PROCEEDINGS OF THE ROYAL SOCIETY A

Proc. R. Soc. A (2007) **463**, 3251–3269
doi:10.1098/rspa.2007.1900
Published online 2 October 2007

Multi-fidelity optimization via surrogate modelling

By Alexander I. J. Forrester*, András Sóbester and Andy J. Keane

Computational Engineering and Design Group, School of Engineering Sciences, University of Southampton, Southampton SO17 1BJ, UK

This paper demonstrates the application of correlated Gaussian process based approximations to optimization where multiple levels of analysis are available, using an extension to the geostatistical method of *co-kriging*. An exchange algorithm is used to choose which points of the search space to sample within each level of analysis. The derivation of the co-kriging equations is presented in an intuitive manner, along with a new variance estimator to account for varying degrees of computational 'noise' in the multiple levels of analysis. A multi-fidelity wing optimization is used to demonstrate the methodology.

Keywords: co-kriging; kriging; noise; subset selection; wing design

1. Introduction

Let $f(\boldsymbol{x})$ denote an *objective function* that maps a vector $\boldsymbol{x}$ of length k to some scalar measure of merit. $\boldsymbol{x}^*$ is sought, such that $f(\boldsymbol{x}^*)$ is the maximum or minimum of $f(\boldsymbol{x})$. Such *optimization* problems (of which this is the simplest form) are encountered in most branches of science. Of interest to us in this paper is the class of optimization problems where $f(\boldsymbol{x})$ is expensive to obtain or we know its value only at a limited number of $\boldsymbol{x}$ values. Thus, efficiency, in terms of the number of $\boldsymbol{x}$ values that must be mapped to their objective function values before we get to within a reasonable distance of $\boldsymbol{x}^*$, is the key feature of the optimization algorithms considered.

While there is often no getting around the fact that a thorough search of a highly nonlinear, multi-dimensional f landscape will require sampling at a large number of sites, a useful shortcut is often employed. Based on a relatively small number of measurements, we can build a statistical approximation of the objective landscape, which, provided f is smooth and continuous and the measurements are reasonably uniformly spread, will be accurate enough to guide the search towards promising areas of the landscape.

If, at the same time, this *surrogate* of the expensive function is very cheap to evaluate, we can be as thorough as we like in searching it. Thus, most of the computational effort will be concentrated on what looks likely to be the

* Author for correspondence (alexander.forrester@soton.ac.uk).

Received 20 March 2007
Accepted 6 September 2007

neighbourhood of the global optimum. An approximation technique that has received much attention in recent years is *kriging*[1]—we shall delve into the details of this later on, as our chosen multi-fidelity optimization technique is an extension of kriging. But what exactly do we mean by multi-fidelity optimization?

The objective function evaluation system may feature lower-fidelity, cheaper models (in addition to the main, high-fidelity calculation of f). In most cases, the fast (but less trustworthy) and the slow (but more accurate) objective values can be obtained independently. Thus, we can learn more about the objective function by additionally measuring the cheap function(s) on a large number of $\boldsymbol{x}$ values. This process is usually referred to as multi-fidelity optimization.

The use of secondary, correlated quantities to improve the accuracy of the model of the primary objective is not a new concept. For example, Hevesi *et al.* (1992) predict average annual precipitation values near a potential nuclear waste disposal site using a sparse set of precipitation measurements from the region along with the correlated and more easily obtainable elevation map of the area. Kennedy & O'Hagan (2000) approach the subject from the perspective of model building using objectives resulting from computational simulations of varying fidelities and costs. This is also the context of the work presented here; additionally, we introduce a method of optimization using correlated models of different costs and fidelities. We focus on combining co-kriging (the multi-response extension to kriging alluded to earlier) with a Bayesian model update criterion designed to balance the exploration of the search space and the quick exploitation of promising basins of attraction on the f landscape. The criterion is based on a newly derived error estimate that reflects the presence of noise in the observed data.

After a brief overview of kriging in §2, we discuss co-kriging in §3, followed by considerations on how to create sampling plans for multiple levels of analyses (§4). We put these pieces together in §5, where we describe an optimization strategy designed for multi-fidelity inputs. An aircraft wing design problem is used to demonstrate and compare the co-kriging approach and other multi-level strategies in §6. Section 7 discusses the remaining issues related to the use of correlated surrogates and summarizes our conclusions.

2. Kriging

We will consider co-kriging as a natural extension to the popular method of kriging and hence begin with a very brief introduction to the kriging method in order to introduce concepts and notation that follow through to co-kriging. Equations are shown without derivations, which are very similar to those presented later for co-kriging; the reader may wish to consult Jones (2001) or Forrester *et al.* (2006*b*) for more information on kriging.

As with all surrogate-based methods, to approximate a function f we start with a set of sample data—usually computed at a set of points in the domain of interest determined by a sampling plan (which will be discussed further in §4).

[1] Originally called *krigeage* by Matheron (1963) after D. G. Krige—a South African mining engineer who pioneered the method in the 1950s for determining ore-grade distributions based on core samples (Krige 1951).

The kriging prediction of f is built from a mean base term, $\hat{\mu}$ (the circumflex denotes a maximum likelihood estimate, MLE) plus a stationary Gaussian process, $Z(\boldsymbol{x})$, representing the local features of f around the n sample points, $\boldsymbol{X}=\{\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(n)}\}^{\mathrm{T}}$, $\boldsymbol{x}\in\mathbb{R}^k$. $Z(\boldsymbol{x})$ has zero mean and covariance

$$\operatorname{cov}[Z(\boldsymbol{x}^{(n+1)}),Z(\boldsymbol{x}^{(i)})]=\sigma^2\psi^{(i)}, \tag{2.1}$$

where $\psi^{(i)}$ are correlations between a random variable $Y(\boldsymbol{x})$ at the point to be predicted $(\boldsymbol{x}^{(n+1)})$ and at the sample data points

$$\psi^{(i)}=\operatorname{corr}[Y(\boldsymbol{x}^{(n+1)}),\,Y(\boldsymbol{x}^{(i)})]=\exp\left(-\sum_{j=1}^{k}\hat{\theta}_j\|x_j^{(n+1)}-x_j^{(i)}\|^{\hat{p}_j}\right). \tag{2.2}$$

The *hyper-parameter,* p_j, can be thought of as determining the smoothness of the function approximation. In many situations, we can assume that there will not be any discontinuities and use $p_j=2$ rather than using an MLE. This means that the basis function is infinitely differentiable through a sample point (when $\|\boldsymbol{x}^{(n+1)}-\boldsymbol{x}^{(i)}\|=0$) and that the function is in the same form as a Gaussian distribution with variance $1/\hat{\theta}_j$. $\hat{\boldsymbol{\theta}}$ therefore can be thought of as determining how quickly the function changes as $\boldsymbol{x}^{(n+1)}$ moves away from $\boldsymbol{x}^{(i)}$, with high and low $\hat{\theta}_j$ values indicating an active or inactive function along dimension j, respectively. The kriging prediction is found as the value at $\boldsymbol{x}^{(n+1)}$ that, maximizes the likelihood, given the sample data and MLEs of the hyper-parameters, and is given by

$$\hat{y}(\boldsymbol{x}^{(n+1)})=\hat{\mu}+\sum_{i=1}^{n}b^{(i)}\psi^{(i)}(\boldsymbol{x}^{(n+1)},\boldsymbol{x}^{(i)}). \tag{2.3}$$

The constants b_i are given by the column vector $\boldsymbol{b}=\boldsymbol{\Psi}^{-1}(\boldsymbol{y}-\boldsymbol{1}\hat{\mu})$, where $\boldsymbol{\Psi}$ is an $n\times n$ symmetric matrix of correlations between the sample data; $\boldsymbol{y}$ is a column vector of responses, $\{y(\boldsymbol{x}^{(1)}),\ldots,\,y(\boldsymbol{x}^{(n)})\}^{\mathrm{T}}$; $\boldsymbol{1}$ is an $n\times 1$ column vector of ones; and the MLE $\hat{\mu}=\boldsymbol{1}^{\mathrm{T}}\boldsymbol{\Psi}^{-1}\boldsymbol{y}/\boldsymbol{1}^{\mathrm{T}}\boldsymbol{\Psi}^{-1}\boldsymbol{1}$.

3. Co-kriging

We now consider how to build an approximation of a function that is expensive to evaluate which is enhanced by data from cheaper analyses of the function. Combining multiple sets of data naturally leads to a complex notation and we will try to simplify this by limiting ourselves to two datasets:[2] the most accurate expensive data with values $\boldsymbol{y}_{\mathrm{e}}$ at points $\boldsymbol{X}_{\mathrm{e}}$ and the less accurate cheap data $\boldsymbol{y}_{\mathrm{c}}$ at points $\boldsymbol{X}_{\mathrm{c}}$ $(\boldsymbol{X}_{\mathrm{e}}\subset\boldsymbol{X}_{\mathrm{c}})$.[3] These data are concatenated to give the combined set of points

$$\boldsymbol{X}=\begin{pmatrix}\boldsymbol{X}_{\mathrm{c}}\\ \boldsymbol{X}_{\mathrm{e}}\end{pmatrix}=(\boldsymbol{x}_{\mathrm{c}}^{(1)},\ldots,\boldsymbol{x}_{\mathrm{c}}^{(n_{\mathrm{c}})},\boldsymbol{x}_{\mathrm{e}}^{(1)},\ldots,\boldsymbol{x}_{\mathrm{e}}^{(n_{\mathrm{e}})})^{\mathrm{T}}.$$

[2] Our methods can be extended to multiple code levels following the notation used by Kennedy & O'Hagan (2000).

[3] We require partially collocated points in our estimation of a scaling parameter between the data. It is possible to construct the presented co-kriging model with completely non-collocated points by using kriging estimates $\hat{\boldsymbol{y}}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{e}})$.

As with kriging, the value at a point in $\boldsymbol{X}$ is treated as if it were the realization of a Gaussian random variable. For co-kriging, we therefore have the random field

$$\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{Y}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{c}}) \\ \boldsymbol{Y}_{\mathrm{e}}(\boldsymbol{X}_{\mathrm{e}}) \end{pmatrix} = (Y_{\mathrm{c}}(\boldsymbol{x}_{\mathrm{c}}^{(1)}), \ldots, Y_{\mathrm{c}}(\boldsymbol{x}_{\mathrm{c}}^{(n_{\mathrm{c}})}), Y_{\mathrm{e}}(\boldsymbol{x}_{\mathrm{e}}^{(1)}), \ldots, Y_{\mathrm{e}}(\boldsymbol{x}_{\mathrm{e}}^{(n_{\mathrm{e}})}))^{\mathrm{T}}.$$

We follow the auto-regressive model of Kennedy & O'Hagan (2000), which assumes that $\mathrm{cov}\{Y_{\mathrm{e}}(\boldsymbol{x}^{(i)}), Y_{\mathrm{c}}(\boldsymbol{x}) | Y_{\mathrm{c}}(\boldsymbol{x}^{(i)})\} = 0$, $\forall \boldsymbol{x} \neq \boldsymbol{x}^{(i)}$. This means that no more can be learnt about $Y_{\mathrm{e}}(\boldsymbol{x}^{(i)})$ from the cheaper code if the value of the expensive function at $\boldsymbol{x}^{(i)}$ is known—a Markov property (i.e. we assume that the expensive simulation is correct and any inaccuracies lie wholly in the cheaper simulation). Gaussian processes $Z_{\mathrm{c}}(\cdot)$ and $Z_{\mathrm{e}}(\cdot)$ represent the local features of the cheap and expensive codes. Using the auto-regressive model, we are essentially approximating the expensive code as the cheap code multiplied by a scaling factor ρ plus a Gaussian process $Z_{\mathrm{d}}(\cdot)$ that represents the difference between $\rho Z_{\mathrm{c}}(\cdot)$ and $Z_{\mathrm{e}}(\cdot)$

$$Z_{\mathrm{e}}(\boldsymbol{x}) = \rho Z_{\mathrm{c}}(\boldsymbol{x}) + Z_{\mathrm{d}}(\boldsymbol{x}). \tag{3.1}$$

Whereas in kriging we have a covariance matrix $\mathrm{cov}\{\boldsymbol{Y}(\boldsymbol{X}), \boldsymbol{Y}(\boldsymbol{X})\} = \sigma^2 \boldsymbol{\Psi}(\boldsymbol{X}, \boldsymbol{X})$, we now have a more complex covariance matrix constructed as follows:

$$\begin{aligned}
\mathrm{cov}\{\boldsymbol{Y}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{c}}), \boldsymbol{Y}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{c}})\} &= \mathrm{cov}\{Z_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{c}}), Z_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{c}})\} \\
&= \sigma_{\mathrm{c}}^2 \boldsymbol{\Psi}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{c}}, \boldsymbol{X}_{\mathrm{c}}), \\
\mathrm{cov}\{\boldsymbol{Y}_{\mathrm{e}}(\boldsymbol{X}_{\mathrm{e}}), \boldsymbol{Y}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{c}})\} &= \mathrm{cov}\{\rho Z_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{c}}) + Z_{\mathrm{d}}(\boldsymbol{X}_{\mathrm{c}}), Z_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{e}})\} \\
&= \rho \sigma_{\mathrm{c}}^2 \boldsymbol{\Psi}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{c}}, \boldsymbol{X}_{\mathrm{e}}), \\
\mathrm{cov}\{\boldsymbol{Y}_{\mathrm{e}}(\boldsymbol{X}_{\mathrm{e}}), \boldsymbol{Y}_{\mathrm{e}}(\boldsymbol{X}_{\mathrm{e}})\} &= \mathrm{cov}\{\rho Z_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{e}}) + Z_{\mathrm{d}}(\boldsymbol{X}_{\mathrm{e}}), \rho Z_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{e}}) + Z_{\mathrm{d}}(\boldsymbol{X}_{\mathrm{e}})\} \\
&= \rho^2 \mathrm{cov}\{Z_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{e}}), Z_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{e}})\} + \mathrm{cov}\{Z_{\mathrm{d}}(\boldsymbol{X}_{\mathrm{e}}), Z_{\mathrm{d}}(\boldsymbol{X}_{\mathrm{e}})\} \\
&= \rho^2 \sigma_{\mathrm{c}}^2 \boldsymbol{\Psi}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{e}}, \boldsymbol{X}_{\mathrm{e}}) + \sigma_{\mathrm{d}}^2 \boldsymbol{\Psi}_{\mathrm{d}}(\boldsymbol{X}_{\mathrm{e}}, \boldsymbol{X}_{\mathrm{e}}).
\end{aligned}$$

The notation $\boldsymbol{\Psi}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{e}}, \boldsymbol{X}_{\mathrm{c}})$, for example, denotes a matrix of correlations of the form ψ_{c} between the data $\boldsymbol{X}_{\mathrm{e}}$ and $\boldsymbol{X}_{\mathrm{c}}$. Our complete covariance matrix is thus

$$\boldsymbol{C} = \begin{pmatrix} \sigma_{\mathrm{c}}^2 \boldsymbol{\Psi}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{c}}, \boldsymbol{X}_{\mathrm{c}}) & \rho \sigma_{\mathrm{c}}^2 \boldsymbol{\Psi}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{c}}, \boldsymbol{X}_{\mathrm{e}}) \\ \rho \sigma_{\mathrm{c}}^2 \boldsymbol{\Psi}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{e}}, \boldsymbol{X}_{\mathrm{c}}) & \rho^2 \sigma_{\mathrm{c}}^2 \boldsymbol{\Psi}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{e}}, \boldsymbol{X}_{\mathrm{e}}) + \sigma_{\mathrm{d}}^2 \boldsymbol{\Psi}_{\mathrm{d}}(\boldsymbol{X}_{\mathrm{e}}, \boldsymbol{X}_{\mathrm{e}}) \end{pmatrix}. \tag{3.2}$$

The correlations are of the same form as equation (2.2), but there are two correlations, ψ_{c} and ψ_{d}, and we therefore have more hyper-parameters to estimate: $\boldsymbol{\theta}_{\mathrm{c}}$; $\boldsymbol{\theta}_{\mathrm{d}}$; $\boldsymbol{p}_{\mathrm{c}}$; $\boldsymbol{p}_{\mathrm{d}}$; and the scaling parameter ρ. Our cheap data are considered to be independent of the expensive data and we can find MLEs for μ_{c}, σ_{c}^2, $\boldsymbol{\theta}_{\mathrm{c}}$ and $\boldsymbol{p}_{\mathrm{c}}$ by maximizing the ln-likelihood (ignoring constant terms)

$$-\frac{n_{\mathrm{c}}}{2}\ln(\sigma_{\mathrm{c}}^2) - \frac{1}{2}\ln|\det(\boldsymbol{\Psi}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{c}}, \boldsymbol{X}_{\mathrm{c}}))| - \frac{(\boldsymbol{y}_{\mathrm{c}} - \boldsymbol{1}\mu_{\mathrm{c}})^{\mathrm{T}} \boldsymbol{\Psi}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{c}}, \boldsymbol{X}_{\mathrm{c}})^{-1} (\boldsymbol{y}_{\mathrm{c}} - \boldsymbol{1}\mu_{\mathrm{c}})}{2\sigma_{\mathrm{c}}^2}. \tag{3.3}$$

By setting the derivatives of equation (3.3) w.r.t. μ_c and σ_c^2 to 0 and solving, we find MLEs of

$$\hat{\mu}_\mathrm{c} = \mathbf{1}^\mathrm{T}\boldsymbol{\Psi}_\mathrm{c}(\boldsymbol{X}_\mathrm{c},\boldsymbol{X}_\mathrm{c})^{-1}\boldsymbol{y}_\mathrm{c}/\mathbf{1}^\mathrm{T}\boldsymbol{\Psi}_\mathrm{c}(\boldsymbol{X}_\mathrm{c},\boldsymbol{X}_\mathrm{c})^{-1}\mathbf{1} \tag{3.4}$$

and

$$\hat{\sigma}_\mathrm{c}^2 = (\boldsymbol{y}_\mathrm{c}-\mathbf{1}\hat{\mu}_\mathrm{c})^\mathrm{T}\boldsymbol{\Psi}_\mathrm{c}(\boldsymbol{X}_\mathrm{c},\boldsymbol{X}_\mathrm{c})^{-1}(\boldsymbol{y}_\mathrm{c}-\mathbf{1}\hat{\mu}_\mathrm{c})/n_\mathrm{c}. \tag{3.5}$$

Substituting equations (3.4) and (3.5) into (3.3) yields the concentrated ln-likelihood

$$-\frac{n_\mathrm{c}}{2}\ln(\hat{\sigma}_\mathrm{c}^2)-\frac{1}{2}\ln|\det(\boldsymbol{\Psi}_\mathrm{c}(\boldsymbol{X}_\mathrm{c},\boldsymbol{X}_\mathrm{c}))| \tag{3.6}$$

and $\hat{\boldsymbol{\theta}}_\mathrm{c}$ and $\hat{\boldsymbol{p}}_\mathrm{c}$ (if not set at $\mathbf{2}$) are found by maximizing this equation.

To estimate the mean, variance, hyper-parameters and scaling parameter of the difference model (μ_d, σ_d^2, $\boldsymbol{\theta}_\mathrm{d}$, $\boldsymbol{p}_\mathrm{d}$ and ρ), we first define

$$\boldsymbol{d} = \boldsymbol{y}_\mathrm{e}-\rho\boldsymbol{y}_\mathrm{c}(\boldsymbol{X}_\mathrm{e}), \tag{3.7}$$

where $\boldsymbol{y}_\mathrm{c}(\boldsymbol{X}_\mathrm{e})$ are the values of $\boldsymbol{y}_\mathrm{c}$ at locations common to those of $\boldsymbol{X}_\mathrm{e}$ (the Markov property implies that we need to consider only this data). If $\boldsymbol{y}_\mathrm{c}$ is not available at $\boldsymbol{X}_\mathrm{e}$, we may estimate ρ at little additional cost by using kriging estimates $\hat{\boldsymbol{y}}_\mathrm{c}(\boldsymbol{X}_\mathrm{e})$, found from equation (2.3), using the already determined hyper-parameters $\hat{\boldsymbol{\theta}}_\mathrm{c}$ and $\hat{\boldsymbol{p}}_\mathrm{c}$. The ln-likelihood of $\boldsymbol{d}$, given $\boldsymbol{y}_\mathrm{c}$, is now

$$-\frac{n_\mathrm{e}}{2}\ln(\sigma_\mathrm{d}^2)-\frac{1}{2}\ln|\det(\boldsymbol{\Psi}_\mathrm{d}(\boldsymbol{X}_\mathrm{e},\boldsymbol{X}_\mathrm{e}))| \\ -\frac{(\boldsymbol{d}-\mathbf{1}\mu_\mathrm{d})^\mathrm{T}\boldsymbol{\Psi}_\mathrm{d}(\boldsymbol{X}_\mathrm{e},\boldsymbol{X}_\mathrm{e})^{-1}(\boldsymbol{d}-\mathbf{1}\mu_\mathrm{d})}{2\sigma_\mathrm{d}^2}, \tag{3.8}$$

yielding MLEs of

$$\hat{\mu}_\mathrm{d} = \mathbf{1}^\mathrm{T}\boldsymbol{\Psi}_\mathrm{d}(\boldsymbol{X}_\mathrm{e},\boldsymbol{X}_\mathrm{e})^{-1}\boldsymbol{d}/\mathbf{1}^\mathrm{T}\boldsymbol{\Psi}_\mathrm{d}(\boldsymbol{X}_\mathrm{e},\boldsymbol{X}_\mathrm{e})^{-1}\mathbf{1}$$

and

$$\hat{\sigma}_\mathrm{d}^2 = (\boldsymbol{d}-\mathbf{1}\hat{\mu}_\mathrm{d})^\mathrm{T}\boldsymbol{\Psi}_\mathrm{d}(\boldsymbol{X}_\mathrm{e},\boldsymbol{X}_\mathrm{e})^{-1}(\boldsymbol{d}-\mathbf{1}\hat{\mu}_\mathrm{d})/n_\mathrm{e}, \tag{3.9}$$

with $\hat{\boldsymbol{\theta}}_\mathrm{d}$, $\hat{\boldsymbol{p}}_\mathrm{d}$ (again, if not set at $\mathbf{2}$) and $\hat{\rho}$ found by maximizing

$$-\frac{n_\mathrm{e}}{2}\ln(\hat{\sigma}_\mathrm{d}^2)-\frac{1}{2}\ln|\det(\boldsymbol{\Psi}_\mathrm{d}(\boldsymbol{X}_\mathrm{e},\boldsymbol{X}_\mathrm{e}))|. \tag{3.10}$$

Equations (3.6) and (3.10) must be maximized numerically using a suitable global search routine such as a genetic algorithm (GA). Depending upon the cost of evaluating the cheap and expensive functions f_c and f_e, for very high dimensional problems, the multiple matrix inversions involved in the likelihood maximization may render the use of the co-kriging model impractical (the size of the matrices depends directly on the quantities of data available, and the number of search steps needed in the MLE process is linked to the number of hyper-parameters being tuned). Typically, a statistical model used as a surrogate will be tuned many fewer times than the number of evaluations of f_e required by a direct search. The cost of tuning the model can therefore be allowed to exceed the cost of computing f_e and still provide significant speed-up. For large k and n, the time required to find MLEs can be reduced by using a constant $\theta_{\mathrm{c},j}$ and $\theta_{\mathrm{d},j}$ for all elements of $\boldsymbol{\theta}_{\mathrm{c}}$ and $\boldsymbol{\theta}_{\mathrm{d}}$ to simplify the maximization, though this may affect the accuracy of the approximation.

With the hyper-parameters estimated, the co-kriging prediction of the expensive function is given by

$$\hat{y}_{\mathrm{e}}(\boldsymbol{x}^{(n_{\mathrm{e}}+1)}) = \hat{\mu} + \boldsymbol{c}^{\mathrm{T}} \boldsymbol{C}^{-1}(\boldsymbol{y} - \mathbf{1}\hat{\mu}), \tag{3.11}$$

where

$$\boldsymbol{c} = \begin{pmatrix} \hat{\rho}\hat{\sigma}_{\mathrm{c}}^2 \psi_{\mathrm{c}}\big(\boldsymbol{X}_{\mathrm{c}}, \boldsymbol{x}^{(n+1)}\big) \\ \hat{\rho}^2\hat{\sigma}_{\mathrm{c}}^2 \psi_{\mathrm{c}}\big(\boldsymbol{X}_{\mathrm{e}}, \boldsymbol{x}^{(n+1)}\big) + \hat{\sigma}_{\mathrm{d}}^2 \psi_{\mathrm{d}}\big(\boldsymbol{X}_{\mathrm{e}}, \boldsymbol{x}^{(n+1)}\big) \end{pmatrix}$$

and $\hat{\mu} = \mathbf{1}^{\mathrm{T}}\boldsymbol{C}^{-1}\boldsymbol{y}/\mathbf{1}^{\mathrm{T}}\boldsymbol{C}^{-1}\mathbf{1}$. The notation $\psi_{\mathrm{c}}\big(\boldsymbol{X}_{\mathrm{e}}, \boldsymbol{x}^{(n+1)}\big)$, for example, denotes a column vector of correlations of the form ψ_{c} between the data $\boldsymbol{X}_{\mathrm{e}}$ and the new point $\boldsymbol{x}^{(n+1)}$. The derivation of equation (3.11) is given in appendix A. If we make a prediction at one of our expensive points, $\boldsymbol{x}^{(n+1)} = \boldsymbol{x}_{\mathrm{e}}^{(i)}$ and $\boldsymbol{c}$ is the $n_{\mathrm{c}} + i$th column of $\boldsymbol{C}$, then $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{C}^{-1}$ is the $n_{\mathrm{c}} + i$th unit vector and $\hat{y}_{\mathrm{e}}(\boldsymbol{x}_{\mathrm{e}}^{(i)}) = \hat{\mu} + \boldsymbol{y}_{(n_{\mathrm{c}}+i)} - \hat{\mu} = y_{\mathrm{e}}^{(i)}$. We see, therefore, that equation (3.11) is an interpolator of the expensive data. If we make a prediction at one of the cheap points, $\boldsymbol{x}^{(n+1)} = \boldsymbol{x}_{\mathrm{c}}^{(i)}$, $\boldsymbol{c}$ is not a column of $\boldsymbol{C}$ and the predictor will in some sense regress $\boldsymbol{y}_{\mathrm{c}}$ unless it coincides with $\boldsymbol{y}_{\mathrm{e}}$.

The estimated mean-squared error in the predictor is given as follows:

$$s^2(\boldsymbol{x}) = \hat{\rho}^2\hat{\sigma}_{\mathrm{c}}^2 + \hat{\sigma}_{\mathrm{d}}^2 - \boldsymbol{c}^{\mathrm{T}}\boldsymbol{C}^{-1}\boldsymbol{c}. \tag{3.12}$$

Again, a derivation is given in appendix A. Since the co-kriging model is an interpolator of $\boldsymbol{y}_{\mathrm{e}}$, we expect the error to be zero at the expensive sample points. For $\boldsymbol{x}^{(n_{\mathrm{e}}+1)} = \boldsymbol{x}_{\mathrm{e}}^{(i)}$, $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{C}^{-1}$ is the $n_{\mathrm{c}} + i$th unit vector, $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{C}^{-1}\boldsymbol{c} = \boldsymbol{c}^{(n_{\mathrm{c}}+i)} = \rho_{\mathrm{c}}^2\sigma_{\mathrm{c}}^2 + \sigma_{\mathrm{d}}^2$ and hence $s^2(\boldsymbol{x})$ is indeed 0. For $\boldsymbol{X}_{\mathrm{c}} \backslash \boldsymbol{X}_{\mathrm{e}}$, $s^2(\boldsymbol{x}) \neq 0$ unless $\boldsymbol{y}_{\mathrm{e}} = \boldsymbol{y}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{e}})$. The error at these points is determined by the character of $\boldsymbol{Y}_{\mathrm{d}}$. If the difference between $\rho\,\boldsymbol{Y}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{e}})$ and $\boldsymbol{Y}_{\mathrm{e}}(\boldsymbol{X}_{\mathrm{e}})$ is simple (characterized by low $\theta_{\mathrm{d},j}$ values), the error will be low, whereas a more complex difference (high $\theta_{\mathrm{d},j}$ values) will lead to high error estimates.

(*a*) *One-variable demonstration*

We will now look at how co-kriging behaves using an example of a simple one-variable function. Imagine that our expensive to compute data are calculated by the function $f_{\mathrm{e}}(x) = (6x-2)^2\sin(12x-4)$, $x \in \{0,1\}$, and a cheaper estimate of this data is given by $f_{\mathrm{c}}(x) = Af_{\mathrm{e}} + B(x-0.5) - C$. We sample the design space extensively using the cheap function at $\boldsymbol{X}_{\mathrm{c}} = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$, but run the expensive function only at four of these points, $\boldsymbol{X}_{\mathrm{e}} = \{0, 0.4, 0.6, 1\}$.

Figure 1 shows the functions f_{e} and f_{c} with $A=0.5$, $B=10$ and $C=-5$. A kriging prediction through $\boldsymbol{y}_{\mathrm{e}}$ gives a poor approximation to the deliberately deceptive function, but the co-kriging prediction lies very close to f_{e}, being better than both the standard kriging model and the cheap data. Despite the considerable differences between f_{e} and f_{c}, a simple relationship has been found between the expensive and the cheap data and the estimated error reduces almost to 0 at $\boldsymbol{X}_{\mathrm{c}}$ (figure 2).

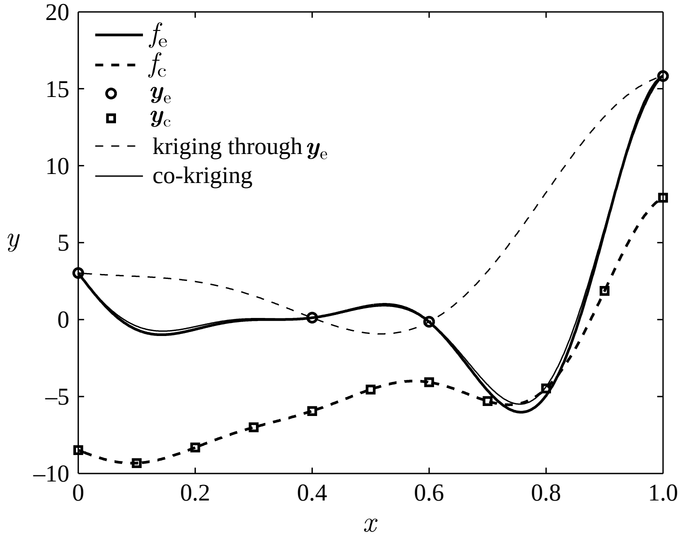

Figure 1. A one-variable co-kriging example. The kriging approximation using four expensive data points ($\boldsymbol{y}_e$) has been significantly improved using extensive sampling from the cheap function ($\boldsymbol{y}_c$).

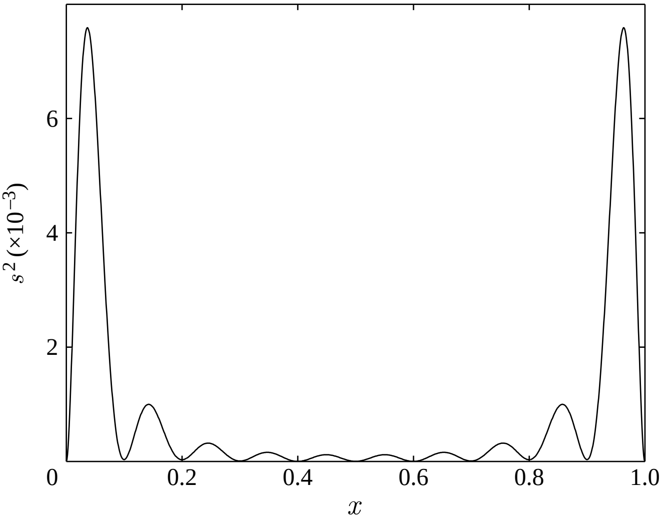

Figure 2. Estimated error in the co-kriging prediction in figure 1. The simple relationship between the data results in low error estimates at $\boldsymbol{X}_c$ as well as $\boldsymbol{X}_e$.

We have chosen the relationship between our low- and high-fidelity test functions in order to show how the hyper-parameters of the co-kriging model behave. The hyper-parameters pertaining to the cheap data, $\boldsymbol{\theta}_c$ and $\boldsymbol{p}_c$, are affected only by this data and behave as described in §2. Moving on to the scaling parameter ρ, if our cheap model parameter A (the multiplying term linking the cheap and expensive functions) is varied such that $1/A \in \{-10, 10\}$, we obtain the values for $\hat{\rho}$ shown in figure 3 and see that $\hat{\rho} \simeq 1/A$. Similar trials show that the parameters B and C have no effect on ρ; thus we see that ρ is purely a scaling parameter. Note that $\hat{\rho}$ is only an *indicator* of the scaling, since this value is estimated based on the data

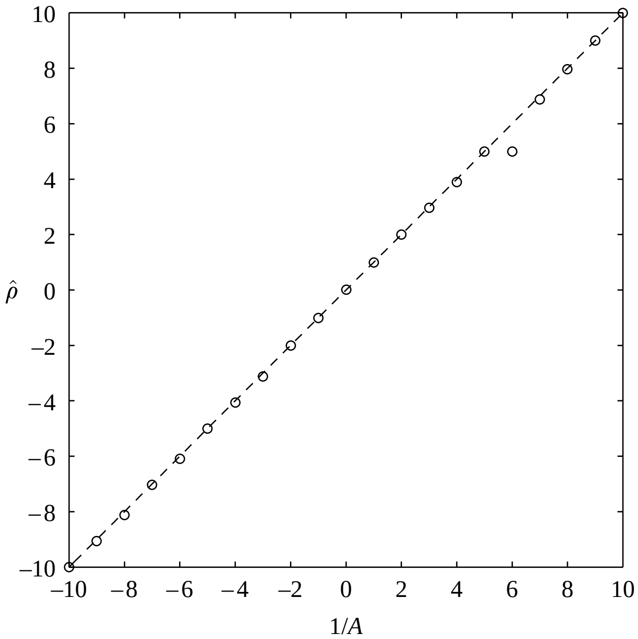

Figure 3. This plot of $\hat{\rho}$ versus $1/A$ shows that the MLE for ρ is a scaling factor between $Z_{\mathrm{c}}(\cdot)$ and $Z_{\mathrm{e}}(\cdot)$, following the formulation in equation (3.1). There is a singularity at $1/A=0$, hence we have used $1/A=0.01$ at this point.

available. For the data in figure 1, $\hat{\rho}=1.87$ (close to the true value of 2), but for small samples of $\boldsymbol{y}_{\mathrm{e}}$ the MLE may be misleading. The slight deviations of the data in figure 3 from $\hat{\rho}=1/A$ (shown as a dashed line), the most significant of which is at $1/A=6$, are where our GA search has not found the true MLE.

Recall that $\boldsymbol{d}=\boldsymbol{y}_{\mathrm{e}}-\rho\boldsymbol{y}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{e}})$ (equation (3.7)) and hence, with $\hat{\rho}\approx\boldsymbol{y}_{\mathrm{e}}/\boldsymbol{y}_{\mathrm{c}}$, $\boldsymbol{d}$ represents the difference in trends between the cheap and the expensive data. Thus, for our one-variable example, when B, $C=0$, $\hat{\mu}_{\mathrm{d}}, \hat{\sigma}_{\mathrm{d}}^2\rightarrow 0$ for all values of A, if $\hat{\rho}$ is estimated accurately. Figure 4 shows how $\hat{\sigma}_{\mathrm{d}}^2$ varies for $B\in\{-10,10\}$ and we see that as $B\rightarrow 0$ and therefore $\boldsymbol{d}\rightarrow\boldsymbol{0}$, $\hat{\sigma}_{\mathrm{d}}^2$ also approaches 0. Note that $\hat{\boldsymbol{\theta}}_{\mathrm{e}}$ and $\hat{\boldsymbol{p}}_{\mathrm{e}}$ will not be affected since the correlation in equation (2.2) is unaffected by the scaling of the objective data (it is, however, affected by the scaling of $\boldsymbol{X}$).

Our choice of cheap function for the above example is somewhat contrived, but this has allowed us to show that the co-kriging model and its parameters are behaving as we would expect. For our test function, the correction process $Z_{\mathrm{d}}(\cdot)$ is linear. Co-kriging will work effectively for more complex correction processes with the proviso that $Z_{\mathrm{d}}(\cdot)$ must be simpler to model than $Z_{\mathrm{e}}(\cdot)$. In §6 we demonstrate the benefits of co-kriging using an engineering design problem.

4. Sampling plans

In §3 we have shown how to build the co-kriging model based on a set of sample data. We now consider how to choose the points $\boldsymbol{X}_{\mathrm{c}}$ and $\boldsymbol{X}_{\mathrm{e}}$ to give us the best prediction $\hat{y}_{\mathrm{e}}$.

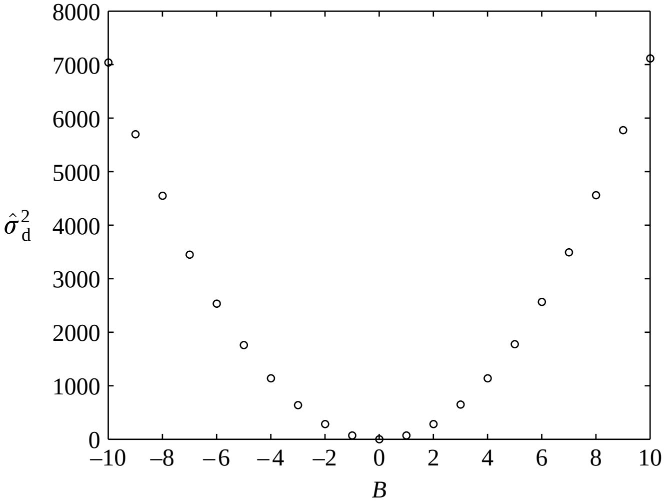

Figure 4. Variance $\hat{\sigma}^2_{\rm d}$ as the cheap function coefficient B is altered. $\hat{\sigma}^2_{\rm d}$, reduces to 0 as the difference between $f_{\rm e}$ and $f_{\rm c}$ can be modelled purely by the scaling parameter ρ.

The results of experiments, whether physical or computational, are corrupted by experimental error, that is, they deviate from the 'true' result. This can be due to human error, systematic deviations (due to a flaw or an inevitable physical or numerical limitation of the experiment) and, *only in physical experiments*, random error (usually linked to the limited accuracy of the instruments). When deciding on the sampling plan, that is, when choosing $\boldsymbol{X}$, there is nothing we can do about the first two error components and, while physical sampling plans often include replicates to reduce the random error, computer experiment plans are simply designed to cover the parameter space reasonably uniformly. The goal is, of course, to enable the model fitted to these points to give an accurate global approximation of the unknown objective function landscape.

The definition of 'uniform coverage' is by no means obvious and a substantial body of literature exists on the subject—here we work with the optimality criterion of Morris & Mitchell (1995). According to this, the plan with the best space-filling properties is that which maximizes the smallest distance between any pair of points within the sample (several additional 'tie-breaker' criteria are given in case of multiple optima). Additionally, we restrict our search to a class of plans known as Latin Hypercubes (LH; McKay *et al.* 1979), which are built to ensure uniformly spread projections of all points on all axes (figure 5*a* shows a random LH). We thus generate an initial Morris–Mitchell optimal LH plan $\boldsymbol{X}_{\rm c}$.

A more interesting question is how do we select an $n_{\rm e}$-element subset $\boldsymbol{X}_{\rm e}$ of $\boldsymbol{X}_{\rm c}$, where the expensive simulations are to be run? Again, we wish to cover the parameter space evenly and hence turn to the Morris–Mitchell criterion, but this time we are dealing with a limited, discrete parameter space and thus the problem becomes a combinatorial one. Since selecting the subset that satisfies this is an *NP-complete* problem and an exhaustive search would have to examine ${}_{n_{\rm c}}C_{n_{\rm e}} = n_{\rm c}!/n_{\rm e}!(n_{\rm c}-n_{\rm e})!$ subsets (clearly infeasible for all but very moderate cardinalities), here we use an exchange algorithm to select $\boldsymbol{X}_{\rm e}$ (Cook & Nachtsheim 1980).

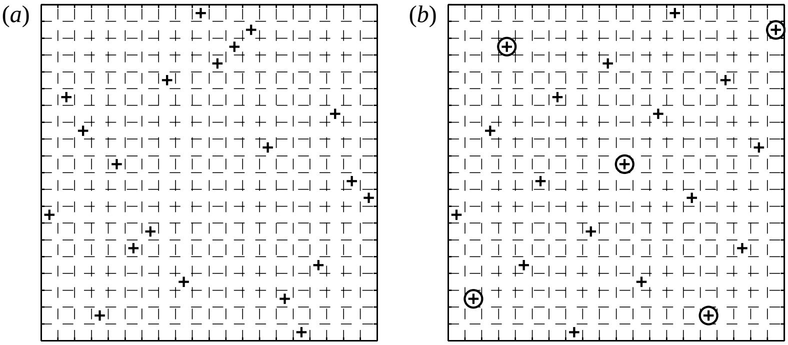

Figure 5. Two 20-point sampling plans: (*a*) a random LH and (*b*) a Morris–Mitchell optimal LH (plus symbol) with a 5-point subset found using the exchange algorithm (circle).

We start from a randomly selected subset $\boldsymbol{X}_\mathrm{e}$ and calculate the Morris–Mitchell criterion. We then exchange the first point $\boldsymbol{x}_\mathrm{e}^{(1)}$ with each of the remaining points in $\boldsymbol{X}_\mathrm{c}\backslash\boldsymbol{X}_\mathrm{e}$ and retain the exchange that gives the best Morris–Mitchell criterion. This process is repeated for each remaining point $\boldsymbol{x}_\mathrm{e}^{(2)}\ldots\boldsymbol{x}_\mathrm{e}^{(n_\mathrm{e})}$. A number of restarts from different initial subsets can be employed to avoid local optima. Figure 5*b* shows a Morris–Mitchell optimal LH with a subset chosen using this exchange algorithm.

A rule of thumb for the number of points that should be used in the sampling plan is $n=10k$. When using a particularly cheap analysis, n_c may be rather greater than this, allowing us to build a more accurate model, and if the relationship between f_c and f_e is simple, n_e may be somewhat fewer—the advantage of the co-kriging method. We will return to this subject in §7.

5. Co-kriging based optimization

In order to confirm and enhance the predictions of a surrogate model, it is usual to update the model with new evaluations in promising areas. Co-kriging (and kriging) treats the value of the function at $\boldsymbol{x}$ as if it were the realization of a Gaussian random variable $Y(\boldsymbol{x})$, with a probability density function

$$\frac{1}{\sqrt{2\pi}s(\boldsymbol{x})}\exp-\frac{1}{2}\left(\frac{Y_\mathrm{e}(\boldsymbol{x})-\hat{y}_\mathrm{e}(\boldsymbol{x})}{s(\boldsymbol{x})}\right)^2,$$

with mean given by the predictor, $\hat{y}_\mathrm{e}(\boldsymbol{x})$ (equation (3.11)) and variance, $s^2(\boldsymbol{x})$ (equation (3.12)). This allows us to model our uncertainty about the predictions we make. The most plausible value at $\boldsymbol{x}$ is $\hat{y}_\mathrm{e}(\boldsymbol{x})$, with the probability decreasing as $Y_\mathrm{e}(\boldsymbol{x})$ moves away from $\hat{y}_\mathrm{e}(\boldsymbol{x})$. Since there is an uncertainty in the value of $\hat{y}_\mathrm{e}(\boldsymbol{x})$ we can calculate the expectation of it being an improvement, $I=\min\{\boldsymbol{y}_\mathrm{e}\}-Y(\boldsymbol{x})$, on the best value calculated so far as

$$\begin{aligned}E[I(\boldsymbol{x})] &= \int_{-\infty}^{\infty}\max\{\min\{\boldsymbol{y}_\mathrm{e}\}-Y_\mathrm{e}(\boldsymbol{x}),0\}\phi(Y_\mathrm{e}(\boldsymbol{x}))\,\mathrm{d}Y_\mathrm{e}\\ &= \begin{cases}(\min\{\boldsymbol{y}_\mathrm{e}\}-\hat{y}_\mathrm{e}(\boldsymbol{x}))\Phi\left(\dfrac{\min\{\boldsymbol{y}_\mathrm{e}\}-\hat{y}_\mathrm{e}(\boldsymbol{x})}{s(\boldsymbol{x})}\right)+s(\boldsymbol{x})\phi\left(\dfrac{\min\{\boldsymbol{y}_\mathrm{e}\}-\hat{y}_\mathrm{e}(\boldsymbol{x})}{s(\boldsymbol{x})}\right) & \text{if} \quad s>0,\\ 0 & \text{if} \quad s=0,\end{cases}\end{aligned}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are the normal cumulative distribution function and probability density function, respectively. By maximizing $E[I(\boldsymbol{x})]$ we can find the best new point at which to sample the design space. Note that $E[I(\boldsymbol{x})]=0$ when $s(\boldsymbol{x})=0$ so that there is no expectation of improvement at a point that has already been sampled and therefore no possibility of re-sampling, which is a necessary characteristic of an updating criterion when using deterministic computer experiments, and guarantees global convergence; given that there is no possibility of re-sampling, as the number of updates based on the maximum $E[I(\boldsymbol{x})]$ increases, the design space will become densely populated and so the global optimum will be found (Locatelli 1997).

(*a*) *Co-kriging regression*

When using multi-fidelity analyses, we can modify the interpolating co-kriging formulation in §3 such that each analysis can be regressed appropriately to filter any noise present in the data. Different levels of filtering may be required for each analysis. For example, data from an empirical low-fidelity computer code may be smooth and require no regression, but could be coupled to a discretized high-fidelity analysis that displays noise that must be filtered using regression. Conversely, a low-fidelity model may be discretized on a much coarser mesh, requiring a higher degree of regression than the high-fidelity code. In our regressing co-kriging formulation, we therefore employ two regression constants, λ_{c} and λ_{e}. These are added to the leading diagonal of the correlation matrices to give the covariance matrix $\boldsymbol{C}+\boldsymbol{\lambda}$,

$$\begin{pmatrix} \sigma_{\mathrm{c}}^2\{\boldsymbol{\Psi}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{c}},\boldsymbol{X}_{\mathrm{c}})+\boldsymbol{I}_{(n_{\mathrm{c}}\times n_{\mathrm{c}})}\lambda_{\mathrm{c}}\} & \rho\sigma_{\mathrm{c}}^2\{\boldsymbol{\Psi}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{c}},\boldsymbol{X}_{\mathrm{e}})+\begin{pmatrix}\boldsymbol{0}_{(n_{\mathrm{c}}-n_{\mathrm{e}}\times n_{\mathrm{e}})}\\ \boldsymbol{I}_{(n_{\mathrm{e}}\times n_{\mathrm{e}})}\end{pmatrix}\lambda_{\mathrm{c}}\} \\ \rho\sigma_{\mathrm{c}}^2\{\boldsymbol{\Psi}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{e}},\boldsymbol{X}_{\mathrm{c}}) & \rho^2\sigma_{\mathrm{c}}^2\{\boldsymbol{\Psi}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{e}},\boldsymbol{X}_{\mathrm{e}})+\boldsymbol{I}_{(n_{\mathrm{e}}\times n_{\mathrm{e}})}\lambda_{\mathrm{c}}\} \\ +\left(\boldsymbol{0}_{(n_{\mathrm{e}}\times n_{\mathrm{c}}-n_{\mathrm{e}})}\boldsymbol{I}_{(n_{\mathrm{e}}\times n_{\mathrm{e}})}\right)\lambda_{\mathrm{c}}\} & +\sigma_{\mathrm{d}}^2\{\boldsymbol{\Psi}_{\mathrm{d}}(\boldsymbol{X}_{\mathrm{e}},\boldsymbol{X}_{\mathrm{e}})+\boldsymbol{I}_{(n_{\mathrm{e}}\times n_{\mathrm{e}})}\lambda_{\mathrm{e}}\} \end{pmatrix}, \qquad (5.1)$$

where $\boldsymbol{I}$ is an identity matrix and $\boldsymbol{0}$ a zero matrix. The values of λ_{c} and λ_{e} are found along with the other hyper-parameters by maximizing equations (3.3) and (3.8). The regressing predictor is now

$$\hat{y}_{\mathrm{e,r}}(\boldsymbol{x}^{(n_{\mathrm{e}}+1)})=\hat{\mu}+\boldsymbol{c}^{\mathrm{T}}(\boldsymbol{C}+\boldsymbol{\lambda})^{-1}(\boldsymbol{y}-\boldsymbol{1}\hat{\mu}), \qquad (5.2)$$

(subscript r denotes regression) with an estimated mean-squared error

$$s_{\mathrm{r}}^2(\boldsymbol{x})=\hat{\rho}^2\hat{\sigma}_{\mathrm{c}}^2\hat{\lambda}_{\mathrm{c}}+\hat{\sigma}_{\mathrm{d}}^2\hat{\lambda}_{\mathrm{e}}-\boldsymbol{c}^{\mathrm{T}}(\boldsymbol{C}+\boldsymbol{\lambda})^{-1}\boldsymbol{c}. \qquad (5.3)$$

Unlike equation (3.12), the regression error does not fall to 0 at sample points, and therefore Locatelli's proof of convergence is invalid because $\max\{E[I(\boldsymbol{x})]\}$ can occur at a previously sampled point. This is satisfactory when f_{c} and/or f_{e} are evaluated through physical experiments, since repeated experiments may be required to reduce measurement error. If f_{c} and f_{e} are found using deterministic computer experiments, re-sampling should be avoided and the convergence properties of $\max\{E[I(\boldsymbol{x})]\}$ (or other error-based sampling strategies) can be restored by modifying the estimated error to eliminate the error due to the noise filtering, since this is the error associated with the data rather than the quality of the model. This is achieved by fitting a kriging interpolation through regressed data, re-interpolation.

(*b*) *Co-kriging re-interpolation*

We now show the derivation of a co-kriging error estimate that reflects the uncertainty in predicting the underlying trend of data, rather than the overall uncertainty, which includes any error in noisy data. The derivation, although naturally complicated by multiple sets of data and regression constants, follows the same theme as for kriging re-interpolation (Forrester *et al.* 2006*b*). As mentioned above, to eliminate the error in the noisy data, we need to calculate the error of an interpolation through a regressed set of data. The column vectors of regressed cheap and expensive data are found from equations (2.3) (with the addition of the regression constant) and (5.2), and can be expressed as

$$\hat{\boldsymbol{y}}_{\mathrm{c,r}} = \mathbf{1}\hat{\mu}_{\mathrm{c}} + \boldsymbol{\Psi}_{\mathrm{c}}(\boldsymbol{\Psi}_{\mathrm{c}} + \hat{\lambda}_{\mathrm{c}}\boldsymbol{I})^{-1}(\boldsymbol{y}_{\mathrm{c}} - \mathbf{1}\hat{\mu}_{\mathrm{c}})$$

and

$$\hat{\boldsymbol{y}}_{\mathrm{e,r}} = \mathbf{1}_{(n_{\mathrm{e}}\times 1)}\hat{\mu} + \{\boldsymbol{c}(\boldsymbol{x}_{\mathrm{e}}^{(1)})^{\mathrm{T}}, \ldots, \boldsymbol{c}(\boldsymbol{x}_{\mathrm{e}}^{(n_{\mathrm{e}})})^{\mathrm{T}}\}^{\mathrm{T}}(\boldsymbol{C} + \boldsymbol{\lambda})^{-1}(\boldsymbol{y} - \mathbf{1}_{(n\times 1)}\hat{\mu}),$$

where the MLEs for the hyper-parameters are those found for the co-kriging model.

Note that only the $\boldsymbol{\sigma}^2$ terms in equation (3.12) depend on $\boldsymbol{y}$. We therefore need to substitute $\hat{\boldsymbol{y}}_{\mathrm{c,r}}$ into equation (3.5) and $\hat{\boldsymbol{y}}_{\mathrm{e,r}}$ into equation (3.9). Beginning with $\hat{\sigma}^2_{\mathrm{c}}$,

$$\hat{\sigma}^2_{\mathrm{c,ri}} = \frac{(\boldsymbol{y}_{\mathrm{c}} - \mathbf{1}\hat{\mu}_{\mathrm{c}})(\boldsymbol{\Psi}_{\mathrm{c}} + \lambda_{\mathrm{c}}\boldsymbol{I})^{-1}\boldsymbol{\Psi}_{\mathrm{c}}(\boldsymbol{\Psi}_{\mathrm{c}} + \lambda_{\mathrm{c}}\boldsymbol{I})^{-1}(\boldsymbol{y}_{\mathrm{c}} - \mathbf{1}\hat{\mu}_{\mathrm{c}})}{n_{\mathrm{c}}} \tag{5.4}$$

(subscript ri denotes re-interpolation). Note that if $\lambda_{\mathrm{c}}=0$, then $\hat{\sigma}^2_{\mathrm{c,ri}} = \hat{\sigma}^2_{\mathrm{c}}$. To find $\hat{\sigma}^2_{\mathrm{d,ri}}$, we substitute $\hat{\boldsymbol{y}}_{\mathrm{e,r}}$ and $\hat{\boldsymbol{y}}_{\mathrm{c,r}}$ into equation (3.7) to give

$$\boldsymbol{d}_{\mathrm{r}} = \mathbf{1}_{(n_{\mathrm{e}}\times 1)}\hat{\mu} + \{\boldsymbol{c}(\boldsymbol{x}_{\mathrm{e}}^{(1)})^{\mathrm{T}}, \ldots, \boldsymbol{c}(\boldsymbol{x}_{\mathrm{e}}^{(n_{\mathrm{e}})})^{\mathrm{T}}\}^{\mathrm{T}}(\boldsymbol{C} + \boldsymbol{\lambda})^{-1}(\boldsymbol{y} - \mathbf{1}_{(n\times 1)}\hat{\mu})$$
$$- \hat{\rho}\left[\mathbf{1}\hat{\mu}_{\mathrm{c}} + \boldsymbol{\Psi}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{e}})(\boldsymbol{\Psi}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{e}}) + \lambda_{\mathrm{c}}\boldsymbol{I})^{-1}(\boldsymbol{y}_{\mathrm{c}}(\boldsymbol{X}_{\mathrm{e}}) - \mathbf{1}\hat{\mu}_{\mathrm{c}})\right].$$

This is substituted into equation (3.9) in place of $\boldsymbol{d}$ (again, note that if $\lambda_{\mathrm{e}}=0$ then $\hat{\sigma}^2_{\mathrm{d,ri}} = \hat{\sigma}^2_{\mathrm{d}}$). The expressions for $\hat{\sigma}^2_{\mathrm{c,ri}}$ and $\hat{\sigma}^2_{\mathrm{d,ri}}$ can now be substituted into equation (3.12) to find the re-interpolation error estimate that reduces to 0 at $\boldsymbol{X}_{\mathrm{e}}$. With the errors due to noisy data removed, the estimated error of the predicted smooth trend is typically very small when compared with an interpolating model. This can lead to problems with floating-point underflow when calculating $E[I(\boldsymbol{x})]$. A scheme for avoiding this problem is presented in appendix B.

6. Example problem

We will demonstrate the benefits of co-kriging, and in particular regressing co-kriging, through the optimization of a generic transonic civil aircraft wing. The wing is defined by 11 variables: area; aspect ratio; kink position; sweep; inboard and outboard taper ratios; root, kink and tip thickness to chord ratio; tip washout; and kink washout. Our cheap analysis is in the form of an empirical drag estimation code, *Tadpole* (Cousin & Metcalf 1990). This returns a drag value based on curve fits to previously analysed wings in approximately 0.6 s. Our ‘expensive’ code is the linearized potential method *VSaero* (Maskew 1982) with viscous coupling (approx. 2 min per drag evaluation). We aim to minimize drag/dynamic pressure (D/q) for a fixed lift (determined by a wing weight calculated by Tadpole).

To allow visualization of the design landscape, we limit the search to the four variables that have the most impact on drag: area, S; aspect ratio, $\mathcal{R}$; sweep, Λ; and inboard taper ratio, T_{in} (these variables were shown to be the most dominant by Keane (2003)). The Tadpole and VSaero design landscapes are shown using hierarchical axis plots in figures 6 and 7. With the fast empirical drag estimation provided by Tadpole, we have been able to build a high-resolution plot from $11^4 = 14\,641$ runs of the code in 2.3 hours, while using the slower physics-based VSaero we have produced a low-resolution plot using $3^2 \times 11^2 = 1089$ evaluations in 34.5 hours.[4] Each tile of the plots shows D/q for $S \in \{150, 250\} m^2$ and $\mathcal{R} \in \{6, 12\}$ for a fixed Λ and T_{in}. Λ and T_{in} vary from tile to tile, with the value at the lower left corner of the tile representing the value for the entire tile. The blank portions of figure 7 are regions where VSaero has failed to return a result for unusual geometries that lead to extreme flow regimes.[5] It is seen that the two codes produce results that follow the same general trend of lower D/q for higher $\mathcal{R}$ and T_{in}, but the global optimum of the VSaero landscape goes against this general trend, with the lowest D/q at $T_{\text{in}} = 0.55$. Z_{d} will not, therefore, be a trivial function as in our one-variable example but, since the cheap and expensive landscapes have the same general trend, should be simpler than Z_{e}.

For this problem, we start with an initial Morris–Mitchell sampling plan for $\boldsymbol{X}_{\text{c}}$ of 100 points (note $n_{\text{c}} > 10k$), to which a kriging model is fitted. A comparison of 100 further Tadpole calculations with predictions from the kriging model yields a high correlation coefficient of 0.96, indicating that the cheap code is approximated well. A subset of $n_{\text{e}} = 20$ points (note $n_{\text{e}} < 10k$) is selected (using the exchange algorithm) from which to build the initial co-kriging model. $\boldsymbol{X}_{\text{c}}$ and $\boldsymbol{X}_{\text{e}}$ are shown in figure 6. Despite the apparent sparseness of the $\boldsymbol{X}_{\text{e}}$ data, a good initial co-kriging prediction of the VSaero landscape is obtained, with a correlation coefficient of 0.96 when compared with the $11^2 \times 3^2$ VSaero dataset. We follow an iterative process of updating the co-kriging model with new VSaero and Tadpole data at $\max\{E[I(\boldsymbol{x})]\}$ and retuning hyper-parameters until the optimum is found. Since n_{c} is large and the correlation with independent data is high, we do not expect the Tadpole landscape to be altered unduly by updates. The GA-based tuning of $\boldsymbol{\theta}_{\text{c}}$ and λ_{c} requires a large number of computationally intensive inversions of the large $n_{\text{c}} \times n_{\text{c}}$ covariance matrix. We therefore save time by limiting our re-tuning of the cheap hyper-parameters $\boldsymbol{\theta}_{\text{c}}$ and λ_{c} to every 10 updates, while we re-tune $\boldsymbol{\theta}_{\text{e}}$ and λ_{e} at each step.

The co-kriging method is compared with a $\max\{E[I(\boldsymbol{x})]\}$ search of a kriging model built using only the VSaero data. A D/q that improves on the minimum from the previously computed $3^2 \times 11^2$ set of data plus one standard deviation of the noise exhibited by the VSaero results (found from 10 VSaero evaluations of the same design with small perturbations), $D/q = 2.57 + 0.027$, is used as a stopping

[4] Generating such plots would, of course, not be possible in most cases due to the high computational cost. Here we have computed this large quantity of data for illustrative purposes.

[5] In our co-kriging and kriging-based searches, we have directed the search away from regions of failure by inputing penalized values of $\hat{y}_{\text{e}}(\boldsymbol{x}_{\text{e}}) + s^2(\boldsymbol{x}_{\text{e}})$ when VSaero fails to return a result (Forrester *et al.* 2006*c*).

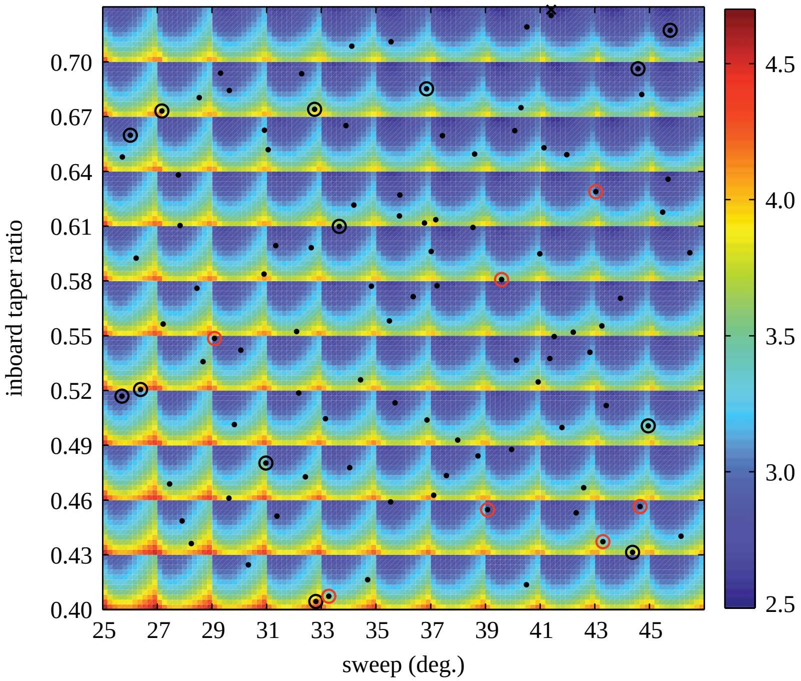

Figure 6. Tadpole calculated D/q with a typical sampling plan. $\boldsymbol{X}_{\mathrm{c}}$ (Tadpole evaluations) are shown as dots and are circled at locations of $\boldsymbol{X}_{\mathrm{e}}$ (VSaero evaluations). Red circles, failed VSaero simulations.

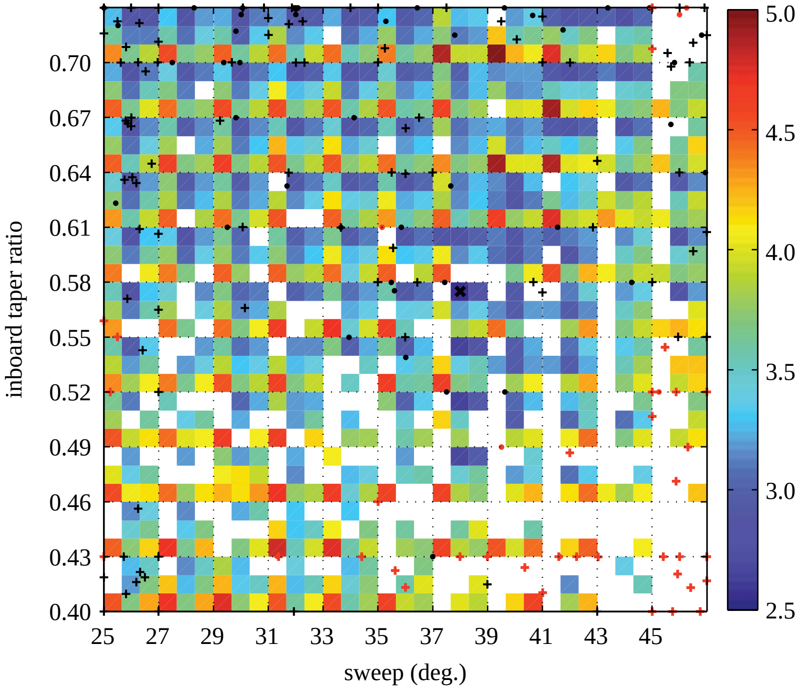

Figure 7. VSaero calculated D/q with co-kriging (dots) and kriging (crosses) updates from a typical search (the search that yielded values closest to the mean results in table 1).

Table 1. Performance comparison for the four-variable transonic wing problem. (Averaged over five searches.)

	initial model		best D/q (m^2)	function evaluations		
	r^2	RMSE	VSaero	Tadpole	succeeded VSaero	failed VSaero
kriging	0.949	0.155	2.621	0	119	40
co-kriging	0.962	0.108	2.556	185	79	27

criterion in both cases, i.e. similar quality answers are achieved from both the searches, allowing a direct comparison of the effort involved. We also limit the size of the kriging and co-kriging models to a maximum of 200 VSaero calculations. Beyond this, the time required for tuning the model becomes equivalent to the time required to run VSaero. Results, averaged over five searches from sampling plans produced using different random number seeds, are shown in table 1.

The co-kriging based search consistently outperforms the kriging-based search; finding better optima for reduced numbers of VSaero evaluations and with fewer failed simulations. The initial prediction of the kriging model is almost as accurate as the co-kriging model (see correlation and RMSE in table 1), but the greater coverage of data in the co-kriging model leads to a better selection of successful update points in promising regions, as seen in figure 7, which shows the distribution of update evaluations for a typical search. Moreover, the co-kriging updates are concentrated in regions of good designs, while the kriging updates are more widespread because, as there is a sparsity of data, there is high error and therefore high expectations of improvement in many areas (i.e. there is an emphasis on exploration over exploitation).

Only discrete values of Λ and T_{in} can be displayed in figure 7 and these design variables are rounded to the nearest 2° and 0.03 respectively, for the purpose of visualization. It should be borne in mind that there is, in fact, a distribution of designs between tiles. The reader should also appreciate that, although the major axis is much larger than the axes of the individual tiles, they both represent the same variation across a variable. Thus, although there seems to be a wide distribution of update points across Λ and T_{in}, for the co-kriging updates these points represent tight clusters in regions of optimal feasible designs.

In §3 we discussed the use of two regression constants λ_{c} and λ_{d} in the co-kriging formulation. The final MLEs of these hyper-parameters for the search in figure 7 were $\hat{\lambda}_{\mathrm{c}} = 1.2 \times 10^{-6}$ and $\hat{\lambda}_{\mathrm{d}} = 6.5 \times 10^{-3}$. These values agree with our intuition that data from an empirical code (Tadpole) will be smooth and require little regression, i.e. a very small $\hat{\lambda}_{\mathrm{c}}$, and data from a dicretized physics-based code (VSaero) will be noisy and require smoothing, i.e. a larger $\hat{\lambda}_{\mathrm{d}}$.

7. Discussion

Our results demonstrate that correlating analyses at multiple levels of fidelity can enhance the accuracy of a surrogate model of the highest level of analysis. This correlated model can be used to find optimal solutions more quickly. We have presented a global optimization strategy using a co-kriging based method that allows for varying levels of noise filtering across multi-fidelity analyses and which converges towards the global optimum using expected improvement maximization.

We have shown a co-kriging based optimization using two levels of CFD code fidelity. The methods are extendable to multiple levels and there are many other examples of analyses that may be coupled. Here, both codes are predicting the same quantity (drag) but, as long as there is a useful correlation, the analyses may be related to completely different quantities as is the case with the work of Hevesi *et al.* (1992), which correlated precipitation with elevation: two different but obviously correlated quantities. Often only one form of analysis is available, although this may come in varying levels of fidelity. For example, Kennedy & O' Hagan (2000) used different finite-element mesh resolutions to produce cheap and expensive results. Forrester *et al.* (2006*a*) showed that partially converged CFD simulations correlate well with their converged counterparts. The convergence could be stopped at any number of points, thus providing many levels of fidelity.

In §4 we discussed the choice of n_c and n_e based on using more or fewer points than an $n=10k$ rule of thumb. In the example problem, we checked that the model built using our choices of n_c and n_e was accurate by assessing the correlation with a second set of data. Naturally the number of points required is problem dependent and additional validation data may be too expensive to compute. A possible way to reduce computational effort in situations when the cost of f_c is considerable is to start with a very low n_c and add data at points where the estimated error is maximized until an accurate cheap model is obtained. Model accuracy may be assessed using a leave-one-out cross-validation procedure. This method will produce a space-filling design with n_c controlled by the desired model accuracy. Maximum expected improvement updates *could* then proceed starting from an initial sample as small as $n_\mathrm{e}=2$, though it may be advantageous to follow a maximum error update strategy until an accurate model is produced.

In our example problem we limited ourselves to four variables. Increasing the number of variables naturally results in a more difficult optimization problem, requiring larger initial sampling plans and more update evaluations, and so the benefits of the co-kriging method are likely to increase. However, since we have noted that co-kriging relies on the relationship between analyses being simpler to model than the analyses themselves, we can assume that for additional variables which have little impact on the design objective the co-kriging model will not offer a significant speed-up. Thus, while we expect co-kriging to perform well in high dimensions, we should still try to isolate the most important variables, either through prior knowledge or through the analysis of elementary effects (Morris 1991).

We limited the total sample size (including updates) to $n_\mathrm{e}=200$ in the example problem due to the expense of tuning the hyper-parameters for large quantities of data overtaking the expense of the CFD simulations. This effectively puts an upper limit on the number of variables that can be considered, since more variables require more points. However, in such cases where f_e is quick to evaluate when compared with the hyper-parameter tuning we can choose to simplify the correlation, and hence the tuning time, by using the same $\theta_{\mathrm{c},j}$ and $\theta_{\mathrm{e},j}$ across all dimensions. The extension of surrogate-based techniques to higher dimensions is a particularly active area of research and further developments in this area would be welcome.

This work has been supported by EPSRC grant code GR/T19209/01.

Appendix A

To derive the co-kriging predictor, we follow a method similar to that for ordinary kriging. The basis of this method is that we wish our prediction of a new expensive point to be consistent with the observed data and the MLEs for the hyper-parameters. We therefore augment the observed data with a predicted value and maximize the likelihood of this augmented dataset by varying our prediction while keeping the hyper-parameters fixed. This gives us an MLE $\hat{y}_\mathrm{e}\big(\boldsymbol{x}^{(n_\mathrm{e}+1)}\big)$.

The augmented dataset is defined as $\tilde{\boldsymbol{X}}=\{\boldsymbol{X}_\mathrm{c}^\mathrm{T}\ \boldsymbol{X}_\mathrm{e}^\mathrm{T}\ \boldsymbol{x}^{(n_\mathrm{e}+1)\mathrm{T}}\}^\mathrm{T}$ and $\tilde{\boldsymbol{y}}=\{\boldsymbol{y}_\mathrm{c}^\mathrm{T}\ \boldsymbol{y}_\mathrm{e}^\mathrm{T}\ \ y^{(n_\mathrm{e}+1)}\}^\mathrm{T}$, with covariance matrix $\tilde{\boldsymbol{C}}$ given by

$$\begin{pmatrix} \hat{\sigma}_\mathrm{c}^2\boldsymbol{\Psi}_\mathrm{c}(\boldsymbol{X}_\mathrm{c},\boldsymbol{X}_\mathrm{c}) & \rho\hat{\sigma}_\mathrm{c}^2\boldsymbol{\Psi}_\mathrm{c}(\boldsymbol{X}_\mathrm{c},\boldsymbol{X}_\mathrm{e}) & \rho\hat{\sigma}_\mathrm{c}^2\psi_\mathrm{c}(\boldsymbol{X}_\mathrm{c},\boldsymbol{x}^{(n_\mathrm{e}+1)}) \\ \rho\hat{\sigma}_\mathrm{c}^2\boldsymbol{\Psi}_\mathrm{c}(\boldsymbol{X}_\mathrm{e},\boldsymbol{X}_\mathrm{c}) & \rho_\mathrm{c}^2\hat{\sigma}_\mathrm{c}^2\boldsymbol{\Psi}_\mathrm{c}(\boldsymbol{X}_\mathrm{e},\boldsymbol{X}_\mathrm{e})+\hat{\sigma}_\mathrm{d}^2\boldsymbol{\Psi}_\mathrm{d}(\boldsymbol{X}_\mathrm{e},\boldsymbol{X}_\mathrm{e}) & \rho\hat{\sigma}_\mathrm{c}^2\psi_\mathrm{c}\big(\boldsymbol{X}_\mathrm{e},\boldsymbol{x}^{(n_\mathrm{e}+1)}\big)+\hat{\sigma}_\mathrm{d}^2\psi_\mathrm{d}\big(\boldsymbol{X}_\mathrm{e},\boldsymbol{x}^{(n_\mathrm{e}+1)}\big) \\ \rho\hat{\sigma}_\mathrm{c}^2\psi_\mathrm{c}(\boldsymbol{X}_\mathrm{c},\boldsymbol{x}^{(n_\mathrm{e}+1)})^\mathrm{T} & \rho\hat{\sigma}_\mathrm{c}^2\psi_\mathrm{c}(\boldsymbol{X}_\mathrm{e},\boldsymbol{x}^{(n_\mathrm{e}+1)})^\mathrm{T}+\hat{\sigma}_\mathrm{d}^2\psi_\mathrm{d}(\boldsymbol{X}_\mathrm{e},\boldsymbol{x}^{(n_\mathrm{e}+1)})^\mathrm{T} & \rho^2\hat{\sigma}_\mathrm{c}^2+\hat{\sigma}_\mathrm{d}^2 \end{pmatrix}$$

which, defining $\boldsymbol{c}$ as a column vector of the covariance of $\boldsymbol{X}$ and $\boldsymbol{x}^{(n_\mathrm{e}+1)}$, can be expressed as

$$\tilde{\boldsymbol{C}}=\begin{pmatrix} \boldsymbol{C} & \boldsymbol{c} \\ \boldsymbol{c}^\mathrm{T} & \rho^2\hat{\sigma}_\mathrm{c}^2+\hat{\sigma}_\mathrm{d}^2 \end{pmatrix}. \tag{A 1}$$

In equations (3.3) and (3.8) it is seen that only the last term of the ln-likelihood contains the sample data, and hence to find an MLE $\hat{y}_\mathrm{e}(\boldsymbol{x}^{(n_\mathrm{e}+1)})$ we need to maximize

$$-\frac{1}{2}(\tilde{\boldsymbol{y}}-\mathbf{1}\mu)^\mathrm{T}\tilde{\boldsymbol{C}}^{-1}(\tilde{\boldsymbol{y}}-\mathbf{1}\mu),$$

which may be expressed as

$$-\frac{1}{2}\begin{pmatrix} \boldsymbol{y}-\mathbf{1}\hat{\mu} \\ \hat{y}_\mathrm{e}(\boldsymbol{x}^{(n_\mathrm{e}+1)})-\hat{\mu} \end{pmatrix}^\mathrm{T}\begin{pmatrix} \boldsymbol{C} & \boldsymbol{c} \\ \boldsymbol{c}^\mathrm{T} & \rho_\mathrm{c}^2\hat{\sigma}_\mathrm{c}^2+\hat{\sigma}_\mathrm{d}^2 \end{pmatrix}^{-1}\begin{pmatrix} \boldsymbol{y}-\mathbf{1}\hat{\mu} \\ \hat{y}_\mathrm{e}(\boldsymbol{x}^{(n_\mathrm{e}+1)})-\hat{\mu} \end{pmatrix}. \tag{A 2}$$

The inverse of the augmented covariance matrix $\tilde{\boldsymbol{C}}^{-1}$ is found using the partitioned inverse formula (Theil 1971)

$$\begin{pmatrix} \boldsymbol{C}^{-1}+\boldsymbol{C}^{-1}\boldsymbol{c}(\rho^2\hat{\sigma}_\mathrm{c}^2+\hat{\sigma}_\mathrm{d}^2-\boldsymbol{c}^\mathrm{T}\boldsymbol{C}^{-1}\boldsymbol{c})^{-1}\boldsymbol{c}^\mathrm{T}\boldsymbol{C}^{-1} & -\boldsymbol{C}^{-1}\boldsymbol{c}(\rho^2\hat{\sigma}_\mathrm{c}^2+\hat{\sigma}_\mathrm{d}^2-\boldsymbol{c}^\mathrm{T}\boldsymbol{C}^{-1}\boldsymbol{c})^{-1} \\ -(\rho^2\hat{\sigma}_\mathrm{c}^2+\hat{\sigma}_\mathrm{d}^2-\boldsymbol{c}^\mathrm{T}\boldsymbol{C}^{-1}\boldsymbol{c})^{-1}\boldsymbol{c}^\mathrm{T}\boldsymbol{C}^{-1} & (\rho^2\hat{\sigma}_\mathrm{c}^2+\hat{\sigma}_\mathrm{d}^2-\boldsymbol{c}^\mathrm{T}\boldsymbol{C}^{-1}\boldsymbol{c})^{-1} \end{pmatrix}. \tag{A 3}$$

Substituting (A 3) into (A 2) and ignoring terms without $\hat{y}_\mathrm{e}(\boldsymbol{x}^{(n_\mathrm{e}+1)})$ we obtain

$$\left(\frac{-1}{2(\rho^2\hat{\sigma}_\mathrm{c}^2+\hat{\sigma}_\mathrm{d}^2-\boldsymbol{c}^\mathrm{T}\boldsymbol{C}^{-1}\boldsymbol{c})}\right)(\hat{y}_\mathrm{e}(\boldsymbol{x}^{(n_\mathrm{e}+1)})-\hat{\mu})^2 + \left(\frac{\boldsymbol{c}^\mathrm{T}\boldsymbol{C}^{-1}(\boldsymbol{y}-\mathbf{1}\hat{\mu})}{(\rho^2\hat{\sigma}_\mathrm{c}^2+\hat{\sigma}_\mathrm{d}^2-\boldsymbol{c}^\mathrm{T}\boldsymbol{C}^{-1}\boldsymbol{c})}\right)(\hat{y}_\mathrm{e}(\boldsymbol{x}^{(n_\mathrm{e}+1)})-\hat{\mu}). \tag{A 4}$$

This expression is maximized by taking the derivative with respect to $\hat{y}_{\rm e}\big(\boldsymbol{x}^{(n_{\rm e}+1)}\big)$ and setting it to 0,

$$\left(\frac{-1}{\rho^2\hat{\sigma}_{\rm c}^2+\hat{\sigma}_{\rm d}^2-\boldsymbol{c}^{\rm T}\boldsymbol{C}^{-1}\boldsymbol{c}}\right)(\hat{y}_{\rm e}(\boldsymbol{x}^{(n_{\rm e}+1)})-\hat{\mu})+\left(\frac{\boldsymbol{c}^{\rm T}\boldsymbol{C}^{-1}(\boldsymbol{y}-\boldsymbol{1}\hat{\mu})}{(\rho^2\hat{\sigma}_{\rm c}^2+\hat{\sigma}_{\rm d}^2-\boldsymbol{c}^{\rm T}\boldsymbol{C}^{-1}\boldsymbol{c})}\right)=0. \quad (A\,5)$$

Solving for $\hat{y}_{\rm e}(\boldsymbol{x}^{(n_{\rm e}+1)})$ now gives

$$\hat{y}_{\rm e}(\boldsymbol{x}^{(n_{\rm e}+1)})=\hat{\mu}+\boldsymbol{c}^{\rm T}\boldsymbol{C}^{-1}(\boldsymbol{y}-1\hat{\mu}). \quad (A\,6)$$

For the estimated mean-squared error in this prediction, we again turn to Jones (2001) for our method of derivation. It is argued that our MLE $\hat{y}_{\rm e}\big(\boldsymbol{x}^{(n_{\rm e}+1)}\big)$ is more accurate if the likelihood has a definite maximum, i.e. a high curvature. Conversely, if different values of $\hat{y}_{\rm e}\big(\boldsymbol{x}^{(n_{\rm e}+1)}\big)$ have similar likelihoods the MLE is less accurate. The error in the predictor is therefore inversely related to the curvature (the second derivative with respect to $\hat{y}_{\rm e}\big(\boldsymbol{x}^{(n_{\rm e}+1)}\big)$ of the augmented likelihood. We have already found the first derivative in equation (A 5) and taking the derivative and inverse of this equation we obtain

$$s^2(\boldsymbol{x})\approx\rho^2\hat{\sigma}_{\rm c}^2+\hat{\sigma}_{\rm d}^2-\boldsymbol{c}^{\rm T}\boldsymbol{C}^{-1}\boldsymbol{c}. \quad (A\,7)$$

This equation is not precisely the classically derived formula, which includes an extra term $(1-\boldsymbol{1}^{\rm T}\boldsymbol{C}^{-1}\boldsymbol{c})^2/\boldsymbol{c}^{\rm T}\boldsymbol{C}^{-1}\boldsymbol{c}$. However, this extra term is so small that it can safely be neglected.

Appendix B

The expected improvement (equation (5.1)) is typically calculated as

$$(\min\{\boldsymbol{y}_{\rm e}\}-\hat{y}_{\rm e}(\boldsymbol{x}))\left[\frac{1}{2}+\frac{1}{2}\mathrm{erf}\left(\frac{\min\{\boldsymbol{y}_{\rm e}\}-\hat{y}_{\rm e}(\boldsymbol{x})}{\sqrt{2}s(\boldsymbol{x})}\right)\right] \\ +\frac{s(\boldsymbol{x})}{\sqrt{2\pi}}\exp\left[-\frac{(\min\{\boldsymbol{y}_{\rm e}\}-\hat{y}_{\rm e}(\boldsymbol{x}))^2}{2s^2(\boldsymbol{x})}\right]. \quad (B\,1)$$

When using re-interpolation, $s(\boldsymbol{x})$ is typically small and so erf$(\cdot)\rightarrow-1$ and $\exp(\cdot)\rightarrow 0$. This often leads to floating-point underflow and $E[I(\boldsymbol{x})]=0$. Using the substitution $a=(\min\{\boldsymbol{y}_{\rm e}\}-\hat{y}_{\rm e}(\boldsymbol{x}))/\sqrt{2}s(\boldsymbol{x})$, when $a\ll-1$, erf(a) can be expressed using a Maclaurin series expansion and the first term of (B 1) becomes

$$(\min\{\boldsymbol{y}_{\rm e}\}-\hat{y}_{\rm e}({\rm x}))\left[\frac{1}{2\sqrt{\pi}}\exp(-a^2)\sum_{n=0}^{\infty}\frac{(-1)^n(2n-1)!!}{2^n}a^{-(2n+1)}\right].$$

Note that $\exp(-a^2)$ appears in the second term in (B 1) and therefore $E[I(\boldsymbol{x})]$ can now be expressed as

$$\left[(\min\{\boldsymbol{y}_{\rm e}\}-\hat{y}_{\rm e}(\boldsymbol{x}))\frac{1}{2\sqrt{\pi}}\sum_{n=0}^{\infty}\frac{(-1)^n(2n-1)!!}{2^n}a^{-(2n+1)}+\frac{s(\boldsymbol{x})}{\sqrt{2\pi}}\right] \\ \times\exp\left[-\frac{(\min\{\boldsymbol{y}_{\rm e}\}-\hat{y}_{\rm e}(\boldsymbol{x}))^2}{2s^2(\boldsymbol{x})}\right]. \quad (B\,2)$$

We can now take natural logarithms and $\ln E[I(\boldsymbol{x})]$ can be searched by the optimizer without problems with floating-point underflow.

References

Cook, R. D. & Nachtsheim, C. J. 1980 A comparison of algorithms for constructing exact D-optimal designs. *Technometrics* **22**, 315–324. (doi:10.2307/1268315)

Cousin, J. & Metcalf M. 1990 The BAE Ltd. transport aircraft synthesis and optimization program. In *AHS, and ASEE, Aircraft Design, Systems and Operations Conference, Dayton, OH, 17–19 September 1990.*

Forrester, A. I. J., Bressloff, N. W. & Keane, A. J. 2006*a* Optimization using surrogate models and partially converged computational fluid dynamics simulations. *Proc. R. Soc. A* **462**, 2177–2204. (doi:10.1098/rspa.2006.1679)

Forrester, A. I. J., Keane, A. J. & Bressloff, N. W. 2006*b* Design and analysis of 'noisy' computer experiments. *AIAA J.* **44**, 2331–2339. (doi:10.2514/1.20068)

Forrester, A. I. J., Sóbester, A. & Keane, A. J. 2006*c* Optimization with missing data. *Proc. R. Soc. A* **462**, 935–945. (doi:10.1098/rspa.2005.1608)

Hevesi, J., Flint, A. & Istok, J. 1992 Precipitation estimation in mountainous terrai using multivariate geostatistics. Part II: isohyetal maps. *J. Appl. Meteorol.* **31**, 677–688. (doi:10.1175/1520-0450(1992)031<0677:PEIMTU>2.0.CO;2)

Jones, D. R. 2001 A taxonomy of global optimization methods based on response surfaces. *J. Global Optim.* **21**, 345–383. (doi:10.1023/A:1012771025575)

Keane, A. J. 2003 Wing optimization using design of experiment, response surface, and data fusion methods. *J. Aircraft* **40**, 741–750.

Kennedy, M. C. & O'Hagan, A. 2000 Predicting the output from complex computer code when fast approximations are available. *Biometrika* **87**, 1–13. (doi:10.1093/biomet/87.1.1)

Krige, D. G. 1951 A statistical approach to some basic mine valuation problems on the Witwatersrand. *J. Chem. Metallurg. Min. Eng. Soc. S. Afr.* **52**, 119–139.

Locatelli, M. 1997 Bayesian algorithms for one-dimensional global optimization. *J. Global Optim.* **10**, 57–76. (doi:10.1023/A:1008294716304)

Maskew, B. 1982 Prediction of subsonic aerodynamic characteristcs: a case for low-order panel methods. *J. Aircraft* **19**, 157–163.

Matheron, G. 1963 Principles of geostatistics. *Econ. Geol.* **58**, 1246–1266.

McKay, M. D., Beckman, R. J. & Conover, W. J. 1979 A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**, 239–245. (doi:10.2307/1268522)

Morris, M. D. 1991 Factorial sampling plans for preliminary computational experiments. *Technometrics* **33**, 161–174. (doi:10.2307/1269043)

Morris, M. D. & Mitchell, T. J. 1995 Exploratory designs for computer experiments. *J. Stat. Plan. Infer.* **43**, 381–402. (doi:10.1016/0378-3758(94)00035-T)

Theil, H. 1971 *Principles of econometrics.* New York, NY: John Wiley.