Gradient-based multifidelity optimisation for aircraft design using Bayesian model calibration

A. March **amarch@mit.edu**
K. Willcox **kwillcox@mit.edu**
Q. Wang **qiqi@mit.edu**
Massachusetts Institute of Technology
Massachusetts, USA

ABSTRACT

Optimisation of complex systems frequently requires evaluating a computationally expensive high-fidelity function to estimate a system metric of interest. Although design sensitivities may be available through either direct or adjoint methods, the use of formal optimisation methods may remain too costly. Incorporating low-fidelity performance estimates can substantially reduce the cost of the high-fidelity optimisation. In this paper we present a provably convergent multifidelity optimisation method that uses Cokriging Bayesian model calibration and first-order consistent trust regions. The technique is compared with a single-fidelity sequential quadratic programming method and a conventional first-order trust-region method on both a two-dimensional structural optimisation and an aerofoil design problem. In both problems adjoint formulations are used to provide inexpensive sensitivity information.

NOMENCLATURE

c	a positive constant
$\mathbf{C}$	output matrix
C_p	pressure coefficient
d	radial distance
e	error model
f	an objective function
$\mathbf{f}$	applied force vector
g	an inequality constraint
h	an equality constraint
$\mathcal{J}$	an objective function
$\mathbf{K}$	stiffness matrix
l	number of state variables
m	surrogate model for high-fidelity function
$\mathbf{M}$	mesh
M_∞	Mach number
n	number of design variables
q	number of calibration points
$\mathbf{r}$	set of residual equations
$\mathbf{R}$	correlation matrix
$\mathbf{s}$	optimisation step
$\mathbf{u}$	vector of state variables
w	structural weight
$\mathbf{x}$	design vector
$\mathbf{y}$	Kriging values to interpolate
$\mathbf{z}$	a vector of ones and zeros
α	flow turning angle
β	Kriging constant
γ	trust region size update ratio
δ	finite change
Δ	trust region size
ε	termination tolerance
θ	a positive constant
Θ	likelihood
κ	bound related to function smoothness
λ	eigenvalue of $\mathbf{R}_c$

Paper No. 3644. Manuscript received 29 October 2010, accepted 2 June 2011.
This paper is based on a presentation given at the 2010 RAeS 2nd Aircraft Structural Design Conference, London, UK.

https://doi.org/10.1017/S0001924000006473 Published online by Cambridge University Press

μ	penalty parameter
ξ	spatial correlation parameter
ρ	trust region update parameter
σ^2	mean square error/variance
$\hat{\sigma}^2$	max. likelihood estimate of δ^2
Υ	penalty function
$\hat{\Upsilon}$	surrogate penalty function
ϕ	correlation function
Φ	correlation vector
Ψ	adjoint variable

Superscript

$^{(p)}$	Vector component
$^{(1:p)}$	Vector components
^{q}	q -component vector

Subscript

$_0$	Initial iterate
$_{bhm}$	Bound for Hessian of the surrogate model
$_c$	Cokriging
$_{high}$	Relating to the high-fidelity function
$_k$	Index of trust-region iteration number
$_{low}$	Relating to a lower-fidelity function
$_{max}$	User-set maximum value

1.0 INTRODUCTION

In the design of complex systems, such as aircraft, a high-fidelity simulation often appears as part of a design objective or constraint. Conveniently, in many applications these simulations provide design sensitivity information. For example, in structural design both direct and adjoint methods have been used to estimate sensitivity to the design parameters[15,16]. Similarly, in aerodynamic design, adjoint solutions are now quite common within computational fluid dynamics solvers and provide a means of estimating design sensitivities[17,28]. The challenge with designing aero-structural and other complex systems is that even with first-order design sensitivities provided inexpensively, the computational expense of these high-fidelity simulations coupled with the number of high-fidelity evaluations required still makes formal design optimisation intractable.

Fortunately, most system designers have other lower-fidelity models available to them that provide estimates of system performance with considerably lower computational requirements. These lower-fidelity models may be simplified physics models, or approximate models generated using methods such as response surfaces[9,33,34], reduced-order models[5], or coarse discretisations[4]. It is desirable to optimise a design with respect to the best available model of reality, typically the high-fidelity model, but lower-fidelity models may provide valuable information that can speed the optimisation process.

There are several different multifidelity optimisation strategies that optimise a high-fidelity function using a low-fidelity surrogate. One class of approaches uses trust regions. These methods are provably convergent to a local optimum of the high-fidelity function, if at the center of the trust region the low-fidelity function value and gradient are scaled or shifted to be equal to the high-fidelity function value and gradient[2-4]. Another multifidelity approach is to combine a pattern-search with optimisation of a surrogate model included as an additional pattern-search location. This derivative-free method is also provably convergent to an optimum of the high-fidelity function[6,7]. A third general approach is Efficient Global Optimisation (EGO) developed by Jones *et al*[18]. In this method, a Bayesian uncertainty approach is used to find regions in the design space with a high likelihood of having an optimal solution. EGO uses a combination of a regression model and an uncertainty estimate based on distance from known points as a way to find better points. An improvement to Jones' approach is to use model calibration techniques to model the difference or quotient between a high- and low-fidelity function as opposed to modeling the high-fidelity function itself[19,20]. In this way, a low-fidelity model can increase the efficiency of finding an optimum of a high-fidelity function in situations where using only a regression surface requires a considerable number of function evaluations for calibration. These techniques have been used for both derivative-free[22] and derivative-based[11] optimisation. Model calibration techniques are generally based on heuristic methods and are not provably convergent to an optimum of the high-fidelity function.

In this paper, we present a gradient-based multifidelity optimisation algorithm using model calibration and trust-regions that is globally convergent to a high-fidelity optimum. The first-order-consistent trust-region methods mentioned above can be thought of as employing model calibration; however, the calibration is only local and temporary, since sample points from previous iterations are not re-used. The challenge we address here is to produce a surrogate model that captures local function behavior sufficiently well to prove convergence, while capturing sufficient global function behavior to speed convergence. Robinson *et al* showed that when using globally calibrated models, for instance, interpolation or least-squares fits, the first-order-consistency requirement is typically detrimental to the performance and the fit of the model to the high-fidelity function[31]. However, Gano *et al* have successfully demonstrated globally calibrated models in a trust-region framework using Kriging-based additive, multiplicative and hybrid correction schemes[12]. We propose to use the model calibration approach of Kennedy and O'Hagan[19]; however, instead of using an ordinary Kriging model for calibration, we use a Cokriging model. A Kriging model interpolates a function exactly at all calibration points, but a Cokriging model interpolates a function exactly at all calibration points and with the same derivative as the function. The Cokriging technique enables us to satisfy the first-order-consistency requirements of a trust-region algorithm while maintaining a 'global' fit of the the high-fidelity function.

Cokriging surrogate models have been used to optimise expensive functions; however, typically in a heuristic manner. For instance, it has been shown that when using the same number of high-fidelity sample points, a Cokriging model will typically provide a better estimate of the optimum compared with other surrogates[13,14]. In addition, by updating Cokriging models during an optimisation using adjoint-based gradient estimates, Cokriging methods converge faster to a high-fidelity optimum than other calibration methods[23]; this was even found to be true when the number of Cokriging calibration points was limited to reduce computational cost[11]. However, with all of these Cokriging optimisation methods, convergence to a high-fidelity optimum is still not guaranteed because arbitrarily poor steps may occur during the optimisation process which could increase the high-fidelity function value, or the algorithm could take an infinite number of small steps and never reach an optimum.

This paper combines the provably convergent optimisation trust region frameworks of Alexandrov *et al*[2-4] and Conn *et al*[8] with Cokriging Bayesian model calibration ideas to create a provably convergent multifidelity optimisation algorithm. The calibration employed is only done within the vicinity of a trust-region to reduce the cost of constructing the surrogate model. In addition, techniques are developed to ensure the conditioning of the Cokriging model satisfy all requirements for convergence in a trust region framework.

Section 2 of this paper provides an overview of the trust-region algorithm, the fundamental ideas of Bayesian model calibration and the technique developed to construct the Cokriging surrogate. Section 3 demonstrates this algorithm on a 26-dimensional structural design problem and Section 4 demonstrates this method on an 11-dimensional aerofoil design problem. Both sample problems estimate gradients using adjoint solutions.

https://doi.org/10.1017/S0001924000006473 Published online by Cambridge University Press

2.0 OPTIMISATION METHOD

In this section we consider a setting where we have two (or more) models that represent the physical system of interest: a high-fidelity model that accurately estimates system metrics of interest but is expensive to solve, denoted $f_{\text{high}}(\mathbf{x})$, and a low-fidelity model with lower accuracy but cheaper to solve, denoted $f_{\text{low}}(\mathbf{x})$. The high-fidelity model is any function that maps a design vector, $\mathbf{x}$, of n design variables, to a scalar output. Accordingly, the optimisation problem considered is to minimise the high-fidelity objective function subject to equality constraints, $h(\mathbf{x})$, and inequality constraints $g(\mathbf{x})$:

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{R}^n} f_{\text{high}}(\mathbf{x}) \\ &\text{s.t. } \mathbf{h}(\mathbf{x}) = 0 \\ &\mathbf{g}(\mathbf{x}) \le 0. \end{aligned} \quad \ldots (1)$$

Of specific interest in this paper is the equivalent optimisation problem when the high-fidelity function comes from the solution of a partial differential equation or includes state variables. For example, we may have a problem with a high-fidelity objective $\mathcal{J}_{\text{high}}(\mathbf{x},\mathbf{u})$ with a vector $\mathbf{u}$ of l state variables set by l residual or state equations $\mathbf{r}_{\text{high}}(\mathbf{x},\mathbf{u}) = 0$. In problems of this form we rewrite the optimisation formulation as,

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{R}^n,\,\mathbf{u}\in\mathbb{R}^l} \mathcal{J}_{\text{high}}(\mathbf{x},\mathbf{u}) \\ &\text{s.t. } \mathbf{r}_{\text{high}}(\mathbf{x},\mathbf{u}) = 0 \\ &\mathbf{h}(\mathbf{x}) = 0 \\ &\mathbf{g}(\mathbf{x}) \le 0. \end{aligned} \quad \ldots (2)$$

When considering problems of the form of Equation (2) we can eliminate the state variables by solving the residual equations. Thus, these systems can be written in the form of Equation. (1), as in the following presentation of the multifidelity optimisation method.

2.1 Trust region method

To solve Equation (1) we may use any of the trust-region algorithms presented in Refs 3 or 4. This paper uses a modified form of sequential quadratic programming (SQP) approximation model management, which generates a sequence of design iterates $\boldsymbol{x}_k$ that converge to an optimum of the high-fidelity problem Equation (1). At each trust-region iteration we minimise a surrogate model $m_k(\mathbf{x})$ of the high-fidelity function. We define the surrogate model to have an additive error model, $e_k(\mathbf{x})$,

$$m_k(\mathbf{x}) = f_{low}(\mathbf{x}) + e_k(\mathbf{x}) \approx f_{high}(\mathbf{x}). \quad \ldots (3)$$

The requirements for convergence of this algorithm are that, (*i*) at each trust-region iteration the surrogate model satisfies first-order consistency requirements,

$$m_k(\boldsymbol{x}_k) = f_{high}(\boldsymbol{x}_k) \quad \ldots (4)$$

$$\nabla m_k(\boldsymbol{x}_k) = \nabla f_{high}(\boldsymbol{x}_k) \quad \ldots (5)$$

and (*ii*) that there exists a constant $k_{bhm} < \infty$ that is an upper bound for the 2-norm of the surrogate model Hessian[2,8].

The trust-region algorithm uses the l_1 penalty function,

$$\Upsilon(\mathbf{x},\mu_k) = f_{high}(\mathbf{x}) + \mu_k \| \mathbf{h}(\mathbf{x}) \|_1 + \mu_k \| \mathbf{g}^+(\mathbf{x}) \|_1, \quad \ldots (6)$$

where $\mathbf{g}^+(\mathbf{x})$ is the inequality constraint violation and μ_k is a penalty parameter. In addition, a surrogate penalty function, $\hat{\Upsilon}(\mathbf{x},\mu_k)$, which replaces the high-fidelity function value with the surrogate model value, is needed for trust region updating,

$$\hat{\Upsilon}(\mathbf{x},\mu_k) = m_k(\mathbf{x}) + \mu_k \| \mathbf{h}(\mathbf{x}) \|_1 + \mu_k \| \mathbf{g}^+(\mathbf{x}) \|_1. \quad \ldots (7)$$

The penalty parameter, μ_k, must be larger than the smallest Lagrange multiplier associated with Equation (1)[4].

From an initial design vector $\mathbf{x}_0$, trust-region size Δ_0, and penalty μ_0, the trust-region iterates are generated by finding a step, $\boldsymbol{s}_k$, that minimises the surrogate model subject to linearised design constraints and trust-region constraint,

$$\begin{aligned} &\min_{\mathbf{s}_k\in\mathbb{R}^n} m_k(\mathbf{x}_k + \mathbf{s}_k) \\ &\text{s.t. } \nabla\mathbf{h}(\mathbf{x}_k)^T\mathbf{s}_k + \mathbf{h}(\mathbf{x}_k) = 0 \\ &\quad \nabla\mathbf{g}(\mathbf{x}_k)^T\mathbf{s}_k + \mathbf{g}(\mathbf{x}_k) \le 0 \\ &\quad \| \mathbf{s}_k \|_\infty \le \Delta_k \end{aligned} \quad \ldots (8)$$

or subject to the full design constraints and trust-region constraint,

$$\begin{aligned} &\min_{\mathbf{s}_k\in\mathbb{R}^n} m_k(\mathbf{x}_k + \mathbf{s}_k) \\ &\text{s.t. } \mathbf{h}(\mathbf{x}_k + \mathbf{s}_k) = 0 \\ &\quad \mathbf{g}(\mathbf{x}_k + \mathbf{s}_k) \le 0 \\ &\quad \| \mathbf{s}_k \|_\infty \le \Delta_k \end{aligned} \quad \ldots (9)$$

Equation (8) is the general subproblem which allows for the current iterate to be infeasible and for first-order consistent surrogate models of the constraints to be used. However, if the constraints are inexpensive compared to the high-fidelity objective function and the current iterate is feasible, Equation (9) can speed finding the optimal high-fidelity design since the constraints are included explicitly in the trust region subproblem. After either subproblem has been solved, the performance of the surrogate model is estimated with the parameter ρ_k, which is the ratio of the actual improvement in the high-fidelity penalty function with the improvement estimated by the surrogate penalty function,

$$\rho_k = \frac{\Upsilon(\mathbf{x}_k,\mu_k) - \Upsilon(\mathbf{x}_k + \mathbf{s}_k,\mu_k)}{\Upsilon(\mathbf{x}_k,\mu_k) - \hat{\Upsilon}(\mathbf{x}_k + \mathbf{s}_k,\mu_k)}. \quad \ldots (10)$$

The size of the trust region is updated based on the performance of the surrogate model. If the surrogate model predicted the high-fidelity behaviour well, the trust region is expanded, if the prediction is poor the trust region is contracted. Specifically, we update the trust region size according to,

$$\Delta_{k+1} = \begin{cases} \min\{\gamma_1\Delta_k, \Delta_{\max}\} & \text{if } \rho_k \ge c_1 \\ \gamma_0 \| \mathbf{s}_k \| & \text{if } \rho_k \le c_2 \\ \Delta_k & \text{otherwise,} \end{cases} \quad \ldots (11)$$

where $0 < c_2 < c_1 < 1$, $\gamma_0 < 1$ and $\gamma_1 > 1$. In addition we move the trust region if the step results in a decrease in the penalty function,

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_k + \mathbf{s}_k & \text{if } \Upsilon(\mathbf{x}_k + \mathbf{s}_k,\mu_k) < \Upsilon(\mathbf{x}_k,\mu_k) \\ \mathbf{x}_k & \text{otherwise.} \end{cases} \quad \ldots (12)$$

We then create a new surrogate model $m_{k+1}(\boldsymbol{x})$ on the new trust region $\{\mathbf{x}: \| \mathbf{x} - \mathbf{x}_{k+1} \|_\infty \le \Delta_{k+1}\}$. If the optimality conditions are not satisfied we repeat the algorithm. In addition, if the size of the trust region becomes sufficiently small we also terminate the algorithm. The trust-region algorithm is summarised as Algorithm 1.

https://doi.org/10.1017/S0001924000006473 Published online by Cambridge University Press

ALGORITHM 1: ITERATION *K* OF A TRUST-REGION ALGORITHM

0. Choose initial design vector $\mathbf{x}_0$, initial trust region size Δ_0, initial penalty μ_0 and initial surrogate model $m_0(\mathbf{x})$ that satisfies Equations (4) and (5) at x_0 (Algorithm 2 can be used to create $m_0(\mathbf{x})$).
1. Solve the trust-region subproblem using nonlinear programming techniques to find the step, $\boldsymbol{s}_k$, that solves,

$$\min_{s_k \in \mathbb{R}^n} m_k(\mathbf{x}_k + \mathbf{s}_k)$$
$$\text{s.t. } \nabla \mathbf{h}(\mathbf{x}_k)^T \boldsymbol{s}_k + \mathbf{h}(\mathbf{x}_k) = 0$$
$$\nabla \mathbf{g}(\mathbf{x}_k)^T \boldsymbol{s}_k + \mathbf{g}(\mathbf{x}_k) \le 0$$
$$\left\| \boldsymbol{s}_k \right\|_\infty \le \Delta_k$$

or

$$\min_{s_k \in \mathbb{R}^n} m_k(\mathbf{x}_k + \mathbf{s}_k)$$
$$\text{s.t. } \mathbf{h}(\mathbf{x}_k + \boldsymbol{s}_k) = 0$$
$$\mathbf{g}(\mathbf{x}_k + \boldsymbol{s}_k) \le 0$$
$$\left\| \boldsymbol{s}_k \right\|_\infty \le \Delta_k.$$

2. Evaluate penalty functions $\Upsilon(\mathbf{x}_k + \mathbf{s}_k, \mu_k)$ and $\hat{\Upsilon}(\mathbf{x}_k + \mathbf{s}_k, \mu_k)$.
3. Compute the ratio of actual improvement to predicted improvement,

$$\rho_k = \frac{\Upsilon(\mathbf{x}_k, \mu_k) - \Upsilon(\mathbf{x}_k + \mathbf{s}_k, \mu_k)}{\Upsilon(\mathbf{x}_k, \mu_k) - \hat{\Upsilon}(\mathbf{x}_k + \mathbf{s}_k, \mu_k)}.$$

4. Update the trust region size according to ρ_k,

$$\Delta_{k+1} = \begin{cases} \min\{\gamma_1 \Delta_k, \Delta_{\max}\} & \text{if } \rho_k \ge c_1 \\ \gamma_0 \| \mathbf{s}_k \| & \text{if } \rho_k \le c_2 \\ \Delta_k & \text{otherwise,} \end{cases}$$

5. Accept or reject the trial point,

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_k + \mathbf{s}_k & \text{if } \Upsilon(\mathbf{x}_k + \mathbf{s}_k, \mu_k) < \Upsilon(\mathbf{x}_k, \mu_k) \\ \mathbf{x}_k & \text{otherwise.} \end{cases}$$

6. Create a new surrogate model $m_{k+1}(\boldsymbol{x})$ that satisfies Equations (4) and (5) using Algorithm 2 on the trust-region $\{\mathbf{x}: \left\| \mathbf{x} - \mathbf{x}_{k+1} \right\|_\infty \le \Delta_{k+1}\}$.
7. Check for convergence. If the solution is not optimal and the trust-region size is sufficient, return to step 1.

ALGORITHM 2: PROCEDURE TO GENERATE A COKRIGING MODEL

0. Select initial calibration point at $\mathbf{x}_k$ and initial correlation vector ξ.
1. Randomly order all high-fidelity sample points within $\left\| \mathbf{x} - \mathbf{x}_k \right\| \le \theta_1 \Delta_k$.
2. Maximise the likelihood function of the Cokriging model using an optimisation algorithm that is robust to non-smoothness, such as a pattern-search or simulated annealing.
   (a) For a given ξ, test each high-fidelity sample point in the order chosen in step 1 to see if after adding that point to the Cokriging model the eigenvalues of $\mathbf{R}_c$ with smallest absolute value and the largest absolute value satisfy, $|\lambda_1| \ge \theta_2$ and $|\lambda_{q(n+1)}| \le \theta_3 |\lambda_1|$. If the conditions are satisfied, add that calibration point (up to user set maximum, q_{max} calibration points), if the conditions are not satisfied reject that calibration point.
   (b) Compute the likelihood of the Cokriging model, Θ, using Equation (21).
3. If more than one calibration point is in the Cokriging model, compute $\mathbf{R}_c^{-1}$ and β_c using the maximum likelihood correlation vector, ξ. Otherwise use the additive correction given in Equation (13).

2.2 Bayesian model calibration

In conventional trust-region algorithms the surrogate model for optimisation is created by correcting the low-fidelity model such that at the center of the trust-region the first-order consistency requirements, Equations (4) and (5), are satisfied. Using the additive correction in this paper, Equation (3), this would correspond to;

$$e_k(\mathbf{x}) = f_{\text{high}}(\mathbf{x}_k) - f_{\text{low}}(\mathbf{x}_k) + [\nabla f_{\text{high}}(\mathbf{x}_k) - \nabla f_{\text{low}}(\mathbf{x}_k)]^T (\mathbf{x} - \mathbf{x}_k) \quad \ldots (13)$$

In the trust-region framework presented in Section 2.1, this calibration approach is provably convergent to a high-fidelity optimum. There are two possible drawbacks with this calibration technique. The first drawback is that the calibration may only be first-order, which means in the worst case the convergence rate could be linear and extremely slow, although we note that quasi-second-order trust region calibration approaches have been proposed in the literature[10]. The second drawback is that both the function value and gradient are known at all previous design iterates encountered by the trust region algorithm and this information is typically not used.

The idea of Bayesian model calibration is to use all available information to estimate unknown high-fidelity information. In the multifidelity setting, this translates to using all previous information about the error between the high- and low-fidelity models to calibrate the low-fidelity model. The calibrated model provides an estimate of the new design that has the largest high-fidelity

https://doi.org/10.1017/S0001924000006473 Published online by Cambridge University Press

improvement. However, the amount of high-fidelity information collected during an optimisation is likely too much to use when creating a surrogate model that will only be used for one trust-region iteration. Therefore, the next section addresses how high-fidelity information is selected for a calibration procedure that guarantees convergence to a high-fidelity optimum and limits the computational effort required to construct a surrogate model.

2.3 Cokriging

The Cokriging Bayesian model calibration technique is demonstrated in Figure 1. The figure shows that at all calibration points the surrogate model exactly interpolates the high-fidelity function and it has the same gradient as the high-fidelity function. The Cokriging model also estimates the uncertainty in its prediction using a maximum likelihood variance estimate. It can be observed that the uncertainty in the Cokriging estimates are zero at all calibration points and increase with distance away from the calibration points. There are two Cokriging formulations, direct[11,14], and indirect[21,25]. The former augments a Kriging model with analytical gradients and the latter adds additional sample points to a Kriging fit such that the Cokriging model interpolates points replicating a Taylor series. This paper uses the direct Cokriging formulation. We first summarise the Cokriging method, then we present the requirements for convergence in a trust-region framework, and we finally present our method to construct Cokriging models that satisfies both the convergence requirements of a trust-region algorithm and limits the computational effort required to construct the surrogate models.

Kriging methods come from the field of Geostatistics and have the underlying assumption that function values from nearby samples are correlated[27]. The correlation function we use is an anisotropic Gaussian,

$$\phi(\mathbf{x}_i,\mathbf{x}_j)=\exp\left[-\sum_{p=1}^{n}\xi^{(p)}\left(x_i^{(p)}-x_j^{(p)}\right)^2\right], \quad \ldots(14)$$

where the correlation between points $\mathbf{x}_i$ and $\mathbf{x}_j$, $\phi(\mathbf{x}_i, \mathbf{x}_j)$, has n spatial tuning parameters, $\xi^{(p)}$ for each design variable $x^{(p)}$. The tuning parameters are estimated using a Bayesian maximum likelihood estimator.

A Cokriging model uses q calibration points in $\mathbb{R}^n$ to generate a correlation vector with length $q(1 + n)$,

$$\mathbf{\Phi}(\mathbf{x})=\left[\phi(\mathbf{x},\mathbf{x}_1),\ldots,\phi(\mathbf{x},\mathbf{x}_q),\nabla\phi(\mathbf{x},\mathbf{x}_1)^T,\ldots,\nabla\phi(\mathbf{x},\mathbf{x}_q)^T\right]^T, \quad \ldots(15)$$

where all derivatives of $\phi(\mathbf{x},\mathbf{x}_j)$ are with respect to $\mathbf{x}$. In addition, we need to define the Cokriging correlation matrix that describes the influence of each calibration point on the other calibration points,

$$\mathbf{R}_c=\begin{array}{c}\overbrace{\phantom{\phi(\mathbf{x}_1,\mathbf{x}_1)\ldots\phi(\mathbf{x}_1,\mathbf{x}_q)}}^{q}\quad\overbrace{\phantom{\nabla\phi(\mathbf{x}_1,\mathbf{x}_1)^T\ldots\nabla\phi(\mathbf{x}_1,\mathbf{x}_q)^T}}^{n\times q}\\ \left[\begin{array}{c|c}\phi(\mathbf{x}_1,\mathbf{x}_1)\ldots\phi(\mathbf{x}_1,\mathbf{x}_q) & \nabla\phi(\mathbf{x}_1,\mathbf{x}_1)^T\ldots\nabla\phi(\mathbf{x}_1,\mathbf{x}_q)^T\\ \vdots\quad\ddots\quad\vdots & \vdots\quad\ddots\quad\vdots\\ \phi(\mathbf{x}_q,\mathbf{x}_1)\ldots\phi(\mathbf{x}_q,\mathbf{x}_q) & \nabla\phi(\mathbf{x}_q,\mathbf{x}_1)^T\ldots\nabla\phi(\mathbf{x}_q,\mathbf{x}_q)^T\\ \hline \nabla\phi(\mathbf{x}_1,\mathbf{x}_1)\ldots\nabla\phi(\mathbf{x}_1,\mathbf{x}_q) & \nabla^2\phi(\mathbf{x}_1,\mathbf{x}_1)\ldots\nabla^2\phi(\mathbf{x}_1,\mathbf{x}_q)\\ \vdots\quad\ddots\quad\vdots & \vdots\quad\ddots\quad\vdots\\ \nabla\phi(\mathbf{x}_q,\mathbf{x}_1)\ldots\nabla\phi(\mathbf{x}_q,\mathbf{x}_q) & \nabla^2\phi(\mathbf{x}_q,\mathbf{x}_1)\ldots\nabla^2\phi(\mathbf{x}_q,\mathbf{x}_q)\end{array}\right]\begin{array}{l}\}q\\ \}n\times q\end{array}\end{array}, \quad \ldots(16)$$

where all first and second derivatives of $\phi(\cdot,\cdot)$ are with respect to the first variable. Using the correlation vector and correlation matrix we can interpolate the values in the vector,

$$y_c=[f_{\text{high}}(\mathbf{x}_1)-f_{\text{low}}(\mathbf{x}_1),\ldots,f_{\text{high}}(\mathbf{x}_q)-f_{\text{low}}(\mathbf{x}_q),\\ \nabla f_{\text{high}}(\mathbf{x}_1)-\nabla f_{\text{low}}(\mathbf{x}_1),\ldots,\nabla f_{\text{high}}(\mathbf{x}_q)-\nabla f_{\text{low}}(\mathbf{x}_q)]^T, \quad \ldots(17)$$

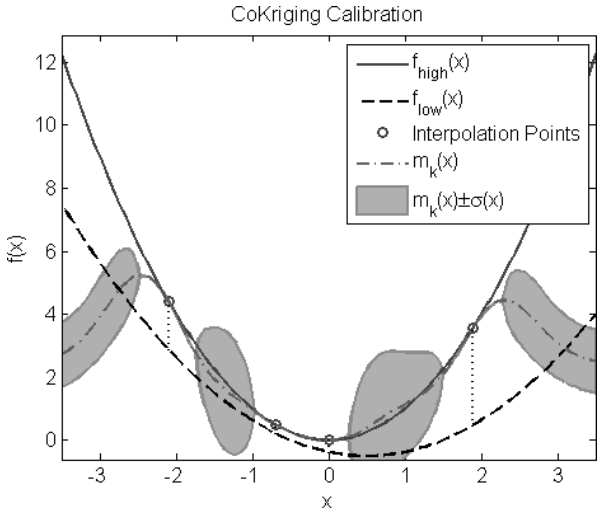


Figure 1. Demonstration of a Cokriging surrogate model for the simple univariate function $f_{high}(x) = x^2$.

by defining the vector of ones and zeros,

$$\mathbf{z}_c=[\mathbf{1}^q,\mathbf{0}^{q,n}]^T \quad \ldots(18)$$

and a generalised least-squares constant term,

$$\beta_c=(\mathbf{z}_c^T\mathbf{R}_c^{-1}\mathbf{z}_c)^{-1}\mathbf{z}_c^T\mathbf{R}_c^{-1}\mathbf{y}_c. \quad \ldots(19)$$

This leads to a final Cokriging error model that has the form,

$$e_k(\mathbf{x})=\beta_c+\Phi(\mathbf{x})^T\mathbf{R}_c^{-1}(\mathbf{y}_c-\mathbf{z}_c\beta_c). \quad \ldots(20)$$

To estimate the spatial correlation parameters ξ we maximise the likelihood function, Θ, which is the probability of observing these data if these data had been generated by a Gaussian process[14,30]. The likelihood is only a function of the observed data and the correlation parameters ξ. Ignoring the contribution of the gradient to the likelihood, the partial likelihood only depends on the Kriging portion of the Cokriging model, so $\mathbf{R}_c^{(1:q\times1:q)}$ indicates that only the first q terms of the Cokriging vectors are used,

$$\Theta=-\frac{q\ln\hat{\sigma}^2+\ln\left|\mathbf{R}_c^{(1:q\times1:q)}\right|}{2}. \quad \ldots(21)$$

Where $\hat{\sigma}^2$ maximises the Kriging portion of the likelihood function, Equation (21)[32].

$$\hat{\sigma}^2=\frac{\left(\mathbf{y}_c^{(1:q)}-\mathbf{1}^q\beta\right)^T\left(\mathbf{R}_c^{(1:q\times1:q)}\right)^{-1}\left(\mathbf{y}_c^{(1:q)}-\mathbf{1}^q\beta\right)}{q}, \quad \ldots(22)$$

and

$$\beta=\left([\mathbf{1}^q]^T\left(\mathbf{R}_c^{(1:q\times1:q)}\right)^{-1}\mathbf{1}^q\right)^{-1}[\mathbf{1}^q]^T\left(\mathbf{R}_c^{(1:q\times1:q)}\right)^{-1}\mathbf{y}_c^{(1:q)}. \quad \ldots(23)$$

To use a Cokriging model within a trust-region algorithm we must ensure the first-order consistency requirements, Equations (4) and (5), are satisfied, and that the norm of the Cokriging Hessian is bounded. The first-order consistency requirements are satisfied provided the current trust-region iterate, $\mathbf{x}_k$, is a calibration point and that $\mathbf{R}_c$ is not ill-conditioned. To satisfy the requirement that the Hessian of the surrogate model has bounded norm, we first assume that both the high- and low-fidelity functions have bounded Hessian norms for all $\mathbf{x}$. What must follow is that the Hessian of $e_k(\mathbf{x})$ has bounded norm, or equivalently the Hessian of $\Phi(\mathbf{x})^T\mathbf{R}_c^{-1}(\mathbf{y}_c - \mathbf{z}_c\beta_c)$ has bounded norm. The vector $(\mathbf{y}_c - \mathbf{z}_c\beta_c)$ has bounded norm by virtue of assumptions on the high- and low-fidelity function. Furthermore, it

https://doi.org/10.1017/S0001924000006473 Published online by Cambridge University Press

can be shown by differentiating the correlation function in Equation (14) four times that the maximum absolute value of any second derivative term for the first q components of $\Phi(\mathbf{x})^T$ is less than $\max_p 2\xi^{(p)}$, and the maximum absolute value of any second derivative term for the remaining qn components is less than $\max_p 3{\cdot}904(\xi^{(p)})^{3/2}$. Therefore, using properties of the 2-norm and ∞-norm, we can establish that the 2-norm of the Hessian of $\Phi(\mathbf{x})^T\mathbf{1}$ is bounded by $[q(1 + n)]^{3/2} \max\{\max_p 2\xi^{(p)}, \max_p 3{\cdot}904(\xi^{(p)})^{3/2}\}$. Now, provided $\left\|\mathbf{R}_c^{-1}\right\|_2$ is bounded, we have ensured that the 2-norm of the Cokriging model Hessian is bounded. We address both this criterion and the conditioning of $\mathbf{R}_c$ during the construction of the Cokriging model.

To construct the Cokriging model, we want to limit the number of calibration points so the size of $\mathbf{R}_c$ remains tractable to invert repeatedly. Therefore, we set a maximum number of calibration points that we may use, $q_{\max}$. In addition, because the trust-region algorithm only requires accuracy within the trust-region we only want to include points in the calibration that will affect the shape of the surrogate model within the trust region. So we allow a user-set distance parameter, θ_1, that controls the extent to which the Cokriging model calibrates locally as opposed to globally. Specifically, any point at which the high-fidelity function value and gradient are known that is located within $\left\|\mathbf{x} - \mathbf{x}_k\right\| \le \theta_1\Delta_k$, $\theta_1 > 0$ is a candidate calibration point. The first-step in constructing the Cokriging model is to select an initial vector of correlation parameters ξ_0 and to select $\mathbf{x}_k$ as the initial calibration point. We then randomly order all candidate calibration points.

To find the Cokriging model with the maximum likelihood correlation parameters and that satisfies all trust-region requirements, we use a greedy approach. We add any candidate calibration point to the Cokriging model provided that we remain able to bound the condition number of $\mathbf{R}_c$ and to bound $\left\|\mathbf{R}_c^{-1}\right\|_2$. These two criteria are satisfied if the eigenvalue of $\mathbf{R}_c$ with the smallest absolute value, $|\lambda_1|$, is greater than a constant, $\theta_2 > 0$, and the eigenvalue with the greatest absolute value, $\lambda_{q(n+1)}$, is less than or equal to $\theta_3|\lambda_1|$, for a constant $\theta_3 > 1$. This means the parameter θ_3 represents the maximum allowable condition number for the correlation matrix. The upper limit for θ_3 depends on the numerical precision and inversion algorithm used. After all the candidate points have been tested or $q_{\max}$ points have been selected, the likelihood of the Cokriging model, Θ is computed. A pattern-search algorithm is used to find the correlation vector, ξ, giving the Cokriging model with the maximum likelihood. An optimisation algorithm that is not highly sensitive to non-smooth functions must be used because for each correlation vector considered the calibration points used may vary. This means that the likelihood function will likely have non-smooth features. A summary of the procedure to generate Cokriging models is presented as Algorithm 2. Table 1 lists the parameter values used for the sample problems presented in the next two sections.

Table 1
Values of the optimisation parameters used

Parameter	Description	value
q_{max}	Maximum calibration points	10
c_1	Trust-region expansion criteria	0·75
c_2	Trust-region contraction criteria	0·25
Δ_0	Initial trust region size	$\max\{5, \left\|\mathbf{x}_0\right\|_\infty\}$
$\Delta_{\max}$	Maximum trust region size	20
$\Delta_{\min}$	Minimum trust region size	10^{-6}
γ_0	Trust region contraction ratio	0·5
γ_1	Trust region expansion ratio	2
ε	Termination tolerances	1×10^{-4}
θ_1	Trust-region neighborhood size	10^3
θ_2	Min eigenvalue	10^{-9}
θ_3	Max condition number	10^8
	Max pattern-search iterations	$10n$

3.0 STRUCTURAL DESIGN PROBLEM

Typical structural design problems have deflection or stress requirements with low-weight as an objective or additional constraint. As an example problem we minimise the deflection of a two-dimensional hook subjected to a bearing load and maximum weight constraint. This objective function is linear in the state variables, however, a nonlinear objective will not change the derivation. In the finite element formulation used, the state variables are nodal displacements, $\mathbf{u}$, and our objective function can be written as;

$$\mathcal{J}_{\text{high}}(\mathbf{x}, \mathbf{u}) = \mathbf{C}(\mathbf{x})\mathbf{u}, \qquad \ldots (24)$$

where $\mathbf{C}(\mathbf{x})$ is an output matrix that only depends on the 26 design variables representing the geometry of the hook, $\mathbf{x}$, and not the state variables. In lieu of computing displacements the output matrix could easily be modified to compute element stresses using Hooke's Law. The equations of state are obtained using a Ritz finite element formulation to minimise the potential energy of the system, and in discretised form are

$$\mathbf{r}_{\text{high}}(\mathbf{x},\mathbf{u}) = \mathbf{K}(\mathbf{x})\mathbf{u} - \mathbf{f} = 0 \qquad \ldots (25)$$

where $\mathbf{K}(\mathbf{x})$ is the stiffness matrix and $\mathbf{f}$ is the vector of applied nodal forces. The formal structural optimisation problem to minimise the deflection of the hook subjected to a bearing load is,

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbf{R}^{26},\,\mathbf{u}\in\mathbf{R}^{l}} \quad \mathbf{C}(\mathbf{x})\mathbf{u} \\ &\text{s.t. } \mathbf{K}(\mathbf{x})\mathbf{u} - \mathbf{f} = 0 \\ &w(\mathbf{x}) - w_{\max} \le 0 \\ &\boldsymbol{g}(\mathbf{x}) \le 0 \end{aligned} \qquad \ldots (26)$$

where $w(\mathbf{x})$ is the weight of the structure, $w_{\max}$ is the maximum allowable weight, and $g(\mathbf{x}) \le 0$ represents 31 constraints ensuring geometric feasibility of the hook – positive thickness and clearance to apply the bearing load.

To use our multifidelity optimisation method, we combine our objective function $\mathcal{J}_{\text{high}}(\mathbf{x},\mathbf{u})$ and state equations $\mathbf{r}_{\text{high}}(\mathbf{x},\mathbf{u}) = 0$ into a single function of the design variables $f_{\text{high}}(\mathbf{x})$ that is the value of our objective function after the state equations have been satisfied. To compute the gradient of $f_{\text{high}}(\mathbf{x})$, we solve the adjoint equation,

$$\frac{\partial \mathcal{J}_{high}}{\partial \mathbf{u}} = \Psi^T \frac{\partial \mathbf{r}_{high}}{\partial \mathbf{u}}, \qquad \ldots (27)$$

for the adjoint variables, Ψ. For this structural optimisation problem the adjoint equation is;

$$\mathbf{K}^T(\mathbf{x})\,\Psi(\mathbf{x}) = \boldsymbol{C}^T(\mathbf{x}). \qquad \ldots (28)$$

The gradient of our objective function with respect to the design variables can be written as,

$$\frac{d\mathcal{J}_{high}}{d\mathbf{x}} = \frac{\partial \mathcal{J}_{high}}{\partial \mathbf{x}} + \frac{\partial \mathcal{J}_{high}}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}} \qquad \ldots (29)$$

and similarly, the gradient of the equations of state with respect to

https://doi.org/10.1017/S0001924000006473 Published online by Cambridge University Press

the design variables can be written,

$$\frac{\mathrm{d}\mathbf{r}_{high}}{\mathrm{d}\mathbf{x}} = \frac{\partial \mathbf{r}_{high}}{\partial \mathbf{x}} + \frac{\partial \mathbf{r}_{high}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{x}} = 0. \quad \ldots (30)$$

Substituting Equations (27) and (30) into Equation (29), we obtain the gradient of our objective function with the state equations satisfied as;

$$\frac{\mathrm{d}f_{high}}{\mathrm{d}\mathbf{x}} = \frac{\partial \mathcal{J}_{high}}{\partial \mathbf{x}} - \Psi^T \frac{\partial \mathbf{r}_{high}}{\partial \mathbf{x}}, \quad \ldots (31)$$

or for this structural optimisation problem,

$$\frac{\mathrm{d}f_{high}}{\mathrm{d}\mathbf{x}} = \left(\frac{\partial \mathbf{C}(\mathbf{x})}{\partial \mathbf{x}} - \Psi^T(\mathbf{x}) \frac{\partial \mathbf{K}(\mathbf{x})}{\partial \mathbf{x}} \right) \mathbf{u}(\mathbf{x}). \quad \ldots (32)$$

The objective function being linear in $\mathbf{u}(\mathbf{x})$ is a simplification, and a typical problem will likely be to minimise the maximum stress in the material. Although maximum stress is a non-smooth objective, Kreisselmeier-Steinhauser functions could be used to lump element stresses into a single smooth maximum stress objective[26,29], or a high-norm, such as $\|\cdot\|_8$, could be used to smoothly approximate the maximum stress[35]. Using these functions as objectives does not alter the derivation of the gradient, provided the chain rule is used in computing the derivatives. In the general case, this adjoint-based gradient calculation requires only two forward solves, or one inversion, of the stiffness matrix to compute the gradient of the objective, regardless of the number of design variables. By comparison, the method of direct sensitivities requires evaluation of the term $\frac{\partial \mathbf{K}^{-1}(\mathbf{x})}{\partial \mathbf{x}}$, and finite difference approximations require at least $n + 1$,where n is the number of design variables, function evaluations.

To find the optimal hook design, we use both high- and low-fidelity finite element models of the hook. The two models are shown in Fig. 2, the high-fidelity model has a state vector with 1,770 degrees of freedom (dof) and the low-fidelity model has a state vector with 276 dof.

For the optimal hook design, the coarse discretisation of the low-fidelity model leads to a 31·6% lower deflection estimate than the high-fidelity model. In addition, for this hook design, Fig. 3 shows that the two models predict significantly different stress distributions.

Table 2 shows that using this coarsely discretised low-fidelity model, the conventional first-order consistent trust-region and our Bayesian calibration approach are able to significantly reduce the number of high-fidelity function calls compared with a single-fidelity SQP method. In addition, the Bayesian model calibration approach reduces the number of high-fidelity function calls by 31% compared with the conventional first-order consistent trust region method. We note that when the geometry of the hook is physically infeasible, it is possible that the stiffness matrix is nearly singular and in these cases the gradient computed using the adjoint approach is inaccurate. Accordingly, our strategy is to not use the gradient for calibration at previously visited designs sites where the geometry was infeasible.

Table 2
The average number of high-fidelity function evaluations to minimise the deflection of a hook subjected to a bearing load from random initial geometries. The numbers in parentheses indicate the percentage reduction in high-fidelity function evaluations relative to SQP

	SQP	First-Order TR	Calibration
Mean	232 (–)	40 (–83%)	27·5 (–88%)
Std Dev	171	12·7	10·2

To analyse the total computational effort required to solve this structural design problem using a multifidelity method in lieu of a single-fidelity scheme, we consider the total effort of the optimisation. As a reference, a general purpose structural finite element solver, Nastran, uses a conjugate gradient method with an incomplete Cholesky factorisation preconditioner to solve the linear finite-element system. So the effort for a forward solve scales with $\mathcal{O}(\text{dof}^2)$[1]. Using this scaling, the forward solution for the high-fidelity structural model is about 41 times as costly as the forward solution for the low-fidelity structural model. In addition, a conservative estimate on the number of low-fidelity function calls used to find the optimal high-fidelity hook design is approximately 3,200. The computational effort for these low-fidelity solves is equivalent to about 80 high-fidelity forward solves. Therefore the total effort of the multifidelity approach is equivalent to approximately 110 high-fidelity forward solves, and for comparison the single-fidelity SQP optimisation required 232 high-fidelity evaluations. Therefore, excluding the cost of constructing the surrogate models the multifidelity approaches correspond to about a 50% decrease in computational effort compared with a single-fidelity approach.

4.0 AERODYNAMIC DESIGN PROBLEM

Aerodynamic design is a computationally expensive process because high-fidelity computational fluid dynamics (CFD) must repeatedly solve nonlinear governing equations for a large number of degrees of freedom. This section presents an adjoint-based formulation for an

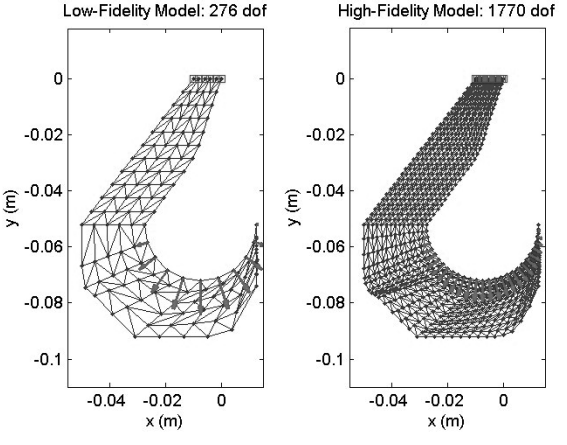


Figure 2. Comparison of the high- and low-fidelity structural models. Both models have the same 26 design variables.

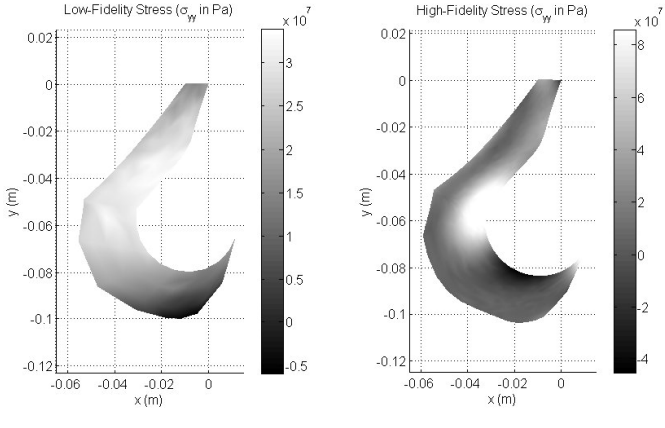


Figure 3. Comparison of the stress estimated by the high- and low-fidelity structural models shown on the deformed hook, the deformation is scaled by a factor of 123·5. The deflection at the midpoint of the bearing load application estimated by the two models differs by 31·6%.

https://doi.org/10.1017/S0001924000006473 Published online by Cambridge University Press

aerodynamic design problem described by a CFD model and then presents results of a multifidelity supersonic aerofoil design problem.

To minimise the drag of a supersonic aerofoil we formulate an optimisation problem in the form given in Equation (2) $\mathcal{J}_{high}(\mathbf{x},\mathbf{u})$ is the surface integral of pressure acting in the flow direction, $\mathbf{r}_{high}(\mathbf{x},\mathbf{u}) = 0$ represents the discretised Euler equations, and $\mathbf{g}(\mathbf{x}) \le 0$ comprises two constraints on the aerofoil geometry. The state variables $\mathbf{u}$ are the primal flow variables for all of the control volumes in a finite volume discretisation of the governing equations. In order to use the multifidelity optimisation technique presented, we need to compute the gradient of $\mathcal{J}_{high}(\mathbf{x},\mathbf{u})$ with respect to the design variables $\mathbf{x}$ given that $\mathbf{r}_{high}(\mathbf{x},\mathbf{u})$. We start by writing the gradient of $\mathcal{J}_{high}(\mathbf{x},\mathbf{u})$,

$$\frac{\mathrm{d}\mathcal{J}_{high}}{\mathrm{d}\mathbf{x}} = \frac{\partial \mathcal{J}_{high}}{\partial \mathbf{x}} + \frac{\partial \mathcal{J}_{high}}{\partial \mathbf{M}}\frac{\partial \mathbf{M}}{\partial \mathbf{x}} + \frac{\partial \mathcal{J}_{high}}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \qquad \ldots (33)$$

where $\mathbf{M}(\mathbf{x})$ represents the dependence of each nodal vertex in the volume mesh on the design vector. For this analysis, a meshing tool was developed that created a volume mesh around the aerofoil with an analytical derivative. Accordingly, for any aerofoil that could be generated with the parameterisation used, the volume mesh and mesh derivative were known *a priori*.

Since the flow residual must always be zero for a converged solution we know that,

$$\frac{\mathrm{d}\mathbf{r}_{high}}{\mathrm{d}\mathbf{x}} = \frac{\partial \mathbf{r}_{high}}{\partial \mathbf{x}} + \frac{\partial \mathbf{r}_{high}}{\partial \mathbf{M}}\frac{\partial \mathbf{M}}{\partial \mathbf{x}} + \frac{\partial \mathbf{r}_{high}}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}} = 0. \qquad \ldots (34)$$

Therefore, combing Equations (33) and (34) with the adjoint equation, Equation (27), we may write the gradient of an objective function with the state equations satisfied as

$$\frac{\mathrm{d}f_{high}}{\mathrm{d}\mathbf{x}} = \frac{\partial \mathcal{J}_{high}}{\partial \mathbf{x}} + \frac{\partial \mathcal{J}_{high}}{\partial \mathbf{M}}\frac{\partial \mathbf{M}}{\partial \mathbf{x}} - \Psi^T\left(\frac{\partial \mathbf{r}_{high}}{\partial \mathbf{x}} + \frac{\partial \mathbf{r}_{high}}{\partial \mathbf{M}}\frac{\partial \mathbf{M}}{\partial \mathbf{x}}\right). \qquad \ldots (35)$$

This reformulation shows how to convert the constrained objective function into an objective function from which the state variables have been eliminated through the solution of the residual equations. Accordingly, the gradient $\mathrm{d}f_{high}/\mathrm{d}\boldsymbol{x}$ represents the gradient of drag with respect to the design variables given that the discretised Euler equations are satisfied. For further discussion see Refs 17, 28. The computational effort required to compute this gradient requires one flow solution, one adjoint solution, one flow iteration per design variable (about 1/500 the effort of a flow solution), and the matrix multiplications shown above. Therefore, the gradient of the objective with respect to all of the design variables requires the computational effort of about two flow solutions plus n/500 flow solutions, where is the number of design variables. For comparison, a finite difference gradient estimate requires at least $n + 1$ flow solutions, so this is a considerable savings and shows the cost of the gradient estimate using an adjoint solution is almost independent of the number of design variables.

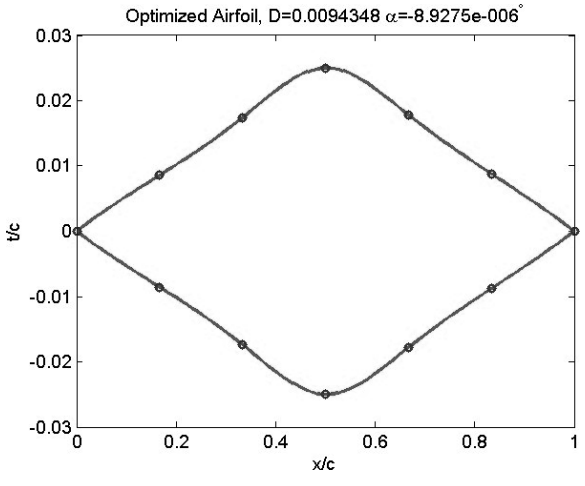


Figure 4. Minimum drag aerofoil computed with the supersonic panel method showing the spline control points.

In addition to the Euler equations as the high-fidelity method, we use a panel method as a low-fidelity analysis. A supersonic panel method can be derived from supersonic small-disturbance theory, and is only a function of the aerofoil geometry, the freestream Mach number, M_∞, and the gas specific heat ratio. In small disturbance theory, the change in the pressure coefficient, δC_p is proportional, to the flow turning angle $\delta\alpha$,

$$\delta C_p = \frac{-2\delta\alpha}{\sqrt{M_\infty^2 - 1}}, \qquad \ldots (36)$$

and by integrating the pressure coefficient around the aerofoil surface the wave drag can be easily estimated[24]. Since the drag coefficient is only a function of the freestream Mach number and aerofoil geometry, the analytical derivative of drag coefficient with respect to the aerofoil shape design parameters is easy to compute.

The aerofoil optimisation problem is to minimise the drag of an aerofoil at $M_\infty = 1{\cdot}5$. by changing the angle of attack, five upper surface spline points, and five lower surface spline points. The aerofoil is required to have positive thickness everywhere, and to have a maximum thickness to chord ratio that is at least five percent. Figure 4 shows the optimal aerofoil and spline control points for the panel method. Figure 5 shows the optimal aerofoil pressure contours and adjoint solution for the streamwise momentum from the Euler method solutions.

Table 3 presents the number of high-fidelity function calls to find the minimum drag aerofoil with respect to the Euler code. The results show that the conventional first-order consistent trust-region and the Bayesian model calibration approach reduce the number of high-fidelity function calls by nearly the same amount, about 80%. In addition, because the low-fidelity model is computationally very inexpensive, and the dimension of the parameter space is small, this is nearly a 70% reduction in wall-clock time. However, it should be noted that the conventional trust-region approach does use on average fewer high-fidelity function calls than the Bayesian calibration method for the ten random initial aerofoils.

Table 3
The average number of high-fidelity function evaluations to minimise the drag of a supersonic aerofoil with respect to an Euler solution using a panel method as a low-fidelity estimate. The numbers in parentheses indicate the percentage reduction in high-fidelity function evaluations relative to SQP

	SQP	First-Order TR	Calibration
Mean	81·5 (-)	12·9 (–84%)	14·7 (–82%)
Std Dev	14·0	2·86	4·19

The same optimisation problem is solved using the same methods, but without any low-fidelity information. The results of this optimisation using $f_{low}(\mathbf{x}) = 0$ as the low-fidelity function are presented in Table 4. In this case the Bayesian model calibration approach performed noticeably better than the conventional trust-region method. The results suggest that when using a 'good' low-fidelity model, the Bayesian calibration approach is not necessary and the computational effort of constructing the Cokriging surrogates is not worthwhile. However, when the low-fidelity function is poor, or when the error between the high- and low-fidelity function behaves in a highly non-linear fashion, then the Bayesian calibration approach may provide a computational savings for low-dimensional

https://doi.org/10.1017/S0001924000006473 Published online by Cambridge University Press

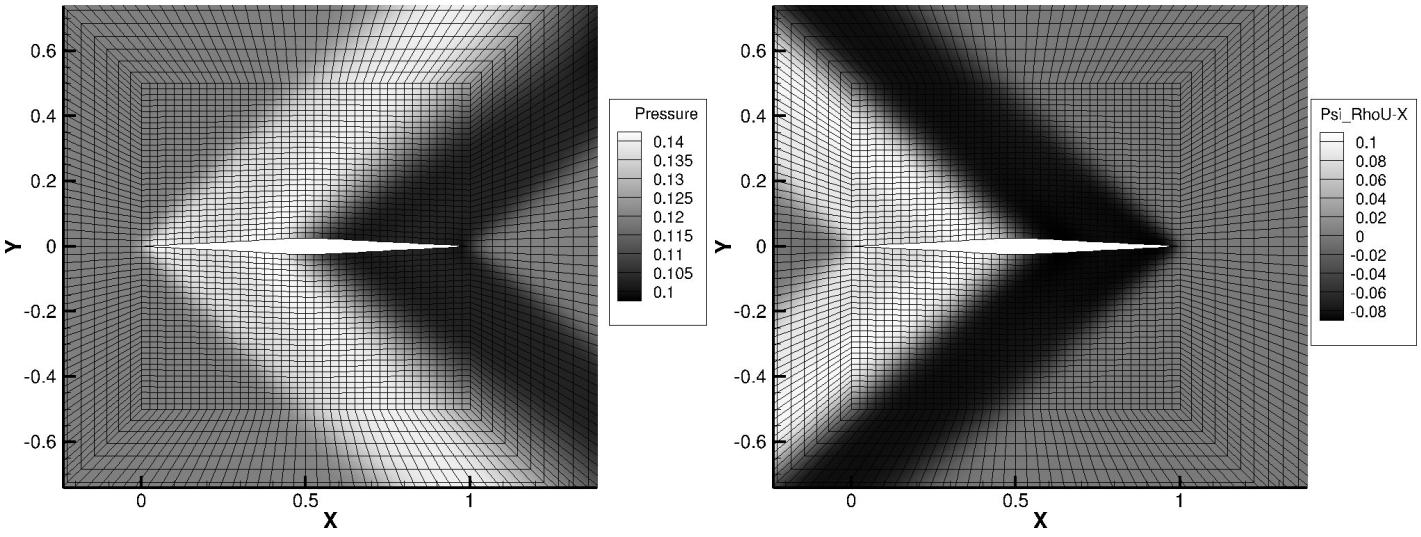


Figure 5. Minimum drag supersonic aerofoil parameterised by five upper surface spline points, five lower surface spline points and angle-of-attack.

optimisation problems. A possible performance issue that affects Bayesian calibration approach significantly more than the conventional trust-region approach is when high-fidelity gradients are computed inaccurately. The calibration approach reuses high-fidelity gradient information, so incorrect gradient information will propagate into the surrogate models generated at future trust-region iterations. In contrast, the conventional approach only uses the high-fidelity gradient to create a single surrogate model so the approach is likely less affected by inaccurate gradient information.

Table 4
The average number of high-fidelity function evaluations to minimise the drag of a supersonic aerofoil with respect to an Euler solution. No low-fidelity information is used. The numbers in parentheses indicate the percentage reduction in high-fidelity function evaluations relative to SQP

	SQP	First-Order TR	Calibration
Mean	81·5(–)	91·1(+12%)	64·2(–21%)
Std Dev	14·0	61·2	21·8

5.0 CONCLUSION

This paper has presented a multifidelity optimisation algorithm using Cokriging-based Bayesian model calibration that is provably convergent to an optimum of the original high-fidelity optimisation problem. The paper demonstrated how to compute gradients inexpensively for both a structural design and an aerofoil shape optimisation problem using adjoint solutions. The derivative estimates and function values from previously visited design sites are used to best calibrate the low-fidelity model to the high-fidelity function. Using this strategy, the optimisation algorithm is likely to quickly find an optimum of the high-fidelity function. The sample results show that for poor low-fidelity models the Bayesian calibration exceeds performance of conventional trust-region algorithms; however, for cases with good low-fidelity models the performance of the two algorithms is similar. However, the multifidelity approaches still significantly outperform single-fidelity sequential quadratic programming methods. The calibration technique developed is recommended for low-dimensional optimisation problems where the quality of the low-fidelity model is unknown or known to be poor in certain portions of the design space.

REFERENCES

1. NX Nastran, Numerical Methods User's Guide. Siemens Product Lifecycle Management Software, 2008.
2. ALEXANDROV, N., DENNIS, J., LEWIS, R. and TORCZON, V. A trust region framework for managing the use of approximation models in optimization. Tech. Rep. CR-201745, NASA, October 1997.
3. ALEXANDROV, N., LEWIS, R., GUMBERT, C., GREEN, L. and NEWMAN, P. Optimization with variable-fidelity models applied to wing design. Tech. Rep. CR-209826, NASA, December 1999.
4. ALEXANDROV, N., LEWIS, R., GUMBERT, C., GREEN, L. and NEWMAN, P. Approximation and model management in aerodynamic optimization with variable-fidelity models, *AIAA J*, November-December, 2001, **38**, (6), pp 1093-1101.
5. ANTOULAS, A. Approximation of Large-Scale Dynamical Systems. SIAM, 2005.
6. BOOKER, A.J., DENNIS, J.E., FRANK, P.D., SERAFINI, D.B., TORCZON, V. and TROSSET, M.W. A rigorous framework for optimization of expensive functions by surrogates, *Structural Optimization*, February 1999, **17**, (1), pp 1-13.
7. CASTRO, J., GRAY, G., GIUNTA, A. and HOUGH, P. Developing a computationally efficient dynamic multilevel hybrid optimization scheme using multi-fidelity model interactions. Tech. Rep SAND2005-7498, Sandia, November 2005.
8. CONN, A., GOULD, N. and TOINT, P. Trust-Region Methods. MPS/SIAM Series on Optimization. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.
9. ELDRED, M. and DUNLAVY, D. Formulations for surrogate-based optimization with data-fit, multifidelity and reduced-order models. 11thAIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Portsmouth, Virginia, USA, 2006. AIAA 2006-7117.
10. ELDRED, M., GIUNTA, A. and COLLIS, S. Second-order corrections for surrogate-based optimization with model hierarchies. 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2004. AIAA 2004-4457.
11. FORRESTER, A., SOBESTER, A. and KEANE, A. Multi-fidelity optimization via surrogate modelling, *J Royal Statistical Society*, September 2007, **463**, (2), pp 3251-3269.
12. GANO, S.E., RENAUD, J.E. and SANDERS, B. Hybrid variable fidelity optimization by using a Kriging-based scaling function, *AIAA J*, November 2005, **43**, (11), 2422-2430.
13. CHUNG, H. and ALONSO, J.J. Design of a low-boom supersonic business jet using cokriging approximation models. In 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, GA, USA, September 2002, AIAA 2002-5598.
14. CHUNG, H. and ALONSO, J.J. Using gradients to construct cokriging approximation models for high-dimensional design optimization problems. In 40th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, USA, January 2002, AIAA 2002-0317.

https://doi.org/10.1017/S0001924000006473 Published online by Cambridge University Press

15. HAFTKA, R.T. and ADELMAN, H.M. Recent developments in structural sensitivity analysis, *Structural Optimization*, 1, 1989, pp 137-151.
16. HAUG, E.J., CHOI, K.K. and KOMKOV, V. *Design Sensitivity Analysis of Structural Systems*, Academic Press Inc, 1986.
17. JAMESON, A. Aerodynamic design via control theory, *J Scientific Computing*, September 1988, **3**, (3), pp 233-260.
18. JONES, D., SCHONLAU, M. and WELCH, W. Efficient global optimization of expensive black-box functions, *J Global Optimization*, 1998, **13**, pp 455-492.
19. KENNEDY, M. and O'HAGAN, A. Predicting the output from a complex computer code when fast approximations are available, *Biometrika*, 2000, **87**, (1), pp 1-13.
20. KENNEDY, M. and O'HAGAN, A. Bayesian calibration of computer models, *J Royal Statistical Society*, **63**, (2), (2001), pp 425-464.
21. LAURENCEAU, J., MEAUX, M., MONTAGNAC, M. and SAGAUT, P. Comparison of gradient-based and gradient-enhanced response-surface-based optimizers, *AIAA J*, May 2010, **48**, (5), pp 981-994.
22. LEARY, S., BHASKAR, A. and KEANE, A. A knowledge-based approach to response surface modelling in multifidelity optimization, *J Global Optimization*, 2003, **26**, pp 297-319.
23. LEARY, S., BHASKAR, A. and KEANE, A. Global approximation and optimization using adjoint computational fluid dynamics codes, *AIAA J*, March 2004, **42**, (3), pp 631-641.
24. LIEPMANN, H.W. and Roshko, A. *Elements of Gasdynamics*, Dover Publications, Mineola, NY, USA, 1993.
25. LIU, W. Development of Gradient-Enhanced Kriging Approximations for Multidisciplinary Design Optimization, PhD thesis, University of Notre Dame, USA, July 2003.
26. MARTINS, J.R.R.A. A Coupled-Adjoint Method for High-Fidelity Aero-Structural Optimization. PhD thesis, Stanford University, USA, October 2002.
27. MATERN, B. *Spatial Variation. Lecture Notes in Statistics*, Springer-Verlag, Berlin, Germany, 1986.
28. NEMEC, M. and AFTOSMIS, M. Aerodynamic shape optimization using a Cartesian adjoint method and cad geometry. In 25th AIAA Applied Aerodynamics Conference, San Francisco, USA, June 2006, AIAA 2006-3456.
29. POON, N.M.K. and MARTINS, J.R.R.A. An adaptive approach to constraint aggregation using adjoint sensitivity analysis. Structural and Multidisciplinary Optimization, 2007, **34**, pp 61-73.
30. RASMUSSEN, C. and WILLIAMS, C. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
31. ROBINSON, T., ELDRED, M., WILLCOX, K. and HAIMES, R. Surrogate-based optimization using multifidelity models with variable parameterization and corrected space mapping, *AIAA J*, November 2008, **46**, (11), pp 2814-2822.
32. SACKS, J., WELCH, W., MITCHELL, T. and WYNN, H. Design and analysis of computer experiments, *Statistical Science*, 1989, **4**, (4), pp 409-435.
33. SIMPSON, T., PEPLINSKI, J., KOCH, P. and ALLEN, J. Metamodels for computer-based engineering design: Survey and recommendations, *Engineering with Computers*, 2001, **17**, pp 129-150.
34. VENTER, G., HAFTKA, R. and STARNES, J. Construction of response surface approximations for design optimization, *AIAA J*, 1998, **36**, (12), 2242-2249.
35. WANG, Q., DURAISAMY, K., ALONSO, J.J. and IACCARINO, G. Risk assessment of scramjet unstart using adjoint-based sampling methods. In 51stAIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Orlando, FL, USA, April 2010. AIAA 2010-2921.

https://doi.org/10.1017/S0001924000006473 Published online by Cambridge University Press