Chapter 6

Surrogate Models

This chapter deals with the theory behind Dakota's surrogate models, which are also known as response surfaces and meta-models.

6.1 Kriging and Gaussian Process Models

In this discussion of Kriging and Gaussian Process (GP) models, vectors are indicated by a single underline and matrices are indicated by a double underline. Capital letters indicate data, or functions of data, that is used to construct an emulator. Lower case letters indicate arbitrary points, i.e. points where the simulator may or may not have been evaluated, and functions of arbitrary points. Estimates, approximations, and models are indicated by hats. For instance, $\hat{f}\left(\underline{x}\right)$ is a model/emulator of the function $f\left(\underline{x}\right)$ and $\hat{y}$ is the emulator's prediction or estimate of the true response $y = f(\underline{x})$ evaluated at the point $\underline{x}$. A tilde indicates a reordering of points/equations, with the possible omission of some but not all points. N is the number of points in the sample design and M is the number of input dimensions.

6.1.1 Kriging & Gaussian Processes: Function Values Only

The set of interpolation techniques known as Kriging, also referred to as Gaussian Processes, were originally developed in the geostatistics and spatial statistics communities to produce maps of underground geologic deposits based on a set of widely and irregularly spaced borehole sites[28]. Building a Kriging model typically involves the

1. Choice of a trend function,
2. Choice of a correlation function, and
3. Estimation of correlation parameters.

A Kriging emulator, $\hat{f}\left(\underline{x}\right)$, consists of a trend function (frequently a least squares fit to the data, $\underline{g}\left(\underline{x}\right)^T \underline{\beta}$) plus a Gaussian process error model, $\epsilon\left(\underline{x}\right)$, that is used to correct the trend function.

$$\hat{f}\left(\underline{x}\right) = \underline{g}\left(\underline{x}\right)^T \underline{\beta} + \epsilon\left(\underline{x}\right)$$

This represents an estimated distribution for the unknown true surface, $f\left(\underline{x}\right)$. The error model, $\epsilon\left(\underline{x}\right)$, makes an adjustment to the trend function so that the emulator will interpolate, and have zero uncertainty at, the data points it was built from. The covariance between the error at two arbitrary points, $\underline{x}$ and $\underline{x'}$, is modeled as

$$\text{Cov}\left(y\left(\underline{x}\right), y\left(\underline{x'}\right)\right) = \text{Cov}\left(\epsilon\left(\underline{x}\right), \epsilon\left(\underline{x'}\right)\right) = \sigma^2\ r\left(\underline{x}, \underline{x'}\right).$$

Here σ^2 is known as the unadjusted variance and $r\left(\underline{x}, \underline{x'}\right)$ is a correlation function. Measurement error can be modeled explicitly by modifying this to

$$\text{Cov}\left(\epsilon\left(\underline{x}\right), \epsilon\left(\underline{x'}\right)\right) = \sigma^2\ r\left(\underline{x}, \underline{x'}\right) + \Delta^2 \delta\left(\underline{x} - \underline{x}'\right)$$

where

$$\delta\left(\underline{x} - \underline{x}'\right) = \begin{cases} 1 & \text{if } \underline{x} - \underline{x}' = \underline{0} \\ 0 & \text{otherwise} \end{cases}$$

and Δ^2 is the variance of the measurement error. In this work, the term "nugget" refers to the ratio $\eta = \frac{\Delta^2}{\sigma^2}$.

By convention, the terms simple Kriging, ordinary Kriging, and universal Kriging are used to indicate the three most common choices for the trend function. In simple Kriging, the trend is treated as a known constant, usually zero, $g\left(\underline{x}\right)\beta \equiv 0$. Universal Kriging [78] uses a general polynomial trend model $\underline{g}\left(\underline{x}\right)^T\ \underline{\beta}$ whose coefficients are determined by least squares regression. Ordinary Kriging is essentially universal Kriging with a trend order of zero, i.e. the trend function is treated as an unknown constant and $g\left(\underline{x}\right) = 1$. N_β is the number of basis functions in $\underline{g}\left(\underline{x}\right)$ and therefore number of elements in the vector $\underline{\beta}$.

For a finite number of sample points, N, there will be uncertainty about the most appropriate value of the vector, $\underline{\beta}$, which can therefore be described as having a distribution of possible values. If one assumes zero prior knowledge about this distribution, which is referred to as the "vague prior" assumption, then the maximum likelihood value of $\underline{\beta}$ can be computed via least squares generalized by the inverse of the error model's correlation matrix, $\underline{\underline{R}}$

$$\underline{\hat{\beta}} = \left(\underline{\underline{G}}^T\underline{\underline{R}}^{-1}\underline{\underline{G}}\right)^{-1}\left(\underline{\underline{G}}^T\underline{\underline{R}}^{-1}\underline{Y}\right).$$

Here $\underline{\underline{G}}$ is a N by N_β matrix that contains the evaluation of the least squares basis functions at all points in $\underline{\underline{X}}$, such that $G_{i,l} = g_l\left(\underline{X_i}\right)$. The real, symmetric, positive-definite correlation matrix, $\underline{\underline{R}}$, of the error model contains evaluations of the correlation function, r, at all pairwise combination of points (rows) in the sample design, $\underline{\underline{X}}$.

$$R_{i,j} = R_{j,i} = r\left(\underline{X_i}, \underline{X_j}\right) = r\left(\underline{X_j}, \underline{X_i}\right)$$

There are many options for r, among them are the following families of correlation functions:

- **Powered-Exponential**

$$r\left(\underline{X_i}, \underline{X_j}\right) = \exp\left(-\sum_{k=1}^{M} \theta_k \left|X_{i,k} - X_{j,k}\right|^\gamma\right) \tag{6.1}$$

  where $0 < \gamma \leq 2$ and $0 < \theta_k$.

- **Matern**

$$r\left(\underline{X_i}, \underline{X_j}\right) = \prod_{k=1}^{M} \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\theta_k \left|X_{i,k} - X_{j,k}\right|\right)^\nu \mathcal{K}_\nu\left(\theta_k \left|X_{i,k} - X_{j,k}\right|\right)$$

  where $0 < \nu$, $0 < \theta_k$, and $\mathcal{K}_\nu(\cdot)$ is the modified Bessel function of order ν; $\nu = s + \frac{1}{2}$ is the smallest value of ν which results in a Kriging model that is s times differentiable.

- **Cauchy**

$$r\left(\underline{X_i}, \underline{X_j}\right) = \prod_{k=1}^{M} \left(1 + \theta_k \left|X_{i,k} - X_{j,k}\right|^{\gamma}\right)^{-\nu}$$

  where $0 < \gamma \leq 2$, $0 < \nu$, and $0 < \theta_k$.

Gneiting et al. [58] provide a more thorough discussion of the properties of and relationships between these three families. Some additional correlation functions include the Dagum family [10] and cubic splines.

The squared exponential or Gaussian correlation function (Equation 6.1 with $\gamma = 2$) was selected to be the first correlation function implemented in Dakota on the basis that its infinite smoothness or differentiability should aid in leveraging the anticipated and hopefully sparse data. For the Gaussian correlation function, the correlation parameters, $\underline{\theta}$, are related to the correlation lengths, $\underline{L}$, by

$$\theta_k = \frac{1}{2\, L_k^2}. \tag{6.2}$$

Here, the correlation lengths, $\underline{L}$, are analogous to standard deviations in the Gaussian or normal distribution and often have physical meaning. The adjusted (by data) mean of the emulator is a best linear unbiased estimator of the unknown true function,

$$\hat{y} = \mathrm{E}\left(\hat{f}\left(\underline{x}\right) | \underline{f}\left(\underline{\underline{X}}\right)\right) = \underline{g}\left(\underline{x}\right)^T \underline{\hat{\beta}} + \underline{r}\left(\underline{x}\right)^T \ \underline{\underline{R}}^{-1} \underline{\epsilon}. \tag{6.3}$$

Here, $\underline{\epsilon} = \left(\underline{Y} - \underline{\underline{G}}\, \underline{\hat{\beta}}\right)$ is the known vector of differences between the true outputs and trend function at all points in $\underline{\underline{X}}$ and the vector $\underline{r}\left(\underline{x}\right)$ is defined such that $r_i\left(\underline{x}\right) = r\left(\underline{x}, \underline{X_i}\right)$. This correction can be interpreted as the projection of prior belief (the least squares fit) into the span of the data. The adjusted mean of the emulator will interpolate the data that the Kriging model was built from as long as its correlation matrix, $\underline{\underline{R}}$, is numerically non-singular.

Ill-conditioning of $\underline{\underline{R}}$ and other matrices is a recognized as a significant challenge for Kriging. Davis and Morris [30] gave a thorough review of six factors affecting the condition number of matrices associated with Kriging (from the perspective of semivariograms rather than correlation functions). They concluded that "Perhaps the best advice we can give is to be mindful of the condition number when building and solving kriging systems."

In the context of estimating the optimal $\underline{\theta}$, Martin [77] stated that Kriging's "three most prevalent issues are (1) ill-conditioned correlation matrices,(2) multiple local optimum, and (3) long ridges of near optimal values." Because of the second issue, global optimization methods are more robust than local methods. Martin used constrained optimization to address ill-conditioning of $\underline{\underline{R}}$.

Rennen [96] advocated that ill-conditioning be handled by building Kriging models from a uniform subset of available sample points. That option has been available in Dakota's "Gaussian process" model (a separate implementation from Dakota's "Kriging" model) since version 4.1 [46]. Note that Kriging/Gaussian-Process models will not exactly interpolate the discarded points. The implicit justification for this type of approach is that the row or columns of an ill-conditioned matrix contain a significant amount of duplicate information, and that when discarded, duplicate information should be easy to predict.

As of version 5.2, Dakota's `kriging` model has a similar "discard near duplicate points" capability. However, it explicitly addresses the issue of unique information content. Points are **not** discarded prior to the construction of the Kriging model. Instead, for each vector $\underline{\theta}$ examined that results in an ill-conditioned correlation matrix, $\underline{\underline{R}}$, a pivoted Cholesky factorization of $\underline{\underline{R}}$ is performed. This ranks the points according to how much unique information they contain. Note that the definition of information content depends on $\underline{\theta}$. Low information points are then discarded until $\underline{\underline{R}}$ is no longer ill-conditioned, i.e. until it tightly meets a constraint on condition number. This can be done efficiently using a bisection search that calls LAPACK's fast estimate of the (reciprocal of the) condition number. The possibly, and often, improper subset of points used to construct the Kriging model is the one associated with the chosen $\underline{\theta}$. Methods for selecting $\underline{\theta}$ are discussed below. Since the points that are discarded are the ones that contain the least unique information, they are the ones that are easiest to predict and provide maximum improvement to the condition number.

Adding a nugget, η, to the diagonal entries of $\underline{\underline{R}}$ is a popular approach for both accounting for measurement error in the data and alleviating ill-conditioning. However, doing so will cause the Kriging model to smooth or approximate rather than interpolate the data. Methods for choosing a nugget include:

- Choosing a nugget based on the variance of measurement error (if any); this will be an iterative process if σ^2 is not known in advance.
- Iteratively adding a successively larger nugget until $\underline{\underline{R}} + \eta\underline{\underline{I}}$ is no longer ill-conditioned.
- Exactly calculating the minimum nugget needed for a target condition number from $\underline{\underline{R}}$'s maximum λ_{max} and minimum λ_{min} eigenvalues. The condition number of $\underline{\underline{R}} + \eta\underline{\underline{I}}$ is $\frac{\lambda_{max}+\eta}{\lambda_{min}+\eta}$. However, calculating eigenvalues is computationally expensive. Since Kriging's $\underline{\underline{R}}$ matrix has all ones on the diagonal, its trace and therefore sum of eigenvalues is N. Consequently, a nugget value of $\eta = \frac{N}{\text{target condition number}-1}$ will always alleviate ill-conditioning. A smaller nugget that is also guaranteed to alleviate ill-conditioning can be calculated from LAPACK's fast estimate of the reciprocal of $\underline{\underline{R}}$'s condition number, $\mathrm{rcond}\left(\underline{\underline{R}}\right)$.
- Treating η as another parameter to be selected by the same process used to choose $\underline{\theta}$. Two such approaches are discussed below.

The Kriging model's adjusted variance is commonly used as a spatially varying measure of uncertainty. Knowing where, and by how much, the model "doubts" its own predictions helps build user confidence in the predictions and can be utilized to guide the selection of new sample points during optimization or to otherwise improve the surrogate. The adjusted variance is

$$\begin{aligned}
\mathrm{Var}\left(\hat{y}\right) &= \mathrm{Var}\left(\hat{f}\left(\underline{x}\right)|\underline{f}\left(\underline{\underline{X}}\right)\right) \\
&= \hat{\sigma}^2\left(1 - \underline{r}\left(\underline{x}\right)^T\ \underline{\underline{R}}^{-1}\underline{r}\left(\underline{x}\right) + ...\right. \\
&\quad \left.\left(\underline{g}\left(\underline{x}\right)^T - \underline{r}\left(\underline{x}\right)^T\ \underline{\underline{R}}^{-1}\underline{\underline{G}}\right)\left(\underline{\underline{G}}^T\underline{\underline{R}}^{-1}\underline{\underline{G}}\right)^{-1}\left(\underline{g}\left(\underline{x}\right)^T - \underline{r}\left(\underline{x}\right)^T\ \underline{\underline{R}}^{-1}\underline{\underline{G}}\right)^T\right)
\end{aligned}$$

where the maximum likelihood estimate of the unadjusted variance is

$$\hat{\sigma}^2 = \frac{\epsilon^T\underline{\underline{R}}^{-1}\epsilon}{N - N_\beta}.$$

There are two types of numerical approaches to choosing $\underline{\theta}$. One of these is to use Bayesian techniques such as Markov Chain Monte Carlo to obtain a distribution represented by an ensemble of vectors $\underline{\theta}$. In this case, evaluating the emulator's mean involves taking a weighted average of Equation 6.3 over the ensemble of $\underline{\theta}$ vectors.

The other, more common, approach to constructing a Kriging model involves using optimization to find the set of correlation parameters $\underline{\theta}$ that maximizes the likelihood of the model given the data. Dakota's `gaussian_process` and `kriging` models use the maximum likelihood approach. It is equivalent, and more convenient to maximize the natural logarithm of the likelihood, which assuming a vague prior is,

$$\begin{aligned}\log\left(\text{lik}\left(\underline{\theta}\right)\right) &= -\frac{1}{2}\Bigg((N-N_\beta)\left(\frac{\hat{\sigma}^2}{\sigma^2}+\log\left(\sigma^2\right)+\log(2\pi)\right)+...\\ &\qquad \log\left(\det\left(\underline{\underline{R}}\right)\right)+\log\left(\det\left(\underline{\underline{G}}^T\underline{\underline{R}}^{-1}\underline{\underline{G}}\right)\right)\Bigg).\end{aligned}$$

And, if one substitutes the maximum likelihood estimate $\hat{\sigma}^2$ in for σ^2, then it is equivalent to minimize the following objective function

$$\text{obj}\left(\underline{\theta}\right)=\log\left(\hat{\sigma}^2\right)+\frac{\log\left(\det\left(\underline{\underline{R}}\right)\right)+\log\left(\det\left(\underline{\underline{G}}^T\underline{\underline{R}}^{-1}\underline{\underline{G}}\right)\right)}{N-N_\beta}.$$

Because of the division by $N-N_\beta$, this "per-equation" objective function is mostly independent of the number of sample points, N. It is therefore useful for comparing the (estimated) "goodness" of Kriging models that have different numbers of sample points, e.g. when an arbitrary number of points can be discarded by the pivoted Cholesky approach described above.

Note that the determinant of $\underline{\underline{R}}$ (and $\left(\underline{\underline{G}}^T\underline{\underline{R}}^{-1}\underline{\underline{G}}\right)$) can be efficiently calculated as the square of the product of the diagonals of its Cholesky factorization. However, this will often underflow, i.e. go to zero, making its log incorrectly go to $-\infty$. A more accurate and robust calculation of $\log\left(\det\left(\underline{\underline{R}}\right)\right)$ can be achieved by taking twice the sum of the log of the diagonals of $\underline{\underline{R}}$'s Cholesky factorization.

Also note, that in the absence of constraints, maximizing the likelihood would result in singular $\underline{\underline{R}}$ which makes the emulator incapable of reproducing the data from which it was built. This is because a singular $\underline{\underline{R}}$ makes $\log\left(\det\left(\underline{\underline{R}}\right)\right)=-\infty$ and the *estimate* of likelihood infinite. Constraints are therefore required. Two types of constraints are used in Dakota's `kriging` models.

The first of these is an explicit constraint on LAPACK's fast estimate of the (reciprocal of the) condition number, $2^{-40}<\text{rcond}\left(\underline{\underline{R}}\right)$. The value 2^{-40} was chosen based on the assumption that double precision arithmetic is used. Double precision numbers have 52 bits of precision. Therefore the $2^{-40}<\text{rcond}\left(\det\left(\underline{\underline{R}}\right)\right)$ implies that at least the leading three significant figures should be uncorrupted by round off error. In Dakota 5.2, this constraint is used to determine how many points can be retained in the pivoted Cholesky approach to subset selection described above.

The second, is a box constraint defining a small "feasible" region in correlation length space to search during the maximum likelihood optimization. Many global optimizers, such as the DIRECT (DIvision of RECTangles) used by Dakota's Gaussian Process (as the only option) and Kriging (as the default option) models, require a box constraint definition for the range of acceptable parameters. By default, Dakota's `kriging` model defines the input space to be the smallest hyper-rectangle that contains the sample design. The user has the option to define a larger input space that includes a region where they wish to extrapolate. Note that the emulator can be evaluated at points outside the defined input space, but this definition helps to determine the extent of the "feasible" region of correlation lengths. Let the input space be normalized to the unit hypercube centered at the origin. The average distance between nearest neighboring points is then

$$d = \left(\frac{1}{N}\right)^{1/M}.$$

Dakota's "feasible" range of correlation lengths, $\underline{L}$, for the Gaussian correlation function is

$$\frac{d}{4} \leq L_k \leq 8d.$$

This range was chosen based on correlation lengths being analogous to the standard deviation in the Gaussian or Normal distribution. If the correlation lengths are set to $L_k = d/4$, then nearest neighboring points "should be" roughly four "standard deviations" away making them almost completely uncorrelated with each other. $\underline{\underline{R}}$ would then be a good approximation of the identity matrix and have a condition number close to one. In the absence of a pathological spacing of points, this range of $\underline{L}$ should contain some non-singular $\underline{\underline{R}}$. $L_k = 8d$ implies approximately 32% trust in what points 8 neighbors away have to say and 5% trust in what points 16 neighbors away have to say. It is possible that the optimal correlation lengths are larger than $8d$; but if so, then either almost all of the same information will be contained in more nearby neighbors, or it was not appropriate to use the squared-exponential/Gaussian correlation function. When other correlation functions are added to the Dakota Kriging implementation, each will be associated with its own range of appropriate correlation lengths chosen by similar reasoning. A different definition of d could be used for non-hypercube input domains.

6.1.2 Gradient Enhanced Kriging

This section focuses on the incorporation of derivative information into Kriging models and challenges in their implementation. Special attention is paid to conditioning issues.

There are at least three basic approaches for incorporating derivative information into Kriging. These are

1. **Indirect**: The sample design is augmented with fictitious points nearby actual sample points which are predicted from derivative information and then a Kriging model is built from the augmented design.

2. **Co-Kriging**: The derivatives with respect to each input variables are treated as separate but correlated output variables and a Co-Kriging model is built for the set of output variables. This would use $\begin{pmatrix} M+2 \\ 2 \end{pmatrix}$ $\underline{\theta}$ vectors.

3. **Direct**: The relationship between the response value and its derivatives is leveraged to use a single $\underline{\theta}$ by assuming

$$\text{Cov}\left(y\left(\underline{x}^1\right), \frac{\partial y\left(\underline{x}^2\right)}{\partial x_k^2}\right) = \frac{\partial}{\partial x_k^2}\left(\text{Cov}\left(y\left(\underline{x}^1\right), y\left(\underline{x}^2\right)\right)\right). \tag{6.4}$$

Dakota 5.2 and later includes an implementation of the direct approach, herein referred to simply as Gradient Enhanced (universal) Kriging (GEK). The equations for GEK can be derived by assuming Equation 6.4 and then taking the same steps used to derive function value only Kriging. The superscript on $\underline{x}$ in Equation 6.4 and below indicates whether it's the 1st or 2nd input to $r\left(\underline{x}^1, \underline{x}^2\right)$. Note that when the first and second arguments are the same, the derivative of $r\left(\ ,\ \right)$ with respect to the first argument is equal in magnitude but opposite in sign compared to the derivative with respect to the second argument. The GEK equations can also be obtained by starting from a Kriging model and making the following substitutions $\underline{Y} \rightarrow \underline{Y_{\nabla}}$, $\underline{\underline{G}} \rightarrow \underline{\underline{G_{\nabla}}}$, $\underline{r} \rightarrow \underline{r_{\nabla}}$, $\underline{\underline{R}} \rightarrow \underline{\underline{R_{\nabla}}}$, and $N \rightarrow N_{\nabla} = N\,(1+M)$, where N_{∇} is the number of equations rather than the number of points,

$$\underline{Y_{\nabla}} = \begin{bmatrix} \underline{Y} \\ \\ \frac{\partial \underline{Y}}{\partial X_{:,1}} \\ \\ \frac{\partial \underline{Y}}{\partial X_{:,2}} \\ \\ \vdots \\ \\ \frac{\partial \underline{Y}}{\partial X_{:,M}} \end{bmatrix}, \quad \underline{\underline{G_{\nabla}}} = \begin{bmatrix} \underline{\underline{G}} \\ \\ \frac{\partial \underline{\underline{G}}}{\partial X_{:,1}} \\ \\ \frac{\partial \underline{\underline{G}}}{\partial X_{:,2}} \\ \\ \vdots \\ \\ \frac{\partial \underline{\underline{G}}}{\partial X_{:,M}} \end{bmatrix}, \quad \underline{r_{\nabla}} = \begin{bmatrix} \underline{r} \\ \\ \frac{\partial \underline{r}}{\partial X_{:,1}} \\ \\ \frac{\partial \underline{r}}{\partial X_{:,2}} \\ \\ \vdots \\ \\ \frac{\partial \underline{r}}{\partial X_{:,M}} \end{bmatrix}$$

$$\underline{\underline{R_{\nabla}}} = \begin{bmatrix} \underline{\underline{R}} & \frac{\partial \underline{\underline{R}}}{\partial X^2_{:,1}} & \frac{\partial \underline{\underline{R}}}{\partial X^2_{:,2}} & \cdots & \frac{\partial \underline{\underline{R}}}{\partial X^2_{:,M}} \\ \\ \frac{\partial \underline{\underline{R}}}{\partial X^1_{:,1}} & \frac{\partial^2 \underline{\underline{R}}}{\partial X^1_{:,1} \partial X^2_{:,1}} & \frac{\partial^2 \underline{\underline{R}}}{\partial X^1_{:,1} \partial X^2_{:,2}} & \cdots & \frac{\partial^2 \underline{\underline{R}}}{\partial X^1_{:,1} \partial X^2_{:,M}} \\ \\ \frac{\partial \underline{\underline{R}}}{\partial X^1_{:,2}} & \frac{\partial^2 \underline{\underline{R}}}{\partial X^1_{:,2} \partial X^2_{:,1}} & \frac{\partial^2 \underline{\underline{R}}}{\partial X^1_{:,2} \partial X^2_{:,2}} & \cdots & \frac{\partial^2 \underline{\underline{R}}}{\partial X^1_{:,2} \partial X^2_{:,M}} \\ \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \\ \frac{\partial \underline{\underline{R}}}{\partial X^1_{:,M}} & \frac{\partial^2 \underline{\underline{R}}}{\partial X^1_{:,M} \partial X^2_{:,1}} & \frac{\partial^2 \underline{\underline{R}}}{\partial X^1_{:,M} \partial X^2_{:,2}} & \cdots & \frac{\partial^2 \underline{\underline{R}}}{\partial X^1_{:,M} \partial X^2_{:,M}} \end{bmatrix}$$

$$\frac{\partial \underline{\underline{R}}}{\partial X^1_{:,I}} = -\left(\frac{\partial \underline{\underline{R}}}{\partial X^1_{:,I}}\right)^T = -\frac{\partial \underline{\underline{R}}}{\partial X^2_{:,I}} = \left(\frac{\partial \underline{\underline{R}}}{\partial X^2_{:,I}}\right)^T$$

$$\frac{\partial^2 \underline{\underline{R}}}{\partial X^1_{:,I} \partial X^2_{:,J}} = \left(\frac{\partial^2 \underline{\underline{R}}}{\partial X^1_{:,I} \partial X^2_{:,J}}\right)^T = \frac{\partial^2 \underline{\underline{R}}}{\partial X^1_{:,J} \partial X^2_{:,I}} = \left(\frac{\partial^2 \underline{\underline{R}}}{\partial X^1_{:,J} \partial X^2_{:,I}}\right)^T$$

Here capital I and J are scalar indices for the input dimension (column) of the sample design, $\underline{\underline{X}}$. Note that for the Gaussian correlation function

$$\frac{\partial^2 R_{j,j}}{\partial X^1_{j,I} \partial X^2_{j,I}} = 2\theta_I$$

and has units of length^{-2}. Two of the conditions necessary for a matrix to qualify as a correlation matrix are that all of its elements must be dimensionless and all of its diagonal elements must identically equal one. Since $\underline{\underline{R_{\nabla}}}$ does not satisfy these requirements, it technically does not qualify as a "correlation matrix." However, referring to $\underline{\underline{R_{\nabla}}}$ as such is a small abuse of terminology and allows GEK to use the same naming conventions as Kriging.

A straight-forward implementation of GEK tends be significantly more accurate than Kriging given the same sample design provided that the

- Derivatives are accurate
- Derivatives are not infinite (or nearly so)

- Function is sufficiently smooth, and
- $\underline{\underline{R_\nabla}}$ is not ill-conditioned (this can be problematic).

If gradients can be obtained cheaply (e.g. by automatic differentiation or adjoint techniques) and the previous conditions are met, GEK also tends to outperform Kriging for the same computational budget. Previous works, such as Dwight[37], state that the direct approach to GEK is significantly better conditioned than the indirect approach. While this is true, (direct) GEK's $\underline{\underline{R_\nabla}}$ matrix can still be, and often is, horribly ill-conditioned compared to Kriging's $\underline{\underline{R}}$ for the same $\underline{\theta}$ and $\underline{\underline{X}}$

In the literature, ill-conditioning is often attributed to the choice of the correlation function. Although a different correlation function may alleviate the ill-conditioning for some problems, the root cause of the ill-conditioning is a poorly spaced sample design. Furthermore, a sufficiently bad sample design could make any interpolatory Kriging model, gradient enhanced or otherwise, ill-conditioned, regardless of the choice of correlation function. This root cause can be addressed directly by discarding points/equations.

Discarding points/equations is conceptually similar to using a Moore-Penrose pseudo inverse of $\underline{\underline{R_\nabla}}$. However, there are important differences. A pseudo inverse handles ill-conditioning by discarding small singular values, which can be interpreted as throwing away the information that is least present while keeping all of what is most frequently duplicated. This causes a Kriging model to not interpolate any of the data points used to construct it while using some information from all rows.

An alternate strategy is to discard additional copies of the information that is most duplicated and keep more of the barely present information. In the context of eigenvalues, this can be described as decreasing the maximum eigenvalue and increasing the minimum eigenvalue by a smaller amount than a pseudo inverse. The result is that the GEK model will exactly fit all of the retained information. This can be achieved using a pivoted Cholesky factorization, such as the one developed by Lucas [75] to determine a reordering $\underline{\underline{\tilde{R}_\nabla}}$ and dropping equations off its end until it tightly meets the constraint on rcond. However, a straight-forward implementation of this is neither efficient nor robust.

In benchmarking tests, Lucas' level 3 pivoted Cholesky implementation was not competitive with the level 3 LAPACK non-pivoted Cholesky in terms of computational efficiency. In some cases, it was an order of magnitude slower. Note that Lucas' level 3 implementation can default to his level 2 implementation and this may explain some of the loss in performance.

More importantly, applying pivoted Cholesky to $\underline{\underline{R_\nabla}}$ tends to sort derivative equations to the top/front and function value equations to the end. This occurred even when $\underline{\underline{R_\nabla}}$ was equilibrated to have ones for all of its diagonal elements. The result was that for many points at least some of the derivative equations were retained while the function values at the same points were discarded. This can (and often did) significantly degrade the accuracy of the GEK predictions. The implication is that derivative equations contain more information than, but are not as reliable as, function value equations.

To address computational efficiency and robustness, Dakota's pivoted Cholesky approach for GEK was modified to:

- Equilibrate $\underline{\underline{R_\nabla}}$ to improve the accuracy of the Cholesky factorization; this is beneficial because $\underline{\underline{R_\nabla}}$ can be poorly scaled. Theorem 4.1 of van der Sluis [107] states that if $\underline{\underline{a}}$ is a real, symmetric, positive definite n by

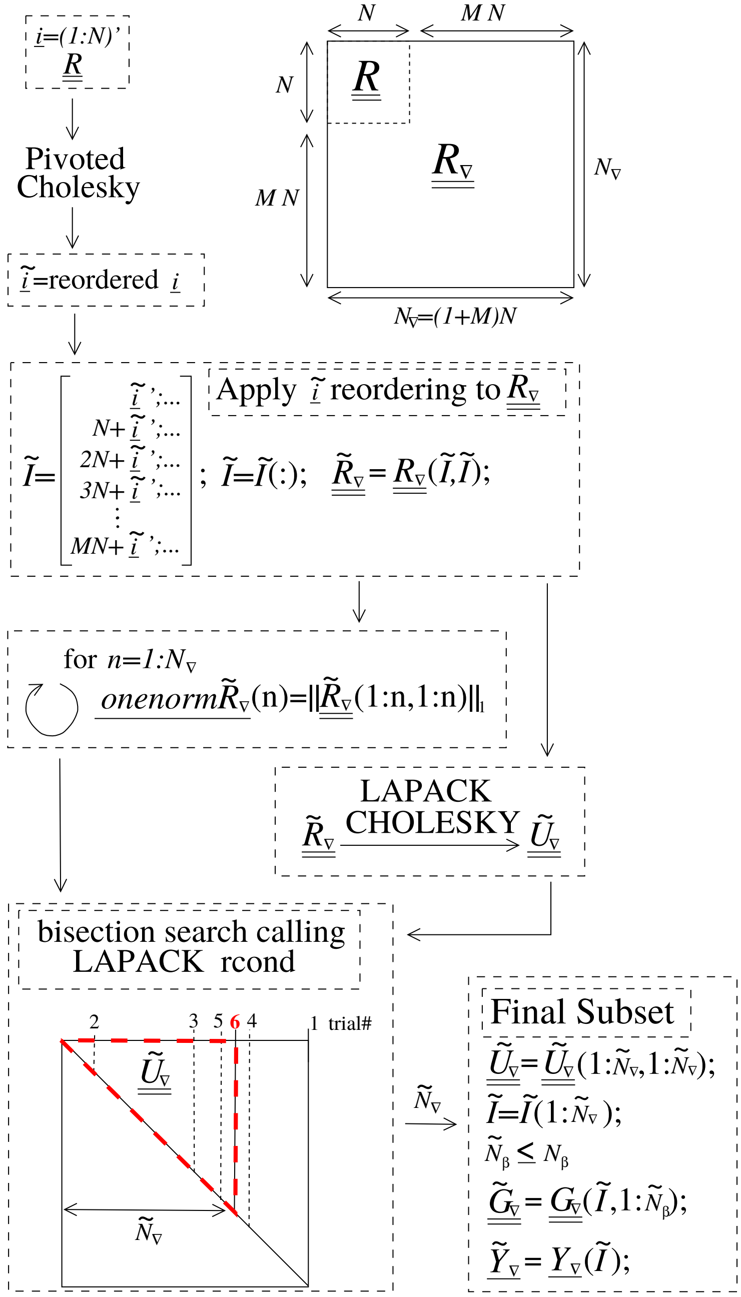
$\underline{i}$=(1:N)'
$\underline{\underline{R}}$
Pivoted Cholesky
$\tilde{\underline{i}}$=reordered $\underline{i}$
N
M N
$\underline{\underline{R}}$
$\underline{\underline{R}}_\nabla$
N_∇
N_∇=(1+M)N
$\tilde{I}$=
$\tilde{\underline{i}}$';...
N+$\tilde{\underline{i}}$';...
2N+$\tilde{\underline{i}}$';...
3N+$\tilde{\underline{i}}$';...
MN+$\tilde{\underline{i}}$';...
Apply $\tilde{\underline{i}}$ reordering to $\underline{\underline{R}}_\nabla$
; $\tilde{I}=\tilde{I}$(:); $\underline{\underline{\tilde{R}}}_\nabla = \underline{\underline{R}}_\nabla(\tilde{I},\tilde{I})$;
for n=1:N_∇
onenorm$\tilde{R}_\nabla$(n)=||$\underline{\underline{\tilde{R}}}_\nabla$(1:n,1:n)||$_1$
LAPACK CHOLESKY
$\underline{\underline{\tilde{R}}}_\nabla$
$\underline{\underline{\tilde{U}}}_\nabla$
bisection search calling LAPACK rcond
2
3
5
6
4
1
trial#
$\underline{\underline{\tilde{U}}}_\nabla$
$\tilde{N}_\nabla$
$\tilde{N}_\nabla$
Final Subset
$\underline{\underline{\tilde{U}}}_\nabla = \underline{\underline{\tilde{U}}}_\nabla(1:\tilde{N}_\nabla,1:\tilde{N}_\nabla)$;
$\tilde{I}=\tilde{I}(1:\tilde{N}_\nabla)$;
$\tilde{N}_\beta \leq N_\beta$
$\underline{\underline{\tilde{G}}}_\nabla = \underline{\underline{G}}_\nabla(\tilde{I},1:\tilde{N}_\beta)$;
$\underline{\tilde{Y}}_\nabla = \underline{Y}_\nabla(\tilde{I})$;

Figure 6.1: A diagram with pseudo code for the pivoted Cholesky algorithm used to select the subset of equations to retain when $\underline{\underline{R}}_\nabla$ is ill-conditioned. Although it is part of the algorithm, the equilibration of $\underline{\underline{R}}_\nabla$ is not shown in this figure. The pseudo code uses MATLAB notation.

n matrix and the diagonal matrix $\underline{\underline{\alpha}}$ contains the square roots of the diagonals of $\underline{\underline{a}}$, then the equilibration

$$\underline{\underline{\breve{a}}} = \underline{\underline{\alpha}}^{-1} \underline{\underline{a}}\, \underline{\underline{\alpha}}^{-1},$$

minimizes the 2-norm condition number of $\underline{\underline{\breve{a}}}$ (with respect to solving linear systems) over all such symmetric scalings, to within a factor of n. The equilibrated matrix $\underline{\underline{\breve{a}}}$ will have all ones on the diagonal.

- Perform pivoted Cholesky on $\underline{\underline{R}}$, instead of $\underline{\underline{R_\nabla}}$, to rank points according to how much new information they contain. This ranking was reflected by the ordering of points in $\underline{\underline{\tilde{R}}}$.
- Apply the ordering of points in $\underline{\underline{\tilde{R}}}$ to whole points in $\underline{\underline{R_\nabla}}$ to produce $\underline{\underline{\tilde{R}_\nabla}}$. Here a whole point means the function value at a point immediately followed by the derivatives at the same point.
- Perform a LAPACK non-pivoted Cholesky on the equilibrated $\underline{\underline{\tilde{R}_\nabla}}$ and drop equations off the end until it satisfies the constraint on rcond. LAPACK's rcond estimate requires the 1-norm of the original (reordered) matrix as input so the 1-norms for all possible sizes of $\underline{\underline{\tilde{R}_\nabla}}$ are precomputed (using a rank one update approach) and stored prior to the Cholesky factorization. A bisection search is used to efficiently determine the number of equations that need to be retained/discarded. This requires $\text{ceil}\left(\log 2\left(N_\nabla\right)\right)$ or fewer evaluations of rcond. These rcond calls are all based off the same Cholesky factorization of $\underline{\underline{\tilde{R}_\nabla}}$ but use different numbers of rows/columns, $\tilde{N}_\nabla$.

This algorithm is visually depicted in Figure 6.1. Because inverting/factorizing a matrix with n rows and columns requires $\mathcal{O}\left(n^3\right)$ flops, the cost to perform pivoted Cholesky on $\underline{\underline{R}}$ will be much less than, i.e. $\mathcal{O}\left((1+M)^{-3}\right)$, that of $\underline{\underline{R_\nabla}}$ when the number of dimensions M is large. It will also likely be negligible compared to the cost of performing LAPACK's non-pivoted Cholesky on $\underline{\underline{\tilde{R}_\nabla}}$.