Computer model calibration as a method of design

Carl Ehrett, Sez Atamturktur, Andrew Brown,
Evan Chodora, Mingzhe Jiang, Christopher Kitchens

October 29, 2018

Abstract

A well-established framework for the calibration of computer models is provided by Kennedy and O'Hagan (2001) and the myriad works that build upon it. Common to those approaches is the assumption that calibration is a matter of estimating unknown and/or uncontrollable parameters by attuning the model to data obtained through physical experimentation. In this work we reconceptualize model calibration as a method for optimization. Rather than calibrating a model to find a posterior distribution of unknown parameters in order to bring the model as much into agreement with reality as possible, we calibrate to find a posterior distribution on controllable model inputs in order to bring the predicted system behavior into agreement with pre-determined performance targets. In essence, we treat performance targets as "desired observations" and use them as the data in the calibration problem. We demonstrate our proposed methodology in both an artificial case and in the case of a finite element model of wind turbine blade performance and cost. In the latter case, we demonstrate how to estimate the Pareto front with uncertainty bands.

Keywords: Uncertainty quantification, Gaussian processes, optimization

1 Introduction

Broadly, in model calibration, one may consider a model to be of the form $\eta(\mathbf{x}, \boldsymbol{\theta})$, where $(\mathbf{x}, \boldsymbol{\theta})$ comprise all inputs to the model. Input vector $\mathbf{x}$ is the collection of inputs that are known and/or under the control of the researcher. The vector of calibration inputs $\boldsymbol{\theta}$ is the collection of parameters the values of which are unknown. These must be estimated for successful simulation. Thus where f describes the true system and y an observation of that system, consider the model to be

$$y(\mathbf{x}) = f(\mathbf{x}) + \epsilon(\mathbf{x}) = \eta(\mathbf{x}, \boldsymbol{\theta}) + \delta(\mathbf{x}) + \epsilon(\mathbf{x}) \tag{1}$$

where $\delta(\cdot)$ describes the model discrepancy – i.e., the bias of the model as an estimate of the real system – and $\epsilon(\cdot)$ is a mean-zero observation error, often i.i.d. Gaussian. To undertake model calibration, one must have access to at least some observations of the real system.

Much interest in the past two decades has centered on Bayesian methods for model calibration. The appeal of a Bayesian approach to model calibration is that the calibration parameters are a source of uncertainty for the model. This uncertainty should be quantified so that its effect on the model can be made explicit. One can thus use Bayesian methods to construct a posterior distribution of the calibration parameters which balances our prior knowledge about the calibration parameters with what can be learned from the available data, and which also allows for accurate uncertainty quantification on the model outputs.

The work of Kennedy and O'Hagan (2001) offers a Bayesian approach to computer model calibration that allows for the uncertainty of the calibration parameters in the predictions of the resulting calibrated model. This area is furthered by Higdon et al. (2004), who develop an approach that undertakes model calibration with quantification of the related uncertainty, as well as explicitly incorporating uncertainty regarding the computer model output, the bias of the computer model, and uncertainty due to observation error (of field data). Other contributions in this area come from Williams et al. (2006), Bayarri et al. (2007a), Liu et al. (2009), Bayarri et al. (2007b), Brynjarsdóttir and O'Hagan (2014), and Brown and Atamturktur (2018).

Common to these approaches is the conception of calibration as a matter of aligning the computer model output to observations of the real system. In this paper, we explore the idea of calibrating a computer model to align with performance targets, in order to find system settings that optimize performance with respect to those targets. In the remainder of Section 1, we describe the calibration framework which we go on in Section 2 to adapt as a method for optimization. In Sections 3 and 4 we apply the proposed methodology to an example involving simulated data, and to an application in which the goal is to find material design settings to optimize the performance of a wind turbine blade. Section 5 concludes with discussion of the results.

1.1 Calibration framework

In describing the calibration framework we use in this work, we here assume the use of a Gaussian process (GP) emulator of the computer model. In principle, model calibration need not rely on a GP emulator, or any other sort of emulator; one could (e.g.) complete a full Bayesian analysis via an MCMC chain that involves running the relevant computer model at each iteration of the chain. Indeed, in Section 3 we perform calibration on our example simulated data without an emulator. However, computer models are frequently too computationally expensive to allow for such profligacy. Instead, a computationally tractable emulator can be constructed using a sample of observations from the computer model. GPs are popular prior distributions on computer model output for three reasons. Firstly, because their use does not require detailed foreknowledge of the model function's parametric form. Secondly, GPs easily interpolate the computer model output, which is attractive when the computer model is deterministic. This is usually the case, although some attention in model calibration has focused specifically on stochastic computer models; see e.g. Pratola and Chkrebtii (2018). Thirdly, GPs facilitate uncertainty quantification through the variance of the posterior GP. This section provides brief background on Gaussian processes and their use in regression broadly and in computer model calibration specifically.

The use of GPs to produce a computationally efficient predictor $\hat{\eta}(\mathbf{x})$ of expensive computer code $\eta(\mathbf{x})$ given observations of code output at $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ is promulgated by Sacks et al. (1989) and explored at length by Santner et al. (2003). Since computer code is typically deterministic, these applications differ from the focus of O'Hagan (1978) in that the updated GP is induced to interpolate the observations $\boldsymbol{\eta} = (\eta(\mathbf{x}_1), \cdots, \eta(\mathbf{x}_n))^T$. Kennedy and O'Hagan (2001) develop an influential framework for Bayesian analysis using GPs specifically for computer model calibration. Kennedy et al. (2006) showcase this use of GP emulators for uncertainty and sensitivity analyses. Bastos and O'Hagan (2009) describe both numerical and graphical diagnostic techniques for assessing when a GP emulator of a computer model is successful, as well as discussion of likely causes of poor diagnostic results. While most work in the area of GP emulation uses stationary covariance functions (in which $\mu(\cdot)$ is constant and $C(\mathbf{x}, \mathbf{x}') \equiv C(\mathbf{x} - \mathbf{x}')$ depends only on the difference between $\mathbf{x}$ and $\mathbf{x}'$, rather than on their location in the input domain) and quantitative inputs, efforts have been made to branch away from these core uses. Gramacy and Lee (2008) use treed partitioning to deal with a nonstationary computer model. Qian et al. (2008) explore methods for using GP emulators that include both quantitative and qualitative inputs.

Suppose that we have inputs $\{\mathbf{x}_i\}_{i=1}^n \subseteq \mathbb{R}^p$ scaled to the unit hypercube, and observations

$$y(\mathbf{x}_i) = f(\mathbf{x}_i) + \epsilon(\mathbf{x}_i), \quad i = 1, \cdots, n, \tag{2}$$

where $f(\cdot)$ is the true system and $\epsilon(\cdot)$ is known measurement error. Then by (1) we have

$$y(\mathbf{x}_i) = \eta(\mathbf{x}_i, \boldsymbol{\theta}) + \delta(\mathbf{x}_i) + \epsilon(\mathbf{x}_i), \quad i = 1, \cdots, n \tag{3}$$

where $\eta(\cdot, \cdot)$ is the computer model, $\boldsymbol{\theta}$ is the bestsetting of the vector of calibration parameters, and $\delta(\cdot)$ is the discrepancy function describing the bias of $\eta(\cdot, \cdot)$ as an estimate of $f(\cdot)$. Williams et al. define the GP prior for modeling $\eta(\cdot, \cdot)$ as follows. Let the mean function $\mu(\mathbf{x}, \mathbf{t}) = c$, c a constant. Set the covariance function in terms of the marginal precision λ_η and a product power exponential correlation function:

$$C((\mathbf{x}, \mathbf{t}), (\mathbf{x}', \mathbf{t}')) = \frac{1}{\lambda_\eta} \prod_{k=1}^{p} \exp\left(-\beta_k^\eta |x_k - x_k'|^{\alpha_\eta}\right) \times \prod_{k=1}^{q} \exp\left(-\beta_{p+k}^\eta |t_k - t_k'|^{\alpha_\eta}\right) \tag{4}$$

where each β_k describes the strength of the GP's dependence on one of the elements of the input vectors $\mathbf{x}, \mathbf{t}$, and α_η determines the smoothness of the GP.

The authors place the following priors on the hyperparameters:

$$\begin{aligned} c &\sim N(0, v) \\ \lambda_\eta &\sim \text{Gamma}(5,5), \quad \lambda_\eta > 0 \\ \rho_k^\eta &\sim \text{Beta}(1, 0.1), \quad k = 1, \cdots, p+q \end{aligned} \tag{5}$$

where $\rho_k^\eta = \exp(-\beta_k^\eta/4)$ for $k = 1, \cdots, p+q$. They set $\alpha_\eta = 2$, which corresponds to the assumption of an infinitely differentiable correlation function. The parameters of the Gamma and Beta distributions are chosen to encourage λ_η to be close to one, and β_k to be low for all k (encouraging strong dependence; i.e., we antecedently expect each of the inputs to be influential). Furthermore, the authors let $v \to 0$, i.e., the GP is assumed to have constant mean $c = 0$.

The authors similarly model the discrepancy term as a GP, also with mean zero, and covariance function

$$C_\delta(\mathbf{x}, \mathbf{x}') = \frac{1}{\lambda_\delta} \prod_{k=1}^{p} \exp\left(-\beta_k^\delta |x_k - x_k'|^{\alpha_\delta}\right), \tag{6}$$

with priors

$$\begin{aligned} \lambda_\delta &\sim \text{Gamma}(a_\delta, b_\delta) \\ \rho_k^\delta &\sim \text{Beta}(1, 0.3). \end{aligned} \tag{7}$$

where $\rho_k^\delta = \exp(-\beta_k^\delta/4)$ for $k = 1, \ldots, p$ and $\alpha_\delta = 2$.

Where $\boldsymbol{\eta} = (\eta(\mathbf{x}_1, \mathbf{t}_1), \cdots, \eta(\mathbf{x}_n, \mathbf{t}_n))^T$ are the simulation observations, $\mathbf{y} = (y(\mathbf{x}_{n+1}), \cdots, y(\mathbf{x}_{n+m}))^T \equiv (y(\mathbf{x}_{n+1}, \boldsymbol{\theta}), \cdots, y(\mathbf{x}_{n+m}, \boldsymbol{\theta}))^T$ are the field observations, $\mathcal{D} = (\boldsymbol{\eta}^T, \mathbf{y}^T)^T$, $\boldsymbol{\beta}^\eta = (\beta_1^\eta, \cdots, \beta_{p+q}^\eta)^T$, and $\boldsymbol{\beta}^\delta = (\beta_1^\delta, \cdots, \beta_{p+q}^\delta)^T$, we then have the distribution of $\mathcal{D}$ as

$$\mathcal{D}|\boldsymbol{\theta}, c, \lambda_\eta, \boldsymbol{\beta}^\eta, \lambda_\delta, \boldsymbol{\beta}^\delta, \mathbf{C}_\mathbf{y} \sim N(c \cdot \mathbf{1}_{n+m}, \mathbf{C}_\mathcal{D}) \tag{8}$$

where $\mathbf{C}_\mathbf{y}$ an $m \times m$ matrix in which the i, j entry is the (known) observation variance $C_{obs}(\mathbf{x}_i, \mathbf{x}_j)$ for $n < i, j \leq n + m$, and $\mathbf{C}_\mathcal{D}$ is a matrix with its i, j entry equal to

$$C((\mathbf{x}_i, \mathbf{t}_i), (\mathbf{x}_j, \mathbf{t}_j)) + I(i, j > n) \cdot (C_{obs}(\mathbf{x}_i, \mathbf{x}_j) + C_\delta(\mathbf{x}_i, \mathbf{x}_j)) \tag{9}$$

Thus, the joint posterior density under the model is

$$\pi(\boldsymbol{\theta}, c, \lambda_\eta, \boldsymbol{\rho}^\eta, \lambda_\delta, \boldsymbol{\rho}^\delta, \mathbf{C}_\mathbf{y}|\mathcal{D}) \propto \pi(\mathcal{D}|\boldsymbol{\theta}, c, \lambda_\eta, \boldsymbol{\beta}^\eta, \lambda_\delta, \boldsymbol{\beta}^\delta, \mathbf{C}_\mathbf{y}) \times \pi(c) \times \pi(\lambda_\eta) \times \pi(\boldsymbol{\rho}^\eta) \times \pi(\lambda_\delta) \times \pi(\boldsymbol{\rho}^\delta) \tag{10}$$

Note that where a discrepancy function is not included in the model and the mean c is treated as a constant, (10) simplifies greatly; where furthermore λ_η and $\boldsymbol{\rho}^\eta$ are estimated via maximum likelihood (as in Kennedy and O'Hagan (2001)), (10) simplifies down merely to $\pi(\mathcal{D}|\boldsymbol{\theta}, c, \lambda_\eta, \boldsymbol{\beta}^\eta, \lambda_\delta, \boldsymbol{\beta}^\delta, \mathbf{C}_\mathbf{y})$. Markov chain Monte Carlo methods are useful for evaluating (10).

1.1.1 Computational difficulties

Consider a GP with constant mean function $\mu(\mathbf{x}) = 0$ for all $\mathbf{x}$. We use the covariance function C to define an $n \times n$ matrix $\mathbf{C}_{\mathbf{X},\mathbf{X}}$ such that the i, j entry of $\mathbf{C}_{\mathbf{X},\mathbf{X}}$ is equal to $C(\mathbf{x}_i, \mathbf{x}_j)$. Training the GP on the n observations $\boldsymbol{\eta}$ at $\mathbf{X} \in \mathbb{R}^{n \times p}$, an updated mean and covariance matrix for the points $\mathbf{X}' = (\mathbf{x}_1', \ldots, \mathbf{x}_m')^T$ is

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{X}'}^* &= \mathbf{C}_{\mathbf{X}',\mathbf{X}} \cdot \mathbf{C}_{\mathbf{X},\mathbf{X}}^{-1} \cdot \boldsymbol{\eta} \\ \mathbf{C}_{\mathbf{X}'}^* &= \mathbf{C}_{\mathbf{X}',\mathbf{X}'} - \mathbf{C}_{\mathbf{X}',\mathbf{X}} \cdot \mathbf{C}_{\mathbf{X},\mathbf{X}}^{-1} \cdot \mathbf{C}_{\mathbf{X},\mathbf{X}'} \end{aligned} \tag{11}$$

Poor conditioning of $\mathbf{C}_{\mathbf{X},\mathbf{X}}$ in (11) can make it difficult to invert and find the determinant of this matrix, as must be done in the course of the MCMC to find the relevant likelihoods. This problem can be alleviated by adding a small nugget to $\mathbf{C}_{\mathbf{X},\mathbf{X}}$. That is, we can set $\mathbf{C}_{\mathbf{X},\mathbf{X}}^\xi = \mathbf{C}_{\mathbf{X},\mathbf{X}} + \xi \cdot \mathbf{I}_{\dim(\mathbf{X})}$ for some very small value

of ξ, e.g., $\xi = 10^{-4}$. Such a simple nugget works quite well in many applications. For a more sophisticated approach to selecting the nugget size, however, see Ranjan et al. (2011). Note that adding a nugget here is equivalent to adding a small amount of observation variance for the simulator observations. That is, in adding this nugget, one no longer requires that the GP emulator precisely interpolate the simulation observations. However, for very small nuggets, this effect is so small as to be negligible, though the computational benefits remain. Furthermore, insofar as the effect is non-negligible, it is argued by Gramacy and Lee (2012) to be beneficial for emulating the computer code (so that for some applications one might prefer a larger nugget).

2 Calibration for design

Suppose that a researcher has a fairly reliable computer model of a given system. Suppose furthermore that some of the parameters of that system can be controlled, and that the researcher hopes to select values for these controllable parameters that will facilitate certain target outcomes from the system.

In traditional calibration as described in Section 1, a computer model is calibrated to physical observations. This is done in order to find settings for the computer model that induce its output to match reality as closely as possible. Similarly, one may seek to "calibrate" a computer model to a set of performance targets, in order to find settings that induce the model's output to match, or approximate, those targets. Hereafter, call performance targets treated as observations for the purpose of calibration "desired observations". Call the calibration procedure proposed here, which uses the model calibration framework of Kennedy and O'Hagan (2001) with desired observations, "calibration to desired observations" (CDO).

Computer models are more malleable than reality, and one might worry that in calibrating a model to unobserved performance targets, the model's fidelity to reality might be mitigated. In many cases, however, one is fortunate to have (perhaps after undertaking traditional model calibration, validation and verification) a computer model such that one is confident that the model is known to be faithful to reality over a given set $\mathcal{T}$ of user specified calibration parameters. In such a circumstance, in calibrating $t \in \mathcal{T}$ to one's desires, one does not risk calibrating the model *away* from agreement with reality, even if one's performance targets are not realistically achievable. Instead, one finds a distribution on the settings that achieve the best realistic approximation to the desired targets.

The tools of model calibration founded in the work of Kennedy and O'Hagan (2001) retain their advantages under the proposed application. Most centrally, such calibration to desired observations y produces not merely a static optimum $t \in \mathcal{T}$, but rather a posterior distribution of $t|y$ reflective of remaining uncertainty about the appropriate value of t. Such uncertainty may have its source in parameter uncertainty (uncertainty about the values of certain model inputs), code uncertainty (uncertainty about how closely the code approximates reality), and that which traditional calibration would consider observation error and model inadequacy. Of course, targets are not actually observations, so the concept of observation error does not cleanly transfer. However, a similar uncertainty would be that due to how close reality *can* come to our desired observations. The model calibration framework of Kennedy and O'Hagan (2001) allows for the quantification of all of these uncertainties. Furthermore, by the use of informative priors on the model discrepancy and observation error, the identifiability concerns of the Kennedy-O'Hagan approach can be mitigated (Bayarri et al., 2007b; Tuo and Jeff Wu, 2016).

2.1 Target observations

2.1.1 Level of target data

Unlike in the case of field observations, when calibrating to performance targets treated as desired observations, the question arises of choosing what exactly those "observations" should be. In many cases, no objectively natural target manifests itself. Indeed, there is no barrier to the use even of impossible targets such as negative values for model outputs known to be nonnegative. Such a target observation in certain situations may be appropriate. However, in general, target observations should aim only a little beyond what is realistically achievable; only as much as is necessary to ensure the targets are at least as ambitious as any true optimum in the system. Three reasons why one should go only a little beyond that are as follows. (1) If target observations are set to be too farfetched, then the calibration can become computationally unstable due to underflow and round-off error, since any value of $\boldsymbol{\theta}$ within its support will have extremely

low likelihood. (2) Increasing the distance of the desired observations from the optimal region reduces the identifiability of that region. The calibration finds the region of the parameter space with output closest to the target observations. If the entire model range is far from the target observations, then the optimal region will in relative terms be only a little closer than the rest of the model range. As a result, the identifiability of the optimal region will suffer. (3) The desired observations lose a measure of interpretability when they delve too far into the fantastical, such as with impossibly negative values. Identifying the appropriate range of outputs for desired observations, which exceed reality only slightly, will often require one to consult a subject matter expert.

A third option is also explored in Section 4's treatment of the material design application. This option is not truly another means of achieving a calibration target, but rather is simply the decision to refrain from doing so. That is, rather than include a desired observation of, say, cost in the model or set a prior that induces low cost, one can simply specify a known cost and calibrate desired performance targets to a design having that cost. If it is antecendently unknown which cost settings are optimal, under this third option one may calibrate to performance targets under each point of a grid of "known" costs. Thus we present a comprehensive picture of optimal parameter distributions and resulting performance under a range of costs, which could inform the process of setting a budget for material construction.

2.2 Model shortcoming

It is not merely likely but often desirable that the performance targets have low probability with respect to the likelihood of the calibrated model. In this way, CDO (calibration to desired observations) is unlike traditional calibration. The reason for this is that if the posterior predictive distribution places substantial probability mass at regions of the parameter space that achieve the target desired observations, then the desired observations may have been insufficiently ambitious. In the wind turbine blade application considered in this work, the ideal material would (impossibly) not deform at all under load. In a different application, one might wish to design a material that deforms in a pre-specified (possible) way. In such a case, it would be appropriate to set desired observations that one indeed does hope to find as the posterior predictive mode after calibration. But in cases such as the wind turbine application, finding the desired observations to be the posterior mode would be an indication that the desired observation could potentially be outperformed, or else a warning (if the desired observation is known to be impossible, such as a material that undergoes zero deformation under load) that the model itself may be unrealistic.

Where the mean of the posterior predictive distribution from CDO fails to interpolate the desired observations, this can be understood in two distinct ways. These correspond to the two distinct sources of error in traditional calibration to field observations. The first such source of error is model discrepancy, or $\delta(\cdot)$ in (1). This is defined to be the difference between the mean of the true system and the output of the computer model. It is thus the extent to which the computer model fails to capture reality. The other source of error is observation error, $\epsilon(\cdot)$ in (1). This source of error cannot be attributed to any failing on the part of the computer model. Neither of these two sources of error, under their traditional interpretations, succeeds in capturing the nature of the gap between desired observations and the posterior predictive mean. These two sources of error can nonetheless serve as a basis for modeling this gap, as we discuss in the next subsection.

The model describes *reality*, not our desires. Thus failure to interpolate our desires is not necessarily model error. Though for convenience and ease of exposition we refer to this gap as "error", we can more properly refer to it as "model shortcoming". This phrasing still (infelicitously) implies failure on the part of the model, whereas in fact the underlying discrepancy is between the performance targets and what is physically possible. Still, the term is appropriate, since the "error" that is observed is a discrepancy between the desired observations and the model, not between the model and the true system.

2.3 Setting the marginal variance for model discrepancy

Hereafter in this work, we will assume that model shortcoming will be captured via a discrepancy term $\delta(\cdot)$, modeled as a mean-zero, stationary GP. In order to successfully calibrate to the optimal region of the parameter space, it is necessary either to place an informative prior on the marginal precision λ_δ of the discrepancy, or else to specify that value outright. Otherwise, identifiability issues can cause the calibration to fail. This is a longstanding concern with the Kennedy-O'Hagan framework, raised in the discussion of

Kennedy and O'Hagan (2001) as well as by Bayarri et al. (2007b), Tuo and Jeff Wu (2015), and Plumlee (2017). How informative one's prior on λ_δ will be depends upon how much one knows about the true Pareto front prior to undertaking CDO. For instance, if in a univariate case it is known with some confidence that the true optimum is nearly constant across control settings and that it occurs in the interval $[10, 11]$, then a constant desired observation of 9 could be used with an informative prior tailored to this prior knowledge of the approximate resulting discrepancy – say $\text{Gamma}(20, \text{rate} = 20)$.

When the true Pareto front cannot be estimated prior to undertaking CDO, the desired observations and the prior on the marginal precision of the discrepancy function must be set to avoid the identifiability problems of the Kennedy-O'Hagan framework. That is, where the prior on λ_δ cannot be chosen to be *accurate* (due to insufficient prior knowledge) it should be chosen to *overestimate* the precision. Otherwise, underestimation of λ_δ may lead to poor identifiability of the optimal region of the parameter space. Again consider the example with the constant optimum in $[10, 11]$, but suppose now that our prior knowledge is much more impoverished – we can confidently hold only that the optimum takes positive values under 20. Then a constant desired observation of -1 could be paired with a $\text{Exp}(1)$ prior on λ_δ, to reflect our hope for a discrepancy of 1 while remaining open to a significantly larger discrepancy. In such a case, by setting a prior that overestimates λ_δ, the posterior distribution of λ_δ becomes less reliable than when the prior derives from substantive prior knowledge. However,even when λ_δ must be overestimated, the posterior distribution of $\boldsymbol{\theta}$ will still peak at the optimal region of the parameter space, since the overestimation of λ_δ only increases the penalty of leaving that region. And so while making do with vague knowledge of the optimum does interfere with one's ability to estimate the true discrepancy of the model from the desired observations, it does *not* interfere with one's ability to locate the posterior mode of $\boldsymbol{\theta}$ and thereby the optimal settings for the model.

Furthermore, when the true Pareto front cannot be estimated prior to undertaking CDO, a preliminary round of CDO can be used to provide such an estimate. E.g., consider again the case where we know only that the Pareto front is in the range $(0, 20)$. We can perform CDO with constant desired observation -1 and a prior on λ_δ that deliberately exploits the identifiability problems of the Kennedy-O'Hagan framework in order to explore large regions of the parameter space – say $\text{Exp}(\text{rate} = 0.1)$. The Pareto front of the resulting CDO sample draws can be used as an estimate of the true Pareto front in the vicinity of the desired observation.

In the case of univariate output, the only reason to perform this preliminary step would be to improve the accuracy of the posterior discrepancy. This is because in the case of univariate output, while there may be uncertainty about the magnitude of the difference between the Pareto front and a given set of desired observations, there is no uncertainty about the direction of this difference. But in the case of multivariate output, both the magnitude and the direction of the difference may be unknown. In such a case, a poorly chosen set of desired observations may not result in calibration to the desired region of the Pareto front. E.g., consider a case of bivariate positive output (y_1, y_2) where within the Pareto front y_2 is continuous and strictly decreasing in y_1. Then any value of y_1 corresponds to some point in the Pareto front. In such a case one often wishes to locate an "elbow" which maximizes $\frac{d^2 y_2}{dy_1^2}$. Depending on its location, selecting (e.g.) $[0, 0]$ as a desired observation might not achieve calibration to this elbow, if some other part of the Pareto front is closer to the desired observation than the elbow is. It is in such a situation that a rough estimate of the Pareto front (if not antecedently available) via a preliminary round of CDO using a vague prior on λ_δ can be used to select a desired observation to which the closest region of the Pareto front is the region to which one wishes to calibrate. Then for the second round of CDO one's desired observations and prior distribution on λ_δ can be chosen informatively, so that one will enjoy the above-described benefits both of calibrating to the desired region of the Pareto front and of improving the accuracy of one's posterior distribution on the discrepancy of the model from the desired observations. Note that a preliminary round of CDO can use the same set of model observations as the subsequent CDO. So performing preliminary CDO to sharpen one's desired observation and λ_δ prior does not add to the total budget of model runs, and can thus be a computationally cheap supplement to CDO.

2.4 Field observations and model discrepancy

In the version of CDO presented thus far, the calibration has been entirely to desired observations. This invites the question of how to proceed when one wants to undertake both traditional calibration and cali-

bration to desired observations. In other words: what happens when we have both desired observations *and* field observations?

The question thus arises as to whether it is possible to undertake both calibrations simultaneously. This would seem to encounter the difficulty that one is allowing one's calibration parameters to be simultaneously "pulled" in two directions: toward the true values (by the field observations) and toward a target outcome (by the desired observations), with the result that neither calibration goal is achieved. But notice that these two sorts of calibration parameters tend not to overlap in the matter of which parameters are considered to be calibration parameters. The purpose of CDO is to find optimal settings for parameters over which we have control. It's no use finding out that (e.g.) a building will be most efficiently evacuated when the occupants have average body dimensions $\mathbf{b}$, since we have no power to mandate the body dimensions of people fleeing a burning building. Instead, given a distribution on body dimensions, we may seek to find the layout that best contributes to efficient evacuation, since the building layout is under our control. By contrast, in traditional calibration, one ordinarily specifically calibrates those parameters over which we have no control, and whose true value we seek to discover.

This separation between what constitutes the calibration parameters of the two procedures opens the possibility of the two calibrations proceeding simultaneously, without undermining one another. This possibility will be pursued in future work on this subject. However, an alternative solution is to undertake traditional calibration prior to CDO, finding both a distribution on the (traditional) calibration parameters and an estimated model discrepancy function. Thus one arrives at CDO (assuming success in the former calibration) with a model that faithfully represents the true system.

2.5 Hyperparameter estimation

Consider the covariance function parameters $\lambda_\eta, \boldsymbol{\beta}^\eta$ in (4). In a full Bayesian analysis, these would be searched over in the MCMC along with the calibration parameters. However, Kennedy and O'Hagan (2001) instead find the MLEs of these hyperparameters prior to calibration. More generally, Bayarri et al. (2007b) and Liu et al. (2009) advocate what they call "modularization". Modularization refers to separating sources of information, so that the model has distinct components or "modules", rather than allowing all information to combine into a single analysis under the umbrella of Bayes' theorem. Liu et al. (2009) focus directly on the use of modularization, exploring its advantages and disadvantages; amongst other potential motivations, they show that modularization can improve the identifiability of calibration parameters. A key motivation of modularization is to protect good components of the model from "suspect" components of the model, and desired observations are, by their very nature, "suspect". In other words, the model most successfully represents reality when the settings for these hyperparameters are guided by accurate and precise information about the true system. Desired observations are deliberately not such information. Attempting a full Bayesian analysis that finds these hyperparameter settings as part of CDO would be violate the principle that CDO should be used to tune only those parameters which are within our control over a range $\mathcal{T}$ such that the model faithfully represents reality over all of $\mathcal{T}$. In the true system, the hyperparameters of the covariance function are not under our control.

Therefore, care should be taken to prevent the desired observations from "infecting" the covariance hyperparameters, since we want the latter to reflect reality rather than our performance targets. The way that this is prevented in the application of Section 4 is by using maximum likelihood estimation from the simulation observations alone to estimate these values. Field observations could be used here as well, either for the maximum likelihood estimation or for a modular Bayesian analysis à la Liu et al. (2009).

3 Example

To illustrate CDO, consider the following artificial problem. Let $(x, \boldsymbol{\theta})$ be the vector of inputs, with scalar control input $x \in [1.95, 2.05]$ and calibration parameters $\boldsymbol{\theta} = (\theta_1, \theta_2) \in [0, 3] \times [0, 6]$. We consider three

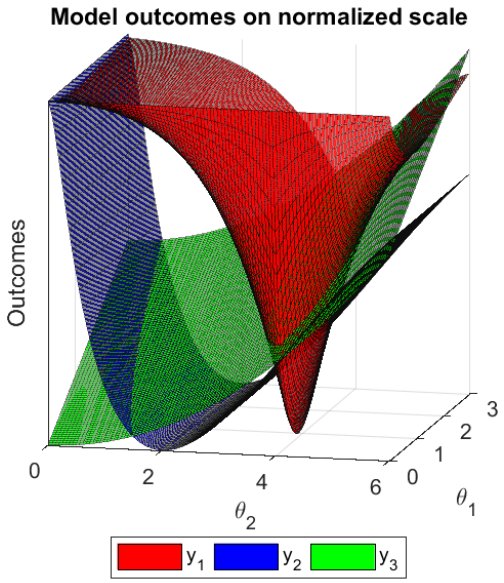

Figure 1: Example model outputs shown on a common scale.

outputs:

$$\begin{aligned} y_1 &= \left(\theta_1 \exp\left(-\left(\theta_1 + |\theta_2 - \frac{\pi \cdot x}{2}|\right)\right) + 1\right)^{-1} \\ y_2 &= \left(\theta_2^{x-1} \exp\left(-0.75\theta_2\right) + 1\right)^{-1} \\ y_3 &= 15 + 2\theta_1 + \frac{\theta_2^2}{4}. \end{aligned} \tag{12}$$

Figure 1 shows the three model outputs on a common scale over the support of the calibration parameters, for $x = 2$. For CDO, the true function was used (rather than a GP emulator). Thus we have

$$\mathbf{y}_j = \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) + \delta(\mathbf{x}) + \epsilon_j$$

for desired observation $\mathbf{y}_j$, where $\mathbf{f}$ is the model output, $\delta(\cdot)$ is the discrepancy function and ϵ_j is independent $N(0, 0.05)$ for all j.

We initially set the desired observations to $[0, 0, 0]$, constant as a function of x. We then estimated the Pareto front via a preliminary round of CDO in order to estimate the distance (after standardizing the outputs to a common scale) of the desired observation from the Pareto front. The distance from the Pareto front to the desired observation was found to be large – at 16 units on the standardized scale, roughly four times the diameter of the Pareto front itself. In order to improve identifiability of the region of the Pareto front closest to the desired observation, we updated the latter to lie closer to the Pareto front (along the same line connecting the Pareto front to the orginal desired observation). We chose a distance of one unit away (roughly one fourth of the diameter of the Pareto front), approaching the (estimated) Pareto front as closely as possible while remaining confident that the new desired observation of $[0.71, 0.71, 17.92]$ still outperforms the true Pareto front. We then set the discrepancy marginal precision λ_δ to 1 for subsequent CDO, corresponding to a degenerate informative prior at the estimated value from preliminary CDO. Observation error $\epsilon(\cdot)$ from (1) was specified as $N(0, 0.05)$ for all x. Figure 2 shows the results, including the marginal distributions of the calibration parameters. The sharply peaked marginals show substantial Bayesian learning compared to the uniform prior distributions on the calibration parameters. The calibration successfully maps the contours of the optimal region, and peaks near the true optimum.

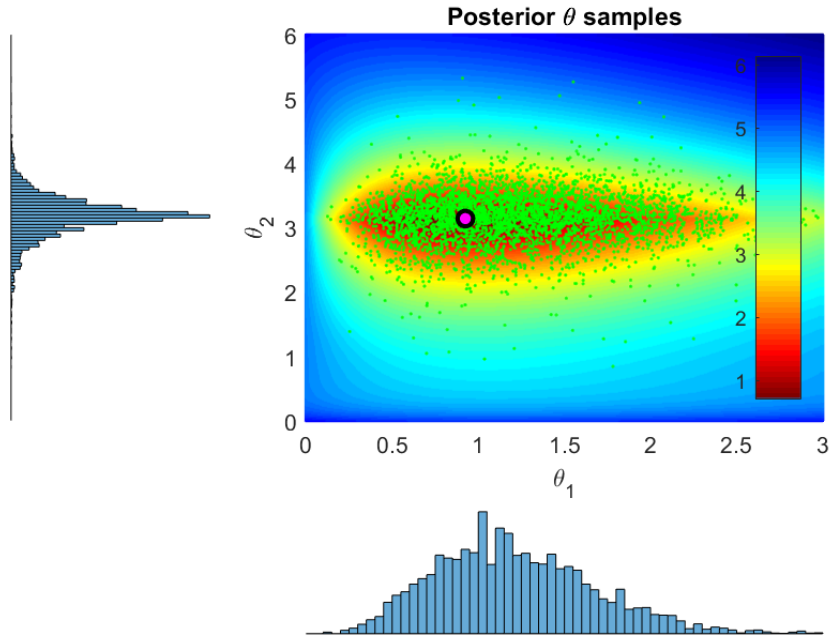

Figure 2: Posterior draws from CDO. The heatmap shows, for each point in the parameter space, the Euclidean distance of the model output at that point from the desired observation. The purple dot shows the true optimum.

4 Application

In this section we describe the use of CDO for the problem of designing a material for constructing a wind turbine blade of fixed geometry. In traditional engineering design, material selection is a matter of choosing a material with appropriate properties for the project at hand from a database of known materials, often as a matter of ad-hoc satisficing. Material design usually occurs separately, and without an eye to specific end-uses. It is desirable to wed these design processes, selecting a material design by modeling its performance outcomes in a particular engineering application. Therefore, here we offer an example of calibrating material design parameters to desired performance targets for a wind turbine blade. This calibration is mediated by a finite element model using `ANSYS` simulation software, which is treated as an accurate representation of reality.

4.1 Project background

Two primary performance targets for the design and construction of wind turbine blades are the distance (in meters) that the blade tip deflects under load from its starting position, and the angle (in radians) that the blade undergoes rotation when under load. Each of these measures should ideally be as close to zero as possible. In selecting the composite material used to build the turbine blade, given a choice of matrix and filler materials, the properties of the material depend on the volume fraction (i.e. the volume ratio of filler material to matrix material used in the composite) and the thickness of the material used to build the blade. The resulting material properties impact the performance of the blade, as well as its cost per square meter.

The finite element model takes as inputs a triplet (h, v, k), where h is the operating temperature of the wind turbine (in kelvin), v is the volume fraction of the material, and k is the thickness of the material (in mm). The outputs of the model are a triplet (d, r, c), where d is tip deflection (in meters), r is rotation (in radians), and c is cost per square meter (USD). The wind turbine should be capable of operating over the range of temperatures 230K-330K. The goal of calibration is thus to find posterior distributions on v and k given outputs from the finite element simulator and desired observations.

4.2 Emulation of finite element simulator

The finite element simulator is too computationally expensive to be suitable for direct use in (e.g.) an MCMC routine. Thus we employ a GP emulator in the manner of Williams et al. (2006). For this purpose, we drew 504 (trivariate) observations from the finite element simulator. These inputs follow a Latin hypercube sampling design (McKay et al., 1979) based on plausible ranges for the three inputs, as identified by expert opinion. We consider the finite element observations to follow a GP with mean 0 and covariance function C as described by (4) above, with $\alpha_\eta = 2$. This is equivalent to assuming smooth, infinitely differentiable sample paths.

The hyperparameters $\lambda_\eta, \boldsymbol{\beta}^\eta$ must be estimated. For the reasons discussed in Section 2.5, we estimated them prior to calibration to the desired observations, via maximum likelihood estimation. We used a gradient descent method (Cauchy, 1847) to maximize the log-likelihood of the simulation observations over the joint (6-dimensional) support of $\boldsymbol{\beta}^\eta, \lambda_\eta$. The result is $\hat{\lambda}_\eta = 0.0152$, $\hat{\boldsymbol{\rho}}^\eta = (0.9358, 0.6509, 0.6736, 0.4797, 0.9673)$ where $\rho_k^\eta = \exp(-\beta_k^\eta/4)$.

4.3 The model

Following the framework laid out in Section **??** and the hyperparameters estimated in Section 4.2, the model takes the trained emulator to be distributed as

$$\mathcal{GP}\left(\mu^*(\mathbf{b}), C^*(\mathbf{b}, \mathbf{b}')\right) \tag{13}$$

where $\mu^*(\mathbf{b}) = \mathbf{C}_{\mathbf{b},\mathbf{B}} \cdot \mathbf{C}_{\mathbf{B},\mathbf{B}}^{-1} \cdot \boldsymbol{\eta}$, $C^*(\mathbf{b}, \mathbf{b}') = \mathbf{C}_{(\mathbf{b}^T,\mathbf{b}'^T)^T,(\mathbf{b}^T,\mathbf{b}'^T)^T} - \mathbf{C}_{(\mathbf{b}^T,\mathbf{b}'^T)^T,\mathbf{B}} \cdot \mathbf{C}_{\mathbf{B},\mathbf{B}}^{-1} \cdot \mathbf{C}_{\mathbf{B},(\mathbf{b}^T,\mathbf{b}'^T)^T}$, $\mathbf{C}_{\Upsilon,\Gamma}$ is the matrix whose i,j element is equal to the covariance between the observation at the i^{th} row of Υ and at the j^{th} row of Γ, $\mathbf{b} = (\mathbf{x}, \mathbf{t})$ is a row vector of control and calibration inputs, $\mathbf{B} = (\mathbf{b}_1^T, \mathbf{b}_2^T, \cdots, \mathbf{b}_n^T)^T$ is the 1512×5 matrix of locations of the 1512 simulation observations, and $\boldsymbol{\eta}$ is a column vector of the 1512 simulation responses: $\eta_i = \eta(\mathbf{b}_i)$. All model inputs are normalized to [0,1] over their supports. All model outputs are standardized so that $\boldsymbol{\eta}$ has mean 0 and standard deviation 1. $C(\cdot,\cdot)$ is given by (4), where I plug in the MLEs given above. The full joint posterior density of the calibration parameters and discrepancy function hyperparameters, from (10), is

$$\pi(\boldsymbol{\theta}, \lambda_\delta, \boldsymbol{\rho}^\delta | \mathcal{D}) \propto \pi(\mathcal{D} | \boldsymbol{\theta}, \lambda_\delta, \boldsymbol{\rho}^\delta) \times \pi(\lambda_\delta) \times \pi(\boldsymbol{\rho}^\delta). \tag{14}$$

The initial desired observations were set to $[0, 0, 0]$, constant as a function of temperature. We carried out an initial round of CDO in order to update the desired observations to ones that lie an estimated distance of 1 (on a standardized scale) from the Pareto front. A total of 20,000 samples were drawn via Metropolis-Hastings-within-Gibbs MCMC, of which 4,000 samples were thrown out as burn-in. During the burn-in period, the covariance of the proposal distributions for $\boldsymbol{\theta}$, λ_δ, and $\boldsymbol{\rho}^\delta$ were all adjusted for optimal acceptance rates. The adjustment took place every 100 iterations of the MCMC, at which point the relevant covariance matrix was set to be equal to the sample covariance of the previous draws, times a scalar multiplier. The level of the scalar multiplier was adaptively adjusted to promote optimal acceptance rates of $\approx 30\%$ for $\boldsymbol{\theta}$ and $\boldsymbol{\rho}$, and $\approx 44\%$ for λ_δ. As expected for the preliminary round of CDO, the posterior distribution of $\boldsymbol{\theta}$ was quite diffuse. We used the GP emulator to estimate the model output for each sample of $\boldsymbol{\theta}$ drawn. We filtered the resulting posterior predictions to retain only the estimated Pareto front. Examining the estimated Pareto front, one finds a distinct "elbow"; see figure 3. We selected this elbow as the target for calibration. To do so, we set the point [deflection = 0.75m, rotation = 0.09 rad, cost = \$130.34] as the desired observation (constant as a function of temperature). The elbow is the closest region of the Pareto front to this point. Based on the estimated PF, the desired observation is approximately 0.2 units away on the standardized scale. Therefore, we set $\lambda_\delta = 1/0.2^2 = 25$.

In the resulting CDO, we employed the same MCMC approach as in the preliminary round, except that λ_δ was now treated as known. The marginal posterior distributions are shown in Figure 4, along with the (uniform) prior distributions. The mean model output under the prior is [deflection = 0.76m, rotation = 0.09 rads, cost = \$207.90/m^2], whereas under the posterior it is [0.76m, 0.09 rad, \$148.68]. Though the performance outcomes are approximately the same under the posterior distribution as under the prior, the cost per square meter has dropped dramatically. If one desires to prioritize gains in performance over cost, this can be accomplished by selecting desired observations that reflect those priorities.

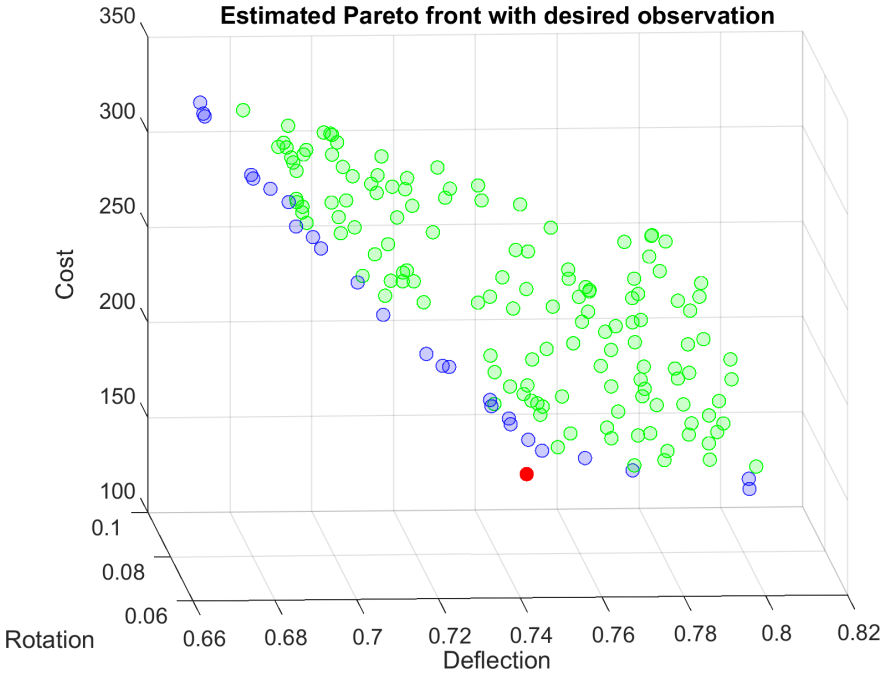

Figure 3: The green points show the estimated range of the model. The blue points show the estimated Pareto front. The red dot is the desired observation selected to calibrate to the "elbow" in the Pareto front.

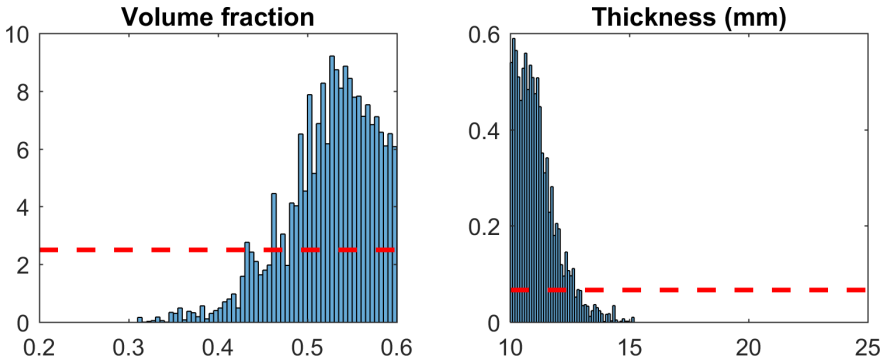

Figure 4: The histograms show the marginal posterior of each calibration parameter. The dotted lines show the priors.

4.4 Removing calibration parameters

Where a model contains several outputs that one wishes to optimize, it can become complicated to juggle one's priorities in selecting targets for each of these outputs. Again to speak from the wind turbine application: we know we want to keep cost, deflection, and rotation low, but we might not have a clear prior conception of exactly what sorts of trade-offs amongst those outcomes we would consider optimal, much less how to implement our priorities in the calibration procedure. Rather than wishing to optimize relative to some particular set of desired observations, one may prefer simply to learn as much as possible about the Pareto front as a whole. In low-dimensional cases this may be achieved using CDO as follows. Where the model output is $d-$dimensional, one may draw a grid over the range of $d-1$ of the model outputs and perform CDO to minimize the remaining output at each point of the grid. The $d-1$ outputs, at each grid point, are treated as known up to observation error (meaning that the discrepancy function $\delta(\cdot)$ is set to 0 in the dimension of these outputs). The resulting estimate is distinguished from other methods of estimating the Pareto front (including from the filtering method employed in preliminary CDO) by including uncertainty quantification.

This procedure is illustrated here using the wind turbine blade application. For simplicity, rotation has been removed as a model output, leaving a 2-dimensional model output of deflection and cost. A 20-point grid was drawn over costs. Using the rough estimate of the Pareto front from preliminary CDO, we found that to cover the Pareto front, the cost grid should cover the range $[\$96, \$352]$. For each point c in the cost grid, we used the point $[0\text{m}, \$c]$ as an initial desired observation (constant with respect to temperature). We then updated this initial desired observation, using the rough estimate from preliminary CDO, to be a point which lies near the Pareto front, to improve identifiability. Here as in Section 4.3, "near" was taken to be 0.2 units on the standardized scale of model outputs.

The result of this strategy is similar to that one can derive a response surface over the $d-1$-dimensional grid of outputs, where that response describes the optimal results across the grid. If the grid covers the Pareto front in $d-1$ dimensions, then the result of the strategy is to provide a response surface with included uncertainty quantification describing, for each point in the grid, the optimal achievable outcome for the output not included in the grid. Thus a decisionmaker can visualize the space of desirable possibilities with associated uncertainty metrics. They can do so without the need for antecedently rigorously determining their exact priorities for weighing gains in each of the outputs against one another, nor (much worse) working out how to translate those priorities into specific choices of desired observations and observation variance schemes.

The result of applying this strategy to the wind turbine blade application is shown in Figure 5. The rightmost plot is included to verify that the posterior model output respected the d-known cost values used in the calibrations.

5 Conclusion

In this work we have described the theoretical background for the use of Gaussian processes to emulate computationally expensive computer model code, and the use of such emulators for computer model calibration under the framework established principally by Kennedy and O'Hagan (2001), Williams et al. (2006) and Bayarri et al. (2007b). CDO is a modification of that framework which calibrates a computer model, not to field observations, but rather to desired observations, i.e., performance targets for the system. Unlike other methods of Bayesian optimization (for an overview of which see Shahriari et al. 2016), CDO does not require the ability to carry out computer model observations adaptively. Instead, it can operate using a batch of observations gathered prior to (and independently of) the calibration procedure. We described the implementation of this approach in an MCMC routine along with considerations to accommodate computational instability.

The use of this methodology is illustrated in the case of material design for a wind turbine blade. We have shown thereby a variety of ways in which CDO can be used to produce a guide that decision-makers can consult in the design process. By expropriating established tools of model calibration, CDO offers a method of optimization which is sensitive to all sources of uncertainty, and which results in an estimate that includes uncertainty quantification.

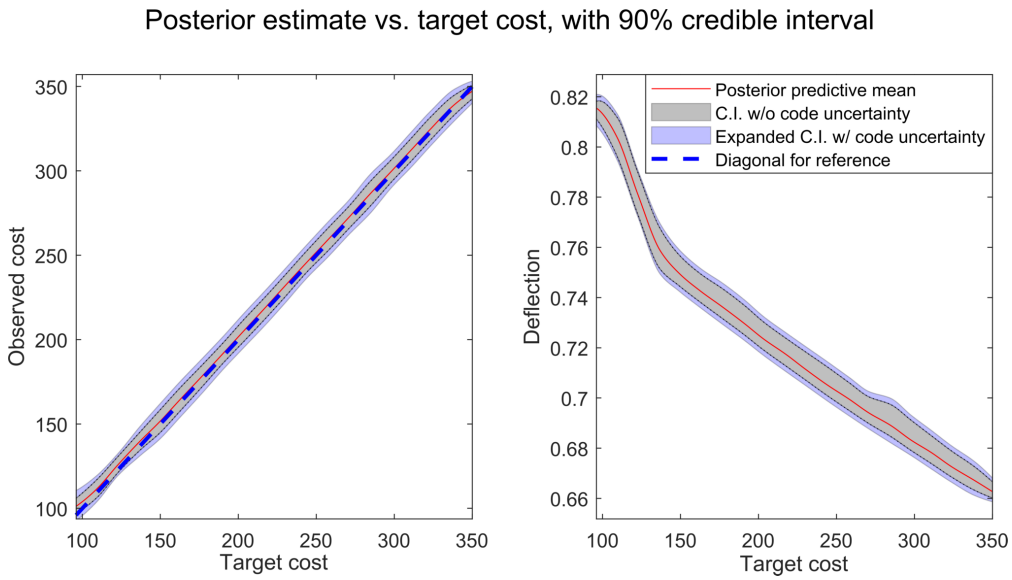

Figure 5: "Pareto bands" of wind turbine blade posterior cost and tip deflection across a range of target costs. The gray region gives a 90% credible interval only considering parameter uncertainty; the blue region extends this to include code uncertainty.

References

Bastos, L. S. and O'Hagan, A. (2009). Diagnostics for Gaussian Process Emulators. *Technometrics*, 51(4):425–438.

Bayarri, M. J., Berger, J. O., Cafeo, J., Garcia-Donato, G., Liu, F., Palomo, J., Parthasarathy, R. J., Paulo, R., Sacks, J., and Walsh, D. (2007a). Computer Model Validation with Functional Output. *The Annals of Statistics*, 35:1874–1906.

Bayarri, M. J., Berger, J. O., Paulo, R., Sacks, J., Cafeo, J. A., Cavendish, J., Lin, C.-H., and Tu, J. (2007b). A Framework for Validation of Computer Models. *Technometrics*, 49(2):138–154.

Brown, D. A. and Atamturktur, S. (2018). Nonparametric Functional Calibration of Computer Models. *Statistica Sinica*, 28:721–742.

Brynjarsdóttir, J. and O'Hagan, A. (2014). Learning about physical parameters: The importance of model discrepancy. *Inverse Problems*, 30(11).

Cauchy, A. (1847). Méthode générale pour la résolution des systèmes d'équations simultanées. *Comptes Rendus Hebdomadaires des Séances de L'Académie des Sciences*, 25(July-December):536–538.

Gramacy, R. B. and Lee, H. K. H. (2008). Bayesian Treed Gaussian Process Models With an Application to Computer Modeling. *Journal of the American Statistical Association*, 103(483):1119–1130.

Gramacy, R. B. and Lee, H. K. H. (2012). Cases for the nugget in modeling computer experiments. *Statistics and Computing*, 22(3):713–722.

Higdon, D., Kennedy, M., Cavendish, J. C., Cafeo, J. A., and Ryne, R. D. (2004). Combining Field Data and Computer Simulations for Calibration and Prediction. *SIAM Journal on Scientific Computing*, 26(2):448–466.

Kennedy, M. C., Anderson, C. W., Conti, S., and O'Hagan, A. (2006). Case studies in Gaussian process modelling of computer codes. *Reliability Engineering & System Safety*, 91(10-11):1301–1309.

Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464.

Liu, F., Bayarri, M. J., and Berger, J. O. (2009). Modularization in Bayesian analysis, with emphasis on analysis of computer models. *Bayesian Analysis*, 4(1):119–150.

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, 21(2):239–245.

O'Hagan, A. (1978). Curve Fitting and Optimal Design for Prediction. *Journal of the Royal Statistical Society. Series B*, 40(1):1–42.

Plumlee, M. (2017). Bayesian Calibration of Inexact Computer Models. *Journal of the American Statistical Association*, 112(519):1274–1285.

Pratola, M. and Chkrebtii, O. (2018). Bayesian Calibration of Multistate Stochastic Simulators. *Statistica Sinica*, 28:693–719.

Qian, P. Z. G., Wu, H., and Wu, C. F. J. (2008). Gaussian Process Models for Computer Experiments With Qualitative and Quantitative Factors. *Technometrics*, 50(3):383–396.

Ranjan, P., Haynes, R., and Karsten, R. (2011). A Computationally Stable Approach to Gaussian Process Interpolation of Deterministic Computer Simulation Data. *Technometrics*, 53(4):366–378.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and Analysis of Computer Experiments. *Statistical Science*, 4(4):409–423.

Santner, T. J., Williams, B. J., and Notz, W. I. (2003). *The Design and Analysis of Computer Experiments*. Springer, New York.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. (2016). Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104(1):148–175.

Tuo, R. and Jeff Wu, C. F. (2015). Efficient calibration for imperfect computer models. *Annals of Statistics*, 43(6).

Tuo, R. and Jeff Wu, C. F. (2016). A Theoretical Framework for Calibration in Computer Models: Parametrization, Estimation and Convergence Properties. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):767–795.

Williams, B., Higdon, D., Gattiker, J., Moore, L., McKay, M., and Keller-McNulty, S. (2006). Combining experimental data and computer simulations, with an application to flyer plate experiments. *Bayesian Analysis*, 1(4):765–792.