Computational Improvements to Estimating Kriging Metamodel Parameters

Jay D. Martin
Research Associate
Applied Research Laboratory,
State College, PA 16804-0030
e-mail: jdm111@psu.edu

The details of a method to reduce the computational burden experienced while estimating the optimal model parameters for a Kriging model are presented. A Kriging model is a type of surrogate model that can be used to create a response surface based a set of observations of a computationally expensive system design analysis. This Kriging model can then be used as a computationally efficient surrogate to the original model, providing the opportunity for the rapid exploration of the resulting tradespace. The Kriging model can provide a more complex response surface than the more traditional linear regression response surface through the introduction of a few terms to quantify the spatial correlation of the observations. Implementation details and enhancements to gradient-based methods to estimate the model parameters are presented. It concludes with a comparison of these enhancements to using maximum likelihood estimation to estimate Kriging model parameters and their potential reduction in computational burden. These enhancements include the development of the analytic gradient and Hessian for the log-likelihood equation of a Kriging model that uses a Gaussian spatial correlation function. The suggested algorithm is similar to the SCORING algorithm traditionally used in statistics. [DOI: 10.1115/1.3151807]

1 Introduction

The Kriging model is a powerful tool that can be used in engineering design optimization to reduce computational burden [1–4]. It is capable of producing a complex response function given a set of observations of a more computationally expensive computer simulation. It is a statistical model that can also provide an estimate of the uncertainty associated with its estimates if a few assumptions are satisfied such as (1) the observations can be represented as a spatial Gaussian process and (2) that spatial process is stationary (i.e., the parameters used to describe the process can be assumed constant over the domain of the model) [5]. This ability to estimate the uncertainty in its estimates has motivated research in adaptive sampling methods using Kriging models [6–9].

The Kriging model has many desirable properties for use as an engineering design tool, but it still has a few drawbacks. One of the major drawbacks is the lack of off-the-shelf software or easy-to-implement algorithms. The only openly available implementation is the design and analysis of computer experiments (DACE) toolbox for MATLAB [10]. A second drawback is the potential computational expense of estimating the Kriging model parameters. This paper does not attempt to address the first drawback. To address the second drawback, algorithms are presented that can efficiently estimate the Kriging model parameters using maximum likelihood estimation (MLE). These are also compared with the method used in the DACE toolbox for MATLAB.

The MLE parameter estimation method is an optimization method that maximizes the logarithm of the likelihood equation. The MLE optimization is often the most computationally expensive element of the successful creation and use of the Kriging model as an approximation model. This work uses the Levenberg–Marquardt (LM) heuristic with a slightly modified Newton–Raphson algorithm to deal with the constraints on the model parameters. One key developments of this work is the derivation of the analytical gradient and Hessian for the log-likelihood equation when a Gaussian spatial correlation function is used, the most common spatial correlation function used when approximating computationally expensive computer models [11]. This derivation, given in the Appendix, provides a significant improvement over using finite difference methods to estimate the gradients. As an application of these derived analytical forms, Sec. 4 compares the relative performance of three different variations of gradient-based optimization algorithms to solve the MLE. These algorithms include the Newton–Raphson method, the traditional SCORING method, and the simplified version of the SCORING method. This paper ends with some conclusions and suggestions for future work in this area.

2 Kriging Model Form

The mathematical form of a Kriging model has two parts, as shown in Eq. (1). The first part is a linear regression of the data with k regressors that model the drift of the process mean, the *trend*, over the domain. Most engineering applications of the Kriging model use a constant trend model over the domain and rely on the second part of the model to "pull" the response surface through the observed data by quantifying the correlation of nearby points [11]. There are three types of Kriging identified in the literature: simple, ordinary, and universal Kriging. They are defined as a Kriging model with no trend function, a constant trend function, and a multiparameter trend function, respectively [12].

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^{k} \beta_i f_i(\mathbf{x}) + Z(\mathbf{x}) \tag{1}$$

The second part, $Z(\mathbf{x})$, is a model of a stationary Gaussian random process with zero mean and covariance

$$\mathrm{Cov}(Z(\mathbf{x}_1), Z(\mathbf{x}_2)) = \sigma^2 R(\mathbf{x}_1, \mathbf{x}_2) \tag{2}$$

The process variance, σ^2, is a scalar coefficient of the spatial correlation function (SCF), $R(\mathbf{x}_1, \mathbf{x}_2)$.

There are two important aspects to choosing the model form within the constraints of Eqs. (1) and (2): (1) choice of the trend function regressors, $\mathbf{f}$, and (2) choice of the SCF. The most common choice of trend function regression, when using a Kriging model as a deterministic approximation of a computer model, is a constant trend function, $\mathbf{f}(\mathbf{x})=\{1\}$. This is due to the ability of the correlation function to effectively model the observed variations of the computer model's output over its domain [13]. A nonconstant trend function can, in general, be used to improve the predictive capabilities and its efficiency, i.e., reduce the predicted variance across the model's domain [14,15].

The purpose of the SCF used in the second part of Eq. (1) is to quantify the observed correlations of the residuals. The Gaussian function is the most commonly used SCF in engineering design since it provides a relatively smooth surface, making it a better choice when used with gradient-based optimization algorithms [16]. It also requires the estimation of a single parameter for each dimension. The multidimensional Gaussian SCF [17,18] is defined by combining one-dimensional Gaussian SCFs with the product correlation rule resulting in

Contributed by the Design Automation Committee of ASME for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received September 13, 2007; final manuscript received March 25, 2009; published online July 9, 2009. Review conducted by Zissimos P. Mourelatos. Paper presented at the ASME 2007 Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2007), Las Vegas, NV, September 4–7, 2007.

$$R(\mathbf{x}_1,\mathbf{x}_2)=\prod_{i=1}^{d} e^{-((\mathbf{x}_{2,i}-\mathbf{x}_{1,i})/\theta_i)^2} \tag{3}$$

where $\mathbf{x}_{1,i}$ is the ith dimension of the vector $\mathbf{x}_1$ and θ_i is the correlation range parameter in the ith dimension. When calculating the multivariate correlation using the Gaussian SCF, it is more computationally efficient to add the univariate distance ratios and calculate the exp function only once rather than d times and multiplying the resulting values, as indicated in Eq. (3).

The locations of a set of n observations of the computer model are defined as $\mathbf{X}=\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n\}\subset\Omega$, where Ω is the set of all possible inputs to the model that result in an output, i.e., the domain of the computer model. The resulting outputs are $\mathbf{y}=\{y(\mathbf{x}_1),y(\mathbf{x}_2),\ldots,y(\mathbf{x}_n)\}$. The Kriging approach treats $\hat{y}(\mathbf{x})$ as a random function and finds the best linear unbiased predictor, $\boldsymbol{\lambda}^T(\mathbf{x})\mathbf{y}$, which minimizes the mean square error of the prediction subject to an unbiasedness constraint.

A few definitions are needed before the solution to the optimization problem subject to the constraint can be given. First, a matrix $\mathbf{F}$ is constructed by evaluating the vector of regressors, $\mathbf{f}(\mathbf{x})$, at each of the n known observations,

$$\mathbf{F}=\{\mathbf{f}(\mathbf{x}_1),\mathbf{f}(\mathbf{x}_2),\ldots,\mathbf{f}(\mathbf{x}_n)\}^T \tag{4}$$

A vector $\mathbf{r}(\mathbf{x})$ that represents the correlation between an unknown point, $\mathbf{x}\in\Omega$, and the n known sample points is defined as

$$\mathbf{r}(\mathbf{x})=\{R(\mathbf{x},\mathbf{x}_1),R(\mathbf{x},\mathbf{x}_2),\ldots,R(\mathbf{x},\mathbf{x}_n)\}^T \tag{5}$$

Given the set of observations of the computer model $\mathbf{X}$, the kth element, $k=1,\ldots,n$ of the vector function $\mathbf{r}(\mathbf{x})$ is $\mathbf{r}_k(\mathbf{x})$. The correlation matrix $\mathbf{R}$ quantifies the correlation of the observations to themselves. It a square $(n\times n)$ matrix that is symmetric about its diagonal. Its kth row (or column) is defined as $\mathbf{R}_k=\mathbf{r}(\mathbf{X}_k)$ where $\mathbf{X}_k$ is the kth observation in $\mathbf{X}$. The covariance function $\mathbf{v}(\mathbf{x})$ quantifies the covariance of an unobserved location $\mathbf{x}\in\Omega$ to the set of observations $\mathbf{X}$. It scales the correlation function $\mathbf{r}(\mathbf{x})$ with the process variance σ^2 and is defined as $\mathbf{v}(\mathbf{x})=\sigma^2\mathbf{r}(\mathbf{x})$. Finally, the covariance matrix $\mathbf{V}$, like the covariance function $\mathbf{v}(\mathbf{x})$, is a scalar multiple of the correlation matrix $\mathbf{R}$ and is defined as $\mathbf{V}=\sigma^2\mathbf{R}$.

The best linear unbiased estimator (BLUE) of $\hat{y}(\mathbf{x})$ is given by

$$\hat{y}(\mathbf{x})=\mathbf{f}^T(\mathbf{x})\hat{\boldsymbol{\beta}}+\mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1}(\mathbf{y}-\mathbf{F}\hat{\boldsymbol{\beta}}) \tag{6}$$

where the generalized least-squares estimate of $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}}=(\mathbf{F}^T\mathbf{R}^{-1}\mathbf{F})^{-1}\mathbf{F}^T\mathbf{R}^{-1}\mathbf{y} \tag{7}$$

The first component of Eq. (6) is the generalized least-squares estimate of a point, $\mathbf{x}\in\Omega$, given the correlation matrix, $\mathbf{R}$. The second component of Eq. (6) "pulls" the generalized least-squares estimate through the observed data points, providing a deterministic response surface that interpolates all of the observations. The BLUE defined in Eq. (6) assumes that the correlation parameters $\boldsymbol{\theta}$, used to define the spatial correlation of the observations, are known a priori. This is seldom the case and therefore the correlation parameters must also be estimated from the set of observations. The most common process of estimating the best model parameters, maximum likelihood estimation, is covered in Sec. 2.1.

2.1 Maximum Likelihood Estimation. Most applications of DACE [5] use the statistics-based method of MLE as an objective estimator of the regression function coefficients, process variance, and SCF parameters, $\boldsymbol{\gamma}=\{\boldsymbol{\beta},\sigma^2,\boldsymbol{\theta}\}$, that are most consistent with the observed data [19–21]. If the output distribution of the computer model comes from a Gaussian distribution, then the likelihood of the model parameters, $\boldsymbol{\gamma}$, is defined as a multivariate normal distribution of the n observations of $\mathbf{y}$ given the model parameters, $\boldsymbol{\gamma}$, and is given as

$$L(\boldsymbol{\gamma}|\mathbf{y})=p(\mathbf{y}|\boldsymbol{\gamma})=\frac{1}{\sqrt{(2\pi\sigma^2)^n|\mathbf{R}_{\boldsymbol{\theta}}|}}e^{-((\mathbf{y}-\mathbf{F}\boldsymbol{\beta})^T\mathbf{R}_{\boldsymbol{\theta}}^{-1}(\mathbf{y}-\mathbf{F}\boldsymbol{\beta}))/2\sigma^2} \tag{8}$$

where the term $\mathbf{R}_{\boldsymbol{\theta}}$ includes the subscript $\boldsymbol{\theta}$ to emphasize the fact that the correlation matrix is a function of the correlation range parameter.

The goal of the MLE method is to select the model parameters, $\boldsymbol{\gamma}$, that maximize the probability of all of the observations. The multivariate normal likelihood function can be difficult to maximize due to potentially large flat regions of near zero values. To improve the optimization process, the logarithm of the likelihood function is taken, since the maximum of the logarithm of the likelihood occurs at the same location as the maximum of the likelihood. The logarithm of the multivariate Gaussian likelihood function is

$$\ell(\boldsymbol{\gamma}|\mathbf{y})=-\frac{n}{2}\ln[2\pi\sigma^2]-\frac{1}{2}\ln[|\mathbf{R}_{\boldsymbol{\theta}}|]-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{F}\boldsymbol{\beta})^T\mathbf{R}_{\boldsymbol{\theta}}^{-1}(\mathbf{y}-\mathbf{F}\boldsymbol{\beta}) \tag{9}$$

The closed-form solution for the optimal value of $\boldsymbol{\beta}$ is the same as Eq. (7) found by solving for the least-squares error. The MLE solution for the process variance σ^2 is

$$\hat{\sigma}^2=\frac{1}{n}(\mathbf{y}-\mathbf{F}\hat{\boldsymbol{\beta}})^T\mathbf{R}_{\boldsymbol{\theta}}^{-1}(\mathbf{y}-\mathbf{F}\hat{\boldsymbol{\beta}}) \tag{10}$$

A closed-form solution does not exist for the optimal parameters of most common SCFs, thus requiring numerical optimization. In order to reduce the number of model parameters determined with the numerical optimization, the known optimal values of $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$ from Eqs. (7) and (10) are substituted into Eq. (9), resulting in the profile log-likelihood equation (Eq. (11)) [21]. The profile log-likelihood is then optimized for the unknown correlation parameters, $\boldsymbol{\theta}$.

$$\text{Maximize}\quad -\frac{n}{2}\ln[2\pi\hat{\sigma}^2]-\frac{1}{2}\ln[|\mathbf{R}_{\boldsymbol{\theta}}|]-\frac{n}{2}$$
$$\text{subject to}\quad \boldsymbol{\theta}>0 \tag{11}$$

Even with the reduction in the dimension of the optimization problem that results from using the profile log-likelihood function, the optimization process can still be a computationally expensive process. The computational expense can be attributed primarily to two calculations: the evaluation of the exp function n^2 times during the evaluation of the correlation matrix and the inversion of the correlation matrix. In addition to the potential computational expense associated with each iteration of an optimization algorithm, the optimization of the log-likelihood equation can have several numerical difficulties. The three most prevalent issues are (1) ill-conditioned correlation matrices, (2) multiple local optimum, and (3) long ridges of near optimal values [22]. The remainder of this paper presents optimization algorithm developments to alleviate some of the computational expense and to deal with the numerical difficulties.

2.2 Optimization Algorithms. There are two primary issues concerned with MLE optimization: (1) computational expense and (2) multiple local optima. This increased computational expense is the result of calculating both a gradient and a Hessian at each iteration of the optimization algorithm that increases significantly as more parameters (dimensions) are added to the Kriging model. Without an analytical representation of the gradient and Hessian of the log-likelihood equation, a finite difference method or quasi-Newton method is required. At each step of the optimization process, the evaluation and inversion of the $n\times n$ correlation matrix are performed at least $d+1$ times, depending on the specific algorithm used [23]. A generalize reduced gradient (GRG) method has also been used to reduce the number of active dimensions in the problem [24]. The second issue of the log-likelihood equation

having multiple local maxima may result in the possible failure of a gradient-based method to determine the global maximum. To resolve both of these issues, it has become common to use a non-gradient-based optimization algorithm such as simulated annealing [11,25] or a modified Hooke and Jeeves simplex algorithm [10]. These options may not provide the best algorithm since they: (1) still require the evaluation and inversion of the correlation matrix at each iteration, (2) may require a significantly larger number of iterations to converge than a gradient-based method, and (3) require the user understand the impact of the tuning parameters associated with the optimization algorithm.

The modified Hooke and Jeeves algorithm used in the DACE toolkit for MATLAB is not concerned with convergence to the exact answer that maximizes the likelihood equation. The approach taken is to perform a grid search to determine the best starting location. The algorithm then performs between 2 and 4 simplex iterations, returning the best answer. This effectively limits the maximum execution time. The belief is that the exact determination of the correlation parameters is not very critical or sensitive to the accuracy of the resulting Kriging model [10]. This belief is often true with respect to the creation of an approximating function but it does not hold true if using the Hessian of the log-likelihood function (the observed information matrix [26]) to determine the standard errors of the model parameters.

2.2.1 Newton–Raphson Method. A commonly used gradient-based method is the Newton–Raphson method. In the Newton–Raphson method, a quadratic surface

$$f(\boldsymbol{\theta}) = \mathbf{a} + \mathbf{b}^T\boldsymbol{\theta} - \frac{1}{2}\boldsymbol{\theta}^T\mathbf{C}\boldsymbol{\theta} \tag{12}$$

is fit to the log-likelihood function at the current model parameter values. The maximum value of this quadratic surface is given analytically as $\boldsymbol{\theta}=\mathbf{C}^{-1}\mathbf{b}$, where $-\mathbf{C}$ is the Hessian of the log-likelihood function and the gradient is equal to $\mathbf{b}-\mathbf{C}\boldsymbol{\theta}$. If the log-likelihood function were quadratic, this method would solve the optimization problem in a single step from any starting value.

In general, the log-likelihood function to be maximized is not quadratic, and as a result an iterative technique is needed to solve the maximization problem. Given the current values of the model parameters $\boldsymbol{\theta}^{(k)}$ and the approximated quadratic function at the current values, the next step $k+1$ is defined as

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - (\mathbf{H}^{(k)})^{-1}\nabla f^{(k)} \tag{13}$$

where $\mathbf{H}^{(k)}$ and $\nabla f^{(k)}$ refer to the Hessian and the gradient evaluated at the current model parameter values $\boldsymbol{\theta}^{(k)}$. The primary drawback to this method is the need to evaluate both the gradient and the Hessian of the log-likelihood function at each iteration of the algorithm, with the Hessian generally being the most computationally expensive to evaluate.

The Levenberg–Marquardt method [27] provides a heuristic to transition from the steepest ascent method to the Newton–Raphson method. In general, a steepest ascent algorithm will converge to the optimal value more rapidly when the starting value is far from the optimal value and a Newton–Raphson method will converge faster as the current value of the iteration closes in on the optimal value [28]. The iteration of Eq. (13) can be altered as

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - (\mathbf{H}^{(k)} + \tau^{(k)}\mathbf{I})^{-1}\nabla f^{(k)} \tag{14}$$

where $\tau^{(k)}$ as a scalar that indirectly determines the step size and $\mathbf{I}$ is the identity matrix. For small values of $\tau^{(k)}$, the method approximates the Newton–Raphson method, and for large values, a small step is taken in nearly the direction of steepest descent (ascent).

2.2.2 Scoring Method. The SCORING method replaces the observed information matrix with the expected information matrix [29]. The expected information matrix is defined as the expected value of the Hessian matrix. The inverse of the expected information matrix provides the asymptotic estimate of the covariance of the model parameters, a fact that is useful when comparing the relative importance of the different model parameters [30]. The evaluation of the expected value of the Hessian can be simplified by the relationship (for a proof see the Appendix of Searle et al. [31])

$$E\left(\frac{\partial^2\ell}{\partial\boldsymbol{\theta}\,\partial\,\boldsymbol{\theta}^T}\right) = -E\left(\frac{\partial\ell}{\partial\boldsymbol{\theta}}\frac{\partial\ell}{\partial\boldsymbol{\theta}^T}\right) \tag{15}$$

The result of this relationship is that the Hessian can be estimated by taking the expected value of the outer product of the gradient. Additional details on how to calculate the expected value of the Hessian are given in the Appendix.

3 Optimization Algorithm

This section provides an outline of the steps required to calculate the maximum value of the log-likelihood function. Details are provided on the specific aspects of each step to enable the easy implementation of the optimization algorithms.

1. *Scale inputs*. Scale all inputs to be within a d-dimensional hypercube. Linear scaling is often sufficient, but nonlinear scaling can also be used if it performs better for the specific data set.
2. *Generate starting value*. For a d-dimensional model, generate d random draws Z from $U[0,1]$ and, assuming a minimum correlation range of 0.01, the following equation is used to generate the starting values that are more heavily weighted toward smaller values for each correlation parameter θ

$$\theta = 10^{Z\log_{10}(\max\ \mathrm{Corr}\times 100)-2} \tag{16}$$

   where maxCorr refers to the maximum correlation constraint. If the starting values result in a correlation matrix that is ill conditioned, they are thrown out and a new random sample is taken.
3. *Calculate gradient and Hessian*. Calculate the gradient of the log-likelihood function using the analytical form given in the Appendix at the current values of the correlation parameters. The Hessian can then either be calculated explicitly or estimated using Eq. (15).
4. *Calculate next iteration*. Using Eq. (14) calculate the next iteration. The Levenburg–Marquardt parameter is initialized as $\tau^{(k)}=0.001$.
   (a) *Test for constraint violation*. Test the next iteration to see if it violates the minimum and maximum allowed values for the correlation parameters. If it does, then move to the constraint.
   The move must also be tested for a resulting ill-conditioned correlation matrix. If the current iteration results in an ill-conditioned matrix, then reduce the step size until the current iteration results in a well conditioned correlation matrix.
   (b) *Test step length*. If the resulting step length after the previous constraint test is very small (e.g., 1×10^{-8}), then start a counter indicating the optimizer is stuck. If algorithm is stuck for four consecutive steps, determine if the current value results in the best log-likelihood observed and start the algorithm over again with a new starting seed in Step 1.
5. *Calculate new model parameters*. Using the current correlation parameters, use Eqs. (7) and (10) to calculate the best trend function coefficients and process variance. These values are plugged into the equation for the profile log-likelihood (Eq. (11)).
6. *Change LM parameter*. Test to determine if the current log-likelihood value is better than the previous value in the current iteration. If it is better, reduce the Levenberg–Marquardt

Table 1 Comparison of model parameters values as determined with gradient-based and DACE toolkit methods

Parameter	Method	2D-1	2D-2
x1	DACE toolkit	0.532	0.447
x1	Gradient-based	0.137	0.069
x1	Std deviation	0.018	0.014
x2	DACE toolkit	1.160	1.064
x2	Gradient-based	1.045	2.729
x2	Std deviation	0.286	3.944

parameter, $\tau^{(k)}$, by a factor of 10; if it is worse increase it by a factor of 10.

7. *Test result of current iteration*. Evaluate the gradient at the current value, if it is close to zero (e.g., the root mean square of each element of the gradient is less than 0.01) for four consecutive iterations, then the algorithm is considered to have converged. Additionally, stop the algorithm if more than 100 iterations are needed to converge, limiting the time needed. If the algorithm has converged, move to the next step, otherwise return to Step 3 and calculate the gradient and Hessian at the current iteration.
8. *Test result for global optimum*. Take the best result from the current starting values and determine if it is the best overall result. Continue the algorithm with new starting values in Step 2 until the same optimal values are returned four times (not necessarily consecutively). The overall algorithm will also stop once 100 valid starting points have been optimized. They are not guaranteed to be the global optimum, but if they are not, then there are most likely other problems with the data set (e.g., under-/oversampling of the domain or the data being poorly estimated by a stationary Gaussian process).

4 Results

The use of the analytical gradient and Hessian for the Kriging model are compared for their computational expense. Three variations on the calculation of the Hessian used in the Newton–Raphson optimization method (see Sec. 2.2) are compared based on the results of fitting a Kriging model to three different sets of data. The test examples include two two-dimensional data set from a mathematical function. The third data set has five dimensions and is from the design of a hydroreactive solid fuel gas generator. The comparisons include the average number of iterations and required to converge to a local optimum, the average number of restarts required to converge to a global optimum, and the total average iterations required to converge to a global optimum. The comparisons will also include an estimate of the computational expense associated with estimating the Hessian used in the three variations.

The first two-dimensional test function was introduced by Osio and Amon [24]. This function is linear in one dimension and highly nonlinear in the second dimension. It is approximated using 21 observations taken from an optimized Latin hypercube design [14]. The second two-dimensional test function is the six-hump camelback function [7]. This function is multimodal and is difficult to fit with a traditional regression function. The five-dimensional test function is the analysis results of a gas generation system [32].

A comparison of the resulting model parameters as determined by the three gradient-based methods presented here and the modified Hooke and Jeeves method used in the DACE toolkit for MATLAB is shown in Table 1. A constant trend model was used for both two-dimensional problems compared. The table includes the standard deviation of the model parameter as determined from the observed information matrix at the MLE point (the parameter values as determined by the gradient-based methods proposed in this work). Though the probability distributions of the correlation parameters are not well represented by a Gaussian distribution [33], the fact that the x1 parameter 2D-1 and 2D-2 example problems vary by more than ten standard deviations suggest that the two methods do not return equivalent results.

Table 2 Comparison of optimization method options for 2D example 1

Metric	Full Hess.	Exp. Hess.	SCORING
Iters. to conv.	28.4	41.7	17.0
Starts to opt.	10.70	6.59	7.89
Total iters.	305.2	278.2	135.1
Time/iter. (ms)	12.10	5.40	4.58
Opt. time (ms)	3690	1500	618

The three proposed methods of calculating or estimating the Hessian as part of a Newton–Raphson optimization of the log-likelihood equation are compared in this section. The methods are compared based on (1) the average number of iterations required to converge to an answer, (2) the average number of starts required to converge to the optimal answer four times, (3) the average number of iterations to determine the optimal answer, (4) the average CPU time per iteration, and (5) the average total CPU time to determine the optimal answer. The results were generated by executing the algorithms enough times to record more than 500 starts of the algorithms and are shown in Tables 2–4. The experiments were run on a 1.3 MHz Pentium M with 1 Gbyte of RAM.

The results of this comparison of three examples give relatively consistent results: the simplified SCORING method appears to perform the best. In all of the cases on average, it converges to an optimal solution in the fewest number of iterations (the first row in the tables). The expected Hessian case tends to converge to the correct optimum with the fewest starts, but it takes more iterations to converge to an answer in both 2D examples. In the 5D example, the expected Hessian method converged faster than the simplified SCORING method. The minimum number of samples possible in each start is 4 and the maximum limit of 101. The maximum limit may tend to bias the results slightly to be lower than they should be. Without using a stopping criterion for the maximum limit of the iterations of 101, it was possible for the Newton–Raphson algorithms to continue without converging in some instances. The full Hessian for the 5D example never converged to the same answer within the maximum limit of 101 iterations.

Table 3 Comparison of optimization method options for 2D example 2

Metric	Full Hess.	Exp. Hess.	SCORING
Iters. to conv.	30.3	24.0	15.2
Starts to opt.	9.88	6.96	7.17
Total iters.	300.6	168.3	108.8
Time/iter. (ms)	11.81	6.73	5.09
Opt. time (ms)	3550	1130	554

Table 4 Comparison of optimization method options for 5D example

Metric	Full Hess.	Exp. Hess.	SCORING
Iters. to conv.	63.8	59.3	53.7
Starts to opt.	n/a	19.30	44.5
Total iters.	n/a	1140	2390
Time/iter. (ms)	194	74.7	27.3
Opt. time (s)	n/a	85.2	65.3

One possible explanation for the observed difference between the full Hessian and expected Hessian results lies in the expected Hessian being a negative semidefinite matrix over the entire range of correlation parameters. The actual Hessian may not be negative semidefinite at all locations. This can cause instability in the Newton–Raphson algorithm. As a result, even though the parameter values are constrained to be within valid ranges, the resulting parameter values may still be located on a constraint that has a very little gradient in other directions, causing the algorithm to stall. The positive results of going to a constraint very quickly is that the algorithm will stop after typically four or five iterations and move on to another starting value.

The results of the difference between the expected Hessian and the simplified Scoring methods may be explained by the amount of correlation that exists between the range parameters. If the range parameters are relatively independent of each other, then the expected Hessian matrix will have near zero values on its off-diagonal terms. The result is that the expected Hessian matrix is well approximated by the simplified SCORING method. This appears to be the case for both 2D examples but is not the case for the 5D example.

The execution times tell a similar story as the number of iterations. The average time per iteration decreases from the full Hessian to the expected Hessian to the simplified SCORING methods as expected. As more dimensions are included, this difference becomes more pronounced. For both 2D examples, this result amplifies the result that the simplified SCORING method appears to determine the global optimum the fastest of the three methods. For the 5D example, the improved speed of execution per iteration resulted in the average time to determine the global optimum to be the lowest for the simplified SCORING method even though it must perform more iterations in the process (2390 versus 1140). The execution times for these example problems are relatively small compared with the expense of generating the observations in the first place. This short execution time is a result of the relatively small number of observations and dimensions used in the model. The execution time will increase exponentially as observations and dimensions increase.

5 Conclusions

This paper presents an analytical form of the gradient and Hessian for the log-likelihood equation of a Kriging model that uses a Gaussian spatial correlation function. These analytical forms are demonstrated for the computational efficiency and effectiveness on three example problems using three variations of the Newton–Raphson optimization method. It was found that a simplified SCORING method was the most computationally efficient method to determine the global optimum of the log-likelihood function. These results can be applied to both simple, ordinary, and universal Kriging (no trend model, constant trend model, and multiple component trend model).

These results do not leave the calculation of the full Hessian without a purpose in the creation of Kriging models. The full Hessian can be used to estimate the covariance matrix of the model parameters, the observed Fisher information matrix. From this, the uncertainty (variance) in the estimate of the model parameters can be determined, their p-values calculated, and a t-test performed to determine if a model parameter should be included in the model. In general, this test would not be used to remove correlation range parameters, but it can be used to remove trend function parameters [30].

Future work should explore developing a heuristic to switch from the simplified SCORING method to the expected or full Hessian as the Newton–Raphson method converges. Once the algorithm approaches the optimum, the use of the full Hessian should in a better approximation of the local behavior of the log-likelihood function.

Acknowledgment

The author would like to thank Dr. Kam Ng of ONR 333 for his generous support on Contract No. N00014-05-G-0106.

Appendix: Derivation of Gradient and Hessian

This section provides the derivation of the analytical first and second partial derivatives of the log-likelihood equation when the Gaussian function is used as the spatial correlation function. This form provides a closed-form analytical solution for the gradient and Hessian that does not require additional evaluations of the spatial correlation function and the inversion of the correlation matrix other than at the current values of the correlation parameters. These forms are exact rather than the finite difference approximations used in the past and are much more computationally efficient to evaluate since they do not require the calculation and inversion of many different correlation matrices $\mathbf{R}$.

1 First Partial Derivatives. If the model parameters $\boldsymbol{\gamma}$ are divided into the coefficients of the regressors $\boldsymbol{\beta}$ and the parameters $\boldsymbol{\theta}$ that define the covariance matrix, then the vector of the first partial derivatives of the log-likelihood function ℓ, with respect to $\boldsymbol{\gamma}$, can be written as $\ell^{(1)}=\{\ell_\beta^T, \ell_\theta^T\}^T$, which is also known as the *score* function. The partial derivative of the log-likelihood function with respect to the regressor coefficients is

$$\ell_\beta = -\mathbf{F}^T\mathbf{V}^{-1}\mathbf{F}\boldsymbol{\beta} + \mathbf{F}^T\mathbf{V}^{-1}\mathbf{y} \tag{A1}$$

and the ith element of ℓ_θ is

$$(\ell_\theta)_i = -\tfrac{1}{2}\mathrm{tr}(\mathbf{V}^{-1}\mathbf{V}_i) - \tfrac{1}{2}\mathbf{W}^T\mathbf{V}^i\mathbf{W} \tag{A2}$$

where the inverse of the covariance matrix is $\mathbf{V}^{-1}=\sigma^{-2}\mathbf{R}^{-1}$, $\mathbf{V}_i = \partial\mathbf{V}/\partial\theta_i$ is a matrix that represents the partial derivative of the covariance matrix with respect to the covariance parameter θ_i, $\mathbf{V}^i=\partial\mathbf{V}^{-1}/\partial\theta_i=-\mathbf{V}^{-1}\mathbf{V}_i\mathbf{V}^{-1}$ is the inverse of the partial derivative of the covariance matrix for each covariance parameter, and $\mathbf{W} = \mathbf{y}-\mathbf{F}\boldsymbol{\beta}$ is the vector of residuals. The results presented in Eqs. (A1) and (A2) were originally presented by Mardia and Marshall [19].

The partial derivative of the covariance matrix $\mathbf{V}$ with respect to the process variance σ^2 is the correlation matrix $\mathbf{R}$.

$$\frac{\partial\mathbf{V}}{\partial\sigma^2} = \mathbf{R} \tag{A3}$$

The partial derivatives of the covariance matrix $\mathbf{V}$ with respect to the correlation parameters are determined by evaluating the partial derivative of the covariance function at each of the k observations in $\mathbf{X}$. The partial derivative of the kth row (or column) of the covariance function is

$$\frac{\partial\mathbf{V}_k}{\partial\theta_i} = \frac{\partial\mathbf{v}(\mathbf{X}_k)}{\partial\theta_i} \tag{A4}$$

The resulting partial derivative matrix with respect to each correlation parameters is positive semidefinite and symmetric about the diagonal, just like the correlation matrix $\mathbf{R}$. Unlike the correlation matrix, the diagonal of the partial derivative of the covariance matrix consists of zeros since, by inspection of Eq. (A6), the derivative of the variance of an observations is zero.

The partial derivative of the covariance function with respect to the process variance σ^2 is the correlation function,

$$\frac{\partial\mathbf{v}(\mathbf{x})}{\partial\sigma^2} = \mathbf{r}(\mathbf{x}) \tag{A5}$$

The results shown in Eqs. (A3)–(A5) are correct for any spatial correlation function. The $i=1,\ldots,d$ partial derivatives of the Gaussian correlation (covariance) function with respect to the correlation parameters are defined as

$$\frac{\partial \mathbf{v}(\mathbf{x})}{\partial \theta_i} = \frac{2\sigma^2 \mathbf{r}(\mathbf{x}) \odot (\mathbf{X}_i - \mathbf{x}_i)^2}{\theta_i^3} \tag{A6}$$

where $\mathbf{X}_i$ is the ith column of the matrix of inputs $\mathbf{X}$, $\mathbf{x}_i$ is the ith dimension of an unobserved point in the domain Ω, and $\odot$ is an operator indicating the elementwise multiplication of the two vectors. The resulting partial derivatives of the covariance parameters are n-element vectors corresponding to each of the observations.

2 Second Partial Derivatives. The second derivative matrix or Hessian of the log-likelihood function can be partitioned into four submatrices [19] as

$$\ell^{(2)} = \begin{bmatrix} \ell_{\beta\beta} & \ell_{\beta\theta} \\ \ell_{\beta\theta}^T & \ell_{\theta\theta} \end{bmatrix} \tag{A7}$$

where $\ell_{\beta\beta} = \mathbf{F}^T\mathbf{V}^{-1}\mathbf{F}$, $\ell_{\beta\theta}$ has ith column $-\mathbf{F}^T\mathbf{V}^i\mathbf{F}\boldsymbol{\beta} + \mathbf{F}^T\mathbf{V}^i\mathbf{y}$, and $\ell_{\theta\theta}$ has (i,j)th term:

$$-\frac{1}{2}(\mathrm{tr}(\mathbf{V}^{-1}\mathbf{V}_{ij} + \mathbf{V}^i\mathbf{V}_j) + \mathbf{W}^T\mathbf{V}^{ij}\mathbf{W}) \tag{A8}$$

where $\mathbf{V}_{ij} = \partial^2\mathbf{V}/\partial\theta_i\partial\theta_j$, and

$$\mathbf{V}^{ij} = \frac{\partial^2\mathbf{V}^{-1}}{\partial\theta_i\,\partial\theta_j} = \mathbf{V}^{-1}(\mathbf{V}_i\mathbf{V}^{-1}\mathbf{V}_j + \mathbf{V}_j\mathbf{V}^{-1}\mathbf{V}_i - \mathbf{V}_{ij})\mathbf{V}^{-1} \tag{A9}$$

There are four cases that must be considered when determining the second partial derivatives of the covariance function $\mathbf{V}_{ij}$ analytically. The first two cases are independent of the specific spatial correlation function chosen. The first case is $\partial\sigma^2\partial\sigma^2$. This case is trivial since the first partial derivative (see Eq. (A5)) is not a function of the process variance σ^2. Therefore, the following results:

$$\frac{\partial^2\mathbf{v}(\mathbf{x})}{\partial\sigma^2\,\partial\sigma^2} = \mathbf{0} \tag{A10}$$

where, in this equation, $\mathbf{0}$ is a vector of n zeros.

As a result of Eq. (A10), the second derivative of the covariance matrix with respect to $\partial\sigma^2\partial\sigma^2$ is an $(n\times n)$ matrix of zeros.

$$\frac{\partial^2\mathbf{V}}{\partial\sigma^2\,\partial\sigma^2} = \mathbf{0} \tag{A11}$$

The second case of interest is $\partial\sigma^2\partial\theta_i$. Since differentiation is a linear operation, the following equivalency of expressions of the derivative can be made:

$$\frac{\partial^2\mathbf{v}(\mathbf{x})}{\partial\sigma^2\,\partial\theta_i} = \frac{\partial^2\mathbf{v}(\mathbf{x})}{\partial\theta_i\,\partial\sigma^2} = \frac{\partial\mathbf{v}(\mathbf{x})}{\partial\theta_i}\frac{1}{\sigma^2} \tag{A12}$$

which results from differentiating the results of Eq. (A6) with respect to the process variance σ^2. The corresponding second partial derivative of the kth row (or column) of the covariance matrix is given as

$$\frac{\partial^2\mathbf{V}_k}{\partial\sigma^2\,\partial\theta_i} = \frac{\partial^2\mathbf{V}_k}{\partial\theta_i\,\partial\sigma^2} = \frac{\partial^2\mathbf{v}(\mathbf{X}_k)}{\partial\sigma^2\,\partial\theta_i} \tag{A13}$$

The third case of interest is $\partial\theta_i\partial\theta_j$, where $i \neq j$. This case and the last case are a result of choosing a Gaussian spatial correlation function; other correlation functions would have a different result. The partial derivative of Eq. (A6) with respect to a different correlation range parameter is given as

$$\frac{\partial^2\mathbf{v}(\mathbf{x})}{\partial\theta_i\,\partial\theta_j} = \frac{\partial\mathbf{v}(\mathbf{x})}{\partial\theta_i} \odot \frac{2(\mathbf{X}_j - \mathbf{x}_j)^2}{\theta_j^3} \tag{A14}$$

Following the same pattern as the previous two cases, the partial derivative of the kth row (or column) of the covariance matrix is given as

$$\frac{\partial^2\mathbf{V}_k}{\partial\theta_i\,\partial\theta_j} = \frac{\partial^2\mathbf{v}(\mathbf{X}_k)}{\partial\theta_i\,\partial\theta_j} \tag{A15}$$

The last case of second partial derivatives of interest in this study is $\partial\theta_i\partial\theta_i$. The partial derivative of Eq. (A6) with respect to the same correlation parameter is given as

$$\frac{\partial^2\mathbf{v}(\mathbf{x})}{\partial\theta_i\,\partial\theta_i} = \frac{\partial\mathbf{v}(\mathbf{x})}{\partial\theta_i} \odot \left(\frac{2(\mathbf{X}_i - \mathbf{x}_i)^2}{\theta_i^3} - \frac{\mathbf{3}}{\theta_i}\right) \tag{A16}$$

where $\mathbf{3}$ is a vector of n 3s. The second partial derivative of the kth row (or column) of the covariance matrix is given as

$$\frac{\partial^2\mathbf{V}_k}{\partial\theta_i\,\partial\theta_i} = \frac{\partial^2\mathbf{v}(\mathbf{X}_k)}{\partial\theta_i\,\partial\theta_i} \tag{A17}$$

3 Expected Value of Hessian. The expected value of the Hessian can be determined by using $E(\mathbf{y}) = \mathbf{F}\boldsymbol{\beta}$ and hence $E(\mathbf{y} - \mathbf{F}\boldsymbol{\beta}) = E(\mathbf{W}) = \mathbf{0}$ and the relationship from Eq. (15). The resulting expected value of the Hessian

$$E\begin{bmatrix} \ell_{\beta\beta} & \ell_{\beta\theta} \\ \ell_{\beta\theta}^T & \ell_{\theta\theta} \end{bmatrix} = -\begin{bmatrix} \mathbf{F}^T\mathbf{V}^{-1}\mathbf{F} & \mathbf{0} \\ \mathbf{0} & \frac{1}{2}\mathrm{tr}(\mathbf{V}^{-1}\mathbf{V}_i\mathbf{V}^{-1}\mathbf{V}_j)_{i,j=0}^d \end{bmatrix} \tag{A18}$$

depends only on the covariance matrix and its first partial derivatives with respect to the covariance parameters. The expected value of the Hessian matrix is much more computationally efficient to calculate than the full Hessian. When the Kriging model is able to correctly specify the distribution of the observations, i.e., the residuals are well represented with a zero mean Gaussian shape, the full Hessian evaluated at the MLE parameter values, and the expected Hessian matrices should be nearly identical [34]. The degree to which the residuals are well represented with a Gaussian distribution can be estimated by $\ell_{\beta\theta}$ term of Eq. (A7) being nearly equal to $\mathbf{0}$.

References

[1] Wang, G. G., and Shan, S., 2007, "Review of Metamodeling Techniques in Support of Engineering Design Optimization," ASME J. Mech. Des., **129**, pp. 370–380.

[2] Qian, Z., Seepersad, C. C., Joseph, V. R., Allen, J. K., and Wu, C. F. J., 2006, "Building Surrogate Models Based on Detailed and Approximate Simulations," ASME J. Mech. Des., **128**, pp. 668–677.

[3] Yang, R. J., Wang, N., Tho, C. H., Bobineau, J. P., and Wang, B. P., 2005, "Metamodeling Development for Vehicle Frontal Impact Simulation," ASME J. Mech. Des., **127**(5), pp. 1014–1020.

[4] Pacheco, J. E., Amon, C. H., and Finger, S., 2003, "Bayesian Surrogates Applied to Conceptual Stages of the Engineering Design Process," ASME J. Mech. Des., **125**(4), pp. 664–672.

[5] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., 1989, "Design and Analysis of Computer Experiments," Stat. Sci., **4**(4), pp. 409–435.

[6] Sasena, M. J., Parkinson, M., Reed, M. P., Papalambros, P. Y., and Goovaerts, P., 2005, "Improving an Ergonomic Testing Procedure Via Approximation-Based Adaptive Experimental Design," ASME J. Mech. Des., **127**, pp. 1006–1013.

[7] Jin, R., Chen, W., and Sudjianto, A., 2002, "On Sequential Sampling for Global Metamodeling in Engineering Design," ASME Paper No. DETC2002/DAC-34092.

[8] Jin, R., Chen, W., and Sudjianto, A., 2005, "An Efficient Algorithm for Constructing Optimal Design of Computer Experiments," J. Stat. Plan. Infer., **134**, pp. 268–287.

[9] Chen, W., Jin, R., and Sudjianto, A., 2005, "Analytical Variance-Based Global Sensitivity Analysis in Simulation-Based Design Under Uncertainty," ASME J. Mech. Des., **127**(5), pp. 875–886.

[10] Lophaven, S. N., Nielsen, B. H., and Sondergaard, J., 2002, "DACE—A Matlab Kriging Toolbox, Version 2.0," Technical University of Denmark, Report No. IMM-REP-2002-12.

[11] Simpson, T. W., Maurey, T. M., Korte, J. J., and Mistree, F., 2001, "Kriging Metamodels for Global Approximation in Simulation-Based Multidisciplinary Design Optimization," AIAA J., **39**(12), pp. 2233–2241.

[12] Goovaerts, P., 1997, *Geostatistics for Natural Resources Evaluation* (Applied Geostatistics Series), Oxford University Press, New York.

[13] Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J., and Morris, M. D., 1992, "Screening, Predicting, and Computer Experiments," Technometrics, **34**(1), pp. 15–25.

[14] Martin, J. D., and Simpson, T. W., 2005, "On the Use of Kriging Models to Approximate Deterministic Computer Models," AIAA J., **43**(4), pp. 853–863.
[15] Joseph, V. R., Hung, Y., and Sudjianto, A., 2008, "Blind Kriging: A New Method for Developing Metamodels," ASME J. Mech. Des., **130**(3), p. 031102.
[16] Simpson, T. W., Poplinski, J. D., Koch, P. N., and Allen, J. K., 2001, "Metamodels for Computer-Based Engineering Design: Survey and Recommendations," Eng. Comput., **17**(2), pp. 129–150.
[17] Booker, A. J., Conn, A. R., Dennis, J. E., Jr., Frank, P. D., Trosset, M., and Torczon, V., 1995, "Global Modeling for Optimization: Boeing/IBM/Rice Collaborative Project 1995 Final Report," The Boeing Company, Report No. ISSTECH-95-032.
[18] Currin, C., Mitchell, T. J., Morris, M. D., and Ylvisaker, D., 1991, "Bayesian Prediction of Deterministic Functions, With Applications to the Design and Analysis of Computer Experiments," J. Am. Stat. Assoc., **86**(416), pp. 953–963.
[19] Mardia, K., and Marshall, R., 1984, "Maximum Likelihood Estimation of Models for Residual Covariance in Spatial Regression," Biometrika, **71**(1), pp. 135–146.
[20] Kitanidis, P. K., 1986, "Parameter Uncertainty in Estimation of Spatial Functions: Bayesian Analysis," Water Resour. Res., **22**, pp. 499–507.
[21] Mardia, K., and Watkins, A. J., 1989, "On Multimodality of the Likelihood in the Spatial Linear Model," Biometrika, **76**(2), pp. 289–295.
[22] Warnes, J. J., and Ripley, B. D., 1987, "Problems With Likelihood Estimation of Covariance Function of Spatial Gaussian Processes," Biometrika, **74**(3), pp. 640–642.
[23] Sacks, J., Schiller, S. B., and Welch, W. J., 1989, "Design for Computer Experiments," Technometrics, **31**(1), pp. 41–47.
[24] Osio, I. G., and Amon, C. H., 1996, "An Engineering Design Methodology With Multistage Bayesian Surrogate and Optimal Sampling," Res. Eng. Des., **8**(4), pp. 189–206.
[25] Booker, A. J., Dennis, J. E., Jr., Frank, P. D., Serafini, D. B., Torczon, V., and Trosset, M., 1999, "A Rigorous Framework for Optimization of Expensive Functions by Surrogates," Struct. Optim., **17**(1), pp. 1–13.
[26] Efron, B., and Hinkley, D., 1978, "Assessing the Accuracy of the Maximum Likelihood Estimator: Observed Versus Expected Fisher Information," Biometrika, **65**(3), pp. 457–482.
[27] Marquardt, D. W., 1963, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," J. Soc. Ind. Appl. Math., **11**(2), pp. 431–441.
[28] Hemmerle, W. J., and Hartley, H. O., 1973, "Computing Maximum Likelihood Estimates for the Mixed A. O. V. Model Using the W Transform," Technometrics, **15**(4), pp. 819–831.
[29] Jennrich, R. I., and Sampson, P. F., 1976, "Newton–Raphson and Related Algorithms for Maximum Likelihood Variance Component Estimation," Technometrics, **18**(1), pp. 11–17.
[30] Martin, J. D., and Simpson, T. W., 2006, "A Methodology to Manage Uncertainty During System-Level Conceptual Design," ASME J. Mech. Des., **128**(4), pp. 959–968.
[31] Searle, S. R., Casella, G., and McCulloch, C. E., 1992, *Variance Components*, Wiley, New York.
[32] Martin, J. D., and Simpson, T. W., 2002, "Use of Adaptive Metamodeling for Design Optimization," AIAA Paper No. AIAA-2002-5631.
[33] Martin, J. D., and Simpson, T. W., 2004, "A Monte Carlo Simulation of the Kriging Model," AIAA Paper No. AIAA-2004-4483.
[34] White, H., 1982, "Maximum Likelihood Estimation of Misspecified Models," Econometrica, **50**(1), pp. 1–26.