A Kriging Method for the Solution of Nonlinear Programs with Black-Box Functions

Eddie Davis and Marianthi Ierapetritou

Dept. of Chemical and Biochemical Engineering, Rutgers—The State University of New Jersey, Piscataway, NJ 08854

In this article, a new methodology is developed for the optimization of black-box systems lacking a closed-form mathematical description. To properly balance the computational cost of building the model against the probability of convergence to global optimum, a hybrid methodology is proposed. A kriging approach is first applied to provide information about the global behavior of the system considered, whereas a response surface method is considered close to the optimum to refine the set of candidate local optima and find the global optimum. The kriging predictor is a global model employing normally distributed basis functions, so both an expected sampling value and its variance are obtained for each test point. The presented work extends the capabilities of existing response surface techniques to address the refinement of optima located in regions described by convex asymmetrical feasible regions containing arbitrary linear and nonlinear constraints. The performance of the proposed algorithm is compared to previously developed stand-alone response surface techniques and its effectiveness is evaluated in terms of the number of function calls required, number of times the global optimum is found, and computational time. © 2007 American Institute of Chemical Engineers AIChE J, 53: 2001–2012, 2007
Keywords: optimization, mathematical modeling, black-box, kriging, response surface

Introduction

Process design problems lacking closed-form model equations are very difficult to solve when noisy input-output data are the only information available. Sampling information can be noisy due to unknown/uncontrollable environmental conditions affecting process operation or as a result of the level of detail employed in a computational model. This problem contains three main challenges. First, what kind of model should be used to approximate the black-box function? Local models provide only a limited understanding of system behavior, but global models are more expensive to build. Second, how can the model capture the true behavior, instead of the noise? If derivative-based techniques are applied to models that fit the noise, this can lead to the discovery of ar-

tificial optima due to iterates becoming trapped by the noise. Third, given that more accurate models are obtained using symmetrically arranged sampling points, how can reliable models be constructed for problems described by convex asymmetrical feasible regions?

A new algorithm is presented in this paper for the solution of nonlinear programs (NLP) containing black-box functions and noisy variables that extends the capabilities of current methods to handle convex feasible regions defined by arbitrary linear and nonlinear constraints. In the proposed method, kriging is first used to construct a global model in order to find the set of regions containing potential optima. Response surface techniques are then applied to local models in order to refine the set of local solutions. The global model building expense is offset by an increased probability of finding the global optimum in contrast to the application of purely local techniques.

A variety of techniques have been applied towards the solution of NLP containing black-box functions and noisy vari-

Correspondence concerning this article should be addressed to M. Ierapetritou at marianth@soemail.rutgers.edu.

ables. The first set of algorithms discussed use sample points and do not build surrogate models, so model building costs are avoided. Sampling-based global zero-order methods such as Nelder-Mead (Barton and Ivey, 1996[1]), Divided Rectangles (Jones et al., 1993[2]), multilevel coordinate search (Huyer and Neumaier, 1999[3]), and differential evolution (Storn and Price, 1997[4]) are algorithms that belong to this class and can be used for the optimization of black box models. However they exhibit the following limitations. First, convergence may be asymptotic, and second, premature termination can occur since no mechanism is in place to overcome the discovery of artificial optima. In order to overcome these limitations, gradient-based methods can be used which apply finite-differencing expressions that depend on the system noise (Brekelmans et al., 2005[5]; Choi and Kelley, 2000[6]; Gilmore and Kelley, 1995[7]). Gradient inaccuracy is minimized by using finite-difference intervals large enough to "step over" the noise. The interval length adaptively changes over the sequence of iterations since the noise may decay as the optimum is approached. The main drawback of using this class of methods is that even though the optimum can be found, an understanding of system behavior is limited, thereby motivating the application of a different set of techniques which perform optimization after a surrogate model has first been constructed.

Response surface techniques find an optimum by applying steepest descent towards inexpensive local models (Myers and Montgomery, 2002[8]). Box and Wilson (1951[9]) and Hoerl (1959[10]) authored seminal papers in this field proposing the use of statistical design of experiments, another developing field at the time, to generate quadratic functionals in order to more rigorously quantify response behavior. In this approach, a local input-output mapping is represented using a low-order polynomial fitted to output data obtained from symmetrically arranged sampling points. After minimization of the local model, a new model is built which is centered on the current iterate, and is then again minimized. The procedure continues until a prespecified stopping criterion such as minimal decrease in the objective function has been obtained. Because response surfaces are usually inexpensive in terms of the computational cost required to build the models, sampling expense is the primary source of the optimization cost. To reduce the computational complexity due to sampling in response surface methods, two algorithms have previously been developed by our group (Davis and Ierapetritou, 2006[11]). In the first method, direct search is employed in the early stages followed by minimization of local models once the vicinity of the optimum has been identified. Sampling points are generated according to a stencil that is a coarse representation of the finer template used to generate the set of points needed to build the response surface at the later stages. This method is based on the idea that the value of the search direction information obtained by response surface minimization is not justified by the additional sampling during the early stages of optimization. The second technique borrows ideas from sequential quadratic programming (SQP) by minimizing the local response surface across the entire feasible region in order to obtain possible global movement towards the optimum. Global movement towards the optimum occurs if sampling at the predicted minimum yields a superior value to the current best candidate solution of the

set of local collocation points. This method performs well when optima reside near feasible region boundaries and the local model generated from a nominal collocation set faraway provides a suitable approximation of the global geometry. The cost of an additional function call per iteration is justified when compared against the equivalent set of sequential response surface minimizations that result in the same search direction being identified over local subregions. Optimization using response surfaces has been applied to many fields including automotive impact engineering and design (Stander and Craig, 2002[12]), structural optimization (Roux et al., 1998[13]), and food engineering (Chen et al., 2005[14]). Tang et al. (2005[15]) employs response surface ideas within a Gauss-Newton algorithmic framework to solve the kinetic inversion problem. Zhang et al. (2006[16]) extended this methodology for parameter estimation of coupled reaction-transport systems under uncertainty. Although local optimization techniques are applied more often to response surfaces, global methods have also been studied (Jones, 2001[17]; Jones et al., 1998[18]).

In contrast to response surfaces which are local models, a kriging predictor is a global model employing normally distributed basis functions, so both an expected sampling value and its variance are obtained for each test point (Cressie, 1993[23]; Goovaaerts, 1997[19]; Isaaks and Srivistava, 1989[20]). Kriging was first developed as an inverse distance weighting method to describe the spatial distribution of mineral deposits. The global model obtained served as a tool to determine additional locations possessing the same grade characteristics, thereby enabling the generation of improved monthly forecast estimates for different mine sections. The seminal article by Krige (1951[21]) was reviewed in context of other classical statistical methods used for geostatistics (Matheron, 1963[22]). Kriging has not only been used to generate models based on field data, but also when input-output information is obtained via computer experiments due to the increasing use of simulation for the study of complex processes (Sacks et al., 1989[24]; Santner et al., 2003[25]). It is to be noted that in the proposed algorithm, the function of kriging is to determine improved locations for local search thereby enabling the response surface techniques to be applied to "warm-start" iterates. Because of the black-box functions present in the problem formulation, it is impossible to guarantee global optimality. The interpolatory and confidence limits of kriging may be similar to those attained using αBB (Maranas and Floudas, 1995[26]) but αBB assumes that the functionality is known, which is not the case in the general class of problems addressed in this article.

Kriging is an interpolation technique whereby a prediction $\tilde{z}_2$ at test point x_k is made according to a weighted sum of the observed function values at nearby sampling points. The kriging predictor is modeled after a normally distributed random function, so a prediction variance is also obtained at the test point. As a result, the sampling value is expected to fall within the interval specified by the prediction and corresponding variance. By discretizing the feasible region, both a prediction and variance mapping can be obtained. The variance mapping describes prediction uncertainty, which will be high in regions with a low number of sampling points. Additional sampling can then be done within these regions to reduce the uncertainty. The main drawback for kriging is

that model building costs are high compared to the construction of response surfaces. The existing literature usually compares kriging and RSM methods but has not combined them together for the purpose of global optimization as proposed in this work. In the next section, the details and algorithmic features of the proposed method are presented. The proposed method is applied to several numerical examples and concluding remarks are then given.

Solution Approach

In the next section, the mathematical formulation of the problem is given followed by the basic idea of the proposed kriging-RSM algorithm. The details of the kriging method are presented followed by ideas for refinement of the optimum using response surfaces when the problem is described by arbitrary linear and nonlinear constraints.

Problem Formulation

The problem addressed in this work has the following form:

$$\min F(x, z_1, z_2)$$
$$\text{s.t. } g(x, z_2) \leq 0$$
$$h(x, z_1) = 0$$
$$z_2(x) = \Gamma(x) + \varepsilon(x)$$
$$\varepsilon(x) \in N(x|\mu, \sigma^2)$$
$$N(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) \quad (1)$$

In this formulation, x represents continuous variables that are process inputs. This set of variables is distinct from input information z_2, which is obtained from sampling at upstream units. Deterministic output variables z_1 describe outputs whose modeling equations $h(x, z_1)$ are known. Stochastic output variables z_2 represent the black-box part of the model and are simulated by deterministic functions $\Gamma(x)$ perturbed with noise $\varepsilon(x)$. The noise is described by a normally distributed error whose properties can change depending upon the spatial location of x. The noise function is stochastic in that replicate experiments performed under the same conditions lead to different values for $\varepsilon(x)$ and in turn $z_2(x)$. Design constraints such as operating specifications are given by $g(x, z_2)$. In the proposed approach, a sequential kriging-RSM algorithm is employed for the optimization of systems containing black-box functions and noisy variables. The basic idea is to first use kriging to obtain a global picture of system behavior by generating predictions at test points, and then to apply RSM to a set of regions containing potential local optima in order to find the global optimum. Although the kriging predictor is iteratively improved as additional sampling is conducted in regions where prediction ability is poor, the CPU cost associated with generating many predictions over the course of the modeling can become high, especially if a set of discretized test points obtained using high-resolution grids are employed to obtain more accurate optima. Since response surfaces can be inexpensively generated, a better strategy for

obtaining potential optima is to generate a kriging surface over a lower-resolution grid and then use sequential response surfaces to refine the best set of candidate vectors. Since it is possible that local optima lie along the boundaries of the feasible region, application of current response surface techniques, which are suited for the solution of box-constrained problems, would fail to improve upon the current values of the possible optima. The new method addresses these issues by applying (1) adaptive experimental designs to retain feasibility, and (2) projection of the n-dimensional response surface onto the feasibility space limited by the problem constraints. Although global optimality is not guaranteed, application of the proposed kriging-RSM algorithm can result in the discovery of both interior and boundary solutions for constrained NLP described by (1) black box functions, (2) noisy variables, and (3) arbitrary convex feasible regions. If any z_2 variables are inputs to downstream process units, the feasible region is explicitly defined after sampling at the respective upstream units. In this case, the feasible region will be affected by noise. This complication is not present for the examples presented in the Examples section but will be addressed in a future work. At the present time, the kriging-RSM algorithm has only been applied towards systems having deterministic feasible regions and only a single black-box unit. The next section provides details for obtaining the kriging predictor.

Kriging Methodology

The kriging predictor of the stochastic output $\tilde{z}_2(x_k)$ is a linear combination of N weighted function values at nearby sampling points as given below:

$$\tilde{z}_2(x_k) = \sum_{i=1}^{N_{\text{Cluster}}} f(x_i)\lambda(x_i) \quad (2)$$

where $\tilde{z}_2(x_k)$ represents the predictor at test point x_k. The main problem in building the kriging predictor lies in determining the appropriate set of weights assigned to each set of N_{Cluster} points near x_k subject to the constraint that the expected error between the prediction and true value is zero, and also that the variance of the error distribution for all test points is minimized. The mechanism of obtaining the weights using kriging follows the same principles as the inverse distance weighting methods in that the weights are generally a decreasing function of distance from the test point x_k. Using kriging, predictions are indirectly obtained by estimating the function difference between x_i and x_k as being similar to corresponding differences for known sampling data. Function values for sampling points x_i, which are close to x_k, will be more influential in determining the prediction $z_2(x_k)$. The converse is true as the distance h between x_i and x_k increases. Consider a sampling point x_i and test point x_k separated by a distance h_1. The corresponding kriging weight will be higher if there are many sampling-point x_i–x_j pairs, also separated by this same distance h_1, whose corresponding $f(x_i)$–$f(x_j)$ differences are approximately the same. The weight will be higher because there is a stronger correlation between what the expected function value difference should be for this distance h. In other words, given n sampling-point pairs, each separated by a distance h_1, if the standard deviation of the function value differences between these n sampling pairs is
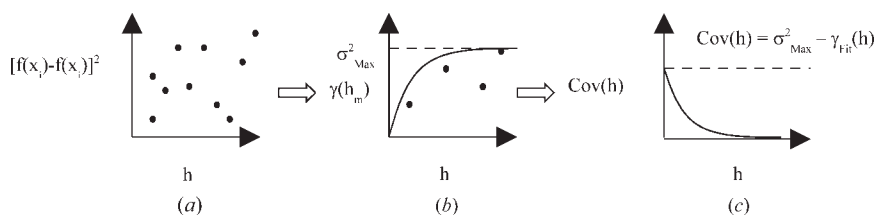
Figure 1. Data smoothing applied to squared function differences (a) to obtain a semivariogram fit (b) and covariance function fit (c).

low, the difference between the function values for an nth + 1 sampling-point pair, or a sampling-point/test-point pair, also separated by h_1, is expected to be close to the average difference obtained for the first n pairs. If the number of sampling pairs n is high, there will be an even higher confidence that the estimated function value difference $f(x_i) - f(x_k)$ should be close to the average difference for the n sampling pairs. The number of n sampling pairs separated by a distance h is usually higher as the value of h decreases, and consequently there will be more information available for estimating the corresponding function value difference if a sampling point is close to the test point. Now consider the case where the sampling point x_i and test point x_k are separated by a distance h_2. After obtaining the set of S sampling pairs also separated by h_2, if there is a high spread observed in the set of corresponding $f(x_i) - f(x_j)$ values, there will be less confidence in estimating the expected difference between the function value at sampling point x_i and test point x_k. The true difference might be closer to the lowest or highest difference instead of the average difference in this case. As a result, the value of the corresponding kriging weight will be lower in this case to emphasize the fact that there is less confidence that the estimate of the expected function value difference between x_i and x_k will be close to the true difference. This idea is quantitatively represented in the form of a fitted covariance function expressing correlation strength between function differences for sampling-point pairs a given distance apart. The first step in obtaining the covariance function is to obtain $(N)(N - 1)/2$ squared function differences F_{ij} for N sampling points x_i, as shown in Eq. 3:

$$F_{ij} = \left[f(x_i) - f(x_j) \right]^2 \qquad i,j = 1,...,N, \qquad i \neq j \quad (3)$$

The covariance function is fitted from a plot of F_{ij} as a function of $x_i - x_j$ distance, also known as lag distance h. Because of the plot complexity as shown in Figure 1a, a resulting fit to one of the established covariance models in the literature is not usually immediately apparent.

To improve the fit, data averaging is performed by clustering F_{ij} within P subintervals each having length h_m, where N_m $x_i - x_j$ pairs fall within the interval $h_m - h_{\text{Tol}} \leq h_m \leq h_m + h_{\text{Tol}}$. The new scatterpoints known as semivariances are then fit to an established semivariance model from the literature as shown in Figure 1b, and may be a linear combination of five major types: spherical, Gaussian, exponential, power, or linear. The corresponding covariance function is then obtained by reflecting the semivariance function between the sill and the x-axis as shown in Figure 1c. The expression for

obtaining the semivariances is given below:

$$\gamma(h_m) = \frac{1}{2N(h_m)} \sum_{i=1}^{N(h_m)} \left[f(x_i) - f(x_j) \right] \qquad m = 1,...,P \quad (4)$$

After obtaining the covariance function, the kriging weights are obtained by substituting in the respective distances. Unlike some inverse distance techniques, the weights also will account for clustering effects by reducing the weight values for sampling points that are the same distance from x_k but close to one another. The weights at test point x_k are then obtained according to Eq. 5, where $\lambda(x_k)$ is an $N_{\text{Cluster}} \times 1$ vector of weights summing to unity. The problem of finding the weights subject to the constraint that they sum to unity is recast in its Lagrangian equivalent, and solution of the resulting set of Lagrangian equations then simultaneously yields both a set of weights and the Lagrange parameter $\mu(x_k)$.

$$\begin{bmatrix} \lambda(x_k) \\ \mu(x_k) \end{bmatrix} = \begin{bmatrix} \text{Cov}(d(x_i, x_j)) & 1 \\ 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \text{Cov}(d(x_i, x_k)) \\ 1 \end{bmatrix}$$
$$i,j = 1,...,N_{\text{Cluster}}, \; i \neq j \quad (5)$$

Once the weights are known for the set of N_{Cluster} nearby sampling points x_i, the kriging prediction $\tilde{z}_2(x_k)$ and its expected variance $\tilde{\sigma}^2(x_k)$ are then obtained as given by Eqs. 5 and 6, respectively:

$$\tilde{\sigma}^2(x_k) = \sigma_{\text{max}}^2 - \sum_{i=1}^{N_{\text{Cluster}}} \lambda(x_i)\text{Cov}[d(x_i, x_k)] \quad (6)$$

After obtaining kriging predictions for all test points, both a prediction and variance mapping can be built. Using the variance mapping, confidence regions around each prediction are then obtained. If there is a tight confidence interval around the kriging prediction at x_k, it means that there is a greater belief that subsequent sampling values will fall within the corresponding interval. The converse is true for the opposite case, since there is less confidence that the kriging prediction accurately represents the mean sampling value for test points x_k located far away from previously sampled points. After obtaining additional sampling data based on predictions satisfying criteria such as maximum variance, minimum value, or high difference in the estimate for consecutive iterations, the kriging model is updated. Over the course of the modeling, the average value of the kriging predictions made over the set of test points can be obtained as $\mu_{m,Pred}$ for iteration m, which converges to a final value. The

average of the prediction variance also drops to a fraction of its corresponding initial value. Once convergence is achieved, a set of regions containing potential local optima can be identified. After identifying all promising regions, it is more efficient to refine the set of best predicted values using local response surface models instead of kriging predictors over finer local grids. The source of the CPU expense required in building and minimizing response surfaces lies in the solution of a single linear system to obtain the fitted response surface coefficients, whereas the construction of refined predictors over an $N_{Fine} \times N_{Fine}$ local grid requires solution of N_{Fine}^2 linear equations as given in Eq. 5, since a unique set of weights is associated with each new local test point.

The kriging algorithm proceeds as follows for obtaining a prediction at x_k. First, the feasible region is described based on $g(x, z_2)$ and $h(x, z_1)$, where $g(x, z_2)$ is assumed to be explicitly defined after having obtained sampling data z_2 from upstream units. A sampling set Ω is generated whose initial size ranges between ten and twenty dispersed data locations for problem dimensions of up to eleven variables. As long as the starting size of Ω is not too small, the number of iterations required to achieve convergence in $\mu_{m,Pred}$ will be relatively insensitive to this number. The initial number of sampling points in Ω is small, placing emphasis on further sampling as needed during the iterative stages of predictor refinement. The location x_k is specified and $N_{Cluster}$ nearest-neighbor sampling points are chosen from Ω based on the L^2-norm relative to x_k. $N_{Cluster}$ usually varies between five and ten regardless of problem dimension although the estimate is potentially skewed if sampling information is sparse. Semivariances are then generated using all sampling data within Ω. The best semivariogram model is obtained using least squares, and the complementary covariance function is generated. The symmetrical matrix $h_{Cluster}$ is built describing sampling-pair distances among the set of $N_{Cluster}$ points. Similarly, the vector h_0 is constructed from sampling-point distances relative to x_k. The covariance matrices C and D are built using information from $h_{Cluster}$ and h_0, respectively. The kriging weights λ are then obtained from solving the linear system of equations as presented in (5) and the prediction
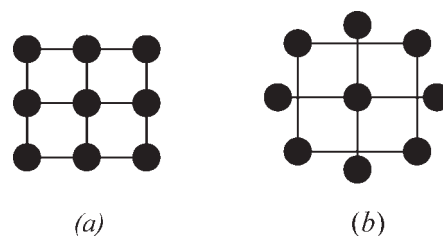

Figure 3. Sample experimental designs in 2-D: (a) Factorial design, (b) Central composite design.

$\tilde{z}_2(x_k)$ and its variance $\tilde{\sigma}^2(x_k)$ are determined. A flowchart of the kriging algorithm is presented in Figure 2.

Refinement of Local Optima Using RSM

The main idea of this step is to utilize RSM to approximate local input–output relationships using polynomial functions fitted to a set of stochastic data, which can then be optimised via conventional techniques. Noninterpolating polynomials are used to avoid fitting the noise, which also cause the search direction to become less sensitive to the process uncertainty at each sample point. A standard RSM algorithm calls for the construction of local models built from a set of collocation points centered on an iterate using an experimental design. Figure 3 shows examples of two-dimensional factorial and central composite designs (CCD). In a factorial design, a set or subset of the input variables is selected to be the factors, and particular levels are chosen for each factor. Experiments are then conducted at all possible factor-level combinations. The total number of sampling points is a^b, where a represents the number of factor levels (usually two or three) and b denotes the number of factors. For a system involving three variables, each with two levels, a total of 2^3 sampling points are needed. Although the same number of sampling points are used for the factorial and central composite design in the 2D case, the number of sampling points required is lower for the CCD in contrast to the factorial design when the system dimensionality increases. The total number of points required for an n-dimensional system using a CCD is $1 + 2n + 2^n$. A factorial design utilizing three variables, each with three levels, results in 27 design points, whereas only 15 are required for the central composite design (8 corner points, 6 axial points, 1 center point). If the number of variables increases to six, the factorial design requires 729 sampling points, whereas the CCD requires only 77. Even though replication at the center point is usually built into the design, the total number of sample points is still lower than that for the factorial design. For a more complete discussion, the reader is referred to Box et al. (2005[27]), and Myers and Montgomery (2002[8]).

At the start of the algorithm, a nominal point serves as the starting iterate; after a response surface has been minimized, the optimum of the local model serves as the next iterate. The procedure of building and minimizing local surfaces continues until a stopping criterion is reached, such as sufficient decrease in the objective or when the Euclidean norm between current and previous iterates falls below a threshold Tol_{RSM}. The parameter Tol_{RSM} is set at $3\sigma x$, since the error is modeled as a normally distributed function. If Tol_{RSM}
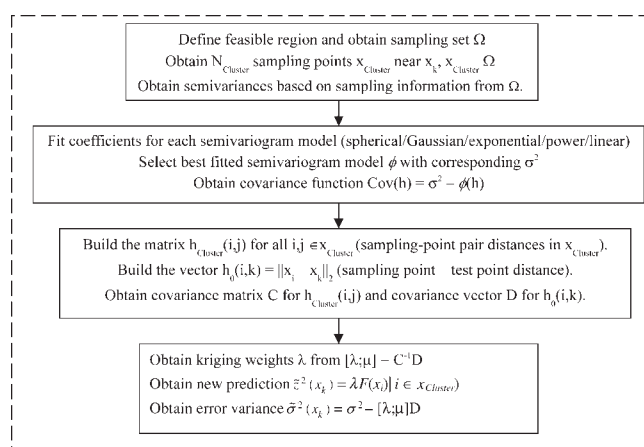

Figure 2. Flowchart of kriging algorithm for obtaining a prediction at test point x_k.
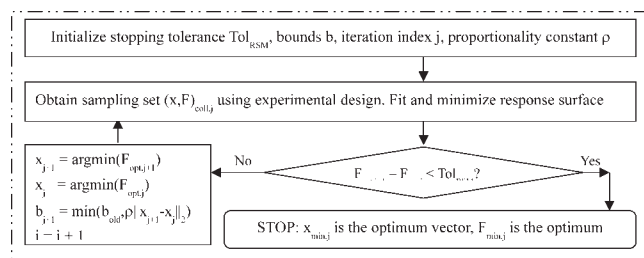
Figure 4. Flowchart of RSM algorithm.

exceeds this value, premature termination may occur. Conversely, a value lower than $3\sigma(x)$ may cause the algorithm to fail as artificial importance will be placed on the noise. A flowchart of the algorithm is presented in Figure 4.

The construction of a response surface at each iteration can lead to high sampling expense; furthermore, since local models are used, the discovered optimum may be suboptimal, since convexity of the black-box function cannot be known a priori. If the convex feasible region is of arbitrary shape and iterates fall near nonlinear or multiple boundaries, the generation of collocation points using symmetrical design stencils can lead to infeasible points being found. Since more reliable search directions are usually obtained from response surfaces constructed over symmetrical design stencils, to retain feasibility, the experimental design for collocation point generation is adaptively changed depending upon iterate proximity to the boundary. If iterates are near nonlinear boundaries or the intersection of linear boundaries, the central composite design is used; conversely, if the iterate is near a single linear boundary, a factorial design is applied, as shown in Figure 5a. The particular designs used provide a more advantageous search of the respective subregions defined by the surrounding constraints over their counterparts. In order to ensure feasibility, the maximum possible design radius is set as the minimum of the distance between the iterate and all constraints, shown in Figure 5b. However, as iterates approach a boundary, this radius can become small, leading to the construction of response surfaces over a small region and small steps taken towards the optimum. When the radius is reduced below a prespecified criterion, such as 1% of the smallest box-constrained operating range for N variables, a lower-dimensional response surface in terms of $(N - 1)$ variables is projected onto the respective constraint to ensure that large steps are still taken toward the optimum, as displayed in Figure 5c.

To further reduce the sampling expense for higher-dimensional problems, line search techniques can also be incorporated into the algorithm whereby sampling is conducted at a sequence of points obtained along the path defined by the current and previous iterates. Although application of response surface techniques can reduce the sampling expense, the optimum found may be sensitive to the nominal point and the solution found may only be a local minimum. When kriging is first used to identify the best regions for local search, this problem no longer exists, since a global model is built.

The kriging-RSM algorithm is now described. First, a set of nominal sampling information is obtained and a set of fea-

sible test points is generated, over which to build the kriging global model. As mentioned before, it is important that the number of test points not be too high as model building costs will increase. Discretization can be used to generate the set of test points for 2- or 3-dimensional problems, but the number may become too high for problems of higher dimension. For the presented examples in which the problem dimensionality was greater than three, it was found that 1000 randomly generated points was sufficient for building the kriging predictor without substantial increase in the model building costs. Next, a test point x_k is selected and both its kriging prediction and variance are obtained according to the kriging algorithm described in Figure 2. After the kriging predictions have been obtained at all test points, the average value of the kriging predictor is then obtained and compared to the corresponding value obtained in the previous iteration. If the difference between these values exceeds a stopping tolerance Tol_{Krig}, additional sampling is performed. The parameter Tol_{Krig} is set at 0.01 and is a generally noise-independent parameter as the number of test points used to build the kriging mapping exceeds the number of points at which sampling data are obtained. It should be emphasized that the global optimum can be missed when using kriging or any other global modeling technique due to the fact that black-box functions prevent knowledge of convexity. To minimize the possibility of missing the neighborhood of the global optimum, the sampling set at each kriging iteration is comprised of three subsets each having an equal number of data points. Each subset has a family of points conforming to one of the following characteristics: (1) high variance, (2) high difference between prediction estimates for consecutive iterations, and (3) minimum prediction values. In addition, the sampling locations within each family subset are separated by a minimum L^2-normed distance to maximize the amount of global information obtained. After performing additional sampling, the kriging predictor is then refined by generating new predictions again at all test points. Once the difference in the average kriging predictor falls below Tol_{Krig} for consecutive iterations, refinement of the current best candidate vectors yielding the lowest kriging predictions in N local regions is performed according to the RSM algorithm described in Figure 4. For the presented examples, whenever an iterate approaches the feasible region boundary, the computational burden associated with identifying the corresponding con-
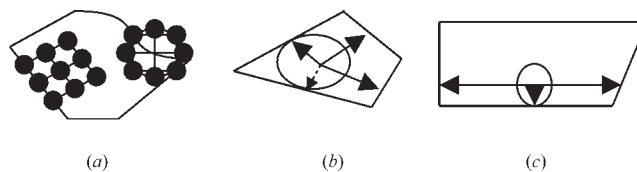

(a)   (b)   (c)

Figure 5. Using adaptive experimental designs to retain feasibility and still obtain good search directions toward an optimum.

(a) Application of CCD or factorial design depending iterate proximity to a linear or nonlinear boundary; (b) Ensuring feasibility by setting the maximum radius as the minimum Euclidean distance between the iterate and boundaries; (c) Projection of the n-dimensional response surface onto constraints to avoid taking small steps toward the optimum.
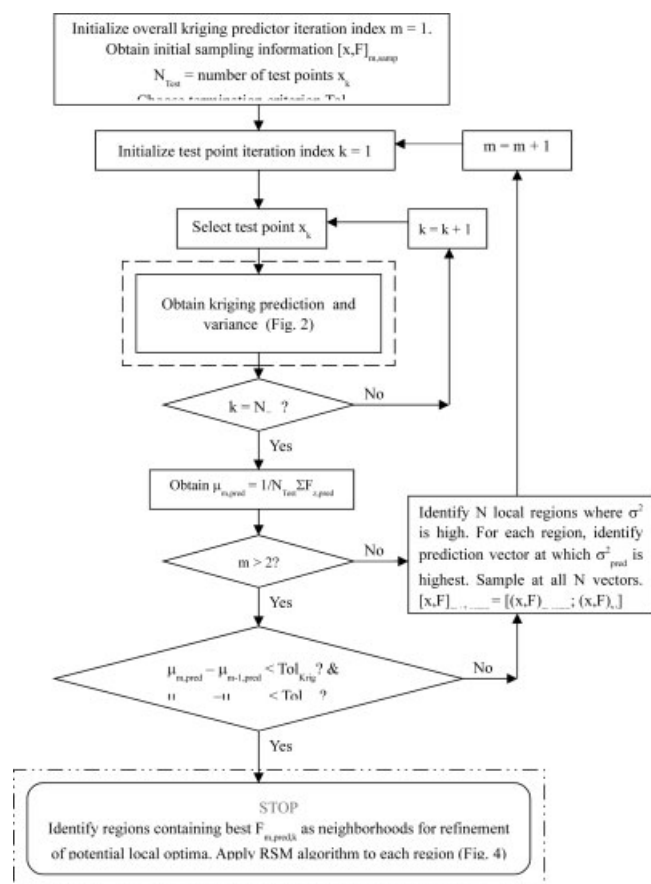
Figure 6. Flowchart of kriging-RSM algorithm.

response surfaces at every iteration. The fourth algorithm refers to the technique of building a response surface and minimizing it in the context of a QP to obtain global steps to the optimum. For each algorithm, 100 trials are performed from a set of randomly selected feasible starting iterates. The percentage of iterates converging to the global optimum is presented in the second column of the respective table. Using information taken from the subset of starting iterates successfully finding the global optimum, the average number of iterations required, function calls needed, and CPU time required are also reported. All computational results are obtained using an HP dv8000 CTO Notebook PC containing a 1.8 GHz AMD Turion 64 processor.

Example 1: Six-Hump Camel Back Function. The six-hump camel back function is a well-known global optimization test function that is box-constrained. Introducing both noise and black-box complications into this example, the output z_2 is simulated according to a normally distributed perturbation of the deterministic function. The problem is formulated as shown below and the deterministic problem is presented in Figure 7:

$$
\begin{aligned}
\min \ & z_2 \\
\text{s.t.} \quad z_2 = & \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1 x_2 \\
& - \left(-4 + 4x_2^2\right)x_2^2 + N(0, 0.05) \\
& -2 \le x_1 \le 2 \\
& -1 \le x_2 \le 1
\end{aligned}
\tag{7}
$$

This problem contains multiple local optima in addition to two global optima and is solved by applying four optimization algorithms. Table 1 presents the results obtained for this problem.

The global minimum is obtained in the fewest number of function evaluations when using the S-SD RSM program, but this is to be expected because the set of all optima are fairly
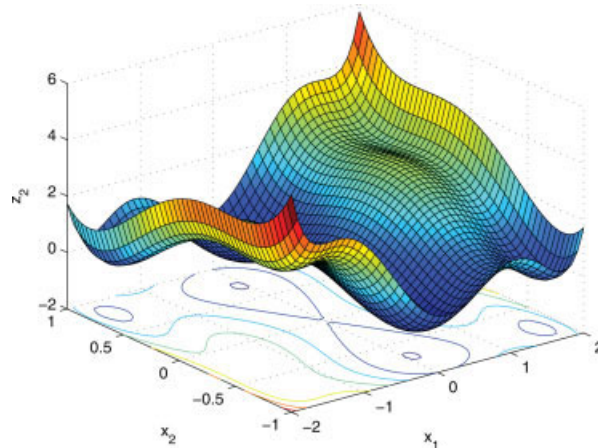
straints is low, since the problem sizes are small. However, modifications to the proposed methodology may be necessary for high-dimensional problems containing many constraints and will be addressed in a future work. The kriging-RSM algorithm terminates after selecting the global optimum as the best candidate value from the set of local optima obtained using RSM. A flowchart of the proposed methodology is presented in Figure 6.

Examples

In this section, the proposed kriging-RSM algorithm is applied to five numerical examples. The first four examples are presented in order of increasing complexity in terms of problem dimensionality and presence of linear/nonlinear constraints. The last example is a kinetics case study intended to provide an application of the new method to a more realistic example. For each example, a table of computational results is provided that illustrates the performance of four optimization algorithms. The kriging-RSM algorithm refers to the new methodology of applying steepest descent to response surfaces after building a global model. The remaining algorithms are stand-alone response surface methods. The second algorithm, S-SD RSM, applies direct search in the early stages followed by application of steepest descent to response surfaces once the neighborhood of a local optimum has been found. The third algorithm applies steepest descent to

Figure 7. Six-hump camel function.

[Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

Table 1. Comparison of the Performance of the Kriging-RSM Algorithm Against Stand-Alone RSM Methods in Finding the Global Optimum for the Six-Hump Camel Back Function

Opt. Algorithm	% Starting Iterates Finding Global Optimum	No. of Iterations		No. of Function Calls			CPU Time (s)		
		Kriging	RSM	Kriging	RSM	Total	Kriging	RSM	Total
Kriging-RSM	81	9	3	47	17	64	5.98	0.09	6.07
S-SD RSM	35	N/A	10	N/A	34	34	N/A	0.23	0.23
Standard RSM	40	N/A	9	N/A	47	47	N/A	0.28	0.28
RSM-SQP	30	N/A	7	N/A	41	41	N/A	0.18	0.18

evenly spread and have wide basins, meaning that the quadratic curvature is not apparent until the iterates are very close to the optimum. The Standard RSM algorithm performs slightly better than the RSM-SQP algorithm because the two symmetrical global optima $(-0.0898, 0.7126)$ and $(0.0898, -0.7126)$ are located near the center of the feasible region $(0,0)$. When applying the RSM-SQP solver, the optimization of many nominal quadratic approximations over the global region leads to identification of local solutions residing near the feasible region boundaries as can be seen in Figure 7. The kriging-RSM program requires $\sim 50\%$ additional function evaluations compared to the Standard RSM and RSM-SQP programs, and almost twice as many as the S-SD RSM solver. However, the additional cost is balanced by the fact that this algorithm leads to global convergence in 81% of the cases. The remaining 19% of the starting iterates successfully found the basin containing either global optimum but terminated at values outside a 2% radius of the optimal solution. The average CPU time required by the Kriging-RSM algorithm is an order of magnitude higher than that of the standalone RSM solvers, but is still relatively low.

Example 2: Schwefel Function. In this example, the four optimization algorithms are applied to the 2D Schwefel test function, a problem also taken from the global optimization literature. This box-constrained problem is modified to include a linear and a nonlinear constraint. The black-box function z_2 depends on both x_1 and x_2 and is noisy, modeled by perturbing its deterministic value by a normally distributed error whose standard deviation is 1% of the deterministic function. The problem is formulated as shown below:

$$\min \; z_2$$
$$\text{s.t.} \quad z_2 = \sum_{i=1}^{2} -x_i \sin\sqrt{|x_i|} + N(0,7)$$
$$-2x_1 - x_2 \leq 800$$
$$0.004x_1^2 - x_1 + x_2 \leq 500$$
$$-500 \leq x_1, x_2 \leq 500 \tag{8}$$

The deterministic equivalent of this problem is shown in Figure 8. This function contains a number of local optima and one global optimum at $(420.97, -302.525)$ with an objective value of -719.53. This problem was selected in order to examine the performance of the stand-alone RSM algorithms when it is the underlying geometry instead of the noise that poses the main complication affecting successful

convergence to the global optimum. In Table 2, results are presented for this example.

For the stand-alone response surface methods, it is observed that the starting iterate easily finds the global optimum if it is within its neighborhood. However, the global optimum is located in a corner of the feasible region and is surrounded by a set of local optima. When applying the Standard RSM and RSM-SQP algorithms, the optimization of the initial response surface can lead to premature termination at the local minima. In contrast, when applying the S-SD RSM algorithm, since response surfaces are not built until the end stages of the algorithm, the global optimum can be attained using a lower number of function calls while also avoiding the local optima. The average number of function calls observed for the Standard RSM and RSM-SQP solvers is approximately the same because the interior candidate values are generally superior to the objective values obtained by sampling at the extremes of the feasible region. Due to the number of local solutions found within the interior of the feasible region, there is a low probability of convergence to the global optimum using the local algorithms. In contrast, application of the Kriging-RSM solver requires almost twice as many function calls as the local methods to find a solution, but global convergence is observed in all cases. The strategy of building a global model before conducting local search is particularly successful for this problem since convergence to a suboptimal solution using the local methods is avoided. The low CPU time required for solution of this
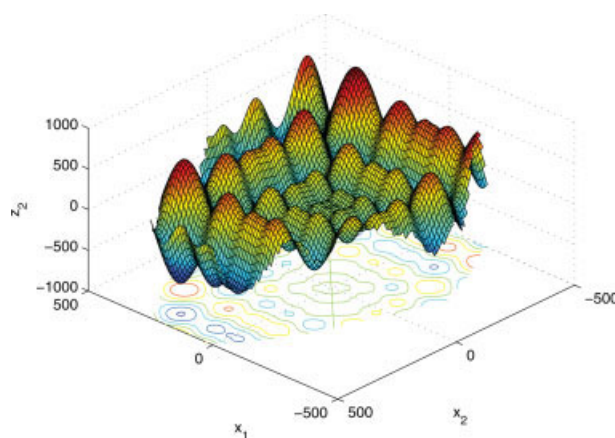

Figure 8. Schwefel function corresponding to the optimization problem presented in Eq. 8.

[Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

Table 2. Comparison of the Performance of the Kriging-RSM Algorithm Against Stand-Alone RSM Methods in Finding the Global Optimum for the Schwefel Problem

Opt. Algorithm	% Starting Iterates Finding Global Optimum	No. of Iterations		No. of Function Calls			CPU Time (s)		
		Kriging	RSM	Kriging	RSM	Total	Kriging	RSM	Total
Kriging-RSM	100	14	4	74	35	109	6.54	0.11	6.65
S-SD RSM	14	N/A	7	N/A	64	64	N/A	0.3	0.3
Standard RSM	11	N/A	5	N/A	53	53	N/A	0.23	0.23
RSM-SQP	11	N/A	5	N/A	49	49	N/A	0.24	0.24

problem using the Kriging-RSM algorithm increases the attractiveness of using this method as an alternative to the stand-alone response surface methods.

Example 3. This example involves five variables with four linear constraints and is an example modified from Floudas (1995). The problem is originally presented as an MINLP, and for this example a relaxed NLP is solved by relaxing the integrality constraints. The black-box variable z_2 is a function of two continuous variables and is noisy according to a normally distributed error with standard deviation 0.01. The problem is formulated as shown in (9) and a plot of z_2 as a deterministic function of the two continuous variables is presented in Figure 9:

$$\min F = y_1 + y_2 + y_3 + 5z_2$$
$$\text{s.t. } z_2 = (8x_1^4 - 8x_1^2 + 1)(2x_2^2 - 1) + N(0, 0.01)$$
$$3x_1 - y_1 - y_2 \leq 0$$
$$5x_2 - 2y_1 - y_2 \leq 0$$
$$-x_1 + 0.1y_2 + 0.25y_3 \leq 0$$
$$x_1 - y_1 - y_2 - y_3 \leq 0$$
$$0.2 \leq x_1, x_2 \leq 1$$
$$0 \leq y_i \leq 1 \quad i = 1, ..., 3 \tag{9}$$

A family of global optimal solutions exists for this problem in terms of the binary variables; however, the global optimum of -0.98688 is not achieved unless the optimal vector for the continuous variables is achieved at (0.2, 0.2). The results for this example are presented in Table 3.

The increased CPU time reported for this example using the Kriging-RSM algorithm is higher than that observed for the earlier presented 2D examples because the kriging predictor is generated over a 5D grid. Even though it appears from the plot that the nominal iterates in the continuous space should converge to (0.2,0.2) easily, movement is restricted by the feasible region defined by the relaxed binary variables, thereby causing many of the nominal iterates to converge to a suboptimal solution when applying the stand-alone response surface methods. Even though the number of function evaluations required is higher when using the Kriging-RSM algorithm, global convergence is observed in nearly all cases with only a modest increase in the overall model building costs.

Example 4. This example is also taken from Floudas (1995) and involves 11 variables with 11 linear constraints and three nonlinear constraints. Although the problem is also formulated as an MINLP, for this example it is solved as a relaxed NLP. The black-box variable z_2 is a function of six continuous variables and is noisy according to a normally distributed error with standard deviation 0.01. The problem

is formulated as shown in (10):

$$\min z_2 + 5y_1 + 8y_2 + 6y_3 + 10y_4 + 6y_5 + 140$$
$$\text{s.t. } z_2 = -10x_3 - 15x_5 - 15x_9 + 15x_{11} + 5x_{13} - 20x_{16} + \exp(x_3)$$
$$\quad + \exp(x_5/1.2) - 60\ln(x_{11} + x_{13} + 1) + N(0, 0.01)$$
$$\quad - \ln(x_{11} + x_{13} + 1) \leq 0$$
$$\exp(x_3) - 10y_1 \leq 0$$
$$\exp(x_5/1.2) - 10y_2 \leq 0$$
$$-x_3 - x_5 - 2x_9 + x_{11} + 2x_{16} \leq 0$$
$$-x_3 - x_5 - 0.75x_9 + x_{11} + 2x_{16} \leq 0$$
$$x_9 - x_{16} \leq 0$$
$$2x_9 - x_{11} - 2x_{16} \leq 0$$
$$-0.5x_{11} + x_{13} \leq 0$$
$$0.2x_{11} - x_{13} \leq 0$$
$$1.25x_9 - 10y_3 \leq 0$$
$$x_{11} + x_{13} - 10y_4 \leq 0$$
$$-2x_9 + 2x_{16} - 10y_5 \leq 0$$
$$y_1 + y_2 = 1$$
$$y_4 + y_5 \leq 1$$
$$a \leq x_i \leq b \quad i = 3, 5, 9, 11, 13, 16$$
$$0 \leq y_i \leq 1 \quad i = 1, ..., 5$$
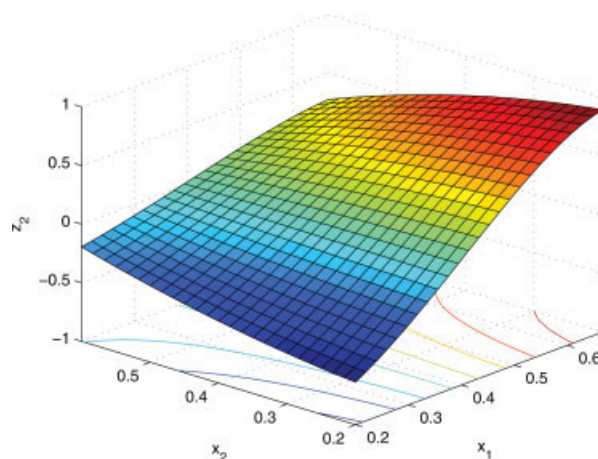$$a^T = (0, 0, 0, 0, 0, 0) \quad b^T = (2, 2, 2, 2, 2, 3) \tag{10}$$

Figure 9. Plot of objective as a function of the continuous variables for the optimization problem described by Eq. 9.

[Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

Table 3. Comparison of the Performance of the Kriging-RSM Algorithm Against Stand-Alone RSM Methods in Finding the Global Optimum for the Problem Presented in Eq. 9

Opt. Algorithm	% Starting Iterates Finding Global Optimum	No. of Iterations		No. of Function Calls			CPU Time (s)		
		Kriging	RSM	Kriging	RSM	Total	Kriging	RSM	Total
Kriging-RSM	97	7	7	40	47	87	10.9	0.5	11.4
S-SD RSM	70	N/A	13	N/A	40	40	N/A	0.24	0.24
Standard RSM	80	N/A	10	N/A	74	74	N/A	0.48	0.48
RSM-SQP	82	N/A	10	N/A	72	72	N/A	0.75	0.75

The solution of the deterministic problem is (1.903,2,2,1.403,0.701,2,0.571,0.429,0.25,0.21,0) with an objective value of -0.554. Because of the higher problem dimensionality, the kriging predictor is generated from a set of 1000 feasible points unevenly dispersed throughout the feasible region rather than from an 11D grid. The results for this example are presented in Table 4.

For this problem, the number of function calls required to obtain the global optimum is higher by two orders of magnitude compared to the corresponding values obtained from the earlier examples. The significantly higher number of function calls needed when using the response surface methods is due to the increased number of collocation points required to build response surfaces in higher dimensions even though the CCD is employed. The average number of function calls required for convergence in the kriging predictor is ~14% of the number needed during the refinement stage of the optimization. This result suggests that the kriging predictor sufficiently captures only the coarse geometry. In contrast to the previous examples, when using the local solvers, a higher number of response surfaces must be obtained before the optimum is found because the problem dimensionality is higher. This leads to increased local model building costs which are on the same order of magnitude as the model building costs for the kriging predictor. However, the required number of function evaluations required using the kriging-RSM algorithm is approximately half of the number required using the stand-alone response surface methods, again emphasizing the success observed when using information obtained from building the kriging global model to guide the local optimization using RSM.

Example 5: Kinetics Example. In this section, a realistic example is considered that involves a kinetics case study taken from Bindal et al. (2006[28]). There are five reaction species, two of which are system inputs. The reactions are assumed to occur in an ideal CSTR and the reaction network considered is a modification of the Fuguitt and Hawkins mechanism (Floudas and Pardalos, 1995[29]) given by: $A \rightarrow E$, $A \rightarrow D$, $B \rightarrow D$, $C \rightarrow 2D$, and $2A \rightarrow C$, where the rate

constants are $k_1^f = 3.33384$ s^{-1}, $k_2^f = 0.26687$ s^{-1}, $k_3^f = 0.29425$ s^{-1}, $k_3^r = 0.14940$ s^{-1}, $k_4^f = 0.011932$ s^{-1}, $k_4^r = 1.8957 \times 10^{-3}$ m^3/(mol s), $k_5^f = 9.598 \times 10^{-6}$ m^3/(mol s), $V = 0.1$ m^3, $F = 0.008$ m^3/s. Only A and C enter the reactor with concentrations C_A^0, C_C^0 in the ranges of $3 \times 10^3 \leq C_A^0$ [mol/m^3] $\leq 3 \times 10^4$ and $0 \leq C_C^0$ [mol/m^3] $\leq 1 \times 10^4$. The dynamic behavior of the system is described by problem (11).

$$\min \ G = 4(x - 0.6)^2 + 4(y - 0.4)^2 + \sin^3(\pi x) + 0.4$$
$$x = 0.1428 C_C^{SS} - 0.357 C_D^{SS}$$
$$y = -0.1428 C_C^{SS} + 2.857 C_D^{SS} + 1.0$$
$$\frac{dC_A}{dt} = \frac{F}{V}\left(C_A^0 - C_A\right) - k_1^f C_A - k_2^f C_A - k_5^f C_A^2$$
$$\frac{dC_B}{dt} = \frac{F}{V}(-C_B) - k_3^f C_B + k_3^r C_D$$
$$\frac{dC_C}{dt} = \frac{F}{V}\left(C_C^0 - C_C\right) - k_4^f C_C + 0.5 k_4^r C_D^2 + 0.5 k_5^f C_A^2$$
$$\frac{dC_D}{dt} = \frac{F}{V}(-C_D) + k_2^f C_A + k_3^f C_B - k_3^r C_D + 2 k_4^f C_C - k_4^r C_D^2$$
$$\frac{dC_E}{dt} = \frac{F}{V}(-C_E) + k_1^f C_A \tag{11}$$

where C_C^{SS} and C_D^{SS} represent the macroscale steady-state values of C_C and C_D, respectively. A contour plot of the objective as a function of the input variables C_A^0 and C_C^0 is shown in Figure 10.

The deterministic solution yields a global optimum of $G = 0.7422$ at $[C_A^0, C_C^0] = [10.117, 8.378]$ and a local optimum of $G = 1.2364$ at $[13.202, 3.163]$. To introduce black-box complications, the rate equations are assumed to be unknown, so a microscopic model is used instead, represented using a lattice containing N molecules. The microscopic model is generated by first translating concentrations to molecular variables, evolving the microscopic system using the Gillespie algorithm, and then mapping back the final measured variables to concentrations. The Gillespie algorithm evolves reaction networks by assigning an event probability to each reaction and choosing one to occur by

Table 4. Comparison of the Performance of the Kriging-RSM Algorithm Against Stand-Alone RSM Methods in Finding the Global Optimum for the Problem Presented in Eq. 10

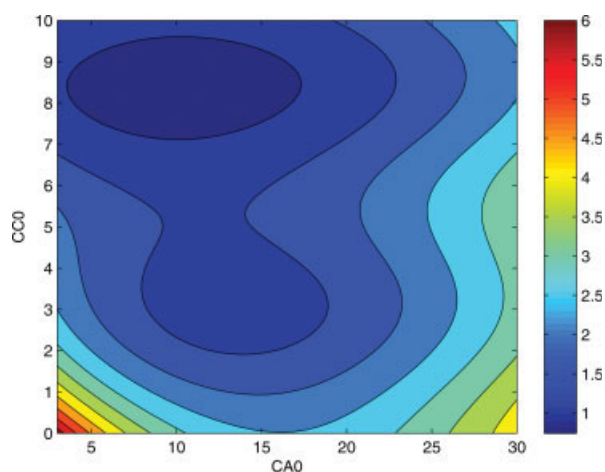| Opt. Algorithm	% Starting Iterates Finding Global Optimum	No. of Iterations		No. of Function Calls			CPU Time (s)		
		Kriging	RSM	Kriging	RSM	Total	Kriging	RSM	Total
Kriging-RSM	76	7	17	62	381	443	11.6	9.6	21.2
S-SD RSM	55	N/A	24	N/A	855	855	N/A	18.8	18.8
Standard RSM	54	N/A	24	N/A	900	900	N/A	20.1	20.1
RSM-SQP	53	N/A	24	N/A	850	850	N/A	19.2	19.2

Figure 10. Kinetics Problem corresponding to the optimization problem presented in Eq. 11.

[Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

generating a random number. After updating the molecular variables, another random number is obtained and the corresponding reaction is selected. This procedure continues occurring until little change is observed in the molecular variables. The noise in the output concentrations thereby arises as a function of how coarse the microscopic model is. Steady-state solution vectors are obtained from an initial point by running the microscale simulations for a long time horizon, after which the objective function can be evaluated. The variance of the objective is then evaluated over k replicate simulations at a nominal point x as:

$$\sigma^2 = \mathrm{Var}(\varepsilon) = \mathrm{Var}\{G(x,N)|i=1,...,k\} \qquad (12)$$

For this example, the noise applied to the objective is in the form of a normally distributed error with a standard deviation of $\sigma = 0.011$, corresponding to a microscale system size of $N = 100{,}000$. Table 5 provides results for the four optimization algorithms applied to this example. The CPU time excludes the time required for each function call in the form of a microscopic model simulation.

It is observed that even though approximately double the number of function calls are required for convergence to an optimum using the kriging-RSM algorithm, the global minimum is achieved in all cases. Since both the local and global optimum are located in wide shallow basins near the center of the feasible region, starting iterates are found to converge to either

optimum quickly when using the stand-alone response surface methods as seen by the low model building costs.

For all the presented examples, it is seen that the modeling and optimization costs are relatively low when applying the Kriging-RSM algorithm due to the solution of $It_{Krig}N_{Test} + It_{RSM}$ systems of linear equations, where It_{Krig} and It_{RSM} represent the number of iterations required for the respective stages of the methodology. In contrast, only It_{RSM} systems of linear equations must be solved when applying any of the S-SD, Standard RSM, or RSM-SQP optimization methods. The value of $It_{Krig}N_{Test}$ is higher than It_{RSM}, so the major source of the required CPU time is attributed to kriging modeling costs. For problems of significantly higher dimensions, such as atomistic modeling applications, the overall problem costs will be dominated by the time required for either real-time lab experiments or molecular simulation.

Conclusions and Future Work

In this work, a new kriging-RSM algorithm is presented for the solution of constrained NLPs containing black-box functions and noisy variables. Using the kriging methodology, a global model is built whereby predictions and variances at discretized test points are obtained using a fitted covariance function built from scattered sampling data. After additional sampling is performed, the covariance function is updated, leading to better predictions and an improved global model. Once convergence in the average value is observed, regions containing possible local optima are identified and RSM is applied to refine the current set of candidate vectors. The global optimum is selected as the best point of the set of local optima. The likelihood of finding the global optimum increases when applying kriging-RSM because the global model obtained using kriging allows the set of regions containing potential local optima to be identified. The proposed methodology extends the capabilities of current kriging and RSM methods to problems containing ten variables whose feasible region is described by arbitrary linear and nonlinear constraints instead of simply being box-constrained. It is found that the proposed approach leads to a substantial increase in the probability of finding the global optimum compared to stand-alone RSM at minimal increased sampling cost due to model construction of the kriging predictor, even though no theoretical guarantees of global optimality are made. The cost of building kriging models can become expensive for problems containing many variables or when two or more black-box functions are present and multiple global models must be built at each iteration. Current work focuses on comparing algorithmic performance when using adaptive versus fixed grid resolutions and also on building kriging

Table 5. Comparison of the Performance of the Kriging-RSM Algorithm Against Stand-Alone RSM Methods in Finding the Global Optimum for the Kinetics Example Presented in Eq. 11

Opt. Algorithm	% Starting Iterates Finding Global Optimum	No. of Iterations		No. of Function Calls			CPU Time (s)		
		Kriging	RSM	Kriging	RSM	Total	Kriging	RSM	Total
Kriging-RSM	100	9	4	46	25	71	5.27	0.09	5.36
S-SD RSM	55	N/A	6	N/A	30	30	N/A	0.09	0.09
Standard RSM	54	N/A	5	N/A	35	35	N/A	0.1	0.1
RSM-SQP	53	N/A	5	N/A	32	32	N/A	0.1	0.1

predictors on a need-to-generate basis for the solution of process synthesis problems.

Acknowledgments

Literature Cited

1. Barton RR, Ivey JS. Nelder-Mead Simplex modifications for simulation optimization. *Manage Sci.* 1996;42:954–973.
2. Jones D, Pertunnen CD, Stuckman BE. Lipschitzian optimization without the Lipschitz constant. *J Optim Theory Appl.* 1993;79:151–181.
3. Huyer W, Neumaier A. Global optimization by multilevel coordinate search. *J Global Optim.* 1999;14:331–355.
4. Storn R, Price K. Differential Evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces. *J Global Optim.* 1997;11:341–359.
5. Brekelmans R, Driessen L, Hamers H, den Hertog D. Gradient estimation schemes for noisy functions. *J Optim Theory Appl.* 2005;126:529–551.
6. Choi TD, Kelley CT. Superlinear Convergence and implicit filtering. *SIAM J Optim.* 2000;10:1149–1162.
7. Gilmore P, Kelley CT. An implicit filtering algorithm for optimization of functions with many local minima. *SIAM J Optim.* 1995;5:269–285.
8. Myers R, Montgomery D. *Response Surface Methodology.* New York, NY: Wiley, 2002.
9. Box G, Wilson K. On the experimental attainment of optimum conditions. *J R Stat Soc B.* 1951;13:1–45.
10. Hoerl A. Optimum solution of many variables equations. *Chem Eng Prog.* 1959;11:69–78.
11. Davis E, Ierapetritou M. Adaptive optimisation of noisy black-box functions inherent in microscopic models. *Comput Chem Eng.* 2007;31:466–476.
12. Stander N, Craig KJ. On the robustness of a simple domain reduction scheme for simulation-based optimization. *Eng. Comp.: Int. J. Comp. Aid. Eng.* 2002;19:431–450.
13. Roux W, Stander N, Haftka R. Response surface approximations for structural optimization. *Int J Numer Methods Eng.* 1998;42:517–534.
14. Chen M, Chen K, Lin C. Optimization on response surface models for the optimal manufacturing conditions of dairy tofu. *J Food Eng.* 2005;68:471–480.
15. Tang W, Zhang L, Linninger A, Tranter R, Brezinsky K. Solving kinetic inversion problems via a physically bounded Gauss-Newton (PGN) method. *Ind Eng Chem Res.* 2005;44:3626–3637.
16. Zhang L, Xue C, Malcolm A, Kulkarni K, Linninger A. Distributed system design under uncertainty. *Ind Eng Chem Res.* 2006;45:8352–8360.
17. Jones D. A taxonomy of global optimization methods based on response surfaces. *J Global Optim* 2001;21:345–383.
18. Jones D, Schonlau M, Welch W. Efficient global optimization of expensive black-box functions. *J Global Optim* 1998;13:455–492.
19. Goovaaerts P. *Geostatistics for Natural Resources Evolution.* New York, NY: Oxford University Press, 1997.
20. Isaaks E, Srivistava R. *Applied Geostatistics.* New York, NY: Oxford University Press, 1989.
21. Krige D. *A Statistical Approach to Some Mine Valuations and Allied Problems at the Witwatersrand*, Master's Thesis. Johannesburg: University of Witwatersrand, 1951.
22. Matheron G. Principles of geostatistics. *Econ Geol.* 1963;58:1246–1266.
23. Cressie N. *Statistics for Spatial Data.* New York, NY: Wiley, 1993.
24. Sacks J, Schiller SB, Welch WJ. Designs for computer experiments. *Technometrics.* 1989;31:41–47.
25. Santner T, Williams B, Notz W. *The Design and Analysis of Computer Experiments.* New York, NY: Springer, 2003.
26. Maranas C, Floudas C. Finding all solutions of nonlinearly constrained systems of equations. *J Global Optim.* 1995;7:143–182.
27. Box G, Hunter S, Hunter WG. *Statistics for Experimenters: Design, Innovation, and Discovery*, 2nd ed. New York, NY: Wiley-Interscience, 2005.
28. Bindal A, Ierapetritou MG, Balakrishnan S, Makeev A, Kevrekidis I, Armaou A. Equation-free, coarse-grained computational optimization using timesteppers. *Chem Eng Sci.* 2006;61:779–793.
29. Floudas CA, Pardalos PM. *A Collection of Test Problems for Constrained Global Optimization Algorithms.* New York, NY: Springer, 1991.