DOI 10.1007/s00158-016-1626-8

RESEARCH PAPER

Gradient based hyper-parameter optimisation for well conditioned kriging metamodels

Jonathan Ollar[1,2] · Charles Mortished[1,3] · Royston Jones[1] · Johann Sienz[3] · Vassili Toropov[2,4]

Received: 17 June 2016 / Revised: 1 October 2016 / Accepted: 17 November 2016

Abstract In this work a two step approach to efficiently carrying out hyper parameter optimisation, required for building kriging and gradient enhanced kriging metamodels, is presented. The suggested approach makes use of an initial line search along the hyper-diagonal of the design space in order to find a suitable starting point for a subsequent gradient based optimisation algorithm. During the optimisation an upper bound constraint is imposed on the condition number of the correlation matrix in order to keep it from being ill conditioned. Partial derivatives of both the condensed log likelihood function and the condition number are obtained using the adjoint method, the latter has been derived in this work. The approach is tested on a number of analytical examples and comparisons are made to other optimisation approaches. Finally the approach is used to construct metamodels for a finite element model of an aircraft wing box comprising of 126 thickness design variables and is then compared with a sub-set of the other optimisation approaches.

Keywords Hyper-parameter optimisation · SQP · Adjoint method · Condition number · Kriging · Gradient-enhanced kriging · Gaussian processes

1 Introduction

Metamodels are frequently used to represent the responses of numerical simulations using less computationally expensive mathematical models. The reader is referred to Barthelemy and Haftka (1993), Wang and Shan (2007), Forrester and Keane (2009), and Viana et al. (2014, 2010) for reviews on metamodel-based optimisation in general.

This paper is concerned with Kriging which is an interpolating metamodelling technique based on spatial correlation that was first proposed by Krige (1951) and later implemented by Matheron (1963) for use within the mining industry. The use of Kriging for replacing expensive computational models with metamodels was shown by Sacks et al. (1989).

In many simulation software products derivatives with respect to the design variables, hereafter referred to as *design sensitivities*, can be cost efficiently obtained using for instance the adjoint method. The reader is referred to Martins and Hwang (2013) for a review on obtaining design sensitivities. If design sensitivities are available, these can be incorporated into the Kriging equations in order to enhance the metamodel quality (Han et al. 2013; Zimmermann 2013; Kim and Lee 2010). This is commonly

Parts of this manuscript have been presented at the 57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Januray, 2016, San Diego, California - Paper number: AIAA 2016-0420.

✉ Jonathan Ollar
jonathan.ollar@uk.altair.com

Charles Mortished
charles.mortished@uk.altair.com

[1] Altair Engineering Ltd, Imperial House Holly walk, CV32 4JG, Leamington Spa, UK

[2] School of Engineering and Materials Science, Queen Mary University of London, Mile End Road, E1 4NS London, UK

[3] College of Engineering, Swansea University, Bay Campus, SA1 8EN Swansea, Wales, UK

[4] Institute of Information Technology, Mathematics and Mechanics, Lobachevsky University of Nizhny Novgorod, Nizhny Novgorod, Russia

Published online: 13 December 2016

known as gradient enhanced kriging (GEK). One of the main challenges of kriging, and gradient enhanced kriging in particular, is the computational cost associated with building the metamodel. This requires optimisation of a condensed log likelihood function with respect to a set of hyper-parameters, one for each design variable. Every evaluation of the condensed log likelihood function requires decomposition of a square correlation matrix, $\mathbf{R} \in \mathbb{R}^{d \times d}$. For kriging $d = p$, where p is the number of training points, and for gradient enhanced kriging $d = p \times (n + 1)$ where n is the number of design variables. For problems with a small number of design variables, gradient based algorithms such as sequential quadratic programming have shown good performance (Zimmermann 2013; Lockwood and Anitescu 2010). To increase the probability of finding a better solution for problem with several optima, multiple start-points have also been proposed, with as few as five points (Lockwood and Anitescu 2010) or as many ten times the number of hyper-parameters (Liu and Batill 2002). For larger problems the optimisation is often carried out using global algorithms such as simulated annealing (Xiong et al. 2007) and genetic algorithm (Forrester et al. 2008). Toal et al. (2011) proposed a Hybrid optimisation scheme where promising points from a particle swarm optimisation were used as starting points for gradient based optimisations using sequential quadratic programming. In the same paper it was also shown how the adjoint method can be used to obtain partial derivatives of the condensed log likelihood function with respect to the correlation matrix, which greatly reduces the computational effort required when compared to finite differences and the direct method.

Ill-conditioning of the correlation matrix can become an issue when building metamodels where training points are located near each other (Haaland et al. 2011), especially for Gaussian correlation matrices (Zimmermann 2015). Attempts have been made to reduce ill conditioning by, for instance, using uniform subsets of the training points Rennen (2008), adding regularisation terms along the diagonal of the correlation matrix which makes the kriging metamodel approximate rather than interpolate the data, and constraining the condition number explicitly during optimisation Dalbey (2013).

In this paper partial derivatives of the condition number of the correlation matrix with respect to the hyper-parameters are obtained making it possible to constrain the condition number directly in a gradient based optimisation approach. A two-step approach is suggested for optimisation of the hyper parameters. In the first step, the optimisation problem is considered as a single variable problem by treating all hyper parameters as one variable. The solution to this problem is then used as a starting point for a gradient based optimisation algorithm. In both cases an upper bound constraint is enforced on the condition number of the correlation matrix. The approach is tested on several analytical examples using two types of gradient based optimisation algorithms, the sequential quadratic programming and the method of feasible directions. The approach is compared to gradient based optimisations starting from random points, multiple starting points and, a genetic algorithm followed by gradient based optimisations. Finally a case study is presented where the responses of an aircraft wing-box with 126 design variables is approximated using the suggested approach and compared to a selection of optimisation methods.

2 Kriging

Kriging is an interpolating metamodel technique based on spatial correlation that was first proposed by Krige (1951) and later implemented by Matheron (1963) for use within the mining industry. The use of kriging for approximation of expensive computational models was shown by Sacks et al. (1989).

Following the notation of Jones (2001), kriging is derived from the assumption that computer simulations are entirely deterministic and any error in the fit of a metamodel is entirely down to missing terms in the model. Given a set of p training points $\mathbf{x}^{(i)} \in \mathbb{R}^n$, $i = 1, p$ and the corresponding function values $y^{(i)}$, $i = 1, p$, the kriging model can in general be written as

$$\hat{y}(\mathbf{x}) = \sum_{j=1}^{h} b_j(\mathbf{x}) a_j + \hat{\epsilon}(\mathbf{x}) \tag{1}$$

where $b_j(\mathbf{x})$ is the j-th regressor, a_j the corresponding regression coefficient, and $\hat{\epsilon}_i(\mathbf{x})$ is the model of the error in the weighted least squares fit. It is assumed that the error, $\epsilon_i(\mathbf{x})$, is continuous for any continuous function $y(\mathbf{x})$, and that the error at two points $\epsilon(\mathbf{x}^{(i)})$ and $\epsilon(\mathbf{x}^{(j)})$ are correlated with their distance according to some model $\psi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$. As the error is modelled explicitly in kriging, the model will exactly interpolate through the training points.

The first part of the model (1), the polynomial regression, can be of arbitrary order, however the order of the regression model will dictate the number of required points which must be at least as many as the number of regressors. Kriging with zero-th order polynomials is usually referred to as ordinary kriging while using first or higher order polynomials are denoted universal kriging. In this work, ordinary kriging is used. The model for ordinary kriging, hereafter referred to as kriging, takes the form

$$\hat{y}(\mathbf{x}) = \hat{\mu} + \hat{\epsilon}(\mathbf{x}) \tag{2}$$

where the estimated mean $\hat{\mu}$ is determined by solving the weighted least squares problem

$$\hat{\mu} = \left(\mathbf{F}^T\mathbf{R}^{-1}\mathbf{F}\right)^{-1}\mathbf{F}^T\mathbf{R}^{-1}\mathbf{y}, \tag{3}$$

and $\mathbf{F}$ is a matrix of regressors, representing a zero-th order basis function in ordinary kriging, is reduced to a vector of ones according to

$$\mathbf{F} = \mathbf{1} = [\underbrace{1, ..., 1}_{\mathrm{p}}]^T. \tag{4}$$

The correlation matrix, $\mathbf{R}$, contains the estimated spatial correlation between all the training points

$$\mathbf{R} = \begin{bmatrix} \psi(\mathbf{x}^{(1)}, \mathbf{x}^{(1)}) & \dots & \psi(\mathbf{x}^{(1)}, \mathbf{x}^{(p)}) \\ \vdots & \ddots & \vdots \\ \psi(\mathbf{x}^{(p)}, \mathbf{x}^{(1)}) & \dots & \psi(\mathbf{x}^{(p)}, \mathbf{x}^{(p)}) \end{bmatrix} \tag{5}$$

here modelled as a Gaussian function according to

$$\psi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = exp\left[\sum_{k=1}^{n} -\theta_k \left(\left|\left|\mathbf{x}_k^{(i)} - \mathbf{x}_k^{(j)}\right|\right|\right)^2\right], \tag{6}$$

where θ_k is a so called hyper-parameter that scales the influence of the k-th design variable. The value of θ_k is obtained through optimisation and will be discussed in Section 5. The second part of (2), the error $\hat{\epsilon}(\mathbf{x})$, treated as a stochastic process, is modelled as

$$\hat{\epsilon}(\mathbf{x}) = \mathbf{w}^T\mathbf{r}(\mathbf{x}) \tag{7}$$

where $\mathbf{r}$ contains basis functions depending on the spatial correlation model (6) between the evaluation point, $\mathbf{x}^{(e)}$, and the training points

$$\mathbf{r} = \left[\psi(\mathbf{x}^{(e)}, \mathbf{x}^{(1)}), ..., \psi(\mathbf{x}^{(e)}, \mathbf{x}^{(p)})\right]^T, \tag{8}$$

and corresponding weights, $\mathbf{w}$, obtained through solving the linear equation system

$$\mathbf{w} = \mathbf{R}^{-1}\left(\mathbf{y} - \mathbf{F}\hat{\mu}\right). \tag{9}$$

The final predicted kriging predictor is given by

$$\hat{y}(\mathbf{x}^{(e)}) = \hat{\mu} + \mathbf{w}^T\mathbf{r}, \tag{10}$$

and a predicted mean squared error of the kriging predictor is obtained as

$$s^2(\mathbf{x}^{(e)}) = \hat{\sigma}^2\left[1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r} + \frac{(1 - \mathbf{F}^T\mathbf{R}^{-1}\mathbf{r})^2}{\mathbf{F}^T\mathbf{R}^{-1}\mathbf{F}}\right], \tag{11}$$

where $\hat{\sigma}$ is the predicted system variance

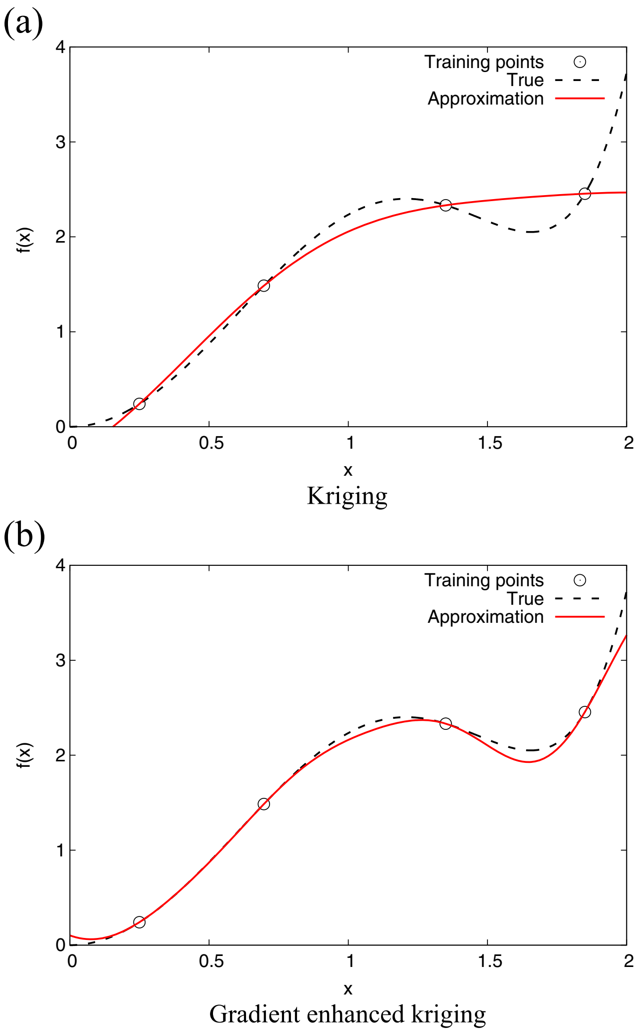
$$\hat{\sigma}^2 = \frac{1}{p}\left(\mathbf{y} - \mathbf{F}\mu\right)^T\mathbf{R}^{-1}\left(\mathbf{y} - \mathbf{F}\mu\right). \tag{12}$$

It should be noted that in order to solve the systems of linear equations in (3), (9), (11) and (12) Cholesky decomposition *(DPOTRF)* and back substitution *(DPOTRS)* routines from the Intel Math Kernel Library 11.2 (Intel 2015) was used.

3 Gradient-enhanced kriging

If, in addition to function values, design sensitivities are available they may be used to improve the accuracy of the kriging metamodel. The method of incorporating gradients in kriging is commonly referred to as gradient enhanced kriging (GEK) and is described in this section. Figure 1 shows an example of a kriging and gradient enhanced kriging fit respectively for a one dimensional function. It is clear that the gradient enhanced fit is of superior quality. The presented implementation is inspired by Han et al. (2013) and Lockwood and Anitescu (2010) to which the reader is referred for further information. In order to create a gradient enhanced kriging fit the correlation matrix is extended to include derivative terms according to

$$\mathbf{R} = \begin{bmatrix} \mathbf{Q}^{1,1} & Q^{1,2} \\ (\mathbf{Q}^{1,2})^T & \mathbf{Q}^{2,2} \end{bmatrix}. \tag{13}$$



Kriging

Gradient enhanced kriging

Fig. 1 Kriging fit in one dimension. Gradient enhanced kriging significantly improves quality of fit

where $\mathbf{Q}^{1,1}$ is the correlation matrix used in the non-gradient case

$$\mathbf{Q}^{1,1} = \begin{bmatrix} \psi(\mathbf{x}^{(1)}, \mathbf{x}^{(1)}) & \dots & \psi(\mathbf{x}^{(1)}, \mathbf{x}^{(p)}) \\ \vdots & \ddots & \vdots \\ \psi(\mathbf{x}^{(p)}, \mathbf{x}^{(1)}) & \dots & \psi(\mathbf{x}^{(p)}, \mathbf{x}^{(p)}) \end{bmatrix}, \tag{14}$$

$\mathbf{Q}^{1,2}$ contains the first derivatives of $\mathbf{R}$ according to

$$\mathbf{Q}^{1,2} = \begin{bmatrix} \frac{\partial\psi(\mathbf{x}^{(1)},\mathbf{x}^{(1)})}{\partial\mathbf{x}^{(1)}} & \dots & \frac{\partial\psi(\mathbf{x}^{(1)},\mathbf{x}^{(p)})}{\partial\mathbf{x}^{(p)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial\psi(\mathbf{x}^{(p)},\mathbf{x}^{(1)})}{\partial\mathbf{x}^{(1)}} & \dots & \frac{\partial\psi(\mathbf{x}^{(p)}\mathbf{x}^{(p)})}{\partial\mathbf{x}^{(p)}} \end{bmatrix}, \tag{15}$$

where

$$\frac{\partial\psi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})}{\partial\mathbf{x}^{(j)}} = \begin{bmatrix} \frac{\partial\psi(\mathbf{x}^{(i)},\mathbf{x}^{(i)})}{\partial x_1^{(j)}} \dots \frac{\partial\psi(\mathbf{x}^{(i)},\mathbf{x}^{(i)})}{\partial x_n^{(j)}} \end{bmatrix}. \tag{16}$$

and, using the Gaussian function in (6)

$$\frac{\partial\psi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})}{\partial x_k^{(j)}} = 2\theta_k \left(x_k^{(i)} - x_k^{(j)}\right) \psi\left(\mathbf{x}_k^{(i)}, \mathbf{x}_k^{(j)}\right). \tag{17}$$

The sub-matrix $\mathbf{Q}^{2,2}$ contains the second derivatives

$$\mathbf{Q}^{2,2} = \begin{bmatrix} \frac{\partial^2\psi(\mathbf{x}^{(1)},\mathbf{x}^{(1)})}{\partial\mathbf{x}^{(1)}\partial\mathbf{x}^{(1)}} & \dots & \frac{\partial^2\psi(\mathbf{x}^{(1)},\mathbf{x}^{(p)})}{\partial\mathbf{x}^{(1)}d\mathbf{x}^{(p)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2\psi(\mathbf{x}^{(p)},\mathbf{x}^{(1)})}{\partial\mathbf{x}^{(p)}\partial\mathbf{x}^{(1)}} & \dots & \frac{\partial^2\psi(\mathbf{x}^{(p)}\mathbf{x}^{(p)})}{\partial\mathbf{x}^{(p)}\partial\mathbf{x}^{(p)}} \end{bmatrix}, \tag{18}$$

where

$$\frac{\partial^2\psi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})}{\partial\mathbf{x}^{(i)}\partial\mathbf{x}^{(j)}} = \begin{bmatrix} \frac{\partial^2\psi(\mathbf{x}^{(i)},\mathbf{x}^{(j)})}{\partial x_1^{(i)}\partial x_1^{(j)}} & \dots & \frac{\partial^2\psi(\mathbf{x}^{(i)},\mathbf{x}^{(j)})}{\partial x_n^{(i)}\partial x_1^{(j)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2\psi(\mathbf{x}^{(i)},\mathbf{x}^{(j)})}{\partial x_1^{(i)}\partial x_n^{(j)}} & \dots & \frac{\partial^2\psi(\mathbf{x}^{(i)},\mathbf{x}^{(j)})}{\partial x_n^{(i)}\partial x_n^{(j)}} \end{bmatrix}, \tag{19}$$

and, using the Gaussian function in (6)

$$\frac{\partial^2\psi(\mathbf{x}^{(i)},\mathbf{x}^{(j)})}{\partial x_l^{(i)}\partial x_k^{(j)}} = \begin{cases} 2\theta_k \left[-2\theta_k \left(x_k^{(i)} - x_k^{(j)}\right)^2 + 1\right] \psi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) & , k = l \\ -4\theta_k\theta_l \left[\left(x_k^{(i)} - x_k^{(j)}\right)\left(x_l^{(i)} - x_l^{(j)}\right)\right] \psi(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) & , k \neq l \end{cases} \tag{20}$$

The vector of spatial correlations between the evaluation points and the training points is extended in the same manner to

$$\mathbf{r} = \begin{bmatrix} \psi(\mathbf{x}^{(e)}, \mathbf{x}^{(1)}) \\ \vdots \\ \psi(\mathbf{x}^{(e)}, \mathbf{x}^{(p)}) \\ \frac{\partial\psi(\mathbf{x}^{(e)},\mathbf{x}^{(p)})}{\partial\mathbf{x}^{(1)}} \\ \vdots \\ \frac{\partial\psi(\mathbf{x}^{(e)},\mathbf{x}^{(p)})}{\partial\mathbf{x}^{(p)}} \end{bmatrix}, \tag{21}$$

using the expression in (17). The vector of function values is extended to include design sensitivities as well as the function values

$$\mathbf{y} = \begin{bmatrix} f(\mathbf{x}^{(1)}) \\ \vdots \\ f(\mathbf{x}^{(p)}) \\ \frac{\partial f(\mathbf{x}^{(1)})}{\partial\mathbf{x}} \\ \vdots \\ \frac{\partial f(\mathbf{x}^{(p)})}{\partial\mathbf{x}} \end{bmatrix}, \tag{22}$$

where

$$\frac{\partial f(\mathbf{x}^{(i)})}{\partial\mathbf{x}} = \begin{bmatrix} \frac{\partial f(\mathbf{x}^{(i)})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\mathbf{x}^{(i)})}{\partial x_n} \end{bmatrix}. \tag{23}$$

Similarly, the basis polynomial vector is extended to include the derivatives of the regression basis function. In the case of a zero-th order basis function, a vector of zeroes according to

$$\mathbf{F} = [\ \mathbf{1}\ \mathbf{0}] = [\underbrace{1, ..., 1}_{\text{p}}, \underbrace{0, ..., 0}_{\text{p} \times \text{n}}]^T. \tag{24}$$

recalling that n is the number of design variables and p is the number of training points. The predicted mean is calculated as

$$\hat{\mu} = \left(\mathbf{F}^T\mathbf{R}^{-1}\mathbf{F}\right)^{-1} \mathbf{F}^T\mathbf{R}^{-1}\mathbf{y}, \tag{25}$$

and the weights vector takes the form

$$\mathbf{w} = \mathbf{R}^{-1}\left(\mathbf{y} - \mathbf{F}\hat{\mu}\right). \tag{26}$$

The final predicted kriging estimation at a point $\mathbf{x}^{(e)}$ is calculated as

$$\hat{y}(\mathbf{x}^{(e)}) = \hat{\mu} + \mathbf{w}^T\mathbf{r} \tag{27}$$

and a predicted mean squared error of the kriging prediction is given by

$$s^2(\mathbf{x}^{(e)}) = \hat{\sigma}^2 \left[1 - \mathbf{r}^T\mathbf{R}^{-1}\mathbf{r} + \frac{(1 - \mathbf{F}^T\mathbf{R}^{-1}\mathbf{r})^2}{\mathbf{F}^T\mathbf{R}^{-1}\mathbf{F}}\right], \tag{28}$$

where the predicted system variance is obtained as

$$\hat{\sigma}^2 = \frac{1}{n(1+p)} \left(\mathbf{y} - \mathbf{F}\mu\right)^T \mathbf{R}^{-1} \left(\mathbf{y} - \mathbf{F}\mu\right). \tag{29}$$

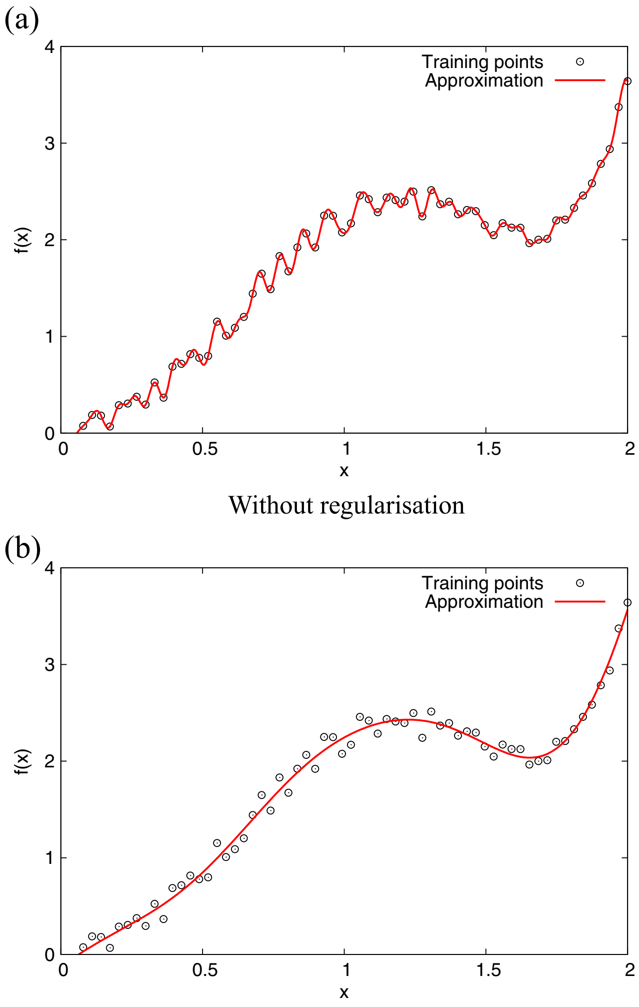
It should be noted that in order to solve the systems of linear equations in (25), (26), (28) and (29) Cholesky decomposition *(DPOTRF)* and back substitution *(DPOTRS)* routines from the Intel Math Kernel Library 11.2 (Intel 2015) was used.

4 Noise regularisation

Although kriging is an interpolation technique it is possible to create a regression-like metamodel through regularisation as shown by Forrester et al. (2006). This is performed by adding a regularisation parameter to the diagonal elements of the correlation matrix according to

$$\mathbf{R} = \mathbf{R} + \lambda \mathbf{I}, \ \lambda > 0, \tag{30}$$

where λ is the regularisation parameter. The higher the value of lambda the more the metamodel is allowed to deviate from the training points. This is demonstrated in Fig. 2 which shows a one-dimensional fit of a noisy function with and without regularisation. It can be seen that the regularisation smooths the noisy function. Adding a regularisation parameter has an added benefit of improving the conditioning of the kriging equation system.



Without regularisation

With regularisation

Fig. 2 Kriging approximation of a noisy function

For the gradient-enhanced case there is a possibility of noise in both the function values and the gradients values. As such it is beneficial to have two regularisation parameters, as proposed by Lukaczyk et al. (2013), one relating to the function values and one for the gradients. This leads to an augmentation of the correlation matrix according to

$$\mathbf{R} = \begin{bmatrix} \mathbf{Q}^{1,1} & Q^{1,2} \\ (\mathbf{Q}^{1,2})^T & \mathbf{Q}^{2,2} \end{bmatrix} + \begin{bmatrix} \lambda_1 \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \lambda_2 \mathbf{I} \end{bmatrix}. \tag{31}$$

By having two regularisation parameters, the regularisation of noisy function values and gradients can be addressed separately.

5 Hyper-parameter tuning

To obtain a good kriging fit it is important to determine suitable values of the hyper-parameters. Failing to do so may result in a sub-standard fit. Figure 3 shows an example of (a) an overestimated hyper parameter and (b) an optimised hyper parameter. The hyper-parameters are determined through maximisation of the condensed log likelihood function (Jones 2001)

$$\phi(\boldsymbol{\theta}) = -\frac{p}{2} \ln(\hat{\sigma}^2(\boldsymbol{\theta})) - \frac{1}{2} \ln(|\mathbf{R}(\boldsymbol{\theta})|), \tag{32}$$

and for the gradient enhanced case (Han et al. 2013)

$$\phi(\boldsymbol{\theta}) = -\frac{p(n+1)}{2} \ln(\hat{\sigma}^2(\boldsymbol{\theta})) - \frac{1}{2} \ln(|\mathbf{R}(\boldsymbol{\theta})|), \tag{33}$$

where $|\mathbf{R}(\boldsymbol{\theta})|$ denotes the determinant of the correlation matrix. To prevent ill conditioning of the correlation matrix the condition number is constrained to be lower than some threshold during optimisation. The condition number is obtained as

$$k(\mathbf{R}) = \|\mathbf{R}^{-1}\| \|\mathbf{R}\|. \tag{34}$$

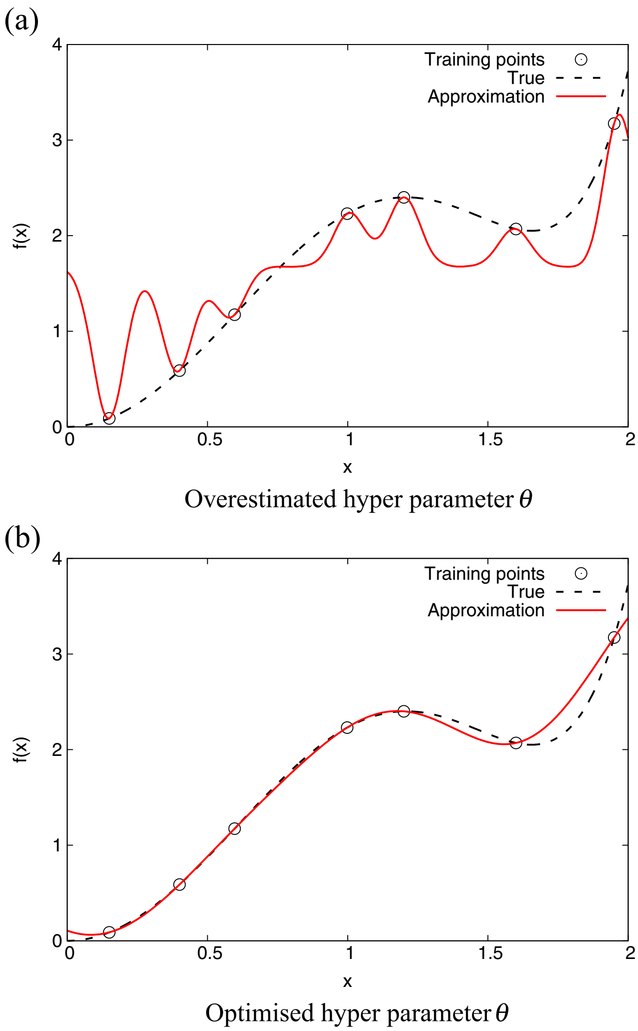
where $\|\mathbf{R}\|$ denotes the norm of the correlation matrix which is here calculated as the Frobenius norm

$$\|\mathbf{R}\| = \|\mathbf{R}\|_F = \sqrt{\sum_i \sum_j R_{i,j}^2}. \tag{35}$$

and $\|\mathbf{R}^{-1}\|$ denotes the norm of the inverse correlation matrix which is calculated using the matrix inversion (*DPOTRI*) routine from the Intel Math Kernel Library 11.2 (Intel 2015) using the matrix decomposition previously obtained for Kriging. Formally, the hyper-parameter optimisation problem takes the form

$$\begin{aligned} &\underset{\boldsymbol{\theta},\boldsymbol{\lambda}}{\text{maximise}} \quad \phi(\boldsymbol{\theta}, \boldsymbol{\lambda}) \\ &\text{subject to} \quad \kappa(\mathbf{R}(\boldsymbol{\theta}, \boldsymbol{\lambda})) \leq \kappa_{max} \end{aligned} \tag{36}$$

where κ_{max} is the upper bound constraint on the condition number. Here, $\kappa_{max} = 10^7$ is used.



Overestimated hyper parameter θ

Optimised hyper parameter θ

Fig. 3 Importance of hyper-parameter optimisation

Because of the computational expense related to hyper parameter optimisation this work is concerned with hyper parameter optimisation using gradient based optimisation techniques. In the following section it is shown how to obtain the gradients of the condensed log likelihood function and of the condition number with respect to the hyper parameters and regularisation parameters. These are then used for a hyper parameter optimisation approach outlined in the subsequent section.

6 Obtaining gradients

In order to use gradient based optimisation techniques, the gradients of the response functions are required. These can be obtained using different methods, with varying associated computational cost, depending on the problem at hand. For a large number of design variables, it may be prohibitively expensive to use finite differences or the direct method as the cost is proportional to the number of design variables. The computational cost of the adjoint method, however, is proportional to the number of response functions, which in this case are two, the condensed likelihood function and the condition number.

Using the chain rule the gradients of the condensed likelihood function with respect to the hyper parameters can be written as

$$\frac{\partial \phi}{\partial \boldsymbol{\theta}} = \sum_{i=1}^{p} \sum_{j=1}^{p} \frac{\partial \phi}{\partial R_{ij}} \frac{\partial R_{ij}}{\partial \boldsymbol{\theta}} \tag{37}$$

and with respect to the regularisation parameters

$$\frac{\partial \phi}{\partial \boldsymbol{\lambda}} = \sum_{i=1}^{p} \sum_{j=1}^{p} \frac{\partial \phi}{\partial R_{ij}} \frac{\partial R_{ij}}{\partial \boldsymbol{\lambda}}. \tag{38}$$

Similarly the gradients of the condition number with respect to the hyper parameters can be written as

$$\frac{\partial \kappa}{\partial \boldsymbol{\theta}} = \sum_{i=1}^{p} \sum_{j=1}^{p} \frac{\partial \kappa}{\partial R_{ij}} \frac{\partial R_{ij}}{\partial \boldsymbol{\theta}} \tag{39}$$

and with respect to the regularisation parameters

$$\frac{\partial \kappa}{\partial \boldsymbol{\lambda}} = \sum_{i=1}^{p} \sum_{j=1}^{p} \frac{\partial \kappa}{\partial R_{ij}} \frac{\partial R_{ij}}{\partial \boldsymbol{\lambda}} \tag{40}$$

In total there are four types of derivatives to establish. The gradients of the condensed likelihood function with respect to the correlation matrix, the gradients of the condition number with respect to the correlation matrix, and the gradients of the correlation matrix with respect to the hyper parameters and regularisation parameters. These are discussed in the following sections.

6.1 Gradients of the condensed likelihood function w.r.t. the correlation matrix

The partial derivatives of the condensed likelihood function with respect to the correlation matrix can be obtained using the adjoint method as shown by Toal et al. (2009) according to

$$\frac{\partial \phi}{\partial \mathbf{R}} = \bar{\mathbf{R}} = \frac{1}{2\hat{\sigma}^2}\mathbf{R}^{-T}\left(\mathbf{y} - \mathbf{F}\hat{\mu}\right)\left(\mathbf{y} - \mathbf{F}\hat{\mu}\right)^T \mathbf{R}^{-T} - \frac{1}{2}\mathbf{R}^{-T}. \tag{41}$$

where $\bar{\mathbf{R}}$ is the adjoint of the correlation matrix. This is applicable to both the non-gradient and gradient-enhanced case.

6.2 Gradients of the condition number w.r.t. the correlation matrix

The adjoint method can also be used for obtaining gradients of the condition number with respect to the correlation matrix. Using the chain rule and recalling (34) the derivatives of the condition number with respect to the correlation matrix can be written as

$$\frac{\partial \kappa}{\partial \mathbf{R}} = \frac{\partial \|\mathbf{R}^{-1}\| \|\mathbf{R}\|}{\partial \mathbf{R}} = \frac{\partial \|\mathbf{R}\|}{\partial \mathbf{R}} \|\mathbf{R}^{-1}\| + \frac{\partial \|\mathbf{R}^{-1}\|}{\partial \mathbf{R}} \|\mathbf{R}\| \tag{42}$$

With this result the intermediate variables for reversed differentiation of the condition number with respect to the correlation matrix can be determined. The intermediate variable for the first term can, given that the intermediate variable for the condition number itself has been initialised to $\bar{\kappa} = 1$, be written as

$$\overline{\|\mathbf{R}\|} = \bar{\kappa} \|\mathbf{R}^{-1}\| = \|\mathbf{R}^{-1}\|. \tag{43}$$

Using the results presented by Giles (2008) which are based on the work of Dwyer and MacPhail (1948) the adjoint of the Frobenius norm can be determined according to

$$\overline{\mathbf{R}} = \overline{\|\mathbf{R}\|} \frac{1}{\|\mathbf{R}\|} \mathbf{R} \tag{44}$$

which together with (43) leads to the adjoint of the correlation matrix for the first term in (42)

$$\overline{\mathbf{R}}_1 = \frac{\|\mathbf{R}^{-1}\|}{\|\mathbf{R}\|} \mathbf{R}. \tag{45}$$

In the second term the intermediate variable from the product rule can be obtained as

$$\overline{\|\mathbf{R}^{-1}\|} = \bar{\kappa} \|\mathbf{R}\| = \|\mathbf{R}\| \tag{46}$$

Again, using the adjoint of the Frobenius norm leads to

$$\overline{\mathbf{R}^{-1}} = \overline{\|\mathbf{R}^{-1}\|} \frac{1}{\|\mathbf{R}^{-1}\|} \mathbf{R}^{-1}. \tag{47}$$

Giles (2008) also presents the Adjoint of the inverse as

$$\overline{\mathbf{R}} = -\mathbf{R}^{-T} \overline{\mathbf{R}^{-1}} \mathbf{R}^{-T} \tag{48}$$

which together with (47) and (46) leads to the adjoint of the correlation matrix for the second term in (42)

$$\overline{\mathbf{R}}_2 = -\mathbf{R}^{-T} \frac{\|\mathbf{R}\|}{\|\mathbf{R}^{-1}\|} \mathbf{R}^{-1} \mathbf{R}^{-T} \tag{49}$$

Adding (45) and (49) yields the gradients of the condition number with respect to the hyper parameters as

$$\frac{\partial \kappa}{\partial \mathbf{R}} = \overline{\mathbf{R}}_1 + \overline{\mathbf{R}}_2 = \frac{\|\mathbf{R}^{-1}\|}{\|\mathbf{R}\|} \mathbf{R} - \frac{|\mathbf{R}\|}{\|\mathbf{R}^{-1}\|} \left(\mathbf{R}^{-T}\, \mathbf{R}^{-1} \mathbf{R}^{-T} \right), \tag{50}$$

which is applicable both for the non-gradient and gradient-enhanced case.

6.3 Gradients of the correlation matrix w.r.t. regularisation parameters

The gradients of the correlation matrix with respect to the regularisation parameters can easily be obtained from (30) for the non-gradient case as

$$\frac{\partial R}{\partial \lambda} = \mathbf{I}_p \tag{51}$$

where $\mathbf{I}_p \in \mathbb{R}^{p \times p}$, is the identity matrix with the number of diagonal elements of p. For the gradient-enhanced case from (31), for the first regularisation parameter as

$$\frac{\partial R}{\partial \lambda_1} = \begin{bmatrix} \mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{52}$$

and for the second regularisation parameter as

$$\frac{\partial R}{\partial \lambda_2} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_d \end{bmatrix} \tag{53}$$

where $d = p \times n$.

6.4 Gradients of the correlation matrix w.r.t. the hyper parameters

For the non-gradient case, the partial derivatives of the correlation matrix with respect to the hyper parameters can be calculated as

$$\frac{\partial R_{i,j}}{\partial \theta_m} = -\left(x_i^m - x_j^m \right)^2 R_{i,j}. \tag{54}$$

For the gradient enhanced case they can be calculated as

$$\frac{\partial \mathbf{R}}{\partial \theta_m} = \begin{bmatrix} \frac{\partial \mathbf{Q}^{1,1}}{\partial \theta_m} & \frac{\partial \mathbf{Q}^{1,2}}{\partial \theta_m} \\ \frac{\partial (\mathbf{Q}^{1,2})^T}{\partial \theta_m} & \frac{\partial \mathbf{Q}^{2,2}}{\partial \theta_m} \end{bmatrix}. \tag{55}$$

where the first quadrant can be calculated according to (54) as

$$\frac{\partial \mathbf{Q}^{1,1}}{\partial \theta_m} = -\left(x_i^m - x_j^m\right)^2 R_{i,j}. \tag{56}$$

and, through the derivation shown in Appendix A, to the following expression for the upper right quadrant

$$\frac{\partial Q_{i,jk}^{1,2}}{\partial \theta_m} = \begin{cases} \left[\frac{1}{\theta_m} - \left(x_i^m - x_j^m\right)^2\right] Q_{i,jk}^{2,1} & , m = k \\ -\left(x_i^m - x_j^m\right)^2 Q_{i,jk}^{1,2} & , m \neq k \end{cases}, \tag{57}$$

and the lower right quadrant

$$\frac{\partial Q_{il,jk}^{2,2}}{\partial \theta_m} = \begin{cases} -\left(x_i^m - x_j^m\right)^2 Q_{il,jk}^{2,2} & , m \neq k, m \neq l \\ \left[\frac{1}{\theta_k} - \left(x_i^m - x_j^m\right)^2\right] Q_{il,jk}^{2,2} & , m = k, m \neq l \\ \left[\frac{1}{\theta_l} - \left(x_i^m - x_j^m\right)^2\right] Q_{il,jk}^{2,2} & , m \neq k, m = l \\ \left[2 - 8\theta_k \left(x_i^m - x_j^m\right)^2\right] R_{i,j} - \left(x_i^m - x_j^m\right)^2 Q_{il,jk}^{2,2} & , m = k, m = l \end{cases}, \tag{58}$$

for $i = 1, ..., p$, $j = 1, ..., p$, $k = 1, ..., n$ and $l = 1, ..., n$.

7 Computational performance

In order to get an idea of the computational cost of obtaining the function values and the partial derivatives of the parameter tuning problem a benchmark example was carried out. The benchmark was carried out using a 76 design variable analytical function with 100 training points. The computational cost of the various routines for the gradient enhanced case are outlined in Table 1. It is shown that the cost of calculating the condensed log likelihood function value and the condition number of the correlation matrix adds up to 6.9 seconds while their partial derivatives with respect to the hyper parameters and regularisation parameters takes 45.3 seconds. This means that for this particular case the cost of the partial derivatives are 6.6 times more expensive than the function values themselves. This is of course less costly than obtaining the gradients through the direct method or finite differences which would incurr a computational cost of around 76 times (the number of design variables) the cost of performing a function evaluation.

8 Proposed optimisation approach

In the proposed approach the aim is to use the derived gradients for hyper parameter optimisation using gradient based methods. Here a method for choosing bounds and a suitable start-point for a gradient based optimisation algorithm is presented.

8.1 Finding a suitable starting point

As seen in, for instance, (Chung and Alonso 2002) it is possible to reduce the complexity of the hyper-parameter

Table 1 Computational cost for evaluation of the various variables in kriging for a test problem with 76 design variables and 100 training points

Variables	Description	Time [s]
Objective and constraint functions		
R,B,f	Pre-processing	0.3
$\mathbf{LL}^T$	Cholesky decomposition	1.7
$\lvert\mathbf{R}\rvert$	Determinant	<0.1
$\hat{\mu}$	System mean	<0.1
$\hat{\sigma}^2$	System variance	<0.1
ϕ	Condensed likelihood function	<0.1
$\mathbf{R}^{-1}$	Inverse of **R**	4.7
κ	Condition number of **R**	0.1
	Total	6.9
Derivatives of the objective and constraint functions		
$\partial\phi/\partial\mathbf{R}$	Partial derivatives of ϕ w.r.t. **R**	0.7
$\partial\kappa/\partial\mathbf{R}$	Partial derivatives of κ w.r.t. **R**	20.8
$\partial\mathbf{R}/\partial\boldsymbol{\theta}$	Partial derivatives of **R** w.r.t. $\boldsymbol{\theta}$	23.9
$\partial\mathbf{R}/\partial\boldsymbol{\lambda}$	Partial derivatives of **R** w.r.t. $\boldsymbol{\lambda}$	<0.1
	Total	45.4

The study was carried out on a computer with the following specifications: Intel(R) Core(TM) i7-4800MQ CPU @ 2.70GHz, and using Intel Math Kernel Library 11.2 (Intel 2015) for matrix multiplication *(DGEMM, DSYRK, DSYMM)*, Cholesky decomposition *(DPOTRF)* and backsubstitution *(DPOTRS)*, matrix inverse *(DPOTRI)*, norm *(DLANGE)* and vector multiplications *(DGEMV)* optimisation problem by considering the set of hyper-parameters as a single variable according to

$$\theta = [\theta_1, ..., \theta_n] = \gamma\,[1, ..., 1] \tag{59}$$

where γ is the single considered variable. This is more commonly known as a radial basis function (RBF). The resulting, reduced, optimisation problem can be solved using a one dimensional line search, in this case a golden search (GS). This greatly reduces the computational cost of the optimisation problem but also limits the optimisation to find a solution on the hyper diagonal of the design space. Here, instead of accepting the resulting point as the final solution, it is used as a starting point for optimisation in full space. The reduced optimisation problem is defined as

$$\begin{aligned} \underset{\gamma}{\text{maximise}} \quad & \phi(\theta) \\ \text{subject to} \quad & \kappa(\mathbf{R}(\theta)) \leq \kappa_{max} \\ & \theta = \gamma\,[1, ..., 1] \\ & 0 < \gamma \leq \gamma_{max} \end{aligned} \tag{60}$$

where κ_{max} is the upper bound constraint on the condition number, chosen as a user input, and γ_{max} is the upper bound of the single hyper parameter, chosen such that all off diagonal elements of the correlation matrix can become sufficiently small, i.e. such that $min(\mathbf{R}_{ij}) = R_{min}, i \neq j$, where R_{min} is a user input. In this work $R_{min} = 10^{-6}$.

8.2 Gradient based optimisation

After a starting point has been found through the golden search a gradient based method is to be used in order to explore the full hyper-parameter and regularisation parameter space. Two gradient based optimisation methods are considered, the method of feasible directions (MFD) developed by Vanderplaats (1973) based on the work of Zoutendijk (1960) and sequential quadratic programming (SQP) developed by Madsen et al. (2002) based on the work of Powell (1978).

To ensure a well conditioned correlation matrix at the solution, the condition number is constrained throughout the optimisation. This is enabled through use of the gradients of the condition number with respect to the hyper and regularisation parameters as outlined in Section 6.

9 Comparative study of optimisation approaches

In this section the proposed approach outlined in the previous section is compared to a selection of optimisation approaches, listed in Table 2. These approaches include sequential quadratic programming (SQP) and method of feasible directions (MFD) from one random (R-) start point, 10 multi (M-) start-points and the proposed method whereby the start-point is found by a golden search (GS-). Furthermore a genetic algorithm (GA) with 5000 evaluations and MDF and SQP starting from the resulting GA solution is included in the study.

Table 2 Considered optimisation methods and corresponding abbreviations

Abbreviation	Optimisation method
GS	Golden search
R-MFD	Random start MFD
R-SQP	Random start SQP
GS-MFD	MFD starting from GS result
GS-SQP	SQP starting from GS result
M-MFD	Multi-start MFD
M-SQP	Multi-start SQP
GA	Genetic algorithm
GA-MFD	MFD starting from GA result
GA-SQP	SQP starting from GA result

The study consists of two parts. The first one is carried out on two dimensional functions and the second on a dimensionally scalable problem.

9.1 Two dimensional benchmark study

This study investigates and compares the performance of the optimisation approaches on a suite of two dimensional analytical functions. The functions used in the case study, selected from those presented in Jamil and Yang (2013), are presented in Table 3. In order to reduce the risk of sporadic solutions 50 design of experiments (DOEs) were generated using different seed. Each of these were used in the optimisation of the tuning parameters for the metamodel.The mean time spent, the mean resulting condensed log likelihood and

Table 3 Two dimensional benchmark functions

Function name	Equation
Six-hump	$f(\mathbf{x}) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (4x_2^2 - 4)x_2^2$
Branin-Hoo	$f(\mathbf{x}) = \left(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6\right)^2$ $+10\left(1 - \frac{1}{8\pi}\right)cos(x_1) + 10$
Himmelblau	$f(\mathbf{x}) = \left(x_1^2 + x_2 - 11\right)^2 + \left(x_1 + x_2^2 - 7\right)^2$
Ursem	$f(\mathbf{x}) = -sin(2x_1 - 0.5\pi) - 3cos(x_2) - 0.5x_1$
Adjiman	$f(\mathbf{x}) = cos(x_1)sin(x_2) - \frac{x_1}{x_2^2+1}$
Keane	$f(\mathbf{x}) = \frac{sin^2(x_1-x_2)sin^2(x1+x2)}{\sqrt{x_1^2+x_2^2}}$

the mean generalisation error of the metamodels, for each of the optimisation approaches, is shown in Table 4.

The GA, GA-MFD and GA-SQP provide the highest condensed log-likelihood values, however these methods take the longest of the tested methods as the number of evaluations carried out by GA within the 2D design variable space is exhaustive. The GS, R-MFD and R-SQP provide the worst results. It is possible to increase the likelihood that a good value is found by the MFD and SQP by using a multi-start strategy as shown, however this increases the amount of time required to build the metamodel. The GS-MFD and GS-SQP provide similar resulting values of the condensed log-likelihood function to the GA-MFD and GA-SQP results at a far lower computation cost. In this case the M-MFD is also finding high values of the condensed log likelihood function to a relatively low computational cost, albeit higher than the GS-MFD and GS-SQP. It is worth noting that for these functions there seems to be a good correlation between a high log likelihood and a low generalisation error.

9.2 Dimensionally scalable benchmark study

This study aims to benchmark the optimisation techniques for functions with higher dimensionality. This was done using the following dimensionally scalable polynomial function

$$\begin{aligned} f(\mathbf{x}) &= \tfrac{i}{n}\sum_i c_1 x_i^3 + c_2 x_i^2 + c_3 x_i + c_4 + c_5 sin(f x_i) \\ c_1 &= 0.5 \\ c_2 &= -2.02\left(i - \tfrac{n}{2}\right) \\ c_3 &= 7.0\left(i - \tfrac{n}{2}\right) \\ c_4 &= 1.0 \\ c_5 &= 35.0 \\ c_6 &= 1.0 \end{aligned} \tag{61}$$

where n is the total number of design variables, chosen as 10, 40 and 60 respectively in this benchmark study. The function has varying degrees of non-linearity between the different design variables and is evaluated in the range 0 to 5. As with the 2D function 50 different training DOEs were evaluated for each of the three cases in order to reduce the risk of sporadic solutions.

Tables 5, 6 and 7 show the results of the parameter tuning for the dimensionally scalable polynomial in the cases of 10, 40 and 60 design variables respectively. For the 10 design variable case, Table 5, it can be seen that the GS-MFD and GS-SQP perform very well in comparison to the other algorithms, providing the highest log-likelihood together with the hybrid GAs. In this case the remaining algorithms do

Table 4 Condensed log-likelihood of gradient enhanced kriging metamodel built using 12 training points averaged over 50 training DoEs

	Time	Mean	RMSE	Time	Mean	RMSE	Time	Mean	RMSE
	(ms)	ϕ	%	(ms)	ϕ	%	(ms)	ϕ	%
	Six Hump			Branin Hoo			Himmelblau		
GS	4.8	−41.05	12.89 %	6.2	−33.21	12.78 %	3.8	−27.29	3.69 %
R-MFD	3.5	−35.20	10.90 %	3.7	−25.13	10.46 %	3.6	−29.68	5.23 %
R-SQP	34.2	−34.49	10.78 %	28.6	−25.18	10.37 %	20.2	−32.38	6.26 %
GS-MFD	7.5	−33.04	9.86 %	9.9	−23.39	9.57 %	7.8	−26.86	3.72 %
GS-SQP	12.1	−33.38	10.07 %	12.4	−23.56	9.67 %	8.6	−26.86	3.71 %
M-MFD	19.7	−33.27	9.81 %	15.5	−23.36	9.69 %	15.4	−25.88	2.94 %
M-SQP	373.8	−33.04	9.91 %	253.9	−23.24	9.64 %	156.6	−26.48	3.10 %
GA	376.4	−33.04	9.92 %	376.3	−23.24	9.64 %	376.2	−25.42	2.50 %
GA-MFD	379.9	−33.03	9.92 %	380.0	−23.24	9.64 %	379.1	−25.39	2.49 %
GA-SQP	383.3	−33.03	9.92 %	382.6	−23.24	9.64 %	382.2	−25.39	2.49 %
	Ursem			Adjiman			Keane		
GS	4.6	−2.26	1.71 %	4.3	51.20	0.19 %	4.9	−31.25	10.10 %
R-MFD	4.6	8.70	4.29 %	4.1	12.86	5.49 %	3.5	−31.11	10.18 %
R-SQP	20.6	17.27	1.60 %	12.2	34.68	2.56 %	16.5	−32.65	11.44 %
GS-MFD	8.8	23.19	0.68 %	7.1	51.97	0.19 %	6.5	−31.11	10.17 %
GS-SQP	10.4	22.28	0.70 %	9.4	52.02	0.18 %	7.7	−31.11	10.17 %
M-MFD	16.5	23.19	0.68 %	20.7	47.86	0.47 %	15.3	−31.09	9.77 %
M-SQP	160.6	23.19	0.68 %	113.7	52.04	0.19 %	137.3	−31.11	10.17 %
GA	373.9	23.19	0.68 %	374.5	52.02	0.19 %	378.4	−31.11	10.17 %
GA-MFD	377.3	23.19	0.68 %	377.7	52.04	0.19 %	381.9	−31.11	10.17 %
GA-SQP	378.9	23.19	0.68 %	379.9	52.03	0.19 %	385.2	−31.11	10.17 %

not perform as well.The GS-MFD and GS-SQP provide the lowest generalised error over the 50 validation DoEs, followed by the GA-MFD and GA-SQP.

When increasing dimensionality of the scalable polynomial function to 40 design variables the benefit of the proposed approach becomes more apparent. The proposed approach delivers a solution with high mean condensed log-likelihood value for a low computational effort when compared to the other evaluated methods. The GS-SQP provides the highest mean condensed log-likelihood value for all numbers of training points. The GS-MFD provides the second highest mean log likelihood over all of the number of training points, followed by the GA-SQP. In the 50 training point case the M-SQP provides the second highest mean condensed log likelihood. However, for the 10 and 20 training point cases does not perform as well. For the 10 and 20 training point cases the GS-MFD and GS-SQP take slightly longer to build than the R-MFD and R-SQP. However, for the 50 training point case the GS-SQP takes less than half the time taken to build the R-SQP. The lowest generalisation error is provided by the GS, GS-MFD and GS-SQP. As more training points are used the GS-MFD and GS-SQP provide a better generalisation error.

In the final case with 60 design variables the GS-MFD and GS-SQP also perform very well. For 10 and 20 training points they provide the highest mean log-likelihood values. For 50 training points the GS-SQP provides the highest mean condensed log likelihood followed closely by the M-SQP, then the GA-SQP and R-SQP, at 25, 15 and 2.5 times the computational effort respectively.

Of the proposed methods the GS-MFD and GS-SQP provide the best results for the time required to build the metamodels. They consistently outperform the random start point, GA-MFD and GA-SQP methods. As the dimensionality of the scalable polynomial function increases the benefit of using the solution of the GS as a starting point for the MFD or SQP increases. Overall the GS-SQP provides the best mean log-likelihood for the time required to build the metamodel, as such will be used in Section 10 for a industrial sized test case.

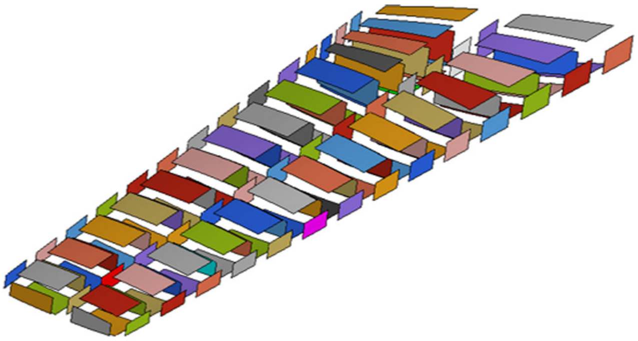
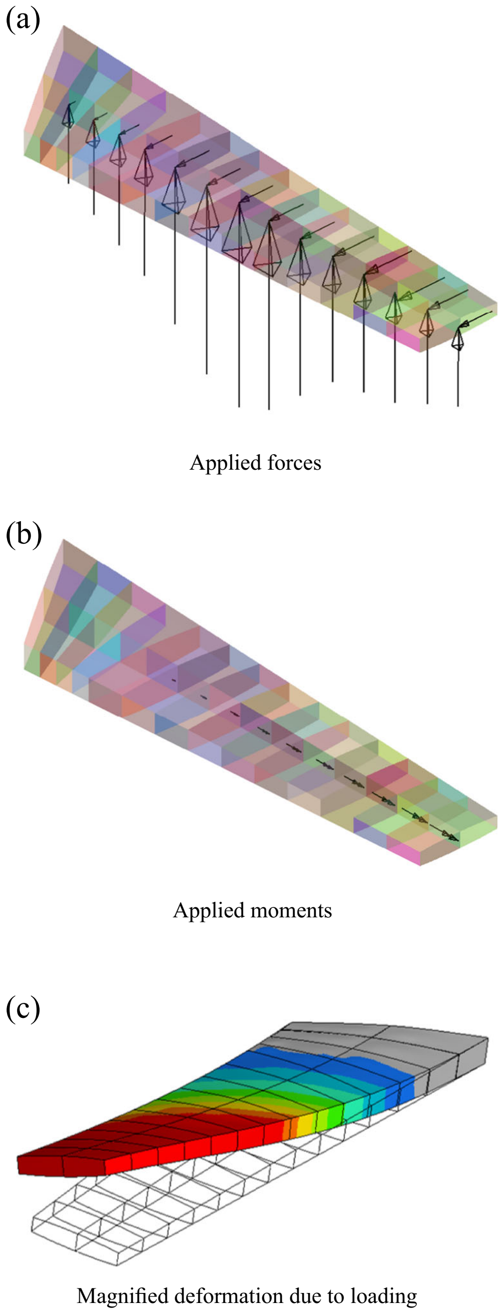


Fig. 4 Wing panel design variables

10 Case study: aircraft wing example

This section presents a study where gradient enhanced kriging metamodels are created for a finite element model of an aircraft wing. The GS-MFD and GS-SQP methods are compared to the GS, R-MFD and R-SQP methods. The multi-start and GA start point methods are not included as the computational effort would be too great.



Applied forces

Applied moments

Magnified deformation due to loading

Fig. 5 Wing loading

Table 5 Scalable polynomial 10 design variables

	10 Training Points			20 Training Points			50 Training Points		
	Time	Mean	RMSE	Time	Mean	RMSE	Time	Mean	RMSE
	$(hh : mm : ss)$	ϕ	%	$(hh : mm : ss)$	ϕ	%	$(hh : mm : ss)$	ϕ	%
GS	<00:00:01	39.12	16.32 %	<00:00:01	107.50	14.93 %	<00:00:01	428.49	10.76 %
R-MFD	<00:00:01	66.90	16.44 %	00:00:02	182.21	12.73 %	00:00:10	618.52	9.77 %
R-SQP	<00:00:01	54.22	16.05 %	<00:00:01	142.08	15.17 %	00:00:03	548.38	12.61 %
GS-MFD	<00:00:01	76.97	13.88 %	00:00:01	202.91	10.66 %	00:00:04	729.28	8.40 %
GS-SQP	<00:00:01	77.19	13.59 %	<00:00:01	197.48	10.99 %	00:00:03	729.72	8.39 %
M-MFD	00:00:01	71.72	15.57 %	00:00:11	193.84	11.28 %	00:01:13	671.56	8.86 %
M-SQP	<00:00:01	74.74	13.96 %	00:00:01	171.87	13.62 %	00:00:31	718.10	8.58 %
GA	00:00:03	74.19	14.47 %	00:00:13	183.26	12.07 %	00:01:06	656.52	8.84 %
GA-MFD	00:00:03	74.74	14.37 %	00:00:14	199.83	10.85 %	00:01:10	728.86	8.39 %
GA-SQP	00:00:03	77.81	13.44 %	00:00:13	193.35	11.39 %	00:01:08	729.68	8.39 %

Table 6 Scalable polynomial 40 design variables

	10 Training Points			20 Training Points			50 Training Points		
	Time	Mean	RMSE	Time	Mean	RMSE	Time	Mean	RMSE
	$(hh : mm : ss)$	ϕ	%	$(hh : mm : ss)$	ϕ	%	$(hh : mm : ss)$	ϕ	%
GS	<00:00:01	519.24	15.67 %	00:00:01	1153.24	14.60 %	00:00:11	3190.31	13.85 %
R-MFD	00:00:05	610.67	17.44 %	00:00:23	1328.32	17.19 %	00:03:07	3647.07	16.38 %
R-SQP	00:00:01	566.24	17.51 %	00:00:12	1329.35	17.22 %	00:03:01	3937.76	12.99 %
GS-MFD	00:00:06	664.50	15.54 %	00:00:25	1435.77	14.31 %	00:03:10	3945.74	12.87 %
GS-SQP	00:00:03	669.91	15.98 %	00:00:12	1437.64	14.51 %	00:01:18	3946.86	12.92 %
M-MFD	00:00:49	632.29	17.38 %	00:03:43	1350.05	17.14 %	00:29:34	3697.19	14.91 %
M-SQP	00:00:13	603.41	17.40 %	00:02:10	1355.04	16.83 %	00:30:59	3946.38	12.92 %
GA	00:00:41	622.84	17.48 %	00:03:00	1321.30	17.26 %	00:20:47	3564.98	17.01 %
GA-MFD	00:00:43	633.20	17.41 %	00:03:14	1347.24	17.18 %	00:23:43	3696.73	14.97 %
GA-SQP	00:00:44	659.21	16.42 %	00:03:04	1352.33	16.84 %	00:22:39	3933.32	13.07 %

Table 7 Scalable polynomial 60 design variables

	10 Training Points			20 Training Points			50 Training Points		
	Time	Mean	RMSE	Time	Mean	RMSE	Time	Mean	RMSE
	$(hh : mm : ss)$	ϕ	%	$(hh : mm : ss)$	ϕ	%	$(hh : mm : ss)$	ϕ	%
GS	<00:00:01	926.49	15.03 %	00:00:03	2048.53	14.13 %	00:00:26	5487.08	13.58 %
R-MFD	00:00:12	1040.72	16.91 %	00:01:03	2291.84	16.66 %	00:07:54	6133.60	16.20 %
R-SQP	00:00:07	1041.60	16.88 %	00:00:43	2329.69	16.59 %	00:10:27	6562.21	13.00 %
GS-MFD	00:00:12	1137.31	14.98 %	00:01:07	2463.50	14.05 %	00:08:04	6559.71	12.92 %
GS-SQP	00:00:09	1156.69	16.00 %	00:00:36	2471.26	14.38 %	00:04:13	6563.42	13.00 %
M-MFD	00:02:12	1082.86	16.87 %	00:10:42	2332.23	16.63 %	01:21:39	6167.31	15.79 %
M-SQP	00:01:16	1101.63	16.62 %	00:07:33	2376.14	15.71 %	01:45:10	6563.21	13.00 %
GA	00:01:30	1050.56	16.95 %	00:07:09	2241.87	16.70 %	00:54:27	5898.04	16.56 %
GA-MFD	00:01:40	1080.82	16.92 %	00:09:47	2320.88	16.67 %	01:03:26	6126.89	16.27 %
GA-SQP	00:01:36	1093.91	16.59 %	00:08:55	2309.32	16.68 %	01:02:19	6562.34	13.00 %

10.1 The wing model

The wing model consists of 126 aluminium sheet panels with designable thickness; as shown in Fig. 4. Each of the design variables are aluminium sheets which are modelled with shell elements with a mesh size of 18mm. The allowable thickness range is from 0.5 mm–5 mm with a nominal thickness of 2.5 mm.

The wing is fully constrained on the wider end to represent attachment to the fuselage. Forces and moments are applied to nodes located at the centroid of each rib, as shown in Fig. 5a and b, and distributed to the edges of the rib using one dimensional distributing coupling (RBE3) elements. An example of the deformation due to the loading is shown in Fig. 5c. Two responses are considered: vertical wing tip deflection and the rotation of the wing tip. Deflection and rotation are measured in horizontal centre and vertical top of the wing tip.

The model is analysed using OptiStruct v13.0.210 (Altair Engineering Inc 2014). OptiStruct provides analytical gradients via either the direct or the adjoint method depending on which is the more efficient choice for the case. In this case the adjoint method is used as the number of design variables is far greater than the number of responses; evaluating the gradients took roughly the same time as evaluating the function values, doubling the total analysis time.

10.2 Study set-up

The study was performed by building the meta-models with 5 points at first, followed by 10, 20, 50 points. For this purpose, sampling was performed using MELS. One benefit of MELS is that any subset of the DoE in sequence from the first point is suitably spaced. This allows the user to assess the approximation quality interactively allowing for a far more flexible approach than would be possible with other space filling techniques such as the Optimal Latin Hypercube (Audze and Eglajs 1977). To leverage this feature a single DoE of was created. 50 points were reserved for training the metamodels and 500 points were reserved for validation.

Table 8 Results for wing tip displacement

No. points	GS	R-MFD	R-SQP	GS-MFD	GS-SQP
(a) Condensed log likelihood (ϕ)					
5	732	798	1082	1731	1780
10	2197	2173	4069	3915	4241
20	5406	5113	6434	8275	8725
50	14696	13912	17415	21737	21853
(b) Generalisation error (RMSE)					
5	14.77 %	14.78 %	14.78 %	13.27 %	10.69 %
10	6.56 %	12.57 %	10.84 %	5.17 %	7.66 %
20	6.34 %	12.64 %	12.59 %	5.01 %	6.66 %
50	5.82 %	12.37 %	12.32 %	6.33 %	7.26 %

Table 9 Results for wing tip rotation

No. points	GS	R-MFD	R-SQP	GS-MFD	GS-SQP
(a) Condensed log likelihood (ϕ)					
5	1360	1384	1500	2091	2157
10	2768	2621	3682	3949	3910
20	5982	5293	6371	7824	7916
50	15998	13494	16003	19123	19236
(b) Generalisation error (RMSE)					
5	10.10 %	8.16 %	8.16 %	9.07 %	9.06 %
10	4.50 %	7.80 %	7.64 %	4.81 %	6.32 %
20	4.55 %	7.62 %	7.60 %	3.91 %	4.16 %
50	4.07 %	7.18 %	7.18 %	3.49 %	3.68 %

10.3 Results

Tables 8 and 9 show the performance of the GS-MFD and GS-SQP is compared with that of the GS, R-MFD and R-SQP. In Table 8a the condensed log likelihood obtained by the different optimisation methods is shown for the wing tip displacement, it can be seen that the GS-SQP outperforms the other methods, closely followed by the GS-MFD which are second best in all cases apart from in the 10 point case where R-SQP provides a slightly better solution.

In Table 8b the generalisation error obtained for the different optimisation methods is shown. It can be seen that GS, GS-MFD and GS-SQP provide solutions which outperform R-MFD and R-SQP. It can also be seen that even though GS-MFD and GS-SQP provide condensed log likelihood values which are higher than the one for GS, the generalisation error is not necessarily improved.

Table 9a shows that, for wing tip rotation, the GS-MFD and GS-SQP out perform the other evaluated methods, which is reflected in the generalisation error, Table 9b. Similarly to the wing tip displacement, the golden search optimisation method shows lower resulting generalisation error than the R-MFD and R-SQP.

11 Conclusions

In this work an approach was suggested for efficient hyper parameter optimisation for building well conditioned gradient enhanced kriging metamodels. The approach consists of two steps, a one dimensional line search where all hyper-parameters are treated as one variable, and a gradient based optimisation starting from the solution of the initial line search. In order to ensure a suitable condition number of the correlation matrix, an upper bound constraint was enforced. Partial derivatives of the condition number with respect to the correlation matrix was derived in order to use this constraint in the gradient based optimisation approach. Both the method of feasible directions and sequential quadratic programming was evaluated within the approach.

The approach was compared to random start point gradient based algorithms, multiple start point gradient based algorithms and a genetic algorithm followed by gradient based algorithms from promising points. It was shown that the approach outperforms random start-point and multi-start gradient based algorithms in terms of both computational performance and quality of solutions. The comparative study shows the SQP to be the better choice of algorithm within the approach as it provides slightly higher condensed log-likelihood values than the MFD for a similar time to build.

The proposed approach, using both the SQP and MFD, was compared to a selection of the other optimisation approaches using an aircraft wing model comprising of 126 thickness design variables. The GS-SQP consistently provides the highest condensed log likelihood value closely followed by the GS-MFD.

In some case it was shown that a big improvement in log likelihood did not necessarily translate to an improvement in generalisation error. This was particularly apparent for the wing tip displacement metamodels where the GS-SQP provided a higher condensed log likelihood than the GS case but the generalisation error was of comparable magnitude.

A possible limitation of the proposed strategy may be the assumption made when finding the starting point, that the optimum lies close to the hyper diagonal of the hyper parameter space. If this assumption is not correct for a given problem then the efficiency of the proposed strategy may be reduced. However this was not shown in the scalable polynomial example which was chosen for it's varying non-linearity between design variables.

Acknowledgments The authors are very grateful to the Altair HyperStudy team for their support throughout this work. Jonathan Ollar, Vassili Toropov and Royston Jones greatly appreciate funding from European Union Seventh Framework Programme FP7-PEOPLE-2012-ITN under grant agreement 316394, Aerospace Multidisciplinarity Enabling DEsign Optimization (AMEDEO) Marie Curie Initial Training Network. Charles Mortished and Johann Sienz greatly appreciate funding from EPSRC EDT (EP/I015507/1) Manufacturing Advances Through Training Engineering Researchers (MATTER). Vassili Toropov is grateful for the support provided by the Russian Science Foundation, project No. 16-11-10150.

Appendix A: Partial derivatives of correlation matrix for GEK

This section explains the derivation of the derivatives of the covariance matrix with respect to the hyper-parameters for the quadrants containing information relating to the design sensitivities.

A.1 Quadrant $Q^{1,2}$ and $Q^{2,1}$

The derivation for the derivatives of quadrant $Q^{1,2}$ that contains the covariance between the design sensitivities and the design function evaluations are shown below:

$$\frac{\partial Q^{1,2}_{i,j\times k}}{\partial \theta_m} = \frac{\partial \left(2\theta_k \left(x_i^k - x_j^k\right) \psi\left(\mathbf{x}_i, \mathbf{x}_j\right)\right)}{\partial \theta_m} \tag{62}$$

- For the case $k = m$:

$$\begin{aligned} &\frac{\partial (2\theta_m \cdot \left(x_i^m - x_j^m\right) \cdot \psi\left(\mathbf{x}_i, \mathbf{x}_j\right))}{\partial \theta_m} \\ &= 2\left(x_i^m - x_j^m\right) \cdot \psi\left(\mathbf{x}_i, \mathbf{x}_j\right) + 2\theta_m \cdot \left(x_i^m - x_j^m\right) \\ &\quad \cdot \left(-\left(x_i^m - x_j^m\right)^2\right) \cdot \psi\left(\mathbf{x}_i, \mathbf{x}_j\right) \\ &= \underbrace{2\theta_m \left(x_i^m - x_j^m\right) \cdot \psi\left(\mathbf{x}_i, \mathbf{x}_j\right)}_{Q^{1,2}_{i,j\cdot k}} \left[\frac{1}{\theta_m} - \left(x_i^m - x_j^m\right)^2\right] \\ &= \left[\frac{1}{\theta_m} - \left(x_i^m - x_j^m\right)^2\right] \cdot Q^{1,2}_{i,j\cdot k} \end{aligned} \tag{63}$$

- For the case $k \neq m$:

$$\begin{aligned} &\frac{\partial \left(2\theta_k \cdot \left(x_i^k - x_j^k\right) \cdot \psi\left(\mathbf{x}_i, \mathbf{x}_j\right)\right)}{\partial \theta_m} \\ &= 2\theta_k \cdot \left(x_i^k - x_j^k\right) \cdot \left(-\left(x_i^m - x_j^m\right)^2\right) \cdot \psi\left(\mathbf{x}_i, \mathbf{x}_j\right) \\ &= \underbrace{2\theta_k \left(x_i^k - x_j^k\right) \cdot \psi\left(\mathbf{x}_i, \mathbf{x}_j\right)}_{Q^{1,2}_{i,j\cdot k}} \left[-\left(x_i^m - x_j^m\right)^2\right] \\ &= \left[-\left(x_i^m - x_j^m\right)^2\right] \cdot Q^{1,2}_{i,j\cdot k} \end{aligned} \tag{64}$$

A.2 Quadrant $Q^{2,2}$

The derivation of the derivatives of quadrant $Q^{2,2}$ that contains the covariance between the design sensitivities are shown below.

$$\frac{\partial Q^{2,2}_{i\cdot l,j\cdot k}}{\partial\theta_m} = \begin{cases} \frac{\partial\left(2\theta_k\left(-2\theta_k\cdot\left(x_i^k-x_j^k\right)^2+1\right)\cdot\psi(\mathbf{x}_i,\mathbf{x}_j)\right)}{\partial\theta_m} & ,k=l \\ \frac{\partial(-4\theta_k\theta_l\left[\left(x_i^k-x_j^k\right)\left(x_i^l-x_j^l\right)\right]\psi(\mathbf{x}_i,\mathbf{x}_j)}{\partial\theta_m} & ,k\neq l \end{cases} \tag{65}$$

- For the case $k = l = m$

$$\begin{aligned} &\frac{\partial\left(2\theta_m\left(-2\theta_m\cdot\left(x_i^m-x_j^m\right)^2+1\right)\cdot\psi(\mathbf{x}_i,\mathbf{x}_j)\right)}{\partial\theta_m} \\ &= \left(4\theta_m^2\left(x_i^m-x_j^m\right)^4-10\theta_m\left(x_i^m-x_j^m\right)^2+2\right)\cdot\psi(\mathbf{x}_i,\mathbf{x}_j) \\ &= \left(x_i^m-x_j^m\right)^2\underbrace{\left(-2\theta_m\left(2\theta_m\left(x_i^m-x_j^m\right)^2-1\right)\right)\cdot\psi(\mathbf{x}_i,\mathbf{x}_j)}_{Q^{2,2}_{i\cdot l,j\cdot k}} \\ &\quad+\left(-8\theta_m\left(x_i^m-x_j^m\right)^2+2\right)\underbrace{\psi(\mathbf{x}_i,\mathbf{x}_j)}_{R_{i,j}} \\ &= \left[2-8\theta_k\left(x_i^m-x_j^m\right)^2\right]\cdot R_{i,j}-\left(x_i^m-x_j^m\right)^2\cdot Q^{2,2}_{i\cdot l,j\cdot k} \end{aligned} \tag{66}$$

- For the case $k \neq m, l \neq m$:

$$\begin{aligned} &\frac{\partial\Big(-4\theta_k\theta_l\left[\left(x_i^k-x_j^k\right)\left(x_i^l-x_j^l\right)\right]\psi(\mathbf{x}_i,\mathbf{x}_j\Big)}{\partial\theta_m} \\ &= \left(x_i^m-x_j^m\right)^2\underbrace{\left(4\theta_k\theta_l\left[\left(x_i^k-x_j^k\right)\left(x_i^l-x_j^l\right)\right]\psi(\mathbf{x}_i,\mathbf{x}_j)\right)}_{-Q^{2,2}_{i\cdot l,j\cdot k}} \\ &= -\left(x_i^m-x_j^m\right)^2\cdot Q^{2,2}_{i\cdot l,j\cdot k} \end{aligned} \tag{67}$$

- For the case $k = m, l \neq m$:

$$\begin{aligned} &\frac{\partial\Big(-4\theta_k\theta_l\left[\left(x_i^k-x_j^k\right)\left(x_i^l-x_j^l\right)\right]\psi(\mathbf{x}_i,\mathbf{x}_j\Big)}{\partial\theta_m} \\ &= \Bigg[\frac{1}{\theta_k}\underbrace{\left(-4\theta_k\theta_l\left(\left(x_i^k-x_j^k\right)\left(x_i^l-x_j^l\right)\right)\right)}_{Q^{2,2}_{i\cdot l,j\cdot k}/\psi(\mathbf{x}_i,\mathbf{x}_j)} \\ &\quad+\left(x_i^m-x_j^m\right)^2\underbrace{4\theta_k\theta_l\left(\left(x_i^k-x_j^k\right)\left(x_i^l-x_j^l\right)\right)}_{-Q^{2,2}_{i\cdot l,j\cdot k}/\psi(\mathbf{x}_i,\mathbf{x}_j)}\Bigg]\psi(\mathbf{x}_i,\mathbf{x}_j) \\ &= \left[\frac{1}{\theta_k}-\left(x_i^m-x_j^m\right)^2\right]\cdot Q^{2,2}_{i\cdot l,j\cdot k} \end{aligned} \tag{68}$$

- The case $k \neq m, l = m$ may be derived in the same manner as the previous case, leading to:

$$\begin{aligned} &\frac{\partial\Big(-4\theta_k\theta_l\left[\left(x_i^k-x_j^k\right)\left(x_i^l-x_j^l\right)\right]\psi(\mathbf{x}_i,\mathbf{x}_j)}{\partial\theta_m} \\ &= \left[\frac{1}{\theta_l}-\left(x_i^m-x_j^m\right)^2\right]\cdot Q^{2,2}_{i\cdot l,j\cdot k} \end{aligned} \tag{69}$$

References

Altair Engineering Inc (2014) Optistruct 13.0 user's guide. Altair Engineering, Inc

Audze P, Eglajs V (1977) New approach for planning out of experiments (in Russian), vol 35. Problems of Dynamics and Strengths

Barthelemy JFM, Haftka RT (1993) Approximation concepts for optimum structural design — a review. Structural Optimization 5(3):129–144

Chung HS, Alonso JJ (2002) Using gradients to construct cokriging approximation models for high-dimensional design optimization problems. AIAA paper 317:14–17

Dalbey KR (2013) Efficient and robust gradient enhanced kriging emulators

Dwyer PS, MacPhail M (1948) Symbolic matrix derivatives. The Annals of Mathematical Statistics:517–534

Forrester AI, Keane AJ (2009) Recent advances in surrogate-based optimization. Prog Aerosp Sci 45(13):50–79

Forrester AIJ, Keane AJ, Bressloff NW (2006) Design and analysis of "noisy" computer experiments. AIAA J 44(10):2331–2339

Forrester A, Sobester A, Keane A (2008) Engineering design via surrogate modelling: a practical guide. Wiley

Giles M (2008) An extended collection of matrix derivative results for forward and reverse mode automatic differentiation. Tech. Rep. 08/01, Oxford University Computing Laboratory

Haaland B, Qian PZ et al (2011) Accurate emulators for large-scale computer experiments. The Annals of Statistics 39(6):2974–3002

Han ZH, Görtz S, Zimmermann R (2013) Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function. Aerosp Sci Technol 25:177–289

Intel (2015) Reference Manual for Intel Math Kernel Library 11.2 Update 3. Intel

Jamil M, Yang XS (2013) A literature survey of benchmark functions for global optimisation problems. International Journal of Mathematical Modelling and Numerical Optimisation 4(2):150–194

Jones DR (2001) A taxonomy of global optimization methods based on response surfaces. J Glob Optim 21(4):345–383

Kim DW, Lee J (2010) An improvement of kriging based sequential approximate optimization method via extended use of design of experiments. Eng Optim 42(12):1133–1149

Krige D (1951) A statistical approach to some mine valuation and allied problems on the witwatersrand. Master's thesis, University of Witwatersrand

Liu W, Batill S (2002) Gradient-enhanced response surface approximations using kriging models. American Institute of Aeronautics and Astronautics, Atlanta

Lockwood BA, Anitescu M (2010) Gradient-enhanced universal kriging for uncertainty propagation. Preprint ANL/MCS-p1833-0111

Lukaczyk T, Taylor T, Palacios F, Alonso J (2013) Managing gradient inaccuracies while enhancing optimal shape design methods.

In: Proceedings of the 51st AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition

Madsen K, Nielsen HB, Sondergaard J (2002) Robust subroutines for non-linear optimization. University of Denmark IMM-REP-2002-02

Martins JRRA, Hwang JT (2013) Review and unification of methods for computing derivatives of multidisciplinary computational models. AIAA J 51(11):2582–2599

Matheron G (1963) Principles of geostatistics. Econ Geol 58:1246–1266

Powell MJD (1978) Numerical analysis: proceedings of the biennial conference held at Dundee, June 28–July 1, 1977:144–157. chap A fast algorithm for nonlinearly constrained optimization calculations

Rennen G (2008) Subset selection from large datasets for kriging modeling. Struct Multidiscip Optim 38(6):545–569

Sacks J, Welch WJ, Mitchell TJ, Wynn HP (1989) Design and analysis of computer experiments. Stat Sci 4(4):409–423

Toal DJJ, Forrester AIJ, Bressloff NW, Keane AJ, Holden C (2009) An adjoint for likelihood maximization. Proc R Soc 465:3267–3287

Toal DJJ, Bressloff NW, Keane AJ, Holden CME (2011) The development of a hybridized particle swarm for krign hyperparameter tuning. Eng Optim 43(6):1–28

Vanderplaats GN (1973) Conmin - a fortran program for constrained function minimization - user's manual, Technical Memorandum TM X-62,282, NASA

Viana FA, Gogu C, Haftka RT (2010) Making the most out of surrogate models: tricks of the trade. In: ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers, pp 587–598

Viana FAC, Simpson TW, Balabanov V, Toropov V (2014) Metamodeling in multidisciplinary design optimization: how far have we really come? Prog Aerosp Sci 52(4):670–690

Wang GG, Shan S (2007) Review of metamodeling techniques in support of engineering design optimization. J Mech Des 129(4):370–380

Xiong Y, Chen W, Apley D, Ding X (2007) A non-stationary covariance-based kriging method for metamodelling in engineering design. Int J Numer Methods Eng 71:733–756

Zimmermann R (2013) On the maximum likelihood training of gradient-enhanced spatial gaussian processes. SIAM J Sci Comput 35(6):2554–2574

Zimmermann R (2015) On the condition number anomaly of gaussian correlation matrices. Linear Algebra Appl 466:512–526

Zoutendijk G (1960) Methods of feasible directions. a study in linear and non-linear programming