Fourth Exam: Computer model calibration as a method of design

Carl Ehrett

June 27, 2018

1 Introduction

1.1 Computer experiments

Suppose that one wishes to improve one's understanding of, say, the movement of people in a crowd escaping from a building in a crisis situation. This is an example of an area in which field data, i.e. real-world observations, are extremely difficult to acquire. Merely assembling a crowd of research subjects in one place is costly and difficult. Asking them to flee a building may result in behaviors which are unlike those in real crisis situations – but which may nonetheless present unacceptable physical risk to the subjects. Inducing them to flee through the generation of a (real or apparent) crisis is similarly infeasible. Observational data are likewise scarce here, since panic-inducing crises are by their nature difficult to predict and chaotic in ways that hinder the orderly collection of data.

In the face of these difficulties, computer models offer a third alternative to the choice between attempting field data collection and giving up on the hope of progress. Using existing theory concerning human psychology and movement, it is possible to construct a computer model simulating the behavior of people evacuating from a large building. For example, the SIMULEX model described by Thompson and Marchant (1995) allows one to observe simulated evacuation behaviors. The user provides as input floor plans for each floor of the building (via CAD-based .dxf files), the locations of staircases and exits, the locations of occupants, the dimensions of their bodies, routes to exits, and a distribution on the response time of each occupant. The user can then observe how long it takes each occupant to exit the building. Thus, computer models provide a means to collect data which might otherwise be largely inaccessible.

The study of computer models from a statistical perspective calls for specialized tools and techniques. Gaussian processes (GPs) are popular for modeling the output of computer code when the code is too computationally expensive to allow for repeated evaluation. The GP then provides a computationally inexpensive emulator for the computer code. There are three reasons the popularity of GPs as emulators: (1) The use of a GP does not require detailed foreknowledge of the approximate parametric form of the computer model. Researchers often lack such foreknowledge in the case of complex computer models. (2) GPs easily interpolate the observed data. This is an advantage when the observations come from deterministic computer code that is free of observation error. (3) The variance of a GP provides a natural form of uncertainty quantification. A Bayesian approach to the study of computer models is undertaken by Currin et al. (1991); the authors approach GPs as prior distributions on the unknown form of the computer model. A frequentist application of GPs to computer models is provided by Sacks et al. (1989), who use GPs not only for estimating uncertainty but also as the basis for their approach to experimental design in the area of deterministic computer models. Santner et al. (2003) offer a comprehensive discussion of the use of GPs for prediction, design, and sensitivity analysis with respect to computer experiments from both frequentist and Bayesian perspectives. It is against the background of these works that the past two decades of research into computer model calibration takes place.

1.2 Computer model calibration

Suppose that we wish to use the SIMULEX model to compare two different proposed building codes to be enforced in a given area. We may use average body dimensions and average interpersonal distance as input parameters for this model, both to settle the initial physical distribution of people throughout the building

and to influence their behavior during evacuation. It is well-established that average body dimensions (Subramanian et al., 2011; Cavelaars et al., 2000) and interpersonal distance (Sorokowska et al., 2017) vary across locales. These values may be unknown for the case of a particular region. Thus we may wish to find the true values for average body dimensions and interpersonal distance in that region. We may wish, in other words, to *calibrate* these parameters in the model.

Broadly, in model calibration, we 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 (in the evacuation example, this would include the building layout). 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 $\mathbf{y}$ our observations 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 lies in the fact 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 arrive at a posterior distribution of the calibration parameters which both balances our prior knowledge about the calibration parameters with what can be learned from the available data and 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). That approach is further refined and exemplified by Williams et al. (2006). Loeppky et al. (2006) offer an MLE-based alternative to the Bayesian approach promulgated by Kennedy and O'Hagan, intending thereby to improve the identifiability of the calibration parameters in the face of model discrepancy. Bayarri et al. (2007b) extend the approach of Kennedy and O'Hagan, providing a framework that is intended to allow for simultaneous validation and calibration of a computer model (using the same training data). Though they work within the Bayesian framework set by Kennedy and O'Hagan, they show how a Bayesian analysis can be profitably attenuated through 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. Bayarri et al. (2007a) further the work of Bayarri et al. (2007b) by applying the latter's methodology to functional data. To do so, the methodology for dealing with scalar data is hierarchically applied to the coefficients of a wavelet representation of the functional data. Similarly, Paulo et al. (2012) focus on applying the lessons of Bayarri et al. (2007b) to computer models with multivariate output.

Above, I have described model calibration as a matter of estimating the true values of unknown parameters. Indeed, that is the sort of model calibration that will form the focus of the present work. However, in a broad sense of the term, model calibration takes a second form as well. For convenience, one may distinguish this second form by referring to it as *tuning*, rather than as calibration. In model tuning, a computer model includes inputs which must be calibrated, but which have no particular interpretation; i.e., tuning parameters do not represent the truth regarding some value of the system being modeled. The work of Brynjarsdóttir and O'Hagan (2014) emphasizes the distinction between calibration and tuning. They show that whereas a well-modeled discrepancy function ($\delta(\cdot)$ in (1)) may not be necessary for tuning, since the tuning process can accomplish much of the work of a discrepancy function, an accurate discrepancy function is crucial for calibration, for the same reason. Letting the calibration process take on the duties of the discrepancy function reduces the identifiability of the calibration parameters.

Another extension of the framework of Kennedy and O'Hagan (2001) and Bayarri et al. (2007b) is *state-aware* calibration (Atamturktur and Brown, 2015; Stevens et al., 2018; Brown and Atamturktur, 2018).

When one knows or suspects that the appropriate value of a calibration parameter $\boldsymbol{\theta}$ is dependent upon the other inputs x, state-aware calibration allows for the calibration procedure to estimate $\boldsymbol{\theta}$ as $\boldsymbol{\theta}(\mathbf{x})$. This may obviate the need for a discrepancy function, either because the true value of the calibration parameter does indeed vary with the other inputs, or (in a tuning problem) because varying the tuning parameter can accomplish the work of a discrepancy function in aligning the computer model response with reality.

1.2.1 Gaussian processes

In principle, model calibration[1] 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. 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. As described in Section 1.1, 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. In this section, I provide brief background on Gaussian processes and their use in regression broadly and in computer model calibration specifically.

Gaussian processes can be thought of as generalizations of multivariate normal distributions. Whereas a multivariate normal random variable is a random vector of finite length, a realization of a Gaussian process can be thought of as a random function. The value of the function at any finite selection of points is itself a multivariate normal random variable. Just as a multivariate Gaussian random variable is characterized by its mean vector and covariance matrix, a Gaussian process is fully characterized by its mean function $\mu : D \to \mathbb{R}$ and covariance function $C : D \times D \to \mathbb{R}$, where D is the domain of the process. Thus for any points $\mathbf{x}, \mathbf{y}$ in the domain of the Gaussian process, $\mu(\mathbf{x})$ gives the mean of the Gaussian process at $\mathbf{x}$, and $C(\mathbf{x}, \mathbf{y})$ gives the covariance between the values of the Gaussian process at points $\mathbf{x}$ and $\mathbf{y}$. As a special case, then, $C(\mathbf{x}, \mathbf{x})$ gives the marginal variance of the GP at $\mathbf{x}$.

O'Hagan (1978) suggested the use of GPs as priors for modeling unknown functions. Given observations $\mathbf{f}$ from the function f being modeled, an updated mean and covariance function $\mu^*(\mathbf{x}) = \mu(\mathbf{x})|\mathbf{f}$ and $C^*(\mathbf{x}, \mathbf{y}) = C(\mathbf{x}, \mathbf{y})|\mathbf{f}$ describe a new GP that has been "trained" on the observations $\mathbf{f}$. Though the approach of O'Hagan (1978) was Bayesian, it did not provide for Bayesian means of selecting a covariance function C. Neal (1998) extends the work of O'Hagan in this direction. Savitsky et al. (2011) describes a method for variable selection in GP regression via spike-and-slab mixture priors on the parameters of the covariance function. Shang and Chan (2013) provide an umbrella generalization of existing methods of GP regression under a framework of what they call generalized GP models. Rasmussen et al. (2006) provide a comprehensive exploration of the use of GPs for regression and classification. All of these works treat GP regression generally, and do not focus specifically on the use of GPs to build emulators for computer models.

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 $\boldsymbol{\eta} = (\eta(\mathbf{x}_1), \cdots, \eta(\mathbf{x}_n))^T$. Much recent attention in the wake of these efforts by Santner et al. and Sacks et al. has focused specifically on the use of GPs for computer model emulation. 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.

[1] In this work, "calibration" will usually be used broadly to include model tuning; where it is used in such a way as to exclude tuning, this will be made clear in context.

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.

As in any instance of GP regression, "training" a GP as a computer model emulator involves finding an updated mean and covariance function given a set of observations. The updated mean allows for prediction of the computer model at points not observed, with uncertainty quantified by the updated covariance function. Suppose, for example, that we begin with a prior GP with constant mean function $\mu(\mathbf{x}) = 0$ for all $\mathbf{x}$. Notice that we may 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 $C(\mathbf{x}_i, \mathbf{x}_j)$. We may wish to train our GP on these n observations, and then examine the resulting posterior GP given the points $\mathbf{X}' = (\mathbf{x}'_1, \ldots, \mathbf{x}'_m)^T$. Recall that the GP at the points $\mathbf{X}'$ is a multivariate normal random variable of length m, which is fully characterized by its mean vector $\mu^*_{\mathbf{X}'}$ and covariance matrix $\mathbf{C}^*_{\mathbf{X}',\mathbf{X}'}$. We can find these as:

$$\begin{aligned} \mu^*_{\mathbf{X}'} &= \mathbf{C}_{\mathbf{X}',\mathbf{X}} \cdot \mathbf{C}^{-1}_{\mathbf{X},\mathbf{X}} \cdot \boldsymbol{\eta} \\ \mathbf{C}^*_{\mathbf{X}'} &= \mathbf{C}_{\mathbf{X}',\mathbf{X}'} - \mathbf{C}_{\mathbf{X}',\mathbf{X}} \cdot \mathbf{C}^{-1}_{\mathbf{X},\mathbf{X}} \cdot \mathbf{C}_{\mathbf{X},\mathbf{X}'} \end{aligned} \tag{2}$$

A visualization of this example is displayed in Figure 1. Here, $n = 5$, and $x' = (-5, -4.75, \ldots, 4.75, 5)$.

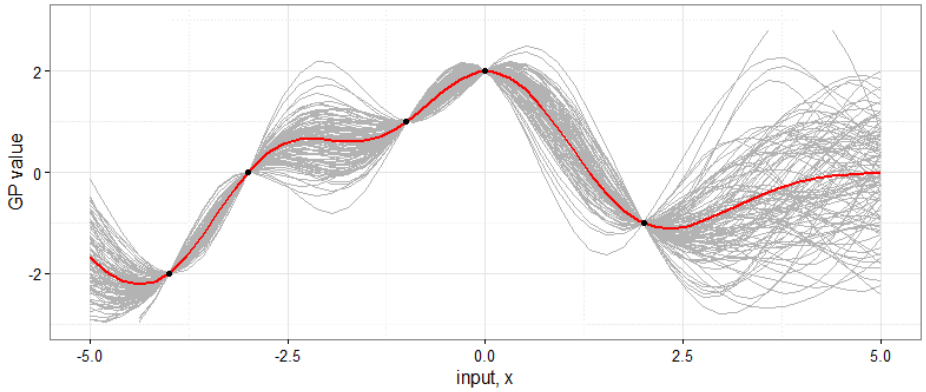

Figure 1: Example of a Gaussian process trained to interpolate five data points (black dots). The posterior mean function is shown in red; the gray lines are 50 draws from the posterior GP.

Aside from Kennedy and O'Hagan (2001), recent applications of GP emulation specifically to problems of calibration have focused largely on the works of Williams et al. (2006) and Bayarri et al. (2007b). It is helpful here to provide an illustrative summary of the approach taken by Williams et al. (2006), both to exemplify the use of GPs for computer model calibration and because the approach utilized in the present work closely follows theirs.

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

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

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

$$y(\mathbf{x}_i) = \eta(\mathbf{x}_i, \boldsymbol{\theta}) + \delta(\mathbf{x}_i) + \epsilon(\mathbf{x}_i), \quad i = 1, \cdots, n \tag{4}$$

where $\eta(\cdot, \cdot)$ is the computer model, $\boldsymbol{\theta}$ is the best[2] setting 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

[2]In the case of calibration, the "best" setting will be the true setting of that parameter; in a case of tuning rather than calibration, the "best" setting would instead be the optimal setting for minimizing model bias.

correlation function:

$$C((\mathbf{x},\mathbf{t}),(\mathbf{x}',\mathbf{t}')) = \frac{1}{\lambda_\eta}\prod_{k=1}^{p}\exp\left(-\beta_k^\eta|x_k - x_k'|^{\alpha_\eta}\right) \times \prod_{k=1}^{q}\exp\left(-\beta_{p+k}^\eta|t_k - t_k'|^{\alpha_\eta}\right) \tag{5}$$

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:

$$\begin{aligned} c &\sim N(0,v) \\ \lambda_\eta &\sim \text{Gamma}(5,5), \quad \lambda_\eta > 0 \\ \rho_k^\eta &\sim \text{Beta}(1,0.1), \quad k = 1,\cdots,p+q \end{aligned} \tag{6}$$

where $\rho_k^\eta = \exp(-\beta_k^\eta/4)$ for $k = 1,\cdots,p+q$. 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

$$C_\delta(\mathbf{x},\mathbf{x}') = \frac{1}{\lambda_\delta}\prod_{k=1}^{p}\exp\left(-\beta_k^\delta|x_k - x_k'|^{\alpha_\delta}\right), \tag{7}$$

with priors

$$\begin{aligned} \lambda_\delta &\sim \text{Gamma}(a_\delta, b_\delta) \\ \rho_k^\delta &\sim \text{Beta}(1,0.3). \end{aligned} \tag{8}$$

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

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

$$\mathcal{D}|\boldsymbol{\theta},c,\lambda_\eta,\boldsymbol{\beta}^\eta,\lambda_\delta,\boldsymbol{\beta}^\delta,\mathbf{C}_\mathbf{y} \sim N(c\cdot\mathbf{1}_{n+m},\mathbf{C}_\mathcal{D}) \tag{9}$$

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 $n < i,j \le n+m$, and $\mathbf{C}_\mathcal{D}$ is a matrix with its i,j entry equal to

$$C((\mathbf{x}_i,\mathbf{t}_i),(\mathbf{x}_j,\mathbf{t}_j)) + I(i,j>n)\cdot(C_{obs}(\mathbf{x}_i,\mathbf{x}_j) + C_\delta(\mathbf{x}_i,\mathbf{x}_j)) \tag{10}$$

Thus, the joint posterior density under the model is

$$\pi(\boldsymbol{\theta},c,\lambda_\eta,\boldsymbol{\rho}^\eta,\lambda_\delta,\boldsymbol{\rho}^\delta,\mathbf{C}_\mathbf{y}|\mathcal{D}) \propto \pi(\mathcal{D}|\boldsymbol{\theta},c,\lambda_\eta,\boldsymbol{\beta}^\eta,\lambda_\delta,\boldsymbol{\beta}^\delta,\mathbf{C}_\mathbf{y})\times\pi(c)\times\pi(\lambda_\eta)\times\pi(\boldsymbol{\rho}^\eta)\times\pi(\lambda_\delta)\times\pi(\boldsymbol{\rho}^\delta) \tag{11}$$

Note that where a discrepancy function is not included in the model and the mean c is treated as a constant, (11) simplifies greatly; where furthermore λ_η and $\boldsymbol{\rho}^\eta$ are estimated via maximum likelihood (as in Kennedy and O'Hagan (2001)), (11) 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 (11). The next section takes up this topic.

1.2.2 Markov chain Monte Carlo methods

The central idea of Markov chain Monte Carlo (MCMC) integration is to construct a Markov chain which has as its equilibrium distribution the target distribution one wishes to explore. The Markov chain is observed, and beyond an initial "burn-in" period during which the chain is allowed to approach its equilibrium distribution, samples are considered to be drawn approximately from the target distribution.

For example, consider a model with posterior distribution given by (11), but where discrepancy is not included and $\lambda_\eta,\boldsymbol{\rho}^\eta$ are found via maximum likelihood estimation. Then the full distribution is given by $\pi(\boldsymbol{\theta},c,\lambda_\eta,\boldsymbol{\rho}^\eta,\lambda_\delta,\boldsymbol{\rho}^\delta|\mathcal{D}) \propto \pi(\mathcal{D}|\boldsymbol{\theta},c,\lambda_\eta,\boldsymbol{\beta}^\eta,\lambda_\delta,\boldsymbol{\beta}^\delta,\mathbf{C}_\mathbf{y})\times\pi(c)$. A simple means of exploring this distribution

via MCMC would begin with an initial guess $\boldsymbol{\theta}^{(1)}, c^{(1)}$. At the i^{th} step for $i = 2, \ldots$, using a proposal distribution $q(\cdot, \cdot|\boldsymbol{\theta}^{(i-1)}, c^{(i-1)})$ from which one may easily sample directly, one draws a new "proposed" sample $\boldsymbol{\theta}^*, c^*$. One then accepts this new proposed sample, setting $(\boldsymbol{\theta}^{(i)}, c^{(i)}) = (\boldsymbol{\theta}^*, c^*)$, with probability

$$\alpha = \frac{\pi(\boldsymbol{\theta}^*, c^*|\mathcal{D})}{\pi(\boldsymbol{\theta}^{(i-1)}, c^{(i-1)}|\mathcal{D})} = \frac{\pi(\mathcal{D}|\boldsymbol{\theta}^*, c^*, \lambda_\eta, \boldsymbol{\beta}^\eta, \lambda_\delta, \boldsymbol{\beta}^\delta, \mathbf{C}_\mathbf{y}) \times \pi(c^*)}{\pi(\mathcal{D}|\boldsymbol{\theta}^{(i-1)}, c^{(i-1)}, \lambda_\eta, \boldsymbol{\beta}^\eta, \lambda_\delta, \boldsymbol{\beta}^\delta, \mathbf{C}_\mathbf{y}) \times \pi(c^{(i-1)})}. \tag{12}$$

Otherwise one rejects the proposed sample and let $(\boldsymbol{\theta}^{(i)}, c^{(i)}) = (\boldsymbol{\theta}^{(i-1)}, c^{(i-1)})$. In defining α in this way, it is assumed that q is a symmetric distribution. The version of MCMC described in this example is known as the Metropolis algorithm. It was initially described by Metropolis et al. (1953). Hastings (1970) generalizes the technique (to what is now called the Metropolis-Hastings algorithm) so that it may utilize non-symmetric proposal distributions. A thorough exposition of the technique, its theoretical foundation, and its relation to other varieties of MCMC is provided by Chib and Greenberg (1995).

A related variant of MCMC – in fact, a special case of Metropolis-Hastings, as shown by Gelman (1992) – is Gibbs sampling, initially described by Geman and Geman (1984). In using Gibbs sampling, one finds the conditional distribution of each of the parameters one wishes to sample, using the most recent samples of all other parameters. For example, we can convert the above Metropolis-Hastings illustration to a simple case of Gibbs sampling as follows. Begin as before, with initial guesses $\boldsymbol{\theta}^{(1)}$ and $c^{(1)}$. Thereafter, sample $\boldsymbol{\theta}$ and c not together, but rather in alternating draws. That is, at the i^{th} step for $i = 2, \cdots$, draw $\boldsymbol{\theta}^{(i)}$ conditional on $c = c^{(i-1)}$, then draw $c^{(i)}$ conditional on $\boldsymbol{\theta} = \boldsymbol{\theta}^{(i)}$. Thus, for example, when drawing $c^{(i)}$, its conditional distribution would be proportional to $\pi(\mathcal{D}|\boldsymbol{\theta}^{(i)}, c^*, \lambda_\eta, \boldsymbol{\beta}^\eta, \lambda_\delta, \boldsymbol{\beta}^\delta, \mathbf{C}_\mathbf{y}) \times \pi(c^*)$. If the conditional distribution is something from which we may easily draw directly, then we can draw $c^{(i)}$ that way; otherwise, we can use a so-called Metropolis-within-Gibbs scheme, and draw $c^{(i)}$ similarly to what was described above for the Metropolis-Hastings algorithm: draw a proposed c^* from a proposal distribution, find the appropriate ratio α_c, and accept or reject c^* accordingly. In the application considered in the present work, Metropolis-within-Gibbs is used.

Recall that a symmetric proposal distribution is required in order for the Metropolis algorithm to proceed as described in the above illustration of that technique. However, the algorithm can accommodate an asymmetric proposal distribution with only slight complication. One reason for using an asymmetric proposal distribution is as a means of accommodating boundary constraints on the parameter being sampled. Thus, for example, in the application considered in the present work, one of the calibration parameters is the thickness (in mm) of the material forming a wind turbine blade. Expert opinion was used to set the support of this parameter to be the range [10mm, 25mm]. Efficient exploration of the parameter space in the MCMC chain is promoted by choosing a proposal density such that the distribution $q(\cdot|\tau)$ has mean τ. An asymmetric distribution can be used to accommodate the boundary constraints. When $q(\cdot|\tau)$ is asymmetric, this changes the above illustration only by way of requiring us to calculate the acceptance probability α as follows:

$$\alpha = \frac{\pi(\boldsymbol{\theta}^*, c^*|\mathcal{D})}{\pi(\boldsymbol{\theta}^{(i-1)}, c^{(i-1)}|\mathcal{D})} \times \frac{q(\boldsymbol{\theta}^{(i-1)}, c^{(i-1)}|\boldsymbol{\theta}^*, c^*)}{q(\boldsymbol{\theta}^*, c^*|\boldsymbol{\theta}^{(i-1)}, c^{(i-1)})}. \tag{13}$$

1.2.3 Computational difficulties

Note that the joint posterior density given in (11) above includes $\mathbf{C}_\mathcal{D}$, which, depending on how many field and simulation observations one has, may be of prohibitively high dimension. This can lead to computational difficulties in calculating α in the course of the MCMC routine. Difficulties arise in two ways. Firstly, the likelihood given in (11) evaluated at a given point can be so small as to be vulnerable to significant round-off error. Secondly, the poor conditioning of $\mathbf{C}_{\mathbf{X},\mathbf{X}}$ in (2) 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. The latter problem can be alleviated by adding a small nugget to $\mathbf{C}_{\mathbf{X},\mathbf{X}}$; i.e., we can set $\mathbf{C}^\xi_{\mathbf{X},\mathbf{X}} = \mathbf{C}_{\mathbf{X},\mathbf{X}} + \xi \cdot \mathbf{I}_{\dim(\mathbf{X})}$ for some very small value of ξ, e.g., $\xi = 10^{-4}$. Such a simple nugget works quite well in many applications, but for a more sophisticated approach to selecting the nugget size, 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. Thus nothing is lost in continuing to think of the GP emulator as interpolating the simulation observations.

The other problem – round-off error due to small likelihoods – can be alleviated through substituting the use of log-likelihoods in the MCMC routine. Thus, rather than finding α in (13) directly, it is preferable to find

$$\log \alpha = \log \pi(\boldsymbol{\theta}^*, c^* | \mathcal{D}) + \log q(\boldsymbol{\theta}^{(i-1)}, c^{(i-1)} | \boldsymbol{\theta}^*, c^*) - \log \pi(\boldsymbol{\theta}^{(i-1)}, c^{(i-1)} | \mathcal{D}) - \log q(\boldsymbol{\theta}^*, c^* | \boldsymbol{\theta}^{(i-1)}, c^{(i-1)}) \quad (14)$$

Of course, in order to enjoy the computational benefits of this approach, one must perform calculations on the log scale from the start; it would not do to, e.g., find $\pi(\boldsymbol{\theta}^*, c^* | \mathcal{D})$ on its original scale and then take its log.

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. An example would be selecting a building layout conducive to efficient evacuation, as modeled using SIMULEX.

We may approach such problems as a matter of calibration. In traditional calibration as described in Section 1, a computer model is calibrated to observations of reality. This is done in order to find settings for the computer model that induce its output to match reality. 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).

Of course, computer models are more malleable than reality, and it is trivial to modify a computer model so that its output matches any given target. It is both easy and pointless to create a model which is a computational "yes man". But in many cases one is fortunate to have (perhaps after undertaking traditional model calibration, validation and verification) a computer model such that one is independently confident that the model is uniformly valid over a given set $\mathcal{T}$ of controllable parameters t, i.e., the model is known to be faithful to reality over $\mathcal{T}$. 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. Instead, one finds 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 in this new domain. 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 especially saliently in this case, that which in traditional calibration would be considered either observation error or model inadequacy. Of course, our targets are not actually observations, and the concept of observation error does not cleanly transfer here. But a relevantly similar uncertainty would be uncertainty over 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.

2.1 Target observations

2.1.1 Level of target data

Unlike in the case of field observations, when calibrating to target observations, the question arises of determining what exactly one's desired targets are. In many cases, no objectively natural target manifests itself. Consider again the case of building evacuation. In the case of a multi-story building, one might plausibly expect to achieve evacuation times of no less than fifteen minutes. But plausibility is no barrier to desire, and it would be a mistake to limit one's target observations to what is antecedently believed to be achievable, if only because to do so would foreclose on the possibility of exceeding those expectations.

In the case of building evacuation, then, one might conclude that the appropriate target observation is in fact instantaneous evacuation – *per impossible*. But, having discarded realism, even this lower bound is not inevitable. Why not calibrate to a *negative* evacuation time, while we're at it?

Such a choice of target observation would indeed be consistent with the method of calibrating to desired observations, and in certain situations may even be appropriate. However, in general, target observations should aim only a little beyond what is realistically achievable, i.e., only as much as is necessary to ensure the targets are at least as ambitious as any true optimum in the system. I described above why it is preferable to go beyond what is achievable. There are two reasons why one should go only a little beyond that. (1) If target observations are set to be too farfetched, then (depending on the calibration settings used; see Section 4) the calibration can become unnecessarily computationally unstable due to underflow and round-off error, since any value of $\boldsymbol{\theta}$ within its support will have extremely low likelihood. (2) The desired observations lose a measure of interpretability when they delve too far into the fantastical, such as with negative evacuation times. Identifying the appropriate range of outputs for desired observations, which exceed reality but slightly, will of course often require one to consult expert opinion.

2.1.2 Set target via desired observations, or prior distribution?

In undertaking model calibration to achieve desired system output, directly setting desired observations is not the only option. For example, in the application considered below in Section 3, the calibration parameters are controllable features of material design, and the cost of the resulting material is one of the outputs for which I establish targets. One approach is simply to include a desired observation of cost, and calibrate to that along with the other desired observation targets. An alternative approach is to remove cost from the model, and place a prior distribution on the remaining calibration parameters that places low probability over those regions of the calibration parameter space for which the cost is high. Of course, this requires prior knowledge of the behavior of cost over the calibration parameter space. Without such knowledge, this alternative would be unavailable.

A third option is also explored in Section (3)'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 (e.g.) cost in the above 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. Since it is antecendently unknown which cost settings are optimal, under this third option I calibrate performance targets under each point of a grid of "known" costs. Thus I 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 desired observations have low likelihood with respect to the posterior predictive distribution of the calibration process. This is another way in which CDO 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 this is a sign that the the desired observations may have been insufficiently ambitious. In some applications, this concern about setting insufficiently ambitious desired observations will not arise. In general, this concern applies whenever one's target output from a system is not limited to what is realistically achievable. In the building design example, we would like to see instantaneous evacuation; in the material design case, we would like to see zero-cost materials that do not deform at all under load. These targets are unattainable, but we wish to induce the system to approach these targets, and so these impossibilities become our desired observations. However, another type of calibration to desired observation may involve searching within the space of what is achievable. In a different application than the one considered in this work (wind turbine blades), it might be ideal to have a material that deforms just so – rather than a material which does not deform under load. 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 and building evacuation systems, finding the desired observations to be the posterior mode would be an indication that the desired observation could potentially be outperformed, or else that the model is itself

unrealistic. In short, if the system can achieve the desired observations, then either the desired observations are realistically achievable (hence insufficiently ambitious) or else the desired observations are not realistically achievable (hence casting doubt on a model which presents them as achievable).

Where the mean of the posterior GP 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, or $\epsilon(\cdot)$ in (1). This is usually taken so that $\epsilon(\mathbf{x}) \equiv \epsilon$ does not depend on $\mathbf{x}$. Note that 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 above-described 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 – see (2.2.1) and (2.2.2) below. Nor does it quite fit even to call this gap "error", or a form of model discrepancy. Even under CDO, the model still describes *reality*, not our desires. Thus failure to interpolate our desires is not error. Though for convenience and ease of exposition I will often slip back into referring to this gap as "error", we can more properly refer to it as "model shortcoming". This is still somewhat infelicitous insofar as it still implies failure on the part of the model, whereas in fact this gap is due to the stubbornness of *reality* in declining to behave according to our desires. The model underperforms with respect to our targets insofar as reality does so. Still, the term is appropriate, since the "error" observed is a discrepancy between the desired observations and the model, not between the model and the true system.

2.2.1 Observation error

Model shortcoming can be accommodated by treating it as observation error $\epsilon(\cdot)$. This approach is flexible and has a number of advantages. First, this approach allows one to specify "known" observation variance. In the framework of CDO, this amounts to specifying how strongly the posterior GP should be drawn to the desired observations. This would be especially useful in the sort of case, described above, in which one hopes the posterior GP to interpolate the desired data. Even in the more ambitious case when the desired observations are realistically unattainable, there is great flexibility to be exercised in selecting observation variances directly. One can thereby, for example, set priorities in one's targets. For example, in the application of Sections 3 and 4, there are three outputs: wind turbine blade tip deflection, rotation, and cost. A plausible scenario for design would be that keeping costs low might be a much greater priority than minimizing deflection and rotation. This can be achieved simply by setting the observation variance of cost to be lower than that of deflection and rotation.

However, one might antecedently have little information about just how close the model can come to the desired observations. This is especially true when multivariate output is considered. The model may be able to approach some elements of the desired observation much more closely than others. Without prior knowledge, it can thus be difficult to specify an appropriate setting for the observation variance. Too low, and computational difficulties may arise from low likelihoods. Too high, and the model may be insufficiently "incentivized" to approach the targets. In such situations, one can simply place a prior on the observation variances, allowing the simulation data to inform us as to how achievable the desired observations are. In Section 4 I implement this, showing that it leads to better performance with respect to the desired observations than specifying a static observation variance.

Finally, these two methods can be combined. One of the approaches used in Section 4 places a prior over the observation variance of deflection and rotation, but specifies a very low observation variance for cost, over a grid of (realistically achievable) desired cost observations, each of which is paired with (unrealistically ambitious) desired observations of zero deflection and rotation. The net result of this is to find distributions of optimally performing materials over a grid of set costs, so that one can see the performance outcomes as variables of a choice of cost over a broad range.

2.2.2 Model discrepancy

A separate means of incorporating model shortcoming would be to treat it as model discrepancy rather than as observation error. Often, model discrepancy $\delta(\cdot)$ is itself modeled as a mean-zero GP; see e.g. Williams

et al. (2006). This has the advantage that it facilitates treating model shortcoming as a function of the control input x, which will often be the case. The deviance of our desired observations from the mean is after all not mere random noise. Performance targets deviate from the true system mean in a systematic but unknown way. Thus model shortcoming is more accurately represented by a discrepancy function than as random observation error.

Disadvantages are that, in contrast with the use of observation error, it is much less convenient to fine-tune the model discrepancy function to the details and priorities of a particular situation. For example, like Williams et al. (2006), in the present work dummy variables are used to construct a univariate Gaussian emulator for our three model outputs. If we now wish to either specify different variances for the three outputs or to allow the data to discover different such variances (as suggested above for observation variances) via a model discrepancy function, we will not be able to use a stationary GP to model that discrepancy. That is, the covariance function for the discrepancy GP, $C_\delta(\mathbf{x}, \mathbf{x}')$, will depend not only on the distance between $\mathbf{x}$ and $\mathbf{x}'$, but on their location within the input space $\mathcal{X}$. This is an unattractive complication, but remains a live option.

2.3 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 calibration 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. Thus, using field data from building evacuations, a researcher might use traditional calibration to try to calibrate SIMULEX to the appropriate distribution on body dimensions – there treating building layout to be fixed (as the layout(s) of whichever buildings were evacuated in the field observations).

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.

As a first approximation, in the unlikely event that the calibration parameters coincide under the two sorts of calibration, it would be possible to undertake a single round of calibration to both field and desired observations. It would be necessary to take care with how to treat model shortcoming in this scenario. Attempting to capture it via the model discrepancy function would likely lead to confounding the true model discrepancy with the model shortcoming, undermining both calibration goals. A more hopeful route would be to use observation variance to incorporate the model shortcoming; this would allow one to specify a larger observation variance for desired observations than for field observations (and that only if these two sets of observations do not share locations in $\mathcal{X}$). Even this strategy would only mitigate the confounding of the two calibration goals, and should only be used if the budget for simulation observations is so small as to make it infeasible to perform two separate calibrations.

2.4 Hyperparameter estimation

Consider the covariance function parameters $\lambda_\eta, \boldsymbol{\beta}^\eta$ in (5). 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, Liu et al. (2009) advocate modularization, described in Section 1.2 above. 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. In essence, attempting a full Bayesian analysis that finds these hyperparameter settings as part of CDO would be committing the same sort of error as described in the previous section: namely, 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 3 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 Application

In this and the following sections I 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 selecting 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 I 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[3] , which is treated as an accurate representation of reality. In this section I describe the emulator, and in Section 4 I describe its use for the calibration procedure.

3.1 Project background

Two primary performance targets for the design and construction of wind turbine blades is 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* – 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.

[3]The finite element code was authored by Evan Chodora.

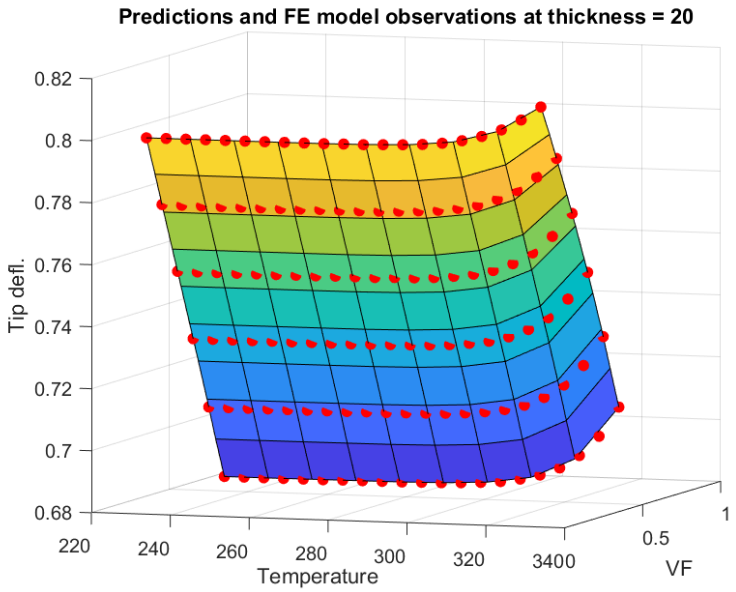

Figure 2: A slice of the GP emulator mean (restricted to the output for tip deflection) at thickness = 20mm. Red dots are observations from the simulator.

3.2 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 I employ a GP emulator in the manner of Williams et al. (2006). For this purpose, 504 observations were drawn 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.

I consider the finite element observations to follow a GP with mean 0 and covariance function C as described by (5) above, with $\alpha_\eta = 2$. This is equivalent to assuming smooth, infinitely differentiable sample paths. I do not include a discrepancy function, per the considerations of Section 2.2.1.

The hyperparameters $\lambda_\eta, \boldsymbol{\beta}^\eta$ must be estimated. I do not estimate these values as part of a full, integrated Bayesian analysis. Instead, they are estimated prior to calibration to the desired observations, via maximum likelihood estimation, for the reasons discussed in Section 2.4. Initially, a grid optimization method was used: a grid of $\boldsymbol{\beta}^\eta$ values was used, finding at each point of the grid the likelihood of the simulation observations integrated over the support of λ_η. However, $\boldsymbol{\beta}^\eta$ is a five-dimensional vector, and a grid fine enough to be useful was too computationally burdensome to be feasible. Instead, a gradient descent method (Cauchy, 1847) was used to maximize the log-likelihood of the simulation observations over the joint (6-dimensional) support of $\boldsymbol{\beta}^\eta, \lambda_\eta$. The result is

$$\hat{\boldsymbol{\rho}}^\eta = (0.9358, 0.6509, 0.6736, 0.4797, 0.9673), \quad \hat{\lambda}_\eta = 0.0152 \tag{15}$$

where $\rho_k^\eta = \exp(-\beta_k^\eta/4)$. A slice of the resulting emulator mean (for thickness = 20mm) for the tip deflection output is shown in Figure 2.

4 MCMC using the emulator

4.1 The model

Following the framework laid out in Section 1.2.1 and the hyperparameters estimated in Section 3.2, the model takes the trained emulator to be distributed as

$$\mathcal{GP}\left(\mu^*(\mathbf{b}), C^*(\mathbf{b}, \mathbf{b}')\right) \tag{16}$$

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 (5), where I plug in the MLEs given in (15).

Generalizing from Kennedy et al. (2006), I expand the Bayesian analysis to include the diagonal observation variance matrix $\mathbf{C}_\mathbf{y}$, rather than requiring this value to be specified as known, and I allow for a non-uniform prior $\pi(\boldsymbol{\theta})$ on the calibration parameters. By (11) on page 5, since I estimate $\boldsymbol{\rho}^\eta, \lambda_\eta$ by maximum likelihood, set $c = 0$, and do not include a discrepancy function, we have for desired observations $\mathbf{y} = (y(b_{n+1}), \cdots, y(b_{n+m}))^T$ and $\mathcal{D} = (\mathbf{y}^T, \boldsymbol{\eta}^T)^T$:

$$\pi(\boldsymbol{\theta}, \mathbf{C}_\mathbf{y} | \mathcal{D}) \propto \pi(\mathcal{D} | \boldsymbol{\theta}, \mathbf{C}_\mathbf{y}) \times \pi(\boldsymbol{\theta}) \times \pi(\mathbf{C}_\mathbf{y}) \tag{17}$$

and $\mathcal{D} | \boldsymbol{\theta}, \mathbf{C}_\mathbf{y} \sim N(\mathbf{0}_{m+n}, \mathbf{C}_\mathcal{D})$,

$$\mathbf{C}_\mathcal{D} = \mathbf{C}_{\boldsymbol{\eta}} + \begin{bmatrix} \mathbf{C}_\mathbf{y} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{18}$$

where $\mathbf{C}_{\boldsymbol{\eta}}$ is a $(m+n) \times (m+n)$ matrix with i,j entry $C(b_i, b_j)$.

The model as described above leaves open several details of the model, options for which are explored in the remainder of Section 4. In 4.3 I consider options for $\pi(\mathbf{C}_\mathbf{y})$, the prior on observation variance, including setting a degenerate prior corresponding to specifying a known observation variance. In 4.4 I take up the question of which data one ought to desire, showing how the goals of one's analysis can support vastly different choices for desired data. In 4.5 I explore options for $\pi(\boldsymbol{\theta})$, the prior on the calibration parameters. I depart thereby from the framework of Kennedy et al. (2006), who (implicitly) restrict the calibration model to the case of using a uniform prior. An example is used to show how setting a non-uniform prior here can allow one to reduce the number of parameters that are subjected to calibration in the model. Before taking up these matters, in 4.2 I take up convergence difficulties that arise due to boundary constraints on the calibration parameters.

4.2 Convergence difficulties

As will often be the case in calibration problems, in the application considered here the calibration parameters (v, k) have compact support: $v \in [.1, .6]$, $k \in [10\text{mm}, 25\text{mm}]$. When the calibration procedure leads to draws near these boundaries, the MCMC routine may suffer poor mixing. For example, consider using proposal density q such that $(v^*, k^*) \sim N((v^{(i)}, k^{(i)}), \Sigma)$ for some proposal covariance Σ. The symmetry of a normal proposal density makes it convenient for MCMC, but also exacerbates the difficulties that come from boundary conditions. Using such a proposal density with $\pi(\boldsymbol{\theta})$ a uniform density on the support of $\boldsymbol{\theta}$ amounts to simply discarding any proposed draw of (v, k) that falls outside of the boundaries for those variables. This is inefficient, as it leads to low acceptance rates during MCMC for new draws of $\boldsymbol{\theta}$. When a new proposed draw is rejected, the previous draw is repeated, leading to extremely high levels of autocorrelation in the MCMC draws.

There are several ways to tackle this problem. One approach seeks to improve acceptance rates through utilizing an adaptive proposal covariance magnitude, so that Σ is periodically updated during the burn-in period in order to achieve an optimal acceptance ratio of around 23%; see Roberts et al. (1997). If Σ is diagonal, then this is especially simple to implement. One simply keeps track of the number of times that

a proposed draw of θ_i is outside of its boundaries, and then every 100 MCMC steps or so, if the number of out-of-boundary proposals exceeds a specified threshold, then one reduces Σ_i, the proposal variance of θ_i. This strategy can easily be paired with a more comprehensive adapative proposal covariance – e.g., one that is set to increase in magnitude whenever more than 30 of the most recent 100 draws have been accepted, and decrease in magnitude whenever fewer than 20 of the most recent draws have been accepted. If the posterior distribution is merely near the boundary, such measures may suffice; however, in calibration to desired observations (which tend to be extreme outliers *qua* observations), often the calibration parameters will be drawn strongly to the boundaries, and in such a situation the sort of adaptive proposal distribution described above may be insufficient to secure good acceptance ratios.

A second strategy is to use an independence chain approach (Tierney, 1994), in which the proposal density is independent of all previous draws. E.g., for compact support $\mathcal{S}$, one could simply draw new proposals as uniform over $\mathcal{S}$. When the posterior distribution is diffuse over its support, this can be effective. Otherwise, it can lead to low acceptance rates, as too many proposals are generated in low-density areas of the posterior distribution.

A third strategy would be to transform the variables to be unbounded. This requires describing the model in terms of these transformed variables. An adaptive covariance can be used here as well, further improving acceptance rates. A fourth, closely related strategy would be to employ some variety of proposal with the same support as the variables. Indeed, the difference between the third and fourth strategies is largely conceptual. Consider, for example, using the logit transform to remove boundary constraints from variables with compact support. Without boundary constraints, one can recover the convenience of a symmetric, Gaussian proposal distribution, without risk of proposing outside of the variables' support. But in addition to being viewed as a symmetric distribution on the transformed variables, this can equally be viewed as a non-symmetric distribution on the untransformed variables. I apply this logit transform to (v, k) in the wind turbine application, where for convenience I adopt the latter conceptual perspective.

The adaptive proposal strategy sketched above changes only the magnitude of the proposal covariance, and depends upon having a diagonal covariance matrix. This diagonality is what allows the easy modification of the magnitude of the variance of all and only those parameters which are being drawn outside of their boundary constraints. But where boundary constraints are no longer a problem, it is just as convenient to establish a non-diagonal proposal covariance. This will, of course, be of especially high value when the calibration parameters are correlated in the posterior distribution, which is frequently the case. Thus, when updating the proposal covariance, one can set it to be equal to the sample covariance of the unique samples accepted thus far in the MCMC routine.

If the acceptance ratio is low early in the MCMC when using this adaptive technique, the resulting proposal covariance may be too small. To promote more efficient exploration of the parameter space, the covariance can be multiplied by a constant τ which is also updated whenever the proposal covariance is updated. One may adjust τ so as to induce acceptance rates close to 23%. Thus, during the burn-in period, every n_0 draws (with n_0 set to, e.g., 100), the proposal covariance Σ and multiplier τ are updated to Σ^* and τ^* as follows, where a is the proportion of the past n_0 draws that were accepted, I is the indicator function, and $\mathbf{S}$ is the set of all samples drawn so far:

$$\begin{aligned} \tau^* &= \tau \cdot (1.25I(a > 25) + 0.75I(a < 20)) \\ \Sigma^* &= \tau^* \cdot \mathrm{Cov}(\mathbf{S}) \end{aligned} \tag{19}$$

The adaptation takes place only during the burn-in period. As a result, the adaptive nature of the algorithm does not change the fact that it is an implementation of the Metropolis-Hastings algorithm given in Hastings (1970). Hence it is guaranteed to converge to the target distribution. However, another option would be to employ the adaptive Metropolis schema of Haario et al. (2001, 2005, 2006), who continue to update the proposal covariance to be the sample covariance of all previous draws throughout the sampling procedure (up to a scaling constant which, unlike τ above, is not itself adaptive). This sampling method loses the Markov property and hence is not an MCMC method. Nonetheless the authors show it is ergodic under the conditions that the target distribution is bounded with bounded support.

Recall (from Section 1.2.2) that abandoning symmetry in one's proposal distribution requires that one employ the more general Metropolis-Hastings algorithm, rather than the Metropolis algorithm of Metropolis et al. (1953). Interpreting a normal distribution on the logit-transformed variables as a distribution on the

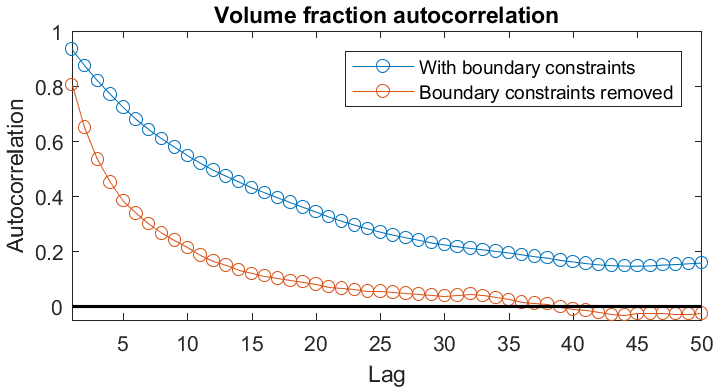

Figure 3: Auto-correlation for draws both with and without the elimination of boundary conditions.

original variables, it is not symmetric. Therefore, calculate the log acceptance probability a proposal for the i^{th} draw of $\boldsymbol{\theta}$ as

$$\log \alpha = \log(\boldsymbol{\theta}^*|\mathcal{D}, \mathbf{C_y}) - \log(\boldsymbol{\theta}^{(i-1)}|\mathcal{D}, \mathbf{C_y}) + \log q(\boldsymbol{\theta}^{(i-1)}|\boldsymbol{\theta}^*) - \log q(\theta^*|\boldsymbol{\theta}^{(i-1)}) \tag{20}$$

where the use of the log scale is required in order to avoid significant round-off and underflow errors due to the dimensionality of $\mathcal{D}$.

The benefits of the logit-transformed, unbounded proposal over a version of the MCMC using a normal proposal (with adaptive covariance) over the untransformed calibration parameters are displayed in Figure 3. Sizeable autocorrelation appears in the untransformed version, whereas in the logit-transformed version the autocorrelation drops off to insignificance within about 30 steps.

4.3 Desired observation variance

In the framework of Kennedy et al. (2006) on which much of the current approach is based, observation variance is assumed to be known. That assumption is less straightforward in the case of CDO. However, as explained below, there are applications in which setting a "known" observation variance is appropriate. In this section, I consider the option of specifying observation variance versus setting a (non-degenerate) prior on observation variance. I also consider the option of whether to set different observation variances for different outputs, or a single observation variance on all (standardized) outputs. Table 1 and the corresponding Figure 4 of MCMC results show how impactful are the choices amongst these options in the wind turbine blade design application. The posterior mean model output varies significantly across the four available strategies, which correspond to different specifications of observation variance settings. The table reflects the choices of (1) whether to specify that the observations are homoskedastic with respect to output type (e.g., whether standardized cost output has the same observation variance as standardized deflection output), and (2) whether the observation variance is set to a constant level or given a (nondegenerate) prior. From (17) above, the full models corresponding to these choices, assuming a uniform prior on the calibration parameters, are as follows. Columns one and two of the table both correspond to specifying $\boldsymbol{\sigma} = (\sigma_1^2, \sigma_2^2, \sigma_3^2)^T$ as the observation variance for each of the three outputs. The two columns differ only in the value of $\boldsymbol{\sigma}$. The prior $\pi(\mathbf{C_y})$ in (17) is thus the degenerate prior $\delta_{\boldsymbol{\sigma}}(\cdot)$, the Dirac delta function centered at $\boldsymbol{\sigma}$. Here, $\mathbf{C_y}$ is a diagonal matrix where the i, i entry is equal to σ_j^2 iff desired observation y_i is an observation of output j. E.g., if y_4 is an observation of rotation, then the fourth diagonal element of $\mathbf{C_y}$ is σ_2^2, the observation variance for rotation. So for these two columns the full model is given by:

$$\begin{aligned} \pi(\boldsymbol{\theta}, \mathbf{C_y}|\mathcal{D}) &\propto \pi(\mathcal{D}|\boldsymbol{\theta}, \mathbf{C_y}) \times \pi(\boldsymbol{\theta}) \times \pi(\mathbf{C_y}) \\ &\propto \pi(\mathcal{D}|\boldsymbol{\theta}, \mathbf{C_y}) \\ &\propto |\mathbf{C}_{\mathcal{D}}|^{-1/2} \exp(\mathcal{D}^T \mathbf{C}_{\mathcal{D}}^{-1} \mathcal{D}) \end{aligned} \tag{21}$$

where $\mathbf{C}_{\mathcal{D}}$ is given in (18), and where in the final expression I do not explicitly include the uniform $\pi(\boldsymbol{\theta})$ or the degenerate $\pi(\mathbf{C}_{\mathbf{y}})$. Columns three and four each place a prior distribution on the observation variances. Column three places independent priors $1/\sigma_i^2$ on each of the σ_i^2, $i = 1, 2, 3$, which serve as the observation variances of the three outputs. Column four places a single prior $1/\sigma_1^2$ on a scalar value σ_1^2 which serves as the observation variance for each of the three outputs. Thus the full models for these columns are:

$$\begin{aligned} \pi(\boldsymbol{\theta}, \mathbf{C}_{\mathbf{y}}|\mathcal{D}) &\propto \pi(\mathcal{D}|\boldsymbol{\theta}, \mathbf{C}_{\mathbf{y}}) \times \pi(\boldsymbol{\theta}) \times \pi(\mathbf{C}_{\mathbf{y}}) \\ &\propto \pi(\mathcal{D}|\boldsymbol{\theta}, \mathbf{C}_{\mathbf{y}}) \times \pi(\mathbf{C}_{\mathbf{y}}) \\ &\propto |\mathbf{C}_{\mathcal{D}}|^{-1/2} \exp(\mathcal{D}^T \mathbf{C}_{\mathcal{D}}^{-1} \mathcal{D}) \times \prod_{i=1}^{k} \frac{1}{\sigma_i^2} \end{aligned} \tag{22}$$

where for column three $k = 3$ and $\mathbf{C}_{\mathbf{y}}$ is defined as in columns one and two, and for column four $k = 1$ and $\mathbf{C}_{\mathbf{y}} = \sigma_1^2 \mathbf{I}_{m \times m}$. In both cases, the final expression does not explicitly include the uniform $\pi(\boldsymbol{\theta})$.

Notice in Table 1 that column three is noticeably distinct from the others. Here is the only case in which the MCMC is free to alter the *proportions* of observation variance allotted to the various outputs. In this case, the desired observation was essentially achieved for deflection and rotation, whereas the posterior cost was far from the desired cost. Thus, using the settings reflected in column three, we are able to learn that it is "easier" to achieve these two performance metrics than to achieve the cost target.

When CDO is used without a discrepancy function and with observation variance sampled under a prior, the posterior distribution on observation variance amounts to an indirect measure of how closely reality can approximate our desired observations. E.g., from the fact that posterior observation variance from the routine represented in column three of Table 1 was near zero, we see immediately that the cost target was achieved. Likewise, extremely high posterior observation variance indicates that the target could not be achieved. The posterior distribution reflects the (false) assumption that the difference between the desired observations and the true system mean is accounted for by random, mean-zero noise.

Matters are different when a set observation variance is specified for each output. Clearly, since the observation variance isn't sampled in the MCMC, there is no Bayesian learning to be had here. However, there is an opportunity to specify priorities through the specification of the variance. When the desired observations are outside the range of what is realistically achievable, the specific level of the observation variance is unlikely to matter much (unless it is so low as to engender computational difficulties). More important (in the case of multivariate output) is the fact that in specifying the observation variance, one is implicitly specifying the relative importance of each output. E.g., if I set a very diffuse observation variance on cost, and a very small one on deflection, then I have told the model that reduction in deflection is to be prioritized over reduction in cost. Whether one chooses to do so will depend on one's goals in calibration, as well as one's level of prior knowledge as to how close to reality one's desired observations are.

Such prioritization can be harnessed to achieve calibration goals in ways that will be explored in Section 4.5. There, one essentially sets the observation variance for all and only those parameters that one *ceases* to treat as calibration parameters. In general, since there is no true value of the observation variance, it is better to let the data inform us about $\mathbf{C}_{\mathbf{y}}$: setting a scale-invariant $1/\sigma_i^2$ prior on each diagonal element σ_i^2 of $\mathbf{C}_{\mathbf{y}}$ yields posterior values of $\mathbf{C}_{\mathbf{y}}$ that inform us as to how achievable our desired observations are.

There is another lesson to be derived from Table 1; namely, a warning that it is important to be aware of the impact of correlated outputs on the calibration procedure. Notice from the third column of the table that the posterior mean output almost achieves the desired observation for deflection and rotation, at the expense of falling wildly short on cost. However, it is known that the true model can achieve the desired cost of \$96/m^2 (for high levels of deflection and rotation). Part of what makes this outcome possible is that deflection and rotation are strongly positively correlated with each other, and negatively with cost. Thus when all three outputs are calibrated to be extremely low values, deflection and rotation will jointly pull against cost.

If one is antecedently unaware of the correlations among the responses, this can undermine one's calibration goals. E.g., if our goal was to balance cost with the performance metrics, then column three of the table fails to achieve this goal, due to the correlation amongst the performance metrics. For that goal, column four would have been a superior strategy, where $\mathbf{C}_{\mathbf{y}} = \sigma_1^2 \mathbf{I}_{m \times m}$ and σ_1^2 is allowed to vary under the $1/\sigma_1^2$ prior. This sets the observation variances of each output to be equal, but more nuanced versions of this strategy could set them to be scalar multiples of one another; in this way different balances could be achieved than

the one displayed in column four of Table 1. One may find, however, that one does not antecedently know what balance best suits one's desires. We want low cost, low deflection and low rotation, but exactly how much of each are we willing to trade for gains in the other? There may be no answer this question. It is for such situations that the strategy of Section 4.5 is proposed.

	Heteroskedastic, constant	Homoskedastic, constant	Heteroskedastic, prior	Homoskedastic, prior
Deflection	0.749	0.729	0.659	0.709
Rotation	0.0904	0.0865	0.0773	0.0843
Cost	276.16	236.11	350.80	233.95

Table 1: Comparison of posterior mean model outputs, where the desired data outputs are assumed to be either homoskedastic or heteroskedastic, with either a specified constant variance or a $1/\sigma^2$ prior. The desired observation was set to [0.65, 0.077, 96] for each control input, which is known to be the lowest achievable value in each of the three outputs (and not jointly achievable).

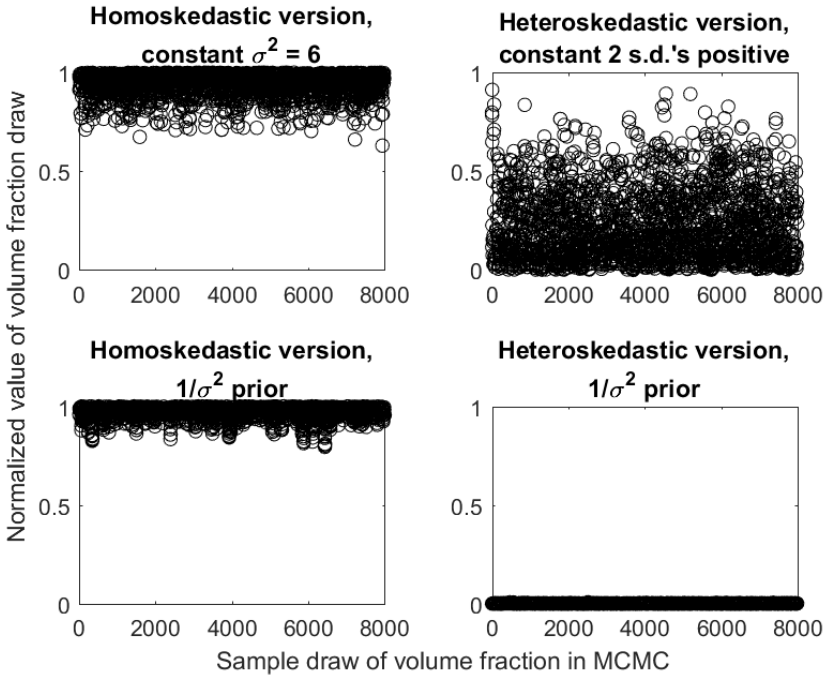

Figure 4: MCMC results of volume fraction at various observation variance settings, on normalized scale. The heteroskedastic constant variance was chosen so that each of the three desired outputs is two standard deviations above zero; e.g., $\sigma_1^2 = \left(\frac{1}{2} \cdot 0.65\right)^2 \approx 0.106$. The homoskedastic constant was set at 6 for each standardized output.

4.4 Which targets to set?

A related question to the setting of the observation variance of the desired observation is the setting of the desired observations themselves. I described some general considerations in Section 2.1.1 above, the upshot of which is that it is generally preferable to set desired observations as close to achievable as possible while remaining confident that the observations are not in fact realistically achievable.

In the case of heteroskedastic observation variance with a (non-degenerate) prior, the specific choice of desired data is not crucial. But where observation variance is either set to a known value or constrained to be equal (up to pre-specified scalar multiple) across outputs, the choice of desired data will matter for the same reason that the constraints on observation variance matter. To see this, consider performing the wind turbine blade calibration with observation variances set equal to $\sigma^2 = 6$ for all model outputs, where the desired deflection, rotation, and cost are set to $\mathbf{d} = [0.65,\ 0.077,\ 96]$. These values are the lower bounds on what is known to be plausible ranges for each output. The MCMC results for volume fraction draws and the resulting posterior mean model output for these settings are shown in Figure 4 and Table 1. Consider now replacing $\mathbf{d}$ with $\mathbf{d}' = [0.65,\ 0.077,\ -500]$. The effect of this would be equivalent to keeping $\mathbf{d}$ and reducing the observation variance for cost from 6 to a much lower value, insofar as each of these two changes amounts to increasing the Mahalanobis distance of each draw from the desired observation along the axis of cost. This, in turn, would have the effect in the calibration procedure of prioritizing cost at the expense of the performance metrics. Thus when one wishes to strike a balance of priorities in the calibration procedure, this will necessarily involve both constraints upon the observation variance and the choice of desired observations. A further illustration of the impact of this choice is in Table 2, where two different levels of desired observations are compared under a pre-specified constant observation variance.

Figure 5 illustrates the sort of difficulty that can arise if one fails to make one's desired observations sufficiently ambitious. Here, a separate $1/\sigma_i^2$ prior was placed on each of the three types of observation variance. The top set of plots show the MCMC results for an appropriately ambitious (i.e. not realistically achievable) desired observations (here rotation has been removed from the model for simplicity). The desired observations yield good mixing in the MCMC routine. In the bottom plots, the desired observation of deflection is still lower than what is realistically achievable, whereas the cost is low but achievable. The model can yield this desired cost, and so draws of the observation variance for cost quickly drift to extremely low values. As a result, the MCMC samples only from the very small region of the parameter space corresponding to cost very near \$163. The extremely strong correlation of VF and thickness within this region can be seen in the bottom right plot of Figure 5. The small size of this region makes it difficult to sample from. But more importantly, samples from this region are not of particular interest, since it is possible to outperform this region in cost without sacrificing other desiderata. If one were antecedently unaware that this region can be outperformed, that fact would be suggested by the ability of the MCMC to find a region that meets the cost target (nearly) exactly without much variation in meeting the other desiderata. Indeed, this outcome of extremely low observation variance – when observation variance is sampled under a prior – serves as a way to detect that one's desired observation is in fact realistically achievable.

When one *knowingly* sets a realistically achievable outcome as a desired observation, it is preferable to specify a constant observation variance, or at least a lower bound on observation variance. In this way, one avoids the convergence difficulties of the lower plots in Figure 5. In essence, setting a lower bound on the observation variance of cost would "fatten" the thin line of accepted draws represented in the rightmost plot, enhancing the ease with which the MCMC routine can explore the parameter space, and improving the acceptance ratio. However, where an analysis calls for strict adherence to a given target observation (so that such "fattening" is unpalatable), tools of low probability estimation can be brought to bear. Sequential Monte Carlo (Doucet et al., 2001; Liu, 2001) or subset simulation (Au and Beck, 2001, 2003; Zuev et al., 2012) can be used to sample from the low-probability space associated with a tight bound on a desired observation.

Desired data $\mathbf{d}$	σ^2_{defl}	σ^2_{rot}	σ^2_{cost}	$\mu_{v\|\mathbf{d}}$	$\mu_{h\|\mathbf{d}}$	$\sigma^2_{v\|\mathbf{d}}$	$\sigma^2_{h\|\mathbf{d}}$
$(0, 0, 0)$	375.45	277.69	2.62	0.215	$4.01 \cdot 10^{-2}$	$4.41 \cdot 10^{-2}$	$1.92 \cdot 10^{-3}$
$(0.65, 0.077, 96)$	16.74	15.25	$4.62 \cdot 10^{-7}$	$1.09 \cdot 10^{-3}$	$3.36 \cdot 10^{-4}$	$1.02 \cdot 10^{-5}$	$9.97 \cdot 10^{-6}$

Table 2: Comparison of results for two different (low) values of $\mathbf{d}$. Values listed are, respectively, the posterior means for the observation variance of each model output, posterior means for volume fraction (v) and thickness (h), and posterior variance of volume fraction and thickness. The differing results reflect different prioritizations of cost over performance metrics, since volume fraction correlates positively with cost and negatively with deflection and rotation.

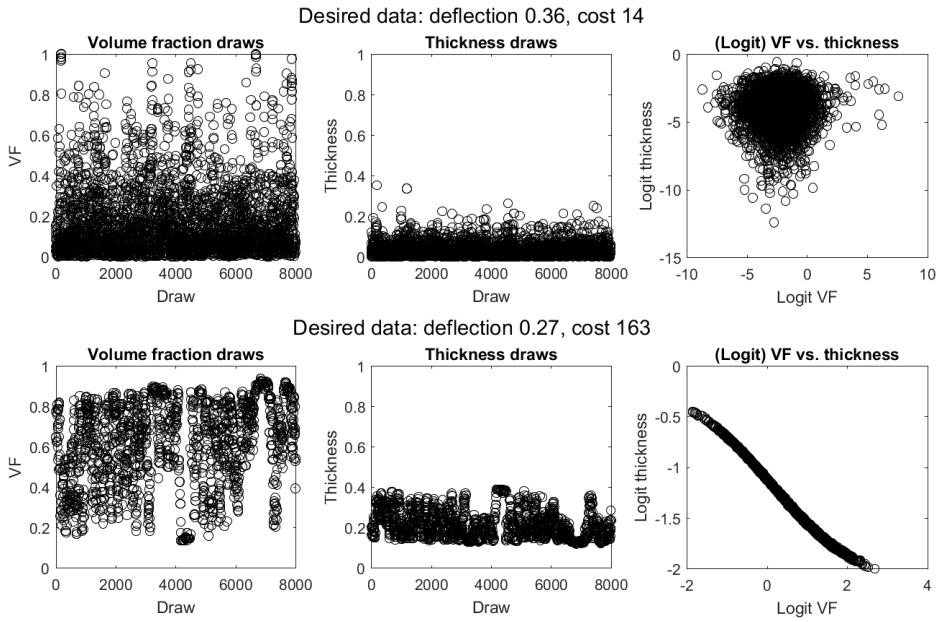

Figure 5: MCMC results for low deflection and cost (top row) and low deflection with easily achievable cost (bottom row).

4.5 Removing calibration parameters

Where a model contains several outputs that one wishes to control, it can become complicated to juggle one's priorities in achieving 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. In this sort of situation, the best strategy is to remove one or more outputs. Two strategies for doing so are explored in this section.

Non-uniform prior on $\boldsymbol{\theta}$: The first option is attractive in those scenarios where one has rough knowledge of the correlation between the calibration parameters and a given model output. The strategy is to exploit this correlation to remove one of the desired observation outputs, where a prior is placed on the calibration parameters that is designed to control the removed model output.

The wind turbine application serves as an example here. It is known that cost correlates strongly with each of the two calibration parameters (volume fraction v and thickness k). Thus, I replace the uniform $\pi(\boldsymbol{\theta})$ of (17) with a prior that penalizes high values of v and k, thereby penalizing high cost output:

$$\pi(\boldsymbol{\theta}) = \pi(v,k) \propto \exp(-\lambda_{cost}\|(v,k)\|^2) \tag{23}$$

where λ_{cost} can be set to any nonnegative value, with higher values resulting in stricter cost control. Pairing this with the $1/\sigma_i^2$ prior on observation variances, the full model becomes:

$$\begin{aligned}\pi(\boldsymbol{\theta}, \mathbf{C_y}|\mathcal{D}) &\propto \pi(\mathcal{D}|\boldsymbol{\theta}, \mathbf{C_y}) \times \pi(\boldsymbol{\theta}) \times \pi(\mathbf{C_y}) \\ &\propto |\mathbf{C}_\mathcal{D}|^{-1/2}\exp(\mathcal{D}^T\mathbf{C}_\mathcal{D}^{-1}\mathcal{D}) \times \exp(-\lambda_{cost}\|(v,k)\|^2) \times \prod_{i=1}^{3}\frac{1}{\sigma_i^2}\end{aligned} \tag{24}$$

This may seem to make little progress in resolving the complication of a model that includes cost, since now instead of specifying a desired cost we must specify a value for λ_{cost}. However, rather than specify a value for this hyperparameter, we can gain a more comprehensive picture of our options by observing the calibration results over a grid of values of λ_{cost}. Thus, rather than specifying beforehand what is our desired

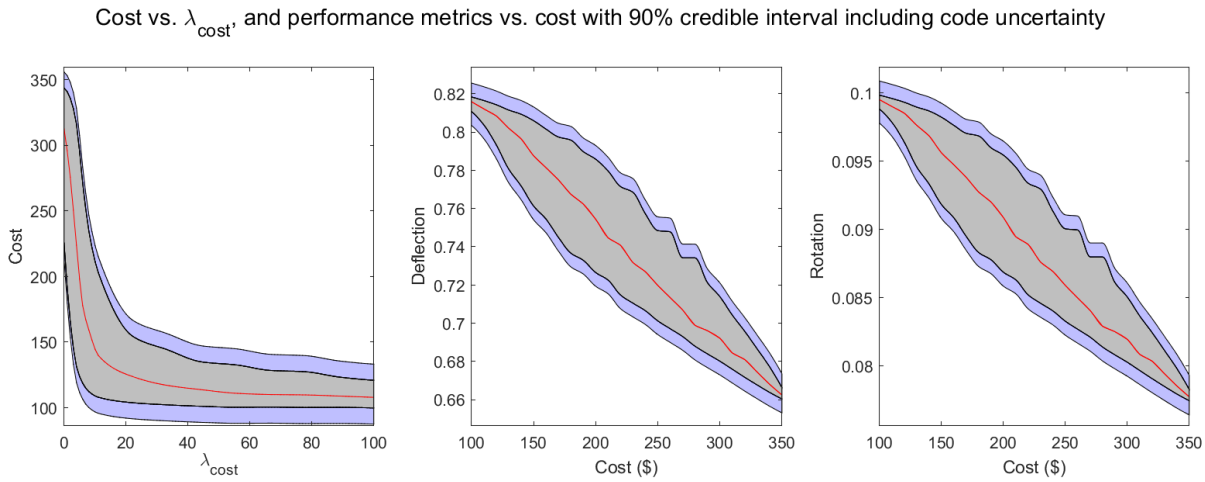

Figure 6: Resulting cost vs. λ_{cost}, and "Pareto bands" of wind turbine blade performance metric over a range of values of cost. Here, cost has been removed from the model and the prior $\exp(-\lambda_{cost}\|\theta\|^2)$ placed on the calibration parameters $\boldsymbol{\theta}$, with desired data 0 deflection and 0 rotation. The gray region gives a 90% credible interval only considering parameter uncertainty. The blue region extends this to include code uncertainty, using the covariance of the posterior GPs corresponding to each draw of $\boldsymbol{\theta}$.

"balance" amongst cost, deflection and rotation, we are able to see a curve describing our options for this trade-off, and to then make an informed choice as to where on that curve we wish to be.

The decision-making value of such an analysis is further improved by including uncertainty quantification, which comes easily due to the nature of the Bayesian approach and GP emulator here. The draws of calibration parameters in the MCMC routine each correspond to a GP with posterior mean serving as an estimate of the true model output for those calibration settings. This captures the parameter uncertainty – i.e., uncertainty remaining in the posterior distribution of the calibration parameters, which is uncertainty about which calibration parameter settings best achieve the desired observations. Furthermore, the GPs themselves are uncertain representations of the true model. For each draw of the calibration parameters in the MCMC routine, (16) gives us the mean and variance of the output at each control input. Thus we can use the posterior covariance of the GP to estimate the code uncertainty – the uncertainty due to the variation of the GP around the true system mean. Thus a sort of "Pareto band" of model performance can guide decision-making in selecting a level for our calibration parameters. Decision-makers can observe surfaces describing the achievable system responses, with included uncertainty measures around those surfaces, and select a target outcome from this comprehensive picture of what is achievable. Figure 6 gives an example of this for the wind turbine application. Figure 7 shows the posterior distributions of volume fraction and thickness for various values of λ_{cost}.

Specifying known cost The above strategy of performing the calibration across a grid and forming a response surface over that grid may be applied even more directly, to largely the same end as in the previous section. Namely: rather than replace one or more desired outputs with a prior on the calibration inputs and drawing a grid over the hyperparameter(s) of that prior, one can instead simply draw a grid over one or more of the desired outputs, treating those outputs as known up to slight deviations at each point of the grid. This strategy has the advantage that, unlike the above strategy, it does not require that we have prior knowledge of any relationship between the calibration parameters and the outputs. Furthermore, whereas the previous strategy requires removing a model output, the current strategy uses the same version of the model, relying solely on changes to the desired observations and observation variance to effect the strategy. In comparison with the previous section's strategy, the approach in the current section also enjoys better interpretability.

For convenience, call an output that is specified as known up to slight deviations in the way suggested

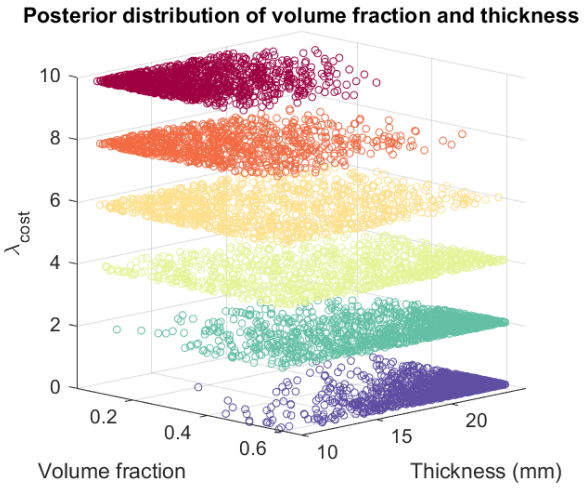

Figure 7: Posterior distributions of volume fraction and thickness for various values of λ_{cost}. Notice that higher values of λ_{cost} push the distribution away from the (high-cost) upper region of the supports of the two variables.

here "d-known". To specify an output as d-known, it suffices to (1) choose a desired observation of that output which is realistically achievable, and (2) set a constant, low observation variance for that output. The precise value of the observation variance for a d-known output can be tuned during the burn-in period of the MCMC routine. It ought to be as low as possible while still allowing for healthy mixing in the MCMC. If the variance is too small, then proposed draws of the calibration parameters will be likely to take the model output too far away from the d-known output, resulting in the draw being rejected, and hence low acceptance rates and poor mixing.

The other desired observations – those that are not d-known – can be treated as usual, with their observation variances each drawn from a $1/\sigma_i^2$ prior, or whichever other treatment of observation variance from Section 4.3 one wishes to use. Thus at the calibration procedure which takes place at each point in the grid of d-known outputs, the d-known outputs have in essence been removed from the CDO, and are treated as known up to a slight variance for facilitating healthy MCMC. This "removal" requires little modification to the model, but still serves to simplify the choice of desired observations and observation variance. The full model is given by

$$\begin{aligned}\pi(\boldsymbol{\theta}, \mathbf{C_y}|\mathcal{D}) &\propto \pi(\mathcal{D}|\boldsymbol{\theta}, \mathbf{C_y}) \times \pi(\boldsymbol{\theta}) \times \pi(\mathbf{C_y}) \\ &\propto \pi(\mathcal{D}|\boldsymbol{\theta}, \mathbf{C_y}) \times \pi(\mathbf{C_y}) \\ &\propto |\mathbf{C}_\mathcal{D}|^{-1/2} \exp(\mathcal{D}^T \mathbf{C}_\mathcal{D}^{-1} \mathcal{D}) \times \prod_{i=1}^{2} \frac{1}{\sigma_i^2}\end{aligned} \tag{25}$$

where $\mathbf{C_y}$ is the diagonal matrix where the j, j entry c_j is equal to σ_1^2 if y_j is an observation of deflection, $c_j = \sigma_2^2$ if y_j is an observation of rotation, and $c_j = s$ if y_j is an observation of cost, for some small pre-selected value of s. In this application, I set $s = 0.05$, so that the observation variance on standardized cost values at each point of the grid was 0.05.

The result of this strategy is similar to that in the previous section: one can derive a response surface over the grid on d-known outputs, where that response describes the achievable results closest to the desired observations (not including the d-known observations) at each point in the space of d-known outputs. 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

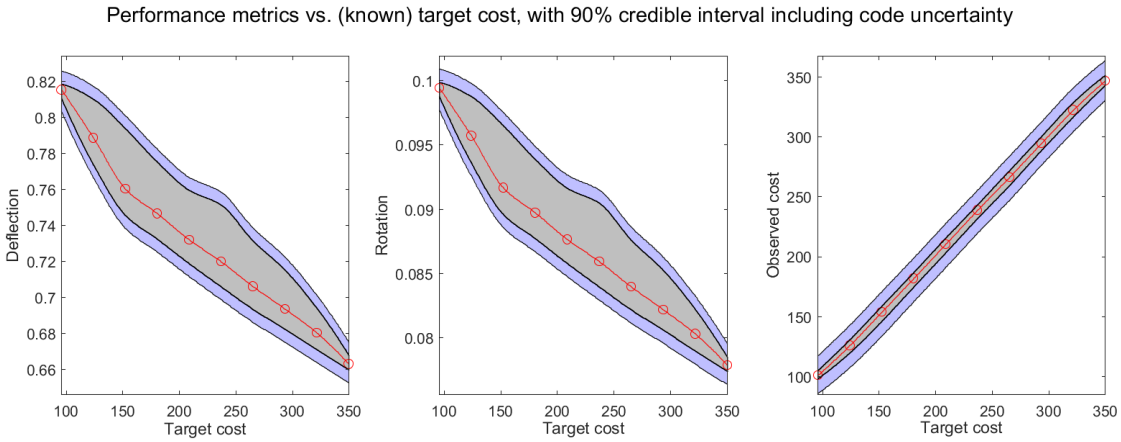

Figure 8: "Pareto bands" of wind turbine blade model outputs over a range of values of cost, where cost is treated as known up to slight deviations, with desired data 0 deflection and 0 rotation. The gray region gives a 90% credible interval only considering parameter uncertainty; the blue region extends this to include code uncertainty.

into specific choices of desired observations and observation variance schemes.

The same sort of uncertainty analysis described in Section 4.5 is available here, since posterior parameter uncertainty and code uncertainty are easily recovered from the results of the MCMC routine. An example of the result of this strategy is shown in Figure 8, where the strategy was applied to the wind turbine application, treating cost as d-known. The rightmost plot is included to verify that the posterior model output respected the d-known cost values used in the calibrations.

5 Future work

5.1 Hamiltonian Monte Carlo

Recall from Figure 3 that the autocorrelation in the MCMC routine, while much reduced by using the Metropolis-Hastings algorithm after applying a logit transformation to remove boundary conditions, is nonetheless still appreciable. In the application considered there, in order to achieve uncorrelated draws, one would save at most every 30^{th} draw. The efficiency of the algorithm can be improved by instead using the Hamiltonian Monte Carlo technique (HMC), also known as hybrid Monte Carlo (Duane et al., 1987).

Initially applied to lattice field theory simulations of quantum chromodynamics, HMC was popularized within the statistical community by Neal (2011). Neal describes the intuition supporting HMC roughly as follows. Consider the Metropolis-Hastings algorithm, in which a proposed new draw is drawn from a proposal distribution which is typically centered on the previous draw. Thus the position of the previous draw informs the next proposed draw. Rather than rely merely on the position of the previous draw, HMC relies both on its position and on its *momentum*. The momentum of a variable moving in $p-$dimensional space can be understood on analogy to a puck moving on a horizontal two-dimensional surface curved in the third dimension to have areas of varying height. Negative likelihood for the variable is the analog for height. As the variable moves from an area of low likelihood to higher likelihood, it gains momentum, increasing the proposal density in the direction of that momentum. Likewise, as the variable moves from an area of high likelihood to low likelihood, its momentum can help carry it further into that space; but moving into the low likelihood space also reduces the variable's momentum, eventually causing the momentum to change direction. The result of these Hamiltonian dynamics in the sampling routine is to improve the efficiency of the exploration of the parameter space. Thus HMC works to reduce autocorellation beyond what is typically possible in Metropolis-Hastings. Therefore its use in the area of CDO can help to improve the efficiency of the sampling routine beyond what is seen in the righthand panel of Figure 3.

5.2 Model discrepancy

Recall from Section 2.2 that we may incorporate model shortcoming in two ways: as a form of observation variance, or as a form of model discrepancy. In Section 4, observation variance serves as the basis for handling model shortcoming. This approach is attractive for its simplicity and for its flexibility in facilitating different calibration goals through modulation of the observation variances and their prior distributions. However, as described in Section 2.2.2, there are advantages to using a model discrepancy $\delta(\cdot)$ instead of observation variance to incorporate model shortcoming. Most importantly, one's desired observation will tend to be systematically biased from the true system mean, and a model that incorporates shortcoming by way of Gaussian white noise, in an observation variance term, fails to capture this systematic bias. For this reason, future work in this area will include replacing observation variance with a prior mean-zero GP discrepancy function in the calibration procedure.

Similarly, the technique of CDO as described in the present work assumes access to a computer model which is known to be accurate throughout the domain $\mathcal{T}$ over which the calibration occurs. Of course, in many applications, this assumption will not hold. Future extensions of the present work will include investigation of combining CDO with traditional calibration. This would be of interest in those situations where one has access to field observations and wishes to employ CDO. If computational expense makes simulation observations scarce, it will be particularly attractive to combine the two calibrations, using the same set of simulations for both calibrations simultaneously.

I intend to pursue two avenues in this area. Firstly, one might capture model inadequacy with a traditional model discrepancy function, while simultaneously capturing model shortcoming via observation variance. In a full Bayesian analysis that treats desired observations as extra field observations, this would certainly lead to a lack of identifiability between the observation variance and the discrepancy function. However, the method of "modularization" (Liu et al., 2009; Bayarri et al., 2007b,a), described above in Section 1.2, can mitigate these difficulties, by partitioning the analysis so that desired observations do not bias the discrepancy function, and field observations do not bias the estimate of the observation variance of the desired observations.

Secondly, one may attempt to capture model inadequacy and model shortcoming each in a separate discrepancy function. This will face the same difficulty as the first approach, and will likewise require modularization in order to retain the identifiability of the two discrepancy functions.

5.3 Comparison with optimization techniques

CDO is a form of optimization, and thus future work will include a thorough comparison of that approach to alternative forms of optimization. As well as comparing results, computational economy should be considered as a desideratum. Furthermore, CDO is a form of optimization under uncertainty, which can easily incorporate uncertainty in the model inputs, and which delivers a result which includes quantification of uncertainty on the posterior model outputs. Therefore, of particular interest are alternative means of optimization that accommodate and quantify uncertainty. Sahinidis (2004) provides a useful overview of existing techniques for optimization under uncertainty; such techniques are the primary alternatives against which CDO should be considered.

6 Conclusion

In this work I 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). I have also described 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. I described the implementation of this approach in an MCMC routine along with considerations to accommodate computational instability. I have also indicated future directions for research in the area of CDO.

The use of this methodology is illustrated in the case of material design for a wind turbine blade. I 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.

Appendix A Full conditional distributions

Using a model that does not employ a discrepancy term, uses maximum likelihood estimation of $\boldsymbol{\beta}^{\eta}$ and λ_{η}, and sets the mean c of the emulator to be 0, we have from (11) above that the joint posterior distribution is

$$\begin{aligned}\pi(\boldsymbol{\theta}, \mathbf{C_y}|\mathcal{D}) &\propto \pi(\mathcal{D}|\boldsymbol{\theta}, \mathbf{C_y}) \times \pi(\boldsymbol{\theta}) \times \pi(\mathbf{C_y}) \\ &\propto |\mathbf{C}_{\mathcal{D}}|^{-1/2} \exp(\mathcal{D}^T \mathbf{C}_{\mathcal{D}}^{-1} \mathcal{D}) \times \pi(\boldsymbol{\theta}) \times \pi(\mathbf{C_y})\end{aligned} \tag{26}$$

where $\mathbf{C}_{\mathcal{D}}$, given in (10), depends on both $\boldsymbol{\theta}$ and on $\mathbf{C_y}$. Different priors on $\boldsymbol{\theta}$ and $\mathbf{C_y}$ are used in the present work to flesh out this distribution. For example, columns one and two of Table 1 correspond to setting a uniform prior on $\boldsymbol{\theta}$ and specifying a set value for $\mathbf{C}_y$ (essentially thereby using a degenerate prior). Thus over the support of $\boldsymbol{\theta}$ at the specified value of $\mathbf{C_y}$, the joint posterior distribution is

$$|\mathbf{C}_{\mathcal{D}}|^{-1/2} \exp(\mathcal{D}^T \mathbf{C}_{\mathcal{D}}^{-1} \mathcal{D}) \times \pi(\boldsymbol{\theta}) \times \pi(\mathbf{C_y}) = |\mathbf{C}_{\mathcal{D}}|^{-1/2} \exp(\mathcal{D}^T \mathbf{C}_{\mathcal{D}}^{-1} \mathcal{D}) \tag{27}$$

and since $\mathbf{C_y}$ is specified in advance, this distribution is also the marginal posterior distribution of $\boldsymbol{\theta}$. By contrast, columns three and four of Table 1 place a reference prior on the observation variance of each model output. Recall that where σ_i^2 gives the observation variance for the i^{th} model output, $\mathbf{C_y}$ is a diagonal matrix with i, i entry σ_i^2. Column three of the table corresponds to a model in which each of the three outputs has its own observation variance, so that the prior $\pi(\mathbf{C_y}) = \prod_{i=1}^{3} \sigma_i^{-2}$. Column four uses a single observation variance for all three outputs, so that the prior is $\pi(\mathbf{C_y}) = \sigma^{-2}$. Letting $K = 3$ for column three and $K = 1$ for column four, then, the joint posterior distribution is

$$|\mathbf{C}_{\mathcal{D}}|^{-1/2} \exp(\mathcal{D}^T \mathbf{C}_{\mathcal{D}}^{-1} \mathcal{D}) \times \pi(\boldsymbol{\theta}) \times \pi(\mathbf{C_y}) \propto |\mathbf{C}_{\mathcal{D}}|^{-1/2} \exp(\mathcal{D}^T \mathbf{C}_{\mathcal{D}}^{-1} \mathcal{D}) \times \prod_{i=1}^{K} \sigma_i^{-2} \tag{28}$$

where $\pi(\boldsymbol{\theta})$ is again set to be uniform. Thus the posterior conditional distributions are

$$\begin{aligned}\pi(\boldsymbol{\theta}|\mathbf{C_y}, \mathcal{D}) &\propto |\mathbf{C}_{\mathcal{D}}|^{-1/2} \exp(\mathcal{D}^T \mathbf{C}_{\mathcal{D}}^{-1} \mathcal{D}) \\ \pi(\mathbf{C_y}|\boldsymbol{\theta}, \mathcal{D}) &\propto |\mathbf{C}_{\mathcal{D}}|^{-1/2} \exp(\mathcal{D}^T \mathbf{C}_{\mathcal{D}}^{-1} \mathcal{D}) \times \prod_{i=1}^{K} \sigma_i^{-2}.\end{aligned} \tag{29}$$

Section 4.5 employs a model which places an informative prior on $\boldsymbol{\theta}$. A reference prior is also placed on the observation variance of each model output. The resulting posterior distribution is thus

$$|\mathbf{C}_{\mathcal{D}}|^{-1/2} \exp(\mathcal{D}^T \mathbf{C}_{\mathcal{D}}^{-1} \mathcal{D}) \times \pi(\boldsymbol{\theta}) \times \pi(\mathbf{C_y}) \propto |\mathbf{C}_{\mathcal{D}}|^{-1/2} \exp(\mathcal{D}^T \mathbf{C}_{\mathcal{D}}^{-1} \mathcal{D}) \times \exp(-\lambda_{cost} \|\boldsymbol{\theta}\|^2) \times \prod_{i=1}^{3} \sigma_i^{-2} \tag{30}$$

and the posterior conditional distributions are given by

$$\begin{aligned}\pi(\boldsymbol{\theta}|\mathbf{C_y}, \mathcal{D}) &\propto |\mathbf{C}_{\mathcal{D}}|^{-1/2} \exp(\mathcal{D}^T \mathbf{C}_{\mathcal{D}}^{-1} \mathcal{D}) \times \exp(-\lambda_{cost} \|\boldsymbol{\theta}\|^2) \\ \pi(\mathbf{C_y}|\boldsymbol{\theta}, \mathcal{D}) &\propto |\mathbf{C}_{\mathcal{D}}|^{-1/2} \exp(\mathcal{D}^T \mathbf{C}_{\mathcal{D}}^{-1} \mathcal{D}) \times \prod_{i=1}^{3} \sigma_i^{-2}.\end{aligned} \tag{31}$$

Appendix B Algorithm for MCMC

The following algorithm describes how to sample from the model given in (30) and (31), which includes the model of (28) and (29) as a special case, by setting $\lambda_{cost} = 0$. Minor and straightforward modifications are required to sample from the model of (27). In the algorithm, single-variable functions applied to multivariate input are applied componentwise; e.g., $\exp\left(MVN([0,0]^T, I_2)\right)$ denotes $[\exp(r_1), \exp(r_2)]$ where $[r_1, r_2]^T \sim \text{MVN}([0,0]^T, I_2)$.

Algorithm 1:

Step 1: Draw $\boldsymbol{\theta}^{(0)} \sim \text{Unif}([0,1]^2)$, $(\sigma_j^2)^{(0)} \sim \text{Gamma}(2,2)$ for $j = 1, 2, \ldots, K$. Set $\mathbf{C_y}$ using $(\sigma_j^2)^{(0)}$, $j = 1, 2, \ldots, K$. Set $\mathbf{C}_\mathcal{D}$ using $\boldsymbol{\theta}^{(0)}$ and $\mathbf{C_y}$. Specify $s^2 = 1$. Specify $\lambda_{cost} \geq 0$. Specify total number of iterations M and burn-in b. Set mult $= 2$. Set $\Sigma = I_2$. Set $A = 0$ and $A_\sigma = 0$. Set $i = 0$.

Step 2: Repeat M times.

- Update $i = i + 1$.
- Draw $\boldsymbol{\theta}^* \sim \text{logit}^{-1}\left(\text{MVN}(\text{logit}(\boldsymbol{\theta}^{(i-1)}), \Sigma)\right)$.
- Set $\mathbf{C}_\mathcal{D}^*$ using $\boldsymbol{\theta}^*$ and $\mathbf{C_y}$.
- Find

$$\begin{aligned}\log\alpha = &\log\left(|\mathbf{C}_\mathcal{D}^*|^{-1/2}\exp(\mathcal{D}^T{\mathbf{C}_\mathcal{D}^*}^{-1}\mathcal{D})\right) - \lambda_{cost}\cdot\|\boldsymbol{\theta}^*\|^2 - \log\left(|\mathbf{C}_\mathcal{D}|^{-1/2}\exp(\mathcal{D}^T\mathbf{C}_\mathcal{D}^{-1}\mathcal{D})\right)\\ &+ \lambda_{cost}\cdot\|\boldsymbol{\theta}^{(i-1)}\|^2 + \log\left(\prod_{j=1}^{2}\left(\theta_j^*(1-\theta_j^*)\right)\right) - \log\left(\prod_{j=1}^{2}\left(\theta_j^{(i-1)}(1-\theta_j^{(i-1)})\right)\right)\end{aligned}$$

- Draw $a \sim \text{Unif}(0,1)$. If $a < \alpha$, set $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^*$, $\mathbf{C}_\mathcal{D} = \mathbf{C}_\mathcal{D}^*$, and increment $A = A + 1$; otherwise, set $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)}$.
- Draw $[\sigma_1^{2^*}, \sigma_2^{2^*}, \sigma_3^{2^*}]^T \sim \exp\left(\text{MVN}\left(\log\left[\sigma_1^{2(i-1)}, \sigma_2^{2(i-1)}, \sigma_3^{2(i-1)}\right]^T, s^2 I_3\right)\right)$.
- Set $\mathbf{C_y^*}$ using $\left[\sigma_1^{2^*}, \sigma_2^{2^*}, \sigma_3^{2^*}\right]^T$. Set $\mathbf{C}_\mathcal{D}^*$ using $\boldsymbol{\theta}^{(i)}$ and $\mathbf{C_y^*}$.
- Find $\log\alpha = \log\left(|\mathbf{C}_\mathcal{D}^*|^{-1/2}\exp(\mathcal{D}^T{\mathbf{C}_\mathcal{D}^*}^{-1}\mathcal{D})\right) - \log\left(|\mathbf{C}_\mathcal{D}|^{-1/2}\exp(\mathcal{D}^T\mathbf{C}_\mathcal{D}^{-1}\mathcal{D})\right)$. (Notice that the log ratio of priors $\log\left(\frac{\pi(\sigma_i^{2\prime})}{\pi(\sigma_i^2)}\right) = \log\frac{\sigma_i^2}{\sigma_i^{2\prime}} = \log\sigma_i^2 - \log\sigma_i^{2\prime}$ is cancelled out by the log Metropolis-Hastings correction for the asymetrical proposal density: $\log\left(\frac{q(\sigma_i^2|\sigma_i^{2\prime})}{q(\sigma_i^{2\prime}|\sigma_i^2)}\right) = \log\left(\frac{\sigma_i^{2\prime}}{\sigma_i^2}\right) = \log\sigma_i^{2\prime} - \log\sigma_i^2$.)
- Draw $a \sim \text{Unif}(0,1)$. If $a < \alpha$, set $\left[\sigma_1^{2(i)}, \sigma_2^{2(i)}, \sigma_3^{2(i)}\right]^T = \left[\sigma_1^{2^*}, \sigma_2^{2^*}, \sigma_3^{2^*}\right]^T$, $\mathbf{C_y} = \mathbf{C_y^*}$, $\mathbf{C}_\mathcal{D} = \mathbf{C}_\mathcal{D}^*$, and increment $A_\sigma = A_\sigma + 1$; otherwise, set $\left[\sigma_1^{2(i)}, \sigma_2^{2(i)}, \sigma_3^{2(i)}\right]^T = \left[\sigma_1^{2(i-1)}, \sigma_2^{2(i-1)}, \sigma_3^{2(i-1)}\right]^T$.
- If $i \leq b$ and $i \pmod{100} = 0$:
  - Update mult $= 1.5 \cdot \text{mult} \cdot \mathbf{1}_{A>30} + 0.75 \cdot \text{mult} \cdot \mathbf{1}_{A<20}$. Set $A = 0$.
  - Update $\Sigma = \text{mult} \cdot \text{Cov}(\Theta)$, where $\Theta = \left[\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(i)}\right]^T$.
  - Update $s^2 = 1.5 \cdot s^2 \cdot \mathbf{1}_{A_\sigma>30} + 0.75 \cdot s^2 \cdot \mathbf{1}_{A_\sigma<20}$. Set $A_\sigma = 0$.

References

Atamturktur, S. and Brown, D. A. (2015). State-aware calibration for inferring systematic bias in computer models of complex systems. *NAFEMS World Congress Proceedings, June 21-24*.

Au, S.-K. and Beck, J. L. (2001). Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, 16(4):263–277.

Au, S.-K. and Beck, J. L. (2003). Subset Simulation and its Application to Seismic Risk Based on Dynamic Analysis. *Journal of Engineering Mechanics*, 129(8):901–917.

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.

Cavelaars, A. E. J. M., Kunst, A. E., Geurts, J. J. M., Crialesi, R., Grötvedt, L., Helmert, U., Lahelma, E., Lundberg, O., Mielck, A., Rasmussen, N. K., Regidor, E., Spuhler, T., and Mackenbach, J. P. (2000). Persistent variations in average height between countries and between socio-economic groups: an overview of 10 European countries. *Annals of Human Biology*, 27(4):407–421.

Chib, S. and Greenberg, E. (1995). Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, 49(4):327.

Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991). Bayesian Prediction of Deterministic Functions, with Applications to the Design and Analysis of Computer Experiments. *Journal of the American Statistical Association*, 86(416):953–963.

Doucet, A., De Freitas, N., and Gordon, N. (2001). *Sequential Monte Carlo methods in practice*. Springer.

Duane, S., Kennedy, A., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222.

Gelman, A. (1992). Iterative and Non-Iterative Simulation Algorithms. *Computing Science and Statistics (Interface Proceedings)*, 24:433–438.

Geman, S. and Geman, D. (1984). IEEE Transactions on Pattern Analysis and Machine Intelligence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741.

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.

Haario, H., Laine, M., Mira, A., and Saksman, E. (2006). DRAM: Efficient adaptive MCMC. *Statistics and Computing*, 16(4):339–354.

Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242.

Haario, H., Saksman, E., and Tamminen, J. (2005). Componentwise adaptation for high dimensional MCMC. *Computational Statistics*, 20(2):265–273.

Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

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.

Liu, J. S. (2001). *Monte Carlo strategies in scientific computing*. Springer.

Loeppky, J. L., Bingham, D., and Welch, W. J. (2006). Computer Model Calibration or Tuning in Practice. *Technometrics*.

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.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

Neal, R. M. (1998). Regression and Classification Using Gaussian Process Priors. In Bernardo, J. M., Berger, J. O., Dawid, A. P., and Smith, A. F. M., editors, *Bayesian Statistics 6*, volume 6, pages 475–501. Oxford University Press, New York.

Neal, R. M. (2011). MCMC Using Hamiltonian Dynamics. In Brooks, S., Gelman, A., Jones, G., and Meng, X.-L., editors, *Handbook of Markov Chain Monte Carlo*, chapter 5, pages 113–162. CRC Press, New York.

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

Paulo, R., García-Donato, G., and Palomo, J. (2012). Calibration of computer models with multivariate output. *Computational Statistics and Data Analysis*, 56:3959–3974.

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.

Rasmussen, C. E., Williams, C. K. I., Sutton, R. S., Barto, A. G., Spirtes, P., Glymour, C., Scheines, R., Schölkopf, B., and Smola, A. J. (2006). *Gaussian Processes for Machine Learning*. MIT Press, Cambridge.

Roberts, G. O., Gelman, A., and Gilks, W. R. (1997). Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms. *The Annals of Applied Probability*, 7(1):110–120.

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.

Sahinidis, N. V. (2004). Optimization under uncertainty: state-of-the-art and opportunities. *Computers & Chemical Engineering*, 28(6-7):971–983.

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

Savitsky, T., Vannucci, M., and Sha, N. (2011). Variable Selection for Nonparametric Gaussian Process Priors: Models and Computational Strategies. *Statistical Science*, 26(1):130–149.

Shang, L. and Chan, A. B. (2013). On Approximate Inference for Generalized Gaussian Process Models. *Technical Report -City University of Hong Kong*.

Sorokowska, A., Sorokowski, P., Hilpert, P., et al. (2017). Preferred Interpersonal Distances: A Global Comparison. *Journal of Cross-Cultural Psychology*, 48(4):577–592.

Stevens, G. N., Atamturktur, S., Brown, D. A., Williams, B. J., and Unal, C. (2018). Statistical inference of empirical constituents in partitioned analysis from integral-effect experiments. *Engineering Computations*, 35(2):672–691.

Subramanian, S. V., Özaltin, E., and Finlay, J. E. (2011). Height of Nations: A Socioeconomic Analysis of Cohort Differences and Patterns among Women in 54 Low- to Middle-Income Countries. *PLoS ONE*, 6(4):e18962.

Thompson, P. A. and Marchant, E. W. (1995). A Computer Model for the Evacuation of Large Building Populations. *Fire Safety Journal*, 24:131–148.

Tierney, L. (1994). Markov Chains for Exploring Posterior Distributions. *The Annals of Statistics*, 22(4):1701–1728.

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.

Zuev, K. M., Beck, J. L., Au, S.-K., and Katafygiotis, L. S. (2012). Bayesian post-processor and other enhancements of Subset Simulation for estimating failure probabilities in high dimensions. *Computers & Structures*, 92-93:283–296.