IEEE Publications, Piscataway, NJ, 1994, pp. 237–244.

[7]Jin, R., Chen, W., and Simpson, T. W., "Comparative Studies of Metamodeling Techniques Under Multiple Modeling Criteria," AIAA Paper 2000-4801, Sept. 2000.

[8]Kleijnen, J. P. C., and Sargent, R. G., "A Methodology for Fitting and Validating Metamodels in Simulation," *European Journal of Operations Research*, Vol. 120, No. 1, 2000, pp. 14–29.

[9]Yesilyurt, S., and Patera, A. T., "Surrogates for Numerical Simulations; Optimization of Eddy-Promoter Heat Exchangers," *Computer Methods in Applied Mechanics and Engineering*, Vol. 121, No. 1–4, 1995, pp. 231–257.

[10]Yesilyurt, S., Paraschivoiu, M., Otto, J., and Patera, A. T., "Computer-Simulation Response Surfaces: A Bayesian-Validated Approach," *11th International Conference on Computational Methods in Water Resources (CMWR96)* edited by A. A. Aldama, Computational Mechanics Publishing, Boston, Vol. 1, 1996, pp. 13–22.

[11]Osio, I. G., and Amon, C. H., "An Engineering Design Methodology with Multistage Bayesian Surrogates and Optimal Sampling," *Research in Engineering Design,* Vol. 8, No. 4, 1996, pp. 189–206.

[12]Eubank, R. L., *Spline Smoothing and Nonparametric Regression*, Marcel Dekker, New York, 1988.

[13]Laslett, G. M., "Kriging and Splines: An Empirical Comparison of their Predictive Performance in Some Applications," *Journal of the American Statistical Association*, Vol. 89, No. 426, 1994, pp. 391–400.

[14]Mitchell, T. J., and Morris, M. D., "Bayesian Design and Analysis of Computer Experiments: Two Examples," *Statistica Sinica*, Vol. 2, 1992, pp. 359–379.

[15]Myers, R. H., and Montgomery, D. C., *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, Wiley, New York, 1995.

[16]Hock, W., and Schittowski, K., *Test Examples for Nonlinear Programming Codes*, Springer-Verlag, Berlin, 1981.

[17]Booker, A. J., "Design and Analysis of Computer Experiments," *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Vol. 1, AIAA, Reston, VA, 1998, pp. 118–128.

[18]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. 409–435.

[19]Simpson, T. W., Mauery, T. M., Korte, J. J., and Mistree, F., "Kriging Metamodels for Global Approximation in Simulation-Based Multidisciplinary Design Optimization," *AIAA Journal*, Vol. 39, No. 12, 2001, pp. 2233–2241.

[20]Hardy, R. L., "Multiquadratic Equations of Topography and Other Irregular Surfaces," *Journal of Geophysical Research,* Vol. 76, 1971, pp. 1905–1915.

[21]Powell, M. J. D., "Radial Basis Functions for Multivariable Interpolation: A Review," *Algorithms for Approximation*, edited by J. C. Mason and M. G. Cox, Oxford Univ. Press, London, 1987, pp. 105–210.

[22]de Boor, C., and Ron, A., "On Multivariate Polynomial Interpolation," *Constructive Approximation*, Vol. 6, No. 3, 1990, pp. 287–302.

[23]Sacks, J., Schiller, S. B., and Welch, W. J., "Designs for Computer Experiments," *Technometrics,* Vol. 31, No. 1, 1989, pp. 41–47.

[24]de Boor, C., and Ron, A., "The Least Solution for the Polynomial Interpolation Problem," *Mathematische Zeitschrift*, Vol. 210, No. 3, 1992, pp. 347–378.

[25]de Boor, C., and Ron, A., "Computational Aspects of Polynomial Interpolation in Several Variables," *Mathematics of Computation*, Vol. 58, No. 198, 1992, pp. 705–727.

[26]Owen, A. B., "Orthogonal Arrays for Computer Experiments, Integration and Visualization," *Statistica Sinica,* Vol. 2, 1992, pp. 439–452.

[27]McKay, M. D., Beckman, R. J., and Conover, W. J., "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis

of Output from a Computer Code," *Technometrics,* Vol. 21, No. 2, 1979, pp. 239–245.

[28]Tang, B., "Orthogonal Array-Based Latin Hypercubes," *Journal of the American Statistical Association*, Vol. 88, No. 424, 1993, pp. 1392–1397.

[29]Kalagnanam, J. R., and Diwekar, U. M., "An Efficient Sampling Technique for Off-Line Quality Control," *Technometrics,* Vol. 39, No. 3, 1997, pp. 308–319.

[30]Fang, K.-T., Lin, D. K. J., Winker, P., and Zhang, Y., "Uniform Design: Theory and Application," *Technometrics,* Vol. 42, No. 2, 2000, pp. 237–248.

[31]Fang, K.-T., and Wang, Y., *Number-Theoretic Methods in Statistics*, Chapman and Hall, New York, 1994.

[32]Simpson, T. W., Lin, D. K. J., and Chen, W., "Sampling Strategies for Computer Experiments: Design and Analysis," *International Journal of Reliability and Applications*, Vol. 2, No. 3, 2001, pp. 209–240.

[33]Efron, B., and Tibshirani, R., *An Introduction to the Bootstrap*, Chapman and Hall, New York, 1993, p. 149.

[34]Meckesheimer, M., "A Framework for Metamodel-Based Design: Subsystem Metamodel Assessment and Implementation Issues," Ph.D. Dissertation, Dept. of Industrial and Manufacturing Engineering, Pennsylvania State Univ., University Park, PA, Dec. 2001; also URL: http://etda.libraries.psu.edu/ [cited 20 Aug. 2001].

A. Messac
Associate Editor