ABSTRACT

Engineering design and optimization efforts using computational systems rapidly become resource intensive. The goal of the surrogate-based approach is to perform a complete optimization with limited resources. In this paper we present a Bayesian–validated approach that informs the designer as to how well the surrogate performs; in particular, our surrogate framework provides precise (albeit probabilistic) bounds on the errors incurred in the surrogate–for–simulation substitution. The theory and algorithms of our computer–simulation surrogate framework are first described. The utility of the framework is then demonstrated through two illustrative examples: maximization of the flowrate of fully developed flow in trapezoidal ducts; and design of an axisymmetric body that achieves a target Stokes drag.

INTRODUCTION

Computational design efforts typically consist of a sequence of optimization problems characterized by different combinations of design parameters and objective functions. Each design problem is solved by means of a nonlinear optimization algorithm. Numerical simulations can be incorporated into the optimization algorithm either by direct–insertion or through a surrogate–based approach. In direct–insertion, the large-scale simulation is inserted into the optimization algorithm as a function call. In the surrogate–based approach, a surrogate (a simple model such as a response surface) for the simulation is first constructed, and the surrogate is then incorporated into the optimization algorithm in place of the full simulation.

Many optimization problems based upon direct–insertion are hindered, first, by the cost of each inquiry to the simulation code, and second, by the algorithmic difficulties of merging the large–scale simulation with the mathematical programming algorithm. Furthermore, the high cost of objective–function evaluations for the direct–insertion method reduces the interactivity of the design process. Finally, the number of objective–function evaluations required to achieve the optimal solution is typically not known a priori, and computational resources may thus be exhausted before a feasible design is achieved. The latter is especially pronounced in the case of multidisciplinary optimization (Newman et al., 1992).

To circumvent these difficulties, we pursue a surrogate–based approach to optimization (McKay et al., 1979; Sacks et al., 1989; Barthelemy and Haftka, 1993; and Yesilyurt and Patera, 1995). The surrogate approach offers a number of advantages. First, by construction, surrogate evaluations require very little computational effort, and are thus easily incorporated into optimization procedures. Second, the very low computational requirement permits a more complete optimization by assuring that computational resources will not be exhausted before the design is complete. Third, surrogates create highly interactive and flexible design environments, allowing multiple optimization tasks characterized by different design parameters to be pursued more easily.

As regards disadvantages, the most important drawback is that surrogate–based optimization introduces a new source of error. A surrogate validation strategy and appropriate error norms must be developed to
quantify the discrepancy between the surrogate and the actual simulation, and to estimate the errors in system predictability and optimality. The former provides a bound on the behavior of the actual objective function in the neighborhood of the surrogate minimizer, while the latter provides a bounding region in which the actual minimizer lies.

In this paper, we first describe the optimization problem which is central to the remainder of the paper. We then briefly review our baseline surrogate framework, and offer references to more detailed descriptions. Next, we present several improved surrogate algorithms that have been developed. In the final section of the paper, we apply the surrogate framework to two design optimization problems: optimization for pressure-driven and other references to more detailed descriptions. Next, then briefly review our baseline surrogate framework, which is central to the remainder of the paper. We
tual minimizer lies.

The optimization then proceeds as for direct-insertion, surrogate-based optimization problem is given by

$$\mathbf{p}^* = \arg \min_{\mathbf{p} \in \Omega} |S(\mathbf{p}) - \lambda|,$$  

(1)

where $\mathbf{p}$ is the vector of $M$ design inputs which lie in the input (or “design”) domain $\Omega \subset \mathbb{R}^M$, $S(\mathbf{p})$ is the input–output relation, and $\lambda$ is the target value. This problem is also referred to as a “discrimination” problem (Seber and Wild, 1989). More general objective functions are examined in Yeşilyurt (1995), and Yeşilyurt and Patera (1995).

As already described, direct insertion of the full simulation for $S(\mathbf{p})$ in equation (1) is impractical. Instead, a surrogate, $\tilde{S}(\mathbf{p})$, for the simulation is inserted into the optimization problem. The minimizer for the resulting surrogate–based optimization problem is given by

$$\tilde{\mathbf{p}}^* = \arg \min_{\mathbf{p} \in \Omega} |\tilde{S}(\mathbf{p}) - \lambda|.$$  

(2)

The optimization then proceeds as for direct-insertion, but the surrogate is evoked instead of the simulation.

The advantages of using a surrogate framework for optimization have already been described. The issues of predictability and optimality must be addressed (Bohlin, 1991). For predictability, we are concerned with how the actual simulation behaves in the vicinity of the surrogate–predicted minimizer, $\tilde{\mathbf{p}}^*$. We seek to be able to bound $|\tilde{S}(\mathbf{p} \approx \tilde{\mathbf{p}}^*) - S(\tilde{\mathbf{p}})|$, and also to ensure that this bound is acceptably small. The proximity of $\tilde{\mathbf{p}}^*$ to $\mathbf{p}^*$, the actual optimizer, (i.e. optimality), is also important, although not as critical in this context as predictability. The validation strategy which yields predictability statements is addressed next.

SURROGATE FRAMEWORK

The Bayesian–validated surrogate methodology presented here has the distinguishing attribute of integrating a complete and rigorous validation framework. The outcome of the validation step is fully incorporated into the a posteriori purposive analysis, and hence the predictability of any optimization is a natural extension of the validation. The validation itself is based on the results of order statistics (David, 1981) where the validation sample size ($N^{va}$) is determined a priori. In this section, we briefly describe the baseline surrogate-validation algorithm and a posteriori error analysis framework. A full description, and additional algorithms, can be found in Paraschivoiu (1995) and Otto et al. (1995).

A Baseline Approach

We first define a model prediction error estimator:

$$U = \max_{\mathbf{p} \in \mathbb{R}^M} |R_j - \tilde{S}(\mathbf{p})|,$$  

(3)

where $\lambda^{va}_{\mathbf{p}}$ is a random set of $N^{va}$ points in $\Omega$ distributed according to an importance function $\rho(\mathbf{p})$. Then, given a vector of “uncertainty” parameters ($\Theta$) we can calculate the required sample size. Two different predictability statements have been developed: “Prediction Region” ($\mathcal{PR}$) and “Proximal–Candidate” ($\mathcal{PC}$). The former gives a predictability statement for a region neighboring the surrogate–based optimum, $\tilde{\mathbf{p}}^*$, while the latter evaluates the predictability of a specific nearby candidate design.

The validation sample sizes for each approach are

$$\mathcal{PR} : \quad N^{va} \equiv \ln \varepsilon_2 / \ln(1 - \varepsilon_1),$$  

(4)

$$\mathcal{PC} : \quad \varepsilon_2 = \frac{1}{\eta(N^{va} + 1)} (1 - (1 - \eta)^{N^{va} + 1}),$$  

(5)

where $\Theta = (\varepsilon_1, \varepsilon_2) \in (0, 1)^2$ in equation (4) and $\Theta = (\eta, \varepsilon_2) \in (0, 1)^2$ in equation (5).

A Posteriori Error Analysis

Two approaches to the error analysis of our surrogate–predicted optimum points are presented. The first is a “Prediction Region” approach and the second is a “Proximal–Candidate” based approach. Both of these analyses seek to address the predictability concerns already described.

Prediction Region. The inputs to the “Prediction Region” error analysis are the model prediction error, $U$, from the surrogate validation, and the surrogate–predicted optimizer $\tilde{\mathbf{p}}^*$. We introduce a parameter, $\eta$,
\( \varepsilon_1 < \eta < 1 \), and construct a region, \( \mathcal{R} \) such that
\[
\int_{\mathcal{R}} \rho(p) dp = \eta . \tag{6}
\]
Here, \( \mathcal{R} \) is typically chosen as that region, \( \mathcal{R}' \), of \( \rho \)-measure \( \eta \) which minimizes \( \Delta_{\text{max}} \), the maximum distance, \( \Delta(p', \hat{p}^*) \), between all points \( p' \in \mathcal{R}' \) and \( \hat{p}^* \). The distance \( \Delta(\cdot, \cdot) \) can be any valid distance between \( p' \) and \( \hat{p}^* \).

Introducing a surrogate sensitivity parameter
\[
\delta = \max_{p \in \mathcal{R}} |\hat{S}(p) - \hat{S}(\hat{p}^*)| , \tag{7}
\]
the following statement can be made: There exists many points, \( \hat{p} \), in the region \( \mathcal{R} \) such that, with probability greater than \( 1 - \varepsilon_2 \),
\[
|S(\hat{p}) - \lambda| \leq |\hat{S}(\hat{p}^*) - \lambda| + \varepsilon', \tag{8}
\]
where \( \varepsilon' = U + \delta \) is the predictability gap. This statement provides a probabilistic bound on the actual system performance, \( |S(\hat{p}) - \lambda| \), for points, \( \hat{p} \), near the surrogate-predicted optimum, \( \hat{p}^* \). Recall that the number of simulations to obtain \( U \) is given by Equation (4).

**Proximal–Candidate.** In many problems it may be difficult, or even impossible, to explicitly construct prediction regions \( \mathcal{R} \subset \Omega \) as described in the last section. Another approach is to only observe the prediction region (and not construct the region) through sample candidate designs, \( \hat{p}^* \), which, with a prescribed probability, will exhibit actual performance within the predictability gap of the surrogate-predicted optimal performance. We refer to this method as a “Proximal–Candidate” approach to a posteriori analysis (Otto, 1995, and Otto et al., 1995).

The two inputs to the validation are \( \eta \) and \( \varepsilon_2 \), where \( 0 < \eta, \varepsilon_2 < 1 \). In the a posteriori analysis, we introduce that region \( \mathcal{R} \subset \Omega \) which satisfies Equation (6) and minimizes \( \Delta_{\text{max}} \), and sample a \( \hat{p}^* \in \mathcal{R} \) according to the probability density
\[
\rho_{\mathcal{R}}(p) = \frac{1}{\eta} \rho(p)|_{\mathcal{R}} \quad \forall p \in \mathcal{R} . \tag{9}
\]
Using acceptance–rejection techniques, the region \( \mathcal{R} \) can be measured and sampled without actually constructing the region.

The a posteriori error statement (similar to the “Prediction Region” result) is
\[
\Pr(|S(\hat{p}^*) - \lambda| \leq |\hat{S}(\hat{p}^*) - \lambda| + \varepsilon') \geq 1 - \varepsilon_2 , \tag{10}
\]
where \( \varepsilon' = U + \delta' \) is the predictability gap and \( \delta' = |\hat{S}(\hat{p}) - \hat{S}(\hat{p}^*)| \). In words, Equation (10) says that, with probability greater than \( 1 - \varepsilon_2 \), there will exist a candidate design \( \hat{p}^* \) for which actual system performance, \( |S(\hat{p}^*) - \lambda| \), is within \( \varepsilon' \) of the surrogate-predicted optimum objective value, \( |\hat{S}(\hat{p}^*) - \lambda| \). The number of validation points needed is given by Equation (5).

Finally, we remark that the “uncertainty” vector \( \Theta \) measures the degree of precision with which the surrogate must be validated. As \( \varepsilon_1 \), \( \varepsilon_2 \), and \( \eta \) decrease, the number of validation points increases, and more information as to the quality of the surrogate is extracted. The probabilistic framework results in a fixed validation sample size, allowing the designer to tune the “uncertainty” vector to the available resources.

**IMPROVED ALGORITHMS**

In this section we describe several improvements to the baseline algorithm that have been developed. A full description of the additional algorithms can be found in Paraschivoiu (1995).

**Elemental Algorithm**

One of the shortcomings of the baseline approach to validation is the global nature of the model prediction error \( U \). That is, a large error at one point of the input space has a dominating influence over the entire input space. One way to address this drawback is to divide the input domain, \( \Omega \), into \( L \) non-overlapping, elements (or sub-domains) (Paraschivoiu, 1995; Otto et al., 1995). Validation is performed independently on each sub-domain yielding a local model prediction error, \( U^\ell, \ell = 1, \ldots, L \), associated with each element, \( \Omega^\ell \). The \( U^\ell \) for each element is then used in the a posteriori error analysis.

For the elemental algorithm, the size of the validation sample for each element, \( N^{\alpha \ell} \), is
\[
P_{\mathcal{R}} : \quad N^{\alpha \ell} = \frac{\ln(1 - (1 - \varepsilon_2)^{1/L})}{\ln(1 - L \varepsilon_1)} , \tag{11}
\]
\[
P_{\mathcal{P}} : \quad \varepsilon_2 = 1 - L N^{\alpha \ell} \int_{0}^{\eta L} \left( 1 - \frac{z}{\eta L} \right) \left( 1 - (1 - z)^{N_{\alpha \ell}} \right)^{L - 1} dz . \tag{12}
\]

The locality improvement in the model prediction error estimator is obtained with a small increase in the number of validation points for small numbers of elements. For a very large number of elements, even fewer validation points are needed than for a single element.
**Cross-Validation, Re-Sampling**

A natural extension of the elemental algorithm for the \( PR \) approach is an intra-model, cross-validation procedure (Paraschivoiu, 1995) which allows the use of the available simulation data for both construction and validation. Indeed, due to the elemental partitioning, the validation data from one element can be used for the construction of a surrogate model in another element. The only restriction is an ordered dependence requirement, which means that validation data can only be used for construction in higher numbered elements (with respect to some ordering).

**Model-Sequential Approach**

To achieve a desired target level for the model prediction error estimator, a model-sequential approach is pursued. We assume that we are given a sequence of \( J \) models, \( S_j(p) \), \( j = 1, \ldots, J \), and a desired model prediction error, \( \alpha > 0 \). The sequential algorithm validates each subsequent model with the same points as used for the previous model plus one new random validation point. If, for the \( j^{th} \) model, the model prediction error estimator is less than or equal to \( \alpha \), then the model is selected. If, on the other hand, all the models have been tested, and none has satisfied the desired model prediction error, then the last model is selected. Respectively, \( \alpha \) or \( U^D \) then serves as the model prediction error estimator in the \textit{a posteriori} analysis. The initial number of validation points is given by

\[
N^{va}(0) = \ln \left( \frac{\varepsilon_1 \varepsilon_2}{1 - (1 - \varepsilon_1)} \right) / \ln(1 - \varepsilon_1) - 1. \tag{13}
\]

**An Adaptive, Model-Sequential Approach**

To integrate construction into the model-sequential approach, an adaptive algorithm is considered. In this algorithm, simulation evaluations used for the validation of previous models are incorporated into the surrogate construction, and a new set of validation points is chosen for each subsequent validation. The size of each sample set (assuming \( J \) finite) is

\[
N^{va} = \ln \left( \frac{\varepsilon_2}{J} \right) / \ln(1 - \varepsilon_1). \tag{14}
\]

The desired model prediction error, \( \alpha \), is used as previously described in the model-sequential approach. In practice, this approach is very powerful for the construction of surrogates that achieve the desired model prediction error. However, the cost of evaluating a new set of validation points at each step is expensive. This drawback is currently under investigation.

We conclude by commenting that the elemental partitioning can be incorporated in both the sequential and the adaptive methods. This combination offers significant advantages in obtaining the desired local model prediction error with a minimal number of validation points. Indeed, in some elements, the desired elemental model prediction error can be reached faster than in others, and the iterative process stopped in these elements.

**APPLICATIONS**

**Flow in a Trapezoidal Duct**

As our first illustrative example, we consider fully developed laminar fluid flow in a duct of constant trapezoidal cross-section. The optimization problem is to determine the geometry, for a given cross-sectional area \( A_0 \), that achieves a target flowrate. In this problem the pressure gradient is fixed. The trapezoidal geometry is defined by two inputs; the angle between the base and the side, and the height normalized with respect to the base length. The non-dimensional form (\(^1\)) of the Poiseuille flow equation inside the duct (\( D \)), with non-slip boundary conditions on the walls (\( B \)) is

\[
\begin{aligned}
\nabla^2 \tilde{u} &= -1 & & \text{in } D, \\
\tilde{u} &= 0 & & \text{on } B. \\
\end{aligned} \tag{15}
\]

Using non-dimensional variables we relate the normalized flowrate to the geometrical properties. We first write the pressure drop (\( \Delta P \)) as a function of the friction factor. Knowing that, for laminar flow, the friction factor is simply a shape-dependent constant \( g \) over the Reynolds number, we can obtain an expression for the dimensional flowrate; here the Reynolds number is taken with respect to the hydraulic diameter of the dimensionless trapezoidal section, \( D_h \). The normalized flowrate can be written

\[
\hat{Q} = \frac{Q}{2\mu D_h^2 A_0^2} = \frac{\hat{D}_h^2}{gA} = \frac{\hat{u}}{2A}, \tag{16}
\]

where \( \hat{L} \) is the length of the duct, \( \mu \) is the dynamic viscosity of the fluid, and \( \hat{u} \) is the non-dimensional cross-sectional average velocity.

Equation (16) is useful in incorporating prior information on particular geometries through different values of \( g \). In addition, we see that our optimization problem is simplified by the absence of scale dependency, that is, our solution is independent of the cross-sectional area, \( A_0 \), of the dimensional duct. The optimization problem is a simple inverse-simulation type: find the input geometry (\( \hat{h}, \theta^* \)) that minimizes the difference between the target \( \lambda \) and the numerical solution.
of the normalized flow rate. That is,

$$\hat{h}^* = \arg \min_{\hat{h}, \theta} |S(\hat{h}, \theta) - \lambda|$$

(17)

where \( \lambda = \frac{Q_{\text{desired}}}{\rho U L} \) and \( S(\hat{h}, \theta) = \frac{\hat{u}}{2A} \).

To obtain the velocity field in the trapezoidal channel, a numerical approximation scheme is used to solve Equation (15). In particular, we employ a second order finite element method on triangular elements in \( \mathbb{R}^2 \) with automatic mesh generation and conjugate gradient iterative solution. For all geometries, approximately 4500 degrees-of-freedom are used, and each evaluation of the flow rate takes approximately 10 seconds on a single processor of a HP9000/735 workstation. The numerical results can be considered to have a normalized \( \ell_\infty \) error norm less than \( 5 \times 10^{-4} \), which motivates us to consider these results to be without “noise” (Yeşilyurt et al., 1995).

**Surrogate approach.** The design (or input) space covers different trapezoidal geometries ranging from a rectangle to an isosceles triangle. Small clearance limits are included to yield the following input space:

$$\Omega = \left\{(\hat{h}, \theta) \mid 0.01 \leq \hat{h} \leq \min(\tan \theta, 4.0); 0.05 \leq \theta \leq \frac{\pi}{2} \right\}.$$  

(18)

In all cases the importance function is taken to be uniform. We first apply the baseline algorithm to find the trapezoidal geometry that achieves a target normalized flow rate. In this first optimization exercise, we consider only variation of the height of the section, while fixing the angle of the trapezoid to \( \frac{\pi}{2} \); the corresponding “slice” of \( \Omega \), \([0.0, 0.5] \), is denoted \( \Omega^s \).

The four steps of the surrogate-based optimization are as follows. First, we construct a surrogate based on prior information only; in particular, we take \( g = 53.33 \) (the known value for an equilateral triangle) in Equation (16). In Figure 1, the normalized flow rate is plotted versus the dimensionless height, and the surrogate is shown by the solid line. Second, we use this surrogate to find the height (\( \hat{p}^* \)) corresponding to the desired target flow rate, \( \lambda = 0.01 \). Third, we wish to know how well our surrogate has performed. To obtain this information, validation is carried out. The number of validation points, that is the number of calculations of the flow rate for random height values, is obtained from Equation (4). For \( \varepsilon_1 = \varepsilon_2 = 0.2 \), eight validation points are needed. These points are represented by circles on Figure 1. The maximum difference between the surrogate and the exact simulation value of the normalized flow rate over all the validation points determines the model prediction error estimator.

![Figure 1: Surrogate for duct flow example using the “baseline” algorithm.](image)

U. Fourth, and finally, we can make the following predictability statements: With probability \( > 0.8 \ (1 - \varepsilon_2) \); (a) the surrogate error is less than \( U = 15.8 \% \) over at least \( 80 \% \ (1 - \varepsilon_1) \) of the input domain \( \Omega^s \), and (b) there exists many points in the prediction neighborhood of the surrogate optimum geometry \( \hat{p}^*, \mathcal{P}_{\Delta}^s \), where the exact simulation is within \( 34.6 \% \) of the target flow rate, normalized by the maximum value of the surrogate.

The predictability statement indicates (a) quantitatively how good the surrogate is over the entire domain, and (b) how well it performs in the neighborhood of the actual optimum. The neighborhood is indicated on the figure by \( \mathcal{P}_{\Delta}^s \). This example is useful to demonstrate not how well we can construct a surrogate, but rather, what can be said about the surrogate once it is constructed. In this case, we see that our model does not perform very well.

One way to decrease the error estimate, for a given construction, is to decompose the input space into elements and validate the surrogate in each element. We divide the input space into eight different elements (shown in Figure 2). In addition, we use the cross-validation algorithm to simultaneously construct and validate as follows: First, element one is constructed using prior information (no flow at \( \hat{h} = 0 \)) and only one construction point. Second, this surrogate is validated with three validation points. Third, the points used to validate the surrogate in the current element are used to construct the surrogate in the next element with a quadratic extrapolation. The second and third steps are repeated until surrogates are constructed and validated in all of the elements. This procedure results in a surrogate with a local \( U \) of less than 0.8% in each de-
The exact solution to this optimization problem is a square cross-section geometry. We can compare this with our surrogate-based optimization on Figure 3 and observe that they are very close. In addition, the difference between the maximum normalized flowrate predicted by the surrogate and predicted by the simulation is less than 0.1%. This is well within the model prediction error estimate. For this optimization problem, we see that sharp results can be obtained at the expense of increased cost. In practice, this cost could be decreased by using combinations of the previous algorithms.

We conclude by acknowledging that this test case falls short of completely illustrating the advantages of the surrogate framework. For example, due to the relatively inexpensive computations, and because the objective function is smooth, a direct insertion approach could be used. However, for multiple design optimizations, direct insertion, even in this case, can be prohibitively expensive compared to the surrogate framework. Furthermore, in practice, most engineering optimization studies require expensive simulations (Yeşilyurt et al., 1995; Yeşilyurt and Patera, 1995).

**Shape Optimization**

Many engineering design problems involve not only configuration optimization, in which optimal values of a small number of parameters (e.g., radius, position, angle, etc.) are desired, but also shape optimization, in which the optimal form of a curve is desired. Given the deterioration of volume-based error estimates present for problems with high input dimensions, it would ap-
pear that shape optimization problems (described exactly by an infinite-dimensional input space) would be intractable in this framework. This is not the case, however, primarily due to two reasons. First, highly irregular shapes will typically be non-optimal, allowing for truncation of the input space. Second, the remaining inputs are typically highly correlated, further reducing the effective volume of the input space.

To demonstrate the surrogate-based shape optimization procedure, we examine the Stokes drag over an axisymmetric, fore–aft symmetric body. The simulation for which the surrogates are substituted is a finite element solver utilizing an Uzawa nested conjugate gradient iteration. The inputs to the problem are a set of coefficients which describe the shape, $b(\theta)$, and the output is the non-dimensional drag, $F_D$. The design optimization problem is to find the optimal shape, $b^*(\theta)$, for a given target drag, $\lambda$:

$$b^*(\theta) = \arg \min_{b(\theta) \in W, \|b(\theta)\| \leq C(\theta)} \left| F_D - \lambda \right|.$$  \hspace{1cm} (19)

The general geometry and the physical constraint on the optimal profile, $C(\theta)$, are shown in Figure 4.

The surrogate for the Stokes drag over the body is simply the Stokes drag over a sphere of equivalent surface area. Other surrogates such as response surfaces constructed from simulation data could be considered instead. The surrogate-based optimization problem is identical to Equation (19), but the simulation result, $F_D$, is replaced by the drag surrogate, $\tilde{F}_D$.

The quarter–body profile of the shape in the first quadrant describes the full three–dimensional body. The shape is expressed as a positive function $b(\theta) \in W_+(0,\pi]$, where $W_+$ is a set of admissible functions. We introduce a basis for $W_+(0,\pi]$ and truncate this basis to $M$ terms. The shape function can then be expressed as

$$b(\theta) = 1 + \sum_{j=1}^{M} A_j \cos(2j\theta).$$  \hspace{1cm} (20)

The result is that the shape profiles are parameterized by $A = \{A_1, \ldots, A_M\}$, which acts as the input, $p$. The shapes in this problem are described by 10 inputs ($M = 10$).

To generate a random ensemble of shapes for validation, and for Monte Carlo sampling for the proximal–candidate error analysis, a random–shape process inducing an importance function on the set of coefficients is employed (Elishakoff, 1983; Otto, 1995; Otto et al., 1995). With this method, certain smoothness conditions at the endpoints of the shape are enforced, as are statistical quantities of the ensemble, such as the mean and covariance. Through the mean and covariance of the radius–function describing the shape, the designer reflects his prejudices as to the class of shapes that will contain the optimizer.

We select $\eta$, $\varepsilon_2$, and $\Delta(\cdot,\cdot)$ and proceed with the validation, the surrogate–based optimization, and the candidate–based $a$ posteriori error analysis. For this problem we choose $\eta = .18$, and $\varepsilon_2 = .18$. The distance criterion chosen, $\Delta(\cdot,\cdot)$, is the Hausdorff distance. With Equation (5), and our values of $\eta$ and $\varepsilon_2$, we find that 30 validation points are needed. The resulting model prediction error is $U = .15$.

For the design problem, we choose a target drag $\lambda = 304$ dynes, and find the surrogate–predicted optimizer, $\tilde{b}^*(\theta)$. Based on $\eta = .18$, we find $\Delta_{\text{max}} = .35$, using a “best scaled” geometry for the acceptance/rejection. The region in the proximity of the $\tilde{b}^*(\theta)$ is then queried for a candidate design $\tilde{B}^*(\theta)$. With this candidate, the following predictability statement can be made: With confidence greater than .82,

$$|F_D(\tilde{B}) - \lambda| \leq |1 - \beta^*| + \beta^* U = .18.$$  \hspace{1cm} (21)

In words, with confidence greater than .82, the absolute difference between the actual drag on the candidate design and the target drag, $\lambda = 304$ dynes, normalized by the target drag, will be less than or equal to .18. The coefficient, $\beta^*$, present in the final predictability statement results from the scale-invariance (self-similarity) in the linear Stokes drag relation. This coefficient is near one and has the effect of increasing the model prediction error, $U$, from .15 to the value .18 present in Equation (21). Three-dimensional plots of the surrogate-predicted optimal design and the proximal–candidate are shown in Figure 5.

A designer can conclude from this result that there is likely to be a design near the surrogate-predicted optimal design which will satisfy the target drag re-
would likely introduce a finite \( \delta \) in (10) (not present for the simple surrogate presented here). To reduce \( \delta \), a smaller prediction region \( \mathcal{R} \) (physically) is necessary, which requires a smaller \( \varepsilon \) and, therefore, either more validation points or a better \( \rho(p) \) (one that more accurately anticipates the “location” of the optimizer).

ACKNOWLEDGMENTS

This work was supported by the ARPA under Grant N00014-91-J-1889, by the ONR under Grants N00014-90-J-4124 and N00014-89-J-1610, by the AFOSR under Grant F49620-94-1-0121, and by NASA under Grant NAG 1-1613. We would also like to acknowledge an FCAR Aeronautical Scholarship (M.P.) and the NASA Graduate Training Program (J.O).

REFERENCES


COMPUTER–SIMULATION SURROGATES FOR OPTIMIZATION: APPLICATION TO TRAPEZOIDAL DUCTS AND AXISYMMETRIC BODIES

by

John C. Otto
Department of Aeronautics and Astronautics, M.I.T.
Multidisciplinary Design Optimization Branch
NASA Langley Research Center
Hampton, Virginia

Marius Paraschivoiu, Serhat Yesilyurt, and Anthony T. Patera
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts

July 10, 1995