Efficient Global Optimization with Gradient Finish for Design Under Uncertainty

Nicholas J. DiGregorio* and Aaron H. Wright†
NASA Langley Research Center, Hampton, VA, 23681, USA

The Efficient Global Optimization (EGO) algorithm is extended to include a gradient-descent-based finish upon reaching a threshold value of the expected improvement function. Emphasis is placed on efficient evaluation of local gradients using Kriging models during the gradient-based finish to enable application to design under uncertainty (DUU) problems. The modified algorithm is applied to both the well-known Rosenbrock function and a more challenging hypersonic inlet design under uncertainty problem. Results demonstrate improvement in locating the global optimum compared to the classical implementation of EGO, as well as a reduced number of true function evaluations compared to pure gradient-based algorithms. For the Rosenbrock function, a global optimum is returned using an average of 12% fewer function calls than a gradient-based optimizer with comparable tolerances. For the design under uncertainty problem, a global optimum is found using an average of 75% fewer function calls than a gradient-based optimizer. Global Pareto fronts of multiobjective DUU problems are obtained at little additional cost after a single optimization is complete.

Nomenclature

a	Uncertainty expansion coefficient	W	Weighting coefficient
B	Interaction polynomial	$\boldsymbol{X}$	Training inputs
$\boldsymbol{D}$	Deterministic variables	$\boldsymbol{x}$	Input variables
E	Expectation value	$\mathcal{Y}$	Kriging model output
F	Kriging model trend	$\boldsymbol{Y}$	Training outputs
I	Improvement function	Z	Stationary, zero-mean Gaussian process
J_{robust}	Robust objective function	α	Deterministic component of random variable
$\boldsymbol{K}$	Kriging correlation matrix	β	Mean of Kriging model
k_ℓ	Kriging correlation function	ϵ_{cv}	Cross-validation error
$\mathcal{L}$	Likelihood function	θ	Inlet ramp angle
ℓ	Kriging length-scale hyperparameters	μ	Mean
M	Mach number	$\hat{\mu}$	Mean value of Kriging predictor
m	Number of variables	ξ	Random variables
m_d	Number of design variables	Π	Total pressure recovery
N_t	Number of terms in expansion	σ	Standard deviation
n	Number of training samples	σ_Z	Standard deviation of Z
OSR	Oversampling ratio	$\hat{\sigma}$	Standard deviation of Kriging predictor
P	Number of dispersed sample points	Φ	Cumulative distribution function
$P_{q,m}$	Polynomial chaos expansion	ϕ	Probability density function
q	Order of polynomial	Ψ	Random variable basis function
R	Response function		

I. Introduction

Hypersonic airbreathing vehicles are highly complex systems composed of multiple components with tightly coupled physics. Due to the nature of this coupling, optimization of a single component may not yield optimal results for the

*Aerospace Engineer, Vehicle Analysis Branch, Systems Analysis and Concepts Directorate, AIAA Member.
†Aerospace Engineer, Vehicle Analysis Branch, Systems Analysis and Concepts Directorate.

overall system performance. In addition, performance margins are typically small enough that system uncertainty should be quantified and preferably minimized, otherwise system uncertainty may be greater than the performance margins. These factors motivate the need for multidisciplinary design optimization and uncertainty quantification of the overall system, rather than individual components. Obtaining accurate predictions of hypersonic vehicle performance often requires expensive high-fidelity simulation. Performing both uncertainty quantification and optimization on such high-fidelity simulations can be computationally cost-prohibitive with current algorithms. Therefore, there is a need to develop improved optimization algorithms suitable for the task.

Although gradient-based optimization approaches are well established, they have the crucial limitation of never exploring in an uphill direction. If the objective function is not convex, a gradient-based optimizer will only find a local minimum, with no guarantee of finding the global optimum. The problem of global optimization is difficult, and the algorithms are correspondingly more complex than gradient-based optimizers. Instead of simply steering the search downhill as quickly as possible, global optimizers must explore the entire domain of the function to assess where the global optimum lies. Numerous approaches have been developed for this problem. Some of the more common techniques include genetic algorithms, basin-hopping algorithms, branch-and-bound algorithms, and surrogate model-based algorithms [1]. This work is based on a surrogate model method known as Efficient Global Optimization (EGO) [2], chosen for its efficiency and derivative-free nature. Derivative-free algorithms are advantageous to use in design under uncertainty (DUU) applications because the computational cost of evaluating the gradients of uncertainty can be avoided.

Previous work by DiGregorio et al. [3] explored the use of EGO for hypersonic inlet design. The authors found that EGO was able to locate the valley containing the global optimum, but the algorithm was insufficiently precise to reproduce known trends [4] of the optimum inlet design. A classical gradient-based optimization algorithm, sequential quadratic programming (SQP), was precise enough to reproduce the known trend in the deterministic problem. However, the authors noted that gradient-based optimization algorithms such as SQP are infeasible for use in design under uncertainty applications due to the extensive computational cost of evaluating the gradients of uncertainty.

The objective of the present work is to develop and test an optimization algorithm that combines the best elements of derivative-free global optimization methods and gradient-based local optimization methods, while remaining computationally efficient enough to be applied to design under uncertainty problems. Ideally, the method should be efficient enough to allow for design space exploration and global optimization of multidisciplinary systems with highly expensive function calls. The method developed is hereafter referred to as efficient global optimization with gradient finish (EGO-GF). The remainder of the paper is organized as follows: Section II describes the theory and methods used in this work, including algorithm development; Section III presents results from applying EGO-GF to two sample problems; and finally, Section IV summarizes the conclusions of this work.

II. Methodology

The following Sections describe the background theory used in the present work. Section II.A contains a brief review of Kriging theory and Section II.B describes EGO. Section II.C describes uncertainty quantification through the use of non-intrusive point collocation polynomial chaos expansions (PCE), to be applied for design under uncertainty. Finally, Section II.D describes the algorithm developed for the present work, EGO-GF, which utilizes material from the preceding three Sections.

A. Kriging: A Brief Overview

A Kriging model is a stochastic interpolation algorithm which assumes that the model output is a realization of a Gaussian process:

$$\mathcal{Y}(\mathbf{x}) = F(\mathbf{x}, \beta) + Z(\mathbf{x}) \tag{1}$$

Here, $F(\mathbf{x}, \beta)$ is the mean value, or trend, of the Gaussian process, and $Z(\mathbf{x})$ is a stationary Gaussian process with zero mean, variance σ_Z^2, and covariance $\sigma_Z^2 k_\ell(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$, where k_ℓ is the correlation function, and $\mathbf{x}$ is an m-dimensional variable input vector. The trend may be assumed to have a known constant value β_0 (simple Kriging), a constant-but-unknown value β (ordinary Kriging), or be a linear combination of polynomials $\sum_{i=1}^{P} \beta_i f_i(\mathbf{x})$) (universal Kriging) [5]. Ordinary Kriging is used in the original formulation of the EGO algorithm [2] and is sufficient for the present work as well. There exist many correlation functions from which k_ℓ may be selected; here, the squared exponential function is

used:

$$k_\ell(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = \sum_{i=1}^{m} \exp\left[-\ell_i (x_i^{(1)} - x_i^{(2)})^2\right] \tag{2}$$

where ℓ is an m-dimensional vector of hyperparameters. The hyperparameters do not represent any physical component of the system; they act as length scale parameters for the correlation. Given a set of training data with input vectors $\mathbf{X}^{(j)}$ and responses $\mathbf{Y}$, one can form the correlation matrix $\mathbf{K}$ according to:

$$\mathbf{K}^{(i,j)} = k_\ell\left(\mathbf{X}^{(i)}, \mathbf{X}^{(j)}\right) \tag{3}$$

The parameters β and σ_Z^2 may then be calculated as:

$$\beta = \left(\mathbf{1}^T \mathbf{K}^{-1} \mathbf{1}\right)^{-1} \mathbf{1}^T \mathbf{K}^{-1} \mathbf{Y} \tag{4}$$

$$\sigma_Z^2 = \frac{1}{n} (\mathbf{Y} - \beta \mathbf{1})^T \mathbf{K}^{-1} (\mathbf{Y} - \beta \mathbf{1}) \tag{5}$$

where n is the number of samples in the data, and $\mathbf{1}$ is a length-n vector of ones. If the values for hyperparameters ℓ are not known a priori, as is often the case, they may be estimated by maximizing the likelihood function:

$$\mathcal{L} = \frac{1}{n} \log\left(\det\left(\mathbf{K}\right)\right) + \log\left(\mathbf{Y}^T \mathbf{K}^{-1} \mathbf{Y}\right) \tag{6}$$

For a new sample $\mathbf{x}^*$ not in the training data, trained Kriging models make predictions in the form of Gaussian variables, with mean $\hat{\mu}$ and variance $\hat{\sigma}^2$ given by:

$$\hat{\mu}(\mathbf{x}^*) = \hat{\beta} + \mathbf{k} \mathbf{K}^{-1} (\mathbf{Y} - \hat{\beta} \mathbf{1}) \tag{7}$$

$$\hat{\sigma}^2(\mathbf{x}^*) = \sigma_Z^2 \left(1 - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k} + \frac{\left(1 - \mathbf{1}^T \mathbf{K}^{-1} \mathbf{k}\right)^2}{\mathbf{1}^T \mathbf{K}^{-1} \mathbf{1}}\right) \tag{8}$$

where $\mathbf{k}$ is shorthand for the vector indicated by $\mathbf{k}_\ell^{(i)} = k_\ell(\mathbf{x}^*, \mathbf{X}^{(i)})$, the correlations between x^* and every point in the training set $\mathbf{X}$.

Finally, once a Kriging model has been trained, its accuracy can be assessed by using k-fold cross validation [6]. For this work, leave-one-out cross validation error ϵ_{cv} is used, normalized by the standard deviation of the training data responses, $\sigma_{\mathbf{Y}}$, to ensure scale-independence.

$$\epsilon_{cv} = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^{n} (\mathbf{Y}_i - \hat{\mu}^{(-i)}(\mathbf{x}^{(i)}))^2}}{\sigma_{\mathbf{Y}}} \tag{9}$$

The easy availability of variance information from the Kriging model sets it apart from other interpolation models, as it provides an analytical, quantitative estimate of the uncertainty in the model's predictions. This property is crucial to EGO, as will be explained in the next Section.

B. Efficient Global Optimization

The EGO algorithm belongs generally to the class of methods known as Bayesian optimization [1], also referred to as active learning or adaptive Kriging methods in the literature. The general structure of these algorithms is to use a Kriging surrogate model to identify points of interest, evaluate them, update the Kriging model with the new data, and then repeat as shown in Fig. 1. The "best" points to explore depend on the nature of the problem. For exploration purposes, the points should generally lie in areas of high model uncertainty; for exploitation, they should fall near an optimum. EGO utilizes the expected improvement function (EIF) to identify the next best point to sample. The EIF is a weighted, linear combination of the Kriging prediction's value and uncertainty. It is derived as the expectation value of the improvement function I:

$$E\left[I(\mathbf{x})\right] = E\left[\max\left(\min(\mathbf{Y}) - \hat{\mu}(\mathbf{x}), 0\right)\right] \tag{10}$$

which evaluates to [2]:

$$E\left[I(\mathbf{x})\right] = \left(\min(\mathbf{Y}) - \hat{\mu}(\mathbf{x})\right)\Phi\left(\frac{\min(\mathbf{Y}) - \hat{\mu}(\mathbf{x})}{\hat{\sigma}(\mathbf{x})}\right) + \sigma\phi\left(\frac{\min(\mathbf{Y}) - \hat{\mu}(\mathbf{x})}{\hat{\sigma}(\mathbf{x})}\right) \tag{11}$$

where $\min(\mathbf{Y})$ is the smallest recorded system response, Φ is the standard normal cumulative distribution function, and ϕ is the standard normal density function. In this way, the algorithm is simultaneously incentivized to both seek likely optima and explore regions where there is insufficient data to determine the rough shape of the function. As a result, areas of the domain unlikely to contain the optimum will be coarsely sampled, whereas areas near the optimum will be more refined.

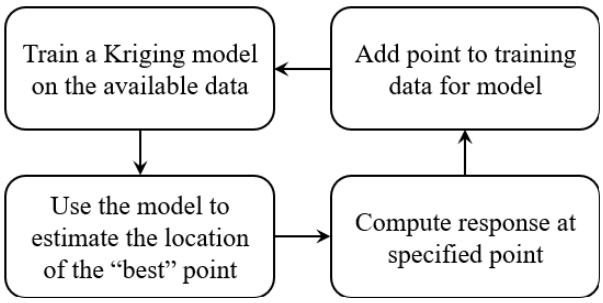


Fig. 1 General structure for active learning algorithms.

C. Uncertainty Quantification via Polynomial Chaos Expansions

The technique used for uncertainty quantification (UQ) in the present work is the polynomial chaos expansion. Polynomial chaos expansions start with the Cameron-Martin theorem [7] by representing a finite variance random variable or response function R as an infinite expansion [8]:

$$R = a_0 B_0 + \sum_{i_1=1}^{\infty} a_{i_1} B_1(\xi_{i_1}) + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1} a_{i_1 i_2} B_2(\xi_{i_1}, \xi_{i_2}) + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1}\sum_{i_3=1}^{i_2} a_{i_1 i_2 i_3} B_3(\xi_{i_1}, \xi_{i_2}, \xi_{i_3}) + \ldots \tag{12}$$

with real coefficients a and interaction polynomial terms B. As stated by Eldred [9], the above expression with polynomial order-based indexing can be replaced with term-based indexing to yield a simpler expression:

$$R(\boldsymbol{D}, \boldsymbol{\xi}) = \sum_{j=0}^{\infty} \alpha_j(\boldsymbol{D})\Psi_j(\boldsymbol{\xi}) \tag{13}$$

where α_j is a deterministic component that is a function of deterministic variables $\boldsymbol{D}$, and Ψ_j is a random variable basis function defined by the random variables $\boldsymbol{\xi}$. The basis functions of the random variables are determined by using the Askey key [10], which relates the distributions of the uncertain parameters to specific orthogonal polynomials. In order to make this approach feasible, the above infinite series of Eq. (13) is truncated at a finite number of terms:

$$R(\boldsymbol{D}, \boldsymbol{\xi}) \approx P_{q,m} = \sum_{j=0}^{N_t - 1} \alpha_j(\boldsymbol{D})\Psi_j(\boldsymbol{\xi}) \tag{14}$$

where $P_{q,m}$ is the polynomial chaos expansion of highest order q for m random dimensions. The number of terms, N_t, is given by:

$$N_t = OSR\frac{(m+q)!}{m!q!} \tag{15}$$

where an oversampling ratio $OSR \geq 1.0$ may be used.

With the random variable basis functions, Ψ_j, known from the Askey key, the expansion coefficients, α_j, must be determined. This can be done through a non-intrusive point collocation approach [11–13], which only requires the

system response values at P dispersed sample points. If the sample points are linearly independent, then the result is the following system of equations:

$$\begin{bmatrix} R(\boldsymbol{D},\boldsymbol{\xi}_0) \\ R(\boldsymbol{D},\boldsymbol{\xi}_1) \\ \vdots \\ R(\boldsymbol{D},\boldsymbol{\xi}_P) \end{bmatrix} \approx \begin{bmatrix} \Psi_0(\boldsymbol{\xi}_0) & \Psi_1(\boldsymbol{\xi}_0) & \cdots & \Psi_P(\boldsymbol{\xi}_0) \\ \Psi_0(\boldsymbol{\xi}_1) & \Psi_1(\boldsymbol{\xi}_1) & \cdots & \Psi_P(\boldsymbol{\xi}_1) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_0(\boldsymbol{\xi}_P) & \Psi_1(\boldsymbol{\xi}_P) & \cdots & \Psi_P(\boldsymbol{\xi}_P) \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_P \end{bmatrix} \tag{16}$$

Note that the minimum value for P is $N_t - 1$ with OSR equal to 1, in which case the linear system may be directly solved. However, if more sample points are available through using an oversampling ratio greater than 1, then the system can be solved with a least squares method.

Once the α_j coefficients have been solved for, the response mean and variance can be determined from closed-form analytical expressions, where angled brackets denote the inner product [14]:

$$\mu = \langle R \rangle \approx \sum_{j=0}^{P} \alpha_j \langle \Psi_j(\boldsymbol{\xi}) \rangle = \alpha_0 \tag{17}$$

$$\sigma^2 = \langle (R-\mu)^2 \rangle \approx \langle (\sum_{j=1}^{P} \alpha_j \Psi_j(\boldsymbol{\xi}))^2 \rangle = \sum_{j=1}^{P} \sum_{k=1}^{P} \alpha_j \alpha_k \langle \Psi_j(\boldsymbol{\xi}) \Psi_k(\boldsymbol{\xi}) \rangle = \sum_{j=1}^{P} \alpha_j^2 \langle \Psi_j(\boldsymbol{\xi})^2 \rangle \tag{18}$$

D. Efficient Global Optimization with Gradient Finish

EGO provides an efficient methodology for approximately locating the global minimum of a function. In practice, however, the ultimate precision of EGO remains inferior to that of gradient-based optimization methods. The issue of accuracy is addressed by running a gradient-based optimizer initialized using the output of EGO. This approach will work well for deterministic problems, but for design under uncertainty problems, computing the gradients of uncertainty can be prohibitively expensive, as shown in Table 1. In order to retain both the efficiency of derivative-free approaches and the accuracy of gradient-based methods for design under uncertainty applications, the EGO-GF algorithm is formulated as shown in Fig. 2.

First, a design of experiments (DOE) is used to coarsely sample the design space and construct the initial Kriging model. As in classic EGO, a Latin hypercube sampling (LHS) scheme is used; this space-filling strategy uses fewer samples than a purely random strategy such as Monte Carlo. Note that the size of this initial DOE will strongly affect the computational cost of the ensuing optimization, and it should be reduced to the extent allowable while maintaining accuracy; common practice is to use ten samples for every independent variable dimension.

Once the points for the initial DOE have been chosen, they are evaluated using the true function. Here, the EGO-GF algorithm begins to depart from classic EGO. If the optimization problem is a design under uncertainty problem, i.e. the objective function being optimized contains any terms for uncertainty statistics μ or σ in it, then polynomial chaos expansion is used to compute the uncertainty statistics at that sample point using Eqs. (17)-(18).

Table 1 Number of function evaluations required at each optimization iteration

Algorithm Type	Design variations per iteration	Function evaluations per variation	Total function evaluations per iteration
Deterministic gradient-based, $O(\Delta \boldsymbol{x})$	$(1 + m_d)$	1	$(1 + m_d)$
Deterministic gradient-based, $O(\Delta \boldsymbol{x}^2)$	$(1 + 2m_d)$	1	$(1 + 2m_d)$
Deterministic derivative-free	1	1	1
Robust gradient-based, $O(\Delta \boldsymbol{x})$	$(1 + m_d)$	$(1 + N_t)$	$(1 + m_d)(1 + N_t)$
Robust gradient-based, $O(\Delta \boldsymbol{x}^2)$	$(1 + 2m_d)$	$(1 + N_t)$	$(1 + 2m_d)(1 + N_t)$
Robust derivative-free	1	$(1 + N_t)$	$(1 + N_t)$

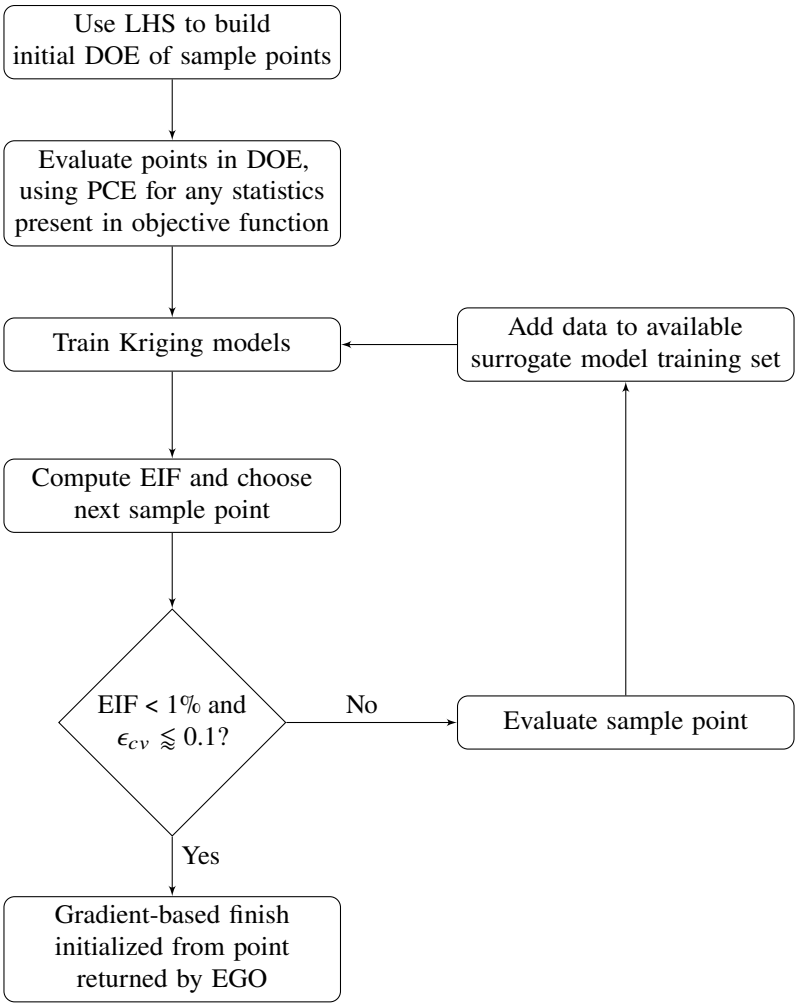


Fig. 2 Flowchart of the EGO-GF process.

After all sample points in the initial DOE have been evaluated, including any necessary uncertainty statistics for the objective function, then Kriging surrogate models of the responses over the design space are built. Whereas classic EGO only builds one Kriging model, EGO-GF builds multiple: one to describe each of the uncertainty metrics within the objective function. Each Kriging model is trained from the polynomial chaos expansions evaluated during the initial DOE. In addition to these component surrogate models, a Kriging model for the objective function is built, similar to classic EGO.

Next, the EIF is evaluated and the next best sample point is identified, exactly the same as in classic EGO. If the expected improvement is under 1% of the current best value, then EGO is considered converged. For EGO-GF, an additional convergence criterion is established: that the standardized error from leave-one-out cross-validation for the component Kriging models, Eq. (9), should be less than approximately 0.1. This additional criterion ensures that the surrogate models are well-trained. Note that relaxing the value for this cross-validation criterion will result in quicker convergence at the cost of accuracy in the surrogate models. If EGO is not converged, then the next sample point is evaluated using the true function and polynomial chaos expansions are used to compute any necessary uncertainty statistics. The data from the function evaluations are added to the training sets for all Kriging models, the models are retrained, and the main EGO loop continues until the EGO convergence criteria are reached.

Once the EGO convergence criteria have been satisfied, the gradient finish starts. In the present work, the L-BFGS-B method [15] is used. Here, the EGO-GF algorithm takes advantage of the Kriging models which have been trained during previous steps. Instead of using the costly true function to evaluate the gradients of uncertainty, the gradient finish is run using the Kriging surrogates generated within the EGO phase. The gradients of uncertainty can then

be easily computed by numerical finite difference. This methodology reduces the cost of evaluating the gradients of uncertainty to the point that gradient-based optimization for design under uncertainty applications becomes feasible. In addition to the efficiency of using Kriging surrogate models for component terms within the objective function, another advantage of this approach is the ease with which weighting coefficients in the objective function can be modified and the gradient finish rerun. Such a strategy enables computing Pareto frontiers of the global optimum for negligible additional computational cost once the first optimization is complete. Note that converging upon an accurate result at the gradient descent stage requires well-trained surrogate models for uncertainty, hence the additional cross-validation criterion placed upon the check for EGO convergence.

III. Results

With the background theory described in the previous Section, this Section describes the results of applying the EGO-GF algorithm to two test problems. First, Section III.A presents results from a well-known canonical deterministic optimization problem: the Rosenbrock function. Then, Section III.B describes an engineering test problem with a higher number of dimensions: the design under uncertainty of a hypersonic inlet previously described by DiGregorio et al. [3].

A. Deterministic Problem

The Rosenbrock function is a well-known two-dimensional mathematical problem defined by [16] as:

$$f(x_1, x_2) = (1 - x_1)^2 + 100\left(x_2 - x_1^2\right)^2 \tag{19}$$

The function has a global minimum at $(1, 1)$, which lies within a long, curved valley as shown in Fig. 3. The gradient is small along the floor of the valley but large pointing up the walls. The disparate scales of the gradient makes it difficult for optimizers to precisely locate the minimum.

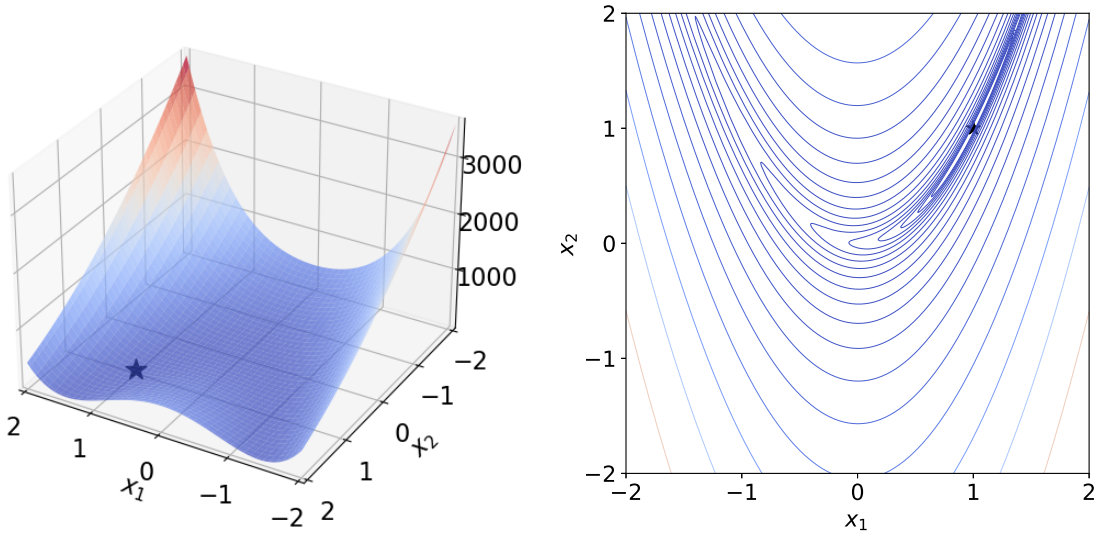
Several optimization algorithms are applied to the Rosenbrock problem, including conjugate gradient, L-BFGS-B, L-BFGS-B with multistart, classic EGO, and EGO-GF. Conjugate gradient is a first-order method and L-BFGS-B is a quasi-Newton method in which the Hessian matrix is approximated and used to speed up convergence [15]. The Rosenbrock function does not strictly require a multistart scheme. However, if a global optimum is desired and no a priori knowledge of the function topology is available, then multistart schemes are generally required. Thus, a gradient-based optimizer with multistart is a reasonable comparison for global optimization algorithms such as EGO and EGO-GF. Each algorithm was run for 100 trials. For conjugate gradient and L-BFGS-B, LHS was used to choose the trial initialization points. For each of the 100 L-BFGS-B multistart trials, 10 different starting points throughout the domain were used, chosen by LHS. Classic EGO and EGO-GF were both initialized using a ten-point Latin hypercube over the domain $x_{1,2} \in [-2, 2]$.

The performance of all algorithms is summarized in Table 2. Metrics for comparison include the average number of function calls across the trials, N_{calls}, the standard deviation of the number of function calls, σ_N, the average error, and the standard deviation of the error, σ_{Error}. The error is calculated as the Euclidean distance between the algorithm-returned result and the known global minimum.

Table 2 Results from the Rosenbrock function

Algorithm	N_{calls}	σ_N	Error	σ_{Error}
Conjugate Gradient	162	62	1.16×10^{-5}	1.32×10^{-5}
L-BFGS-B	99	29	6.35×10^{-6}	1.94×10^{-6}
L-BFGS-B with multistart	1031	107	4.06×10^{-6}	2.14×10^{-6}
EGO	49	20	4.32×10^{-2}	6.71×10^{-2}
EGO-GF	87	49	7.67×10^{-6}	4.53×10^{-6}

Taking the quasi-Newton L-BFGS-B method as a baseline, results indicate that the first-order conjugate gradient method typically takes more function calls, an average of 162 compared to 99, and has approximately 1.83 times as much error. When L-BFGS-B with multistart is considered, the number of function calls greatly increases, as expected, and the average error drops by a small amount. The benefit of such a multistart method is greater assurance that a global



(a) (b)

Fig. 3 The Rosenbrock function, with the global minimum denoted by a star, including (a) three-dimensional view and (b) two-dimensional contour.

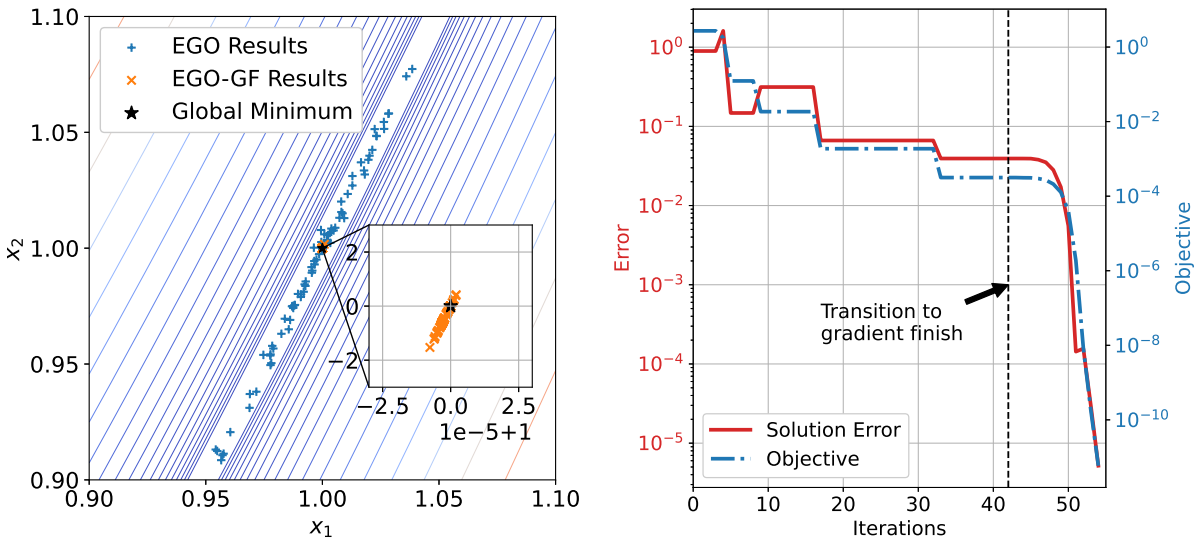
optimum has been obtained when function topology is a priori unknown. When classic EGO is compared to the pure gradient-based optimizers, a global optimum is returned in under half the average number of function calls of L-BFGS-B without multistart; however, the average error is four orders of magnitude higher than the gradient-based optimizers. This loss in accuracy may not be significant in all applications, in which case classic EGO remains an excellent choice of algorithm. Finally, EGO-GF is able to obtain a global optimum using approximately 12% fewer function calls than L-BFGS-B, with approximately 21% more error. Therefore, if a global optimum is desired, EGO-GF is significantly more efficient than multi-start optimization schemes and appears competitive with a single gradient-descent trial for this unimodal optimization problem. Figure 4(a) shows a visualization of the improvement in results obtained from EGO-GF compared to classic EGO. Note that for some trials of classic EGO, the error approaches 5% of the width of the initial LHS domain. In comparison, an inset in the figure is required to discern the error in the results returned from EGO-GF.

A typical convergence history of EGO-GF may be seen in Fig. 4(b). While EGO-GF is in EGO phase, the error converges slowly because it does not seek to immediately optimize the function value but rather the expected improvement, balancing exploration and exploitation. Once the EGO phase converges and the gradient finish takes over, the error begins to drop quickly.

B. Design Under Uncertainty Problem

Although the results of the previous subsection demonstrate the utility of EGO-GF for deterministic problems, accounting for the vast majority of optimization problems, the EGO-GF method was developed specifically for application to design under uncertainty problems. To assess algorithm performance on this type of problem, the hypersonic inlet design under uncertainty problem of DiGregorio et al. [3] is repeated here. For the sake of computational efficiency, only the low-fidelity inviscid analysis is performed.

The multiobjective problem, Eq. (20), consists of designing a 2D hypersonic inlet composed of m_d ramps such that the freestream flow at Mach number M_∞ is compressed down to a target value of M_{target} at the inlet throat while maximizing the total pressure recovery at the throat, $\Pi_{throat} = \frac{p_{t,throat}}{p_{t,\infty}}$. An image of the geometry being optimized is shown in Fig. 5. The design variables consist of five successive ramp angles, $\boldsymbol{\theta}$, the first three external to the inlet and the last two internal to the inlet. The ramp angles are constrained such that the net turning of the inlet is zero degrees, Eq. (20b), and bounded such that all ramp angles are between 2° and 12°, Eq. (20c). Due to EGO being an algorithm for unconstrained optimization, the net turning constraint of Eq. (20b) is converted into a quadratic penalty function, Eq. (20f), and added onto the objective function. The uncertain parameters include freestream Mach number, M_∞, angle of attack, α, and flight dynamic pressure, q. All uncertain variables are notionally modeled with normal



(a) (b)

Fig. 4 Results for Rosenbrock function optimization, including (a) EGO and EGO-GF output from 100 trials and (b) EGO-GF convergence history for a single trial.

distributions with means and standard deviations given in Table 3. Polynomial chaos expansions are used to quantify the uncertainty in the Mach number and total pressure recovery at the throat, M_{throat} and Π_{throat}, respectively, resulting in μ_M, σ_M, μ_Π, and σ_Π. A set of deterministic weighting coefficients, W_M and W_Π, allows the designer to trade off between the amount of thermodynamic compression and the efficiency of compression, while a set of robust weighting coefficients, W_μ and W_σ, allows for trading off between desired mean performance and low variance of performance with respect to uncertain parameters. The mean performance quantities μ_M and μ_Π are normalized by dividing the difference between the mean throat quantity and the target value by the difference between the worst value and the target value, then squaring the result, shown in Eq. (20a) [17]. A normalization scheme ensures that both objective terms are scaled to be between zero and one. For the present work, the worst value of throat Mach number for an inlet designed to compress the flow is taken to be equal to the freestream Mach number, $M_{worst} = M_\infty$, corresponding to an inlet that achieves zero compression. The worst value for the throat total pressure recovery is taken to be the total pressure recovery that would be present after a normal shock at the freestream Mach number, $\Pi_{worst}(M_\infty = 6) \approx 0.03$. The worst value for the net turning penalty term, θ_{worst}, is equal to the maximum achievable value of the right-hand side of Eq. (20b) given the bounds of the design variables, Eq. (20c). For this problem, θ_{worst} is equal to 32.0°. The target values are $M_{target} = 3.0$ and $\Pi_{target} = 1.0$. The physics of the problem are governed by inviscid analytical oblique shock relations. Further details are available in the original paper [3].

$$\min J_{robust} = W_M \left[W_\mu \left(\frac{(\mu_M - M_{target})}{(M_{worst} - M_{target})} \right)^2 + W_\sigma \sigma_M \right] + W_\Pi \left[W_\mu \left(\frac{(\mu_\Pi - \Pi_{target})}{(\Pi_{worst} - \Pi_{target})} \right)^2 + W_\sigma \sigma_\Pi \right], \tag{20a}$$

$$\text{subject to} \sum_{k=1}^{m_{d,ext}} \theta - \sum_{k=m_{d,ext}+1}^{m_d} \theta = 0, \tag{20b}$$

$$2° \leq \theta_i \leq 12°, \tag{20c}$$

$$W_M + W_\Pi = 1. \tag{20d}$$

$$W_\mu + W_\sigma = 1. \tag{20e}$$

$$J_{penalty} = \frac{1}{\theta_{worst}^2} \left(\sum_{k=1}^{m_{d,ext}} \theta - \sum_{k=m_{d,ext}+1}^{m_d} \theta \right)^2 \tag{20f}$$

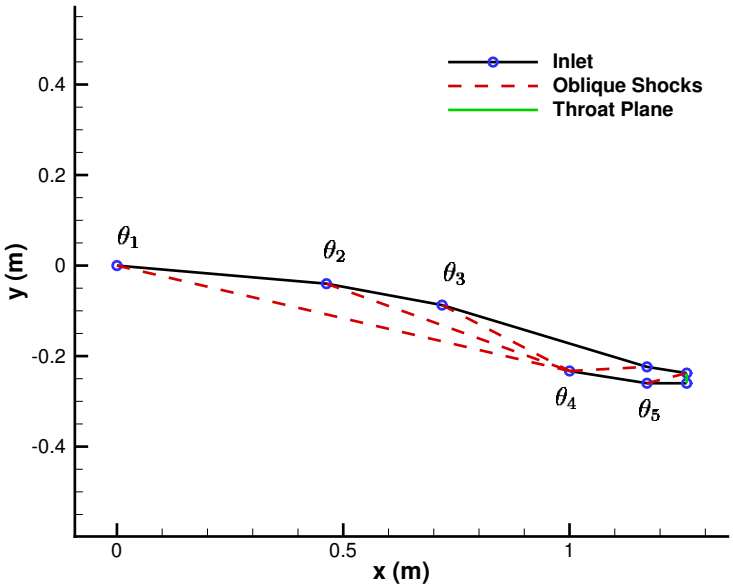


Fig. 5 Two-dimensional hypersonic inlet geometry for the sample DUU problem.

Table 3 Freestream condition uncertainties

Freestream Parameter	μ	σ
Mach number	6.0	0.1
Angle of attack (deg)	0.0	0.25
Dynamic pressure (psf)	1500.0	15.0

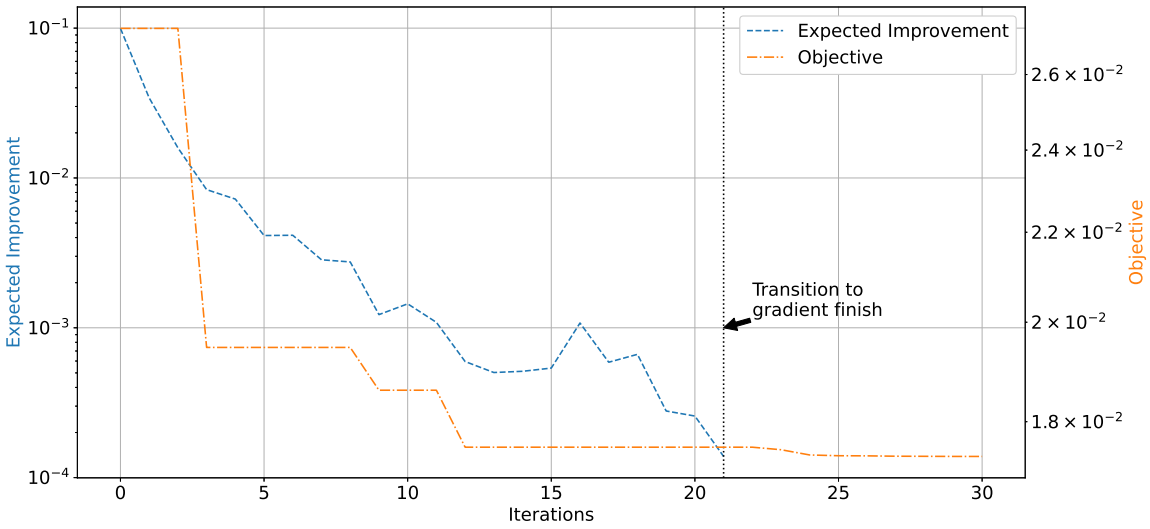


Fig. 6 Convergence history for EGO-GF applied to the inlet design under uncertainty problem.

The convergence history of a sample optimization using EGO-GF is shown in Fig. 6. The iteration number counts each sampling step after the initial DOE is complete. The objective function initially drops quickly as EGO-GF is exploiting its knowledge of the design space to locate a point with near-optimal performance. After exploiting knowledge of the design space, the objective function maintains a roughly constant value for several iterations. During this phase of the algorithm, EGO-GF is exploring the design space. As more points are sampled during exploration, the objective function value is not expected to change, but the expected improvement typically drops as EGO-GF continues to sample the design space and reduce the uncertainty in its own model. Expected improvement will not necessarily drop monotonically in all cases, which can be caused by significant disagreement between the true function evaluation and the expected value for a sample point, resulting in an update to the model. After several iterations of further exploration,

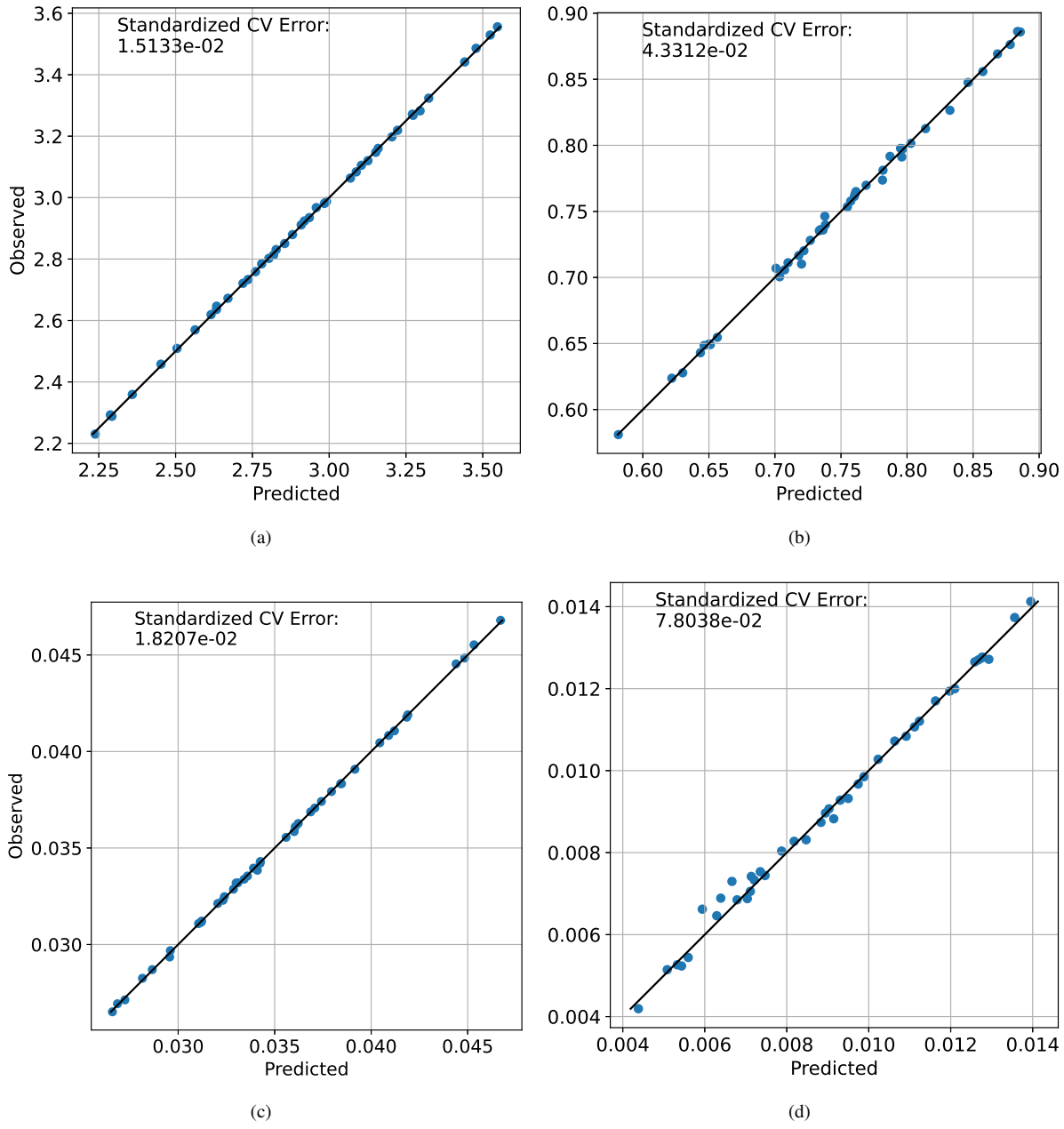


Fig. 7 Results of leave-one-out cross-validation applied to the surrogate models of (a) μ_M, (b) μ_Π, (c) σ_M, and (d) σ_Π after surrogate model training is complete.

the expected improvement drops to below 1% of the best recorded objective function value and ϵ_{cv} of the component Kriging models is within allowable tolerances, shown in Fig. 7, thus EGO-GF transitions from EGO mode to the gradient finish. The gradient-based optimizer, L-BFGS-B in this case, is initialized from the best point identified by EGO mode and the component Kriging models are used to evaluate the uncertainty statistics in the objective function, μ_M, σ_M, μ_Π, and σ_Π. The design variables are finite differenced and the component Kriging models are re-evaluated, allowing for efficient computation of the gradients of uncertainty with respect to the design variables: $\nabla\mu_M, \nabla\sigma_M, \nabla\mu_\Pi$, and $\nabla\sigma_\Pi$. If a quasi-Newton method, such as L-BFGS-B, is used, then the Hessian matrix is approximated and the second derivatives of uncertainty with respect to the design variables are used to further speed up convergence.

The search that EGO-GF conducts can be visualized with a pairwise plot, as shown in Fig. 8, which shows the relations between every input variable. All plots in a column share the same x axis variable, and all plots in a row share the same y axis variable. Points colored in dark blue are samples from the initial DOE, points colored in light blue are sampled during EGO mode, and the orange point is the point converged upon by EGO-GF after the gradient finish. For

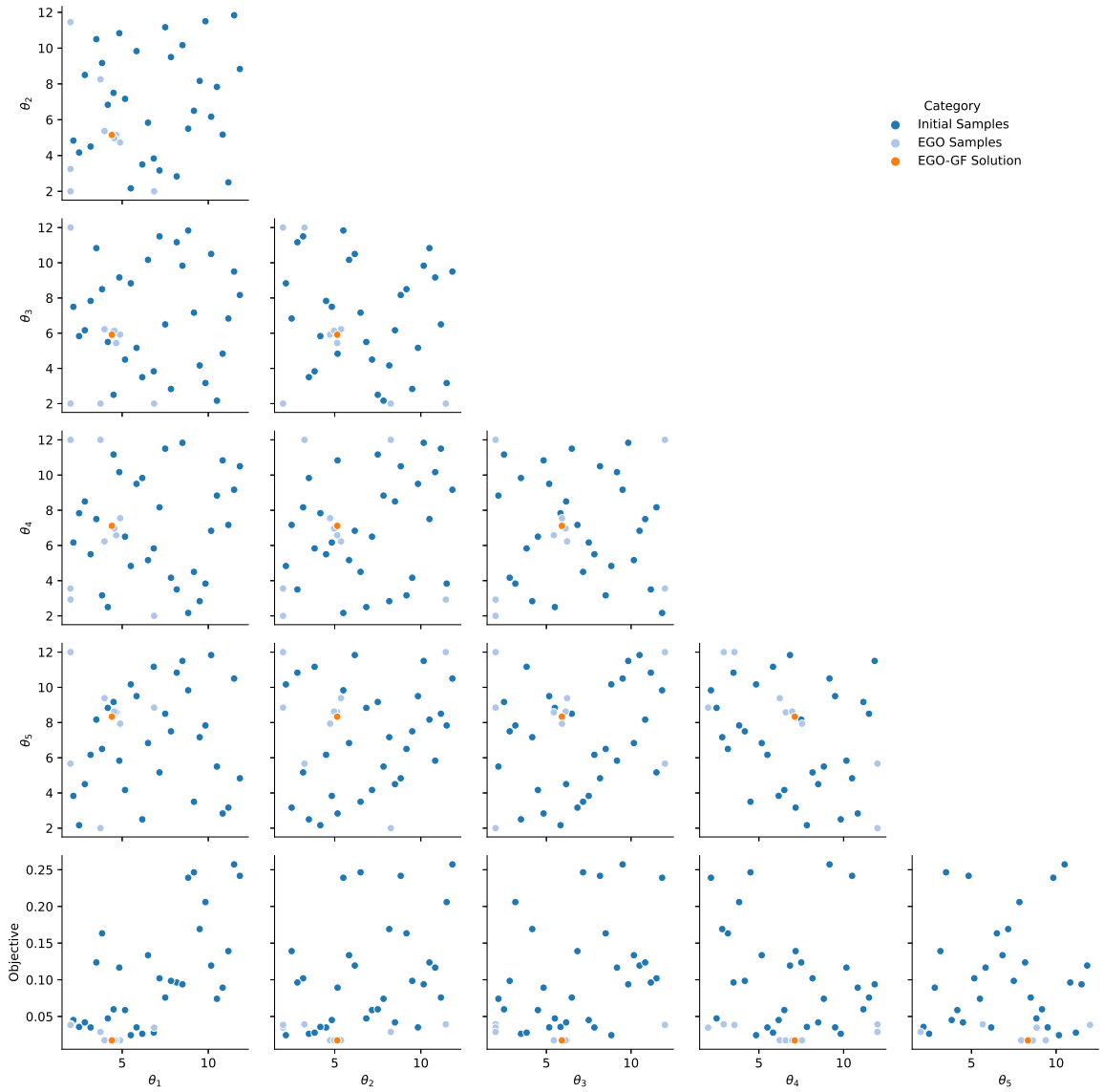
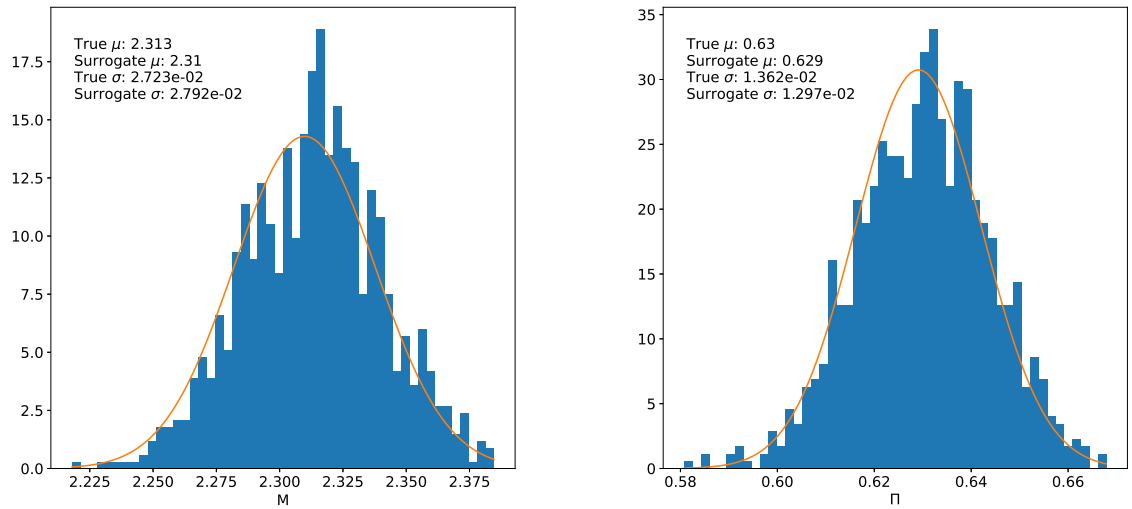


Fig. 8 Pairwise plots showing the relations between all sampled points for the design variables and the objective function.

this problem, the pairwise plots show that the full extent of the design space $2^\circ \leq \theta_i \leq 12^\circ$ is well-sampled by the initial DOE. After the initial DOE is complete, EGO tends to explore by sampling the corners of the design space and reducing model uncertainty. Model exploitation results in several sample points being evaluated near what is eventually identified as the optimum.

In order for the gradient finish of EGO-GF to be trustworthy, the surrogate models trained during design space exploration must be sufficiently accurate. This need is what drives the addition of the ϵ_{cv} criterion to the transition from EGO mode to gradient-finish mode. Example surrogate model verification results for this sample problem are shown in Fig. 9. After EGO mode completed, Monte Carlo sampling was performed to produce 1000 samples, and the true statistics of uncertainty were determined for a design point. A histogram of results is plotted alongside a normal distribution PDF generated from the statistics output from the surrogate models. Results demonstrate that the relative error for mean throat Mach number and mean throat total pressure recovery are under 1%, while the relative error for standard deviation is approximately 2.5% and 4.8% for throat Mach number and throat total pressure recovery, respectively.

In addition to solving for a unique optimum design for a given objective function, EGO-GF also has utility as a design space exploration tool. Since the component Kriging models for each of the unique terms in the objective function are saved separately, any of the weighting coefficients, W_M, W_Π, W_μ or W_σ, can be changed and the gradient finish can be rerun from the point returned by EGO mode at a trivial computational cost. This enables efficient construction of Pareto fronts, illustrating the design trade-offs that can be made for the problem being analyzed, including design trade-offs affecting uncertainty. Figure 10 shows the optimal design variables returned from EGO-GF compared to classic EGO for sweeps through both deterministic and robust weighting coefficients. Using EGO, it is difficult to discern any underlying trends within the design space due to the small amount of discrepancy that may exist between the result returned by EGO and the true optimum. The discrepancy, termed early exit error in previous work [3], is caused by EGO exiting when the expected improvement is less than or equal to 1% of the current best result. Depending on the topology and sensitivity of the optimization problem, a 1% improvement in the objective function may correspond to a significant difference in the design variables. Figures 10(a) and 10(c) illustrate this difficulty in discerning a design trend from EGO results when sweeping through deterministic and robust weighting coefficients, respectively. In contrast, the EGO-GF results in Figs. 10(b) and 10(d) illustrate that the addition of a gradient finish to EGO is able to give insight on design changes which will result in desired system performance. Some noise is still present in the EGO-GF results, likely due to imperfect surrogate models, but a design engineer can get clear insight on design trends. The Oswatitsch



(a) (b)

Fig. 9 Comparison of Monte Carlo results to surrogate model output for (a) M_{throat} and (b) Π_{throat}.

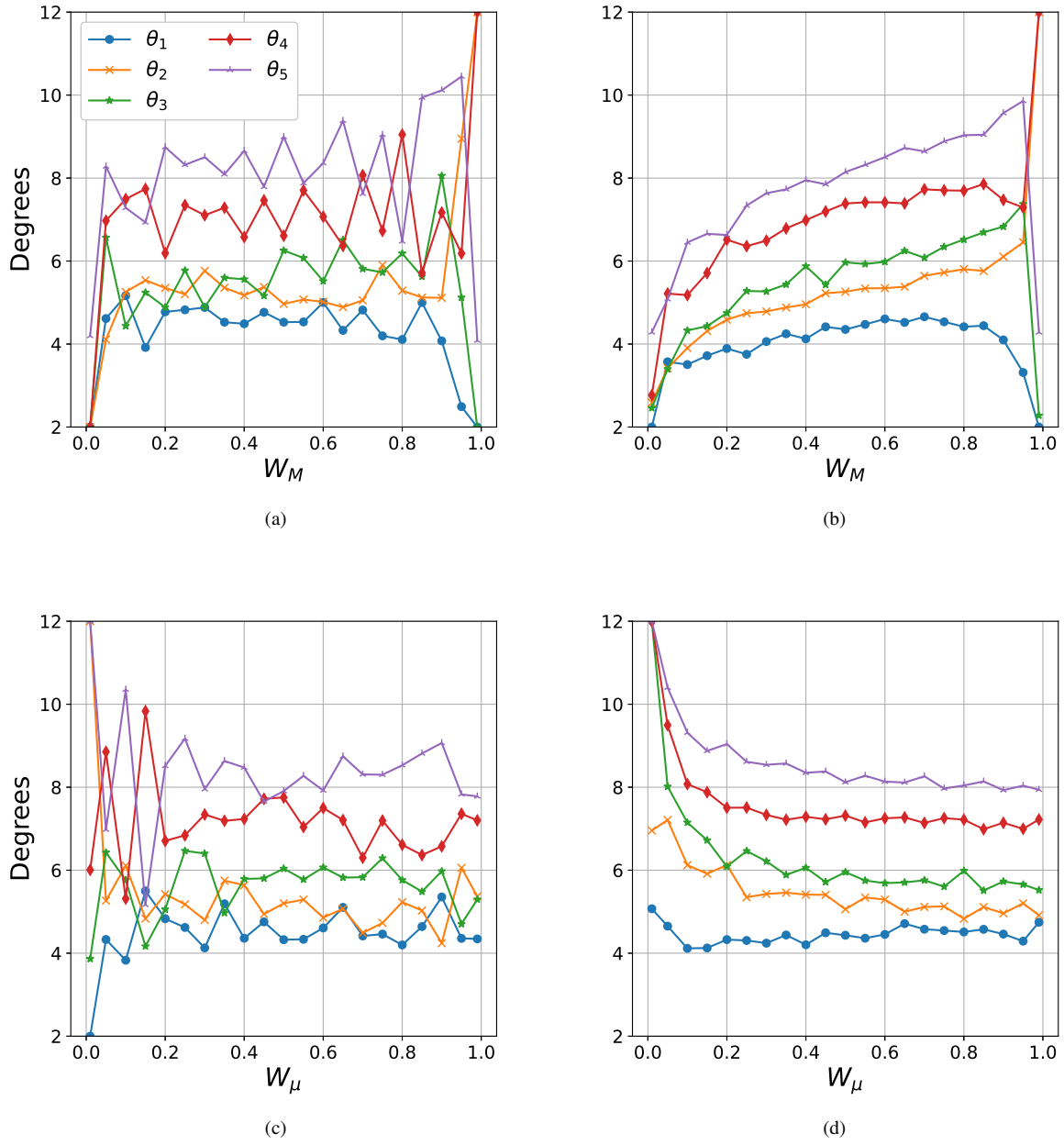


Fig. 10 Optimal design variables for a sweep through deterministic weighting coefficients with robust weighting coefficients held fixed at $W_\mu = W_\sigma = 0.5$ for (a) EGO and (b) EGO-GF, and a sweep through robust weighting coefficients with deterministic weighting coefficients held fixed at $W_M = W_\Pi = 0.5$ for (c) EGO, and (d) EGO-GF.

trend [4], in which each successive ramp angle in an inlet should be slightly larger than the previous ramp angle such that all shocks are of equal strength and result in equal losses, is clearly evident in the stacked results of EGO-GF for both deterministic and robust weighting coefficient sweeps. The classic EGO results, however, give little indication of such a trend existing. Without a priori knowledge of such a design guideline, it would be difficult to conclude such a trend exists using just classic EGO due to noise in the results.

A similar analysis is performed for the performance metrics, in this case μ_M and μ_Π, as shown in the comparisons between classic EGO and EGO-GF results in Fig. 11. For this problem, a trade-off exists due to the thermodynamics

of oblique shock wave compression. As the objective function is weighted more heavily towards achieving the target Mach number at the throat, i.e., optimizing the amount of compression, the total pressure at the throat will naturally decrease as ramp angles increase and shocks become stronger to provide the desired compression. Conversely, as the objective function is weighted more towards achieving the target total pressure recovery at the throat, i.e., optimizing the efficiency of compression, the ramp angles will decrease and shocks will become weaker. As with the design variables, EGO-GF produces much smoother trends in performance metrics than classic EGO. EGO-GF results clearly illustrate both the change in performance as the weighting coefficients are varied and the existence of the fundamental trade-off

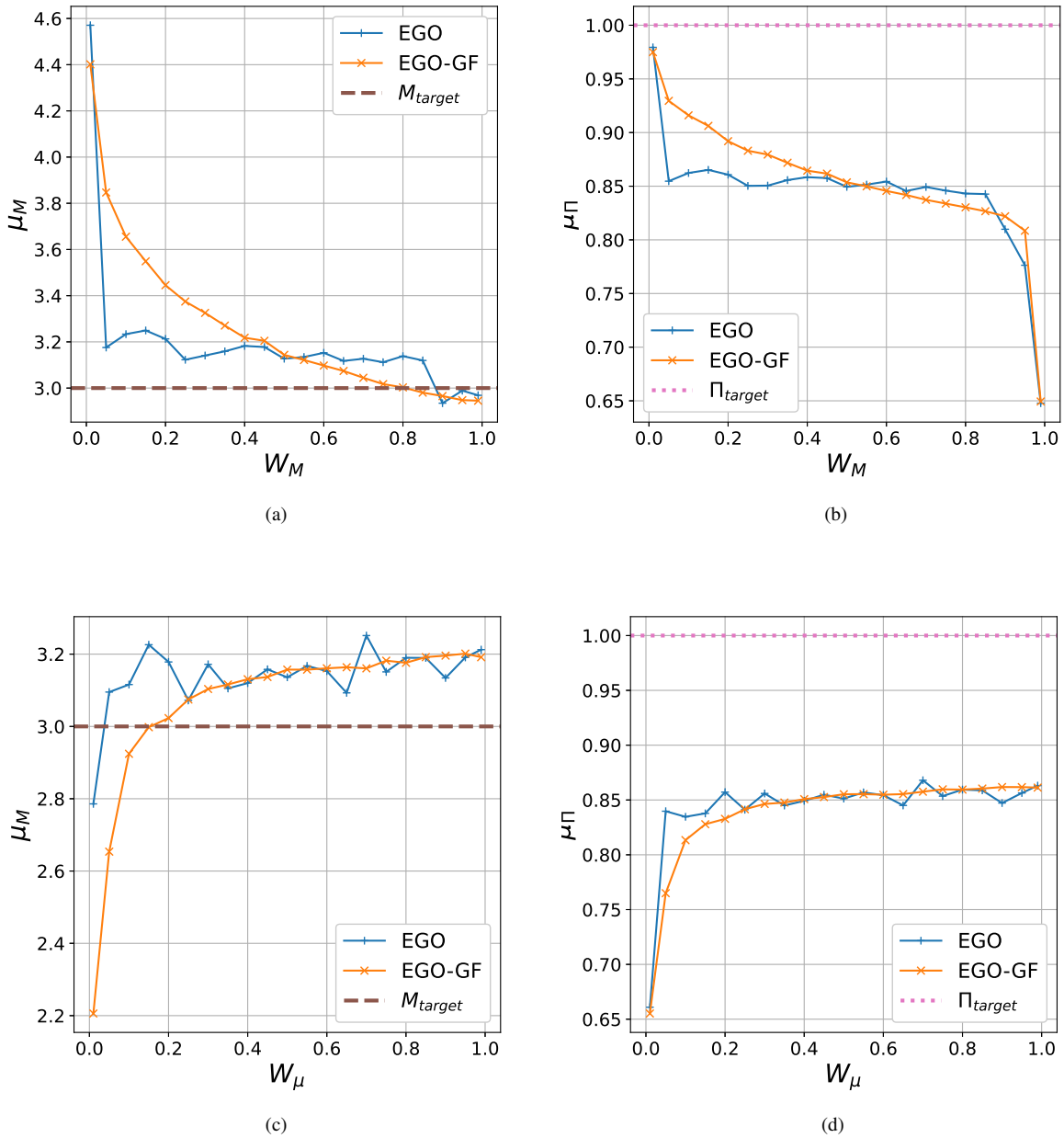


Fig. 11 Comparison of classic EGO to EGO-GF results for a sweep through deterministic weighting coefficients such that $W_\mu = W_\sigma = 0.5$ for (a) μ_M and (b) μ_Π, and a sweep through robust weighting coefficients with deterministic weighting coefficients held fixed at $W_M = W_\Pi = 0.5$ for (c) μ_M and (d) μ_Π.

that exists due to the thermodynamics of the problem. Classic EGO is once again able to obtain results fairly close to EGO-GF, but not accurately or consistently enough to give insight into fundamental trade-offs that exist due to physics of the problem. Although this may make EGO satisfactory for a single point optimization, it is lacking in its ability to give deeper insight to a design engineer.

Another important metric to assess for EGO-GF is its efficiency, judged by the number of true function calls needed for it to arrive at the optimum. Here, the distinction is made between calls to the true function and calls to the surrogate Kriging model functions. The true function may be a simple analytical closed-form function, a computationally expensive external code running on many CPUs over many days, or anything in between. The assumption behind EGO-GF is that the true function is highly expensive and needs to be run as few times as possible, whereas the Kriging models built within EGO-GF have negligible computational cost and can be called as much as desired once they are fully trained. Table 4 compares the average number of calls, N_{calls}, and the standard deviation of the number of calls, σ_N, made to the true function across 100 trials for several types of optimization algorithms for the hypersonic inlet DUU test problem. For each algorithm, the average objective function value, J_{robust}, and the difference between the average objective function value and the minimum average objective function value, ΔJ_{robust}, are also compared. Both classic EGO and EGO-GF required an average of 371 calls to the true function, a substantial improvement compared to the purely gradient-based algorithms, conjugate gradient and L-BFGS-B, which required on average 2950 and 1472 calls to the true function, respectively. Here, the additional function calls incurred by the pure gradient-based approaches, compared to EGO-GF, do not result in any improvements to the optimization result, J_{robust}, and the result is not guaranteed to be a global optimum. Note that EGO-GF does not require any extra calls to the true function compared to classic EGO, because the gradient finish is performed purely on Kriging surrogate models developed during the course of running EGO-GF. When a multistart analysis of 30 runs with L-BFGS-B was applied to the inlet DUU problem to ensure a global optimum, the true function was called an average of 44, 160 times, exceeding the average number of true function calls required by classic EGO and EGO-GF by two orders of magnitude. These results demonstrate that EGO-GF offers significant improvements in accuracy compared to EGO, while remaining computationally efficient enough to be well-suited to design space exploration.

Table 4 Number of function evaluations required for several optimization algorithms

Algorithm	$\mathbf{N_{calls}}$	$\sigma_\mathbf{N}$	$\mathbf{J_{robust}}$	$\mathbf{\Delta J_{robust}}$
Conjugate gradient	2950	63.4	1.7332×10^{-2}	1.1×10^{-5}
L-BFGS-B	1472	49.1	1.7331×10^{-2}	1.0×10^{-5}
L-BFGS-B with multistart	44,160	11,935	1.7321×10^{-2}	0
EGO	371	42.1	1.7432×10^{-2}	1.1×10^{-4}
EGO-GF	371	42.1	1.7327×10^{-2}	6.0×10^{-6}

IV. Conclusions

The optimization algorithm EGO is extended to include a gradient-based optimization finish, EGO-GF, with a particular focus on maintaining high efficiency for design under uncertainty applications. Results indicate that for deterministic problems, EGO-GF accuracy is on par with gradient-based algorithms while efficiency is in line with both classic EGO and a quasi-Newton gradient-based optimizer without multistart. For a sample design under uncertainty problem, accuracy is once again improved over EGO and comparable to gradient-based optimizers, while efficiency is one to two orders of magnitude improved over gradient-based optimizers. Due to EGO-GF algorithm formulation, weighting coefficients in the objective function may be changed and Pareto fronts of the DUU problem may be obtained at negligible additional cost after the first optimization is complete.

References

[1] Locatelli, M., and Schoen, F., "(Global) Optimization: Historical notes and recent developments," *EURO Journal on Computational Optimization*, Vol. 9, 2021, p. 100012. doi:https://doi.org/10.1016/j.ejco.2021.100012.

[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:https://doi.org/10.1023/A:1008306431147.

[3] DiGregorio, N. J., West, T. K., and Choi, S., "Robust Design Under Uncertainty of Hypersonic Inlets," *AIAA Paper 2022-3862*, June 2022. doi:10.2514/6.2022-3862.

[4] Oswatitsch, K., "Pressure Recovery for Missiles with Reaction Propulsion at High Supersonic Speeds (The Efficiency of Shock Diffusers)," Tech. Rep. NACA-TM-1140, 1947.

[5] Schöbi, R., Kersaudy, P., Sudret, B., and Wiart, J., "Combining Polynomial Chaos Expansions and Kriging," Research report, ETH Zurich, Switzerland ; Orange Labs research, 2014.

[6] Forrester, A., Sobester, A., and Keane, A., *Engineering Design via Surrogate Modelling: A Practical Guide*, John Wiley & Sons, 2008.

[7] Cameron, R., and Martin, W., "The Orthogonal Development of Nonlinear Functionals in Series of Fourier-Hermite Functionals," *Annals of Mathematics*, Vol. 48, 1947, pp. 385–392.

[8] Smith, R. C., *Uncertainty Quantification: Theory, Implementation, and Applications*, SIAM, 2013.

[9] Eldred, M. S., and Elman, H. C., "Design Under Uncertainty Employing Stochastic Expansion Methods," *International Journal for Uncertainty Quantification*, Vol. 1, No. 2, 2011, pp. 119–146. doi:https://doi.org/10.2514/6.2008-6001.

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

[11] Hosder, S., Walters, R. W., and Balch, M., "Point-Collocation Nonintrusive Polynomial Chaos Method for Stochastic Computational Fluid Dynamics," *AIAA Journal*, Vol. 48, No. 12, 2010, pp. 2721–2730. doi:https://doi.org/10.2514/1.39389.

[12] Ng, L. W.-T., and Eldred, M., "Multifidelity Uncertainty Quantification using Non-Intrusive Polynomial Chaos and Stochastic Collocation," *AIAA Paper 2012-1852*, April 2012. doi:10.2514/6.2012-1852.

[13] West IV, T. K., and Phillips, B. D., "Multifidelity Uncertainty Quantification of a Commercial Supersonic Transport," *Journal of Aircraft*, Vol. 57, No. 3, 2020, pp. 491–500. doi:https://doi.org/10.2514/1.C035496.

[14] Eldred, M., "Recent Advances in Non-Intrusive Polynomial Chaos and Stochastic Collocation Methods for Uncertainty Analysis and Design," *AIAA Paper 2009-2274*, May 2009. doi:https://doi.org/10.2514/6.2009-2274.

[15] Arora, J., *Introduction to Optimum Design*, Elsevier, 2017.

[16] Rosenbrock, H. H., "An Automatic Method for Finding the Greatest or Least Value of a Function," *The Computer Journal*, Vol. 3, No. 3, 1960, pp. 175–184. doi:10.1093/comjnl/3.3.175.

[17] Vanderplaats, G., *Multidiscipline Design Optimization*, Vanderplaats Research & Development, Incorporated, 2007.