Fourth Exam: Computer model calibration as a method of design

Carl Ehrett

July 12, 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.

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). Other contributions in this area come from Williams et al. (2006), Bayarri et al. (2007a), Liu et al. (2009), Bayarri et al. (2007b), Brynjarsdóttir and O'Hagan (2014), and Brown and Atamturktur (2018).

1.2.1 Gaussian processes

In principle, model calibration need not rely on a GP emulator, or any other sort of emulator; one could (e.g.) complete a full Bayesian analysis via an MCMC chain that involves running the relevant computer model at each iteration of the chain. However, computer models are frequently too computationally expensive to allow for such profligacy. Instead, a computationally tractable emulator can be constructed using a sample of observations from the computer model. GPs are popular prior distributions on computer model output for three reasons. Firstly, because their use does not require detailed foreknowledge of the model function's parametric form. Secondly, GPs easily interpolate the computer model output, which is attractive when the computer model is deterministic. This is usually the case, although some attention in model calibration has focused specifically on stochastic computer models; see e.g. Pratola and Chkrebtii (2018). Thirdly, GPs facilitate uncertainty quantification through the variance of the posterior GP. In this section, I provide brief background on Gaussian processes and their use in regression broadly and in computer model calibration specifically.

The use of GPs to produce a computationally efficient predictor $\hat{\eta}(\mathbf{x})$ of expensive computer code $\eta(\mathbf{x})$ given observations of code output at $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ is promulgated by Sacks et al. (1989) and explored at length by Santner et al. (2003). Since computer code is typically deterministic, these applications differ from the focus of O'Hagan (1978) in that the updated GP is induced to interpolate the observations $\boldsymbol{\eta} = (\eta(\mathbf{x}_1), \cdots, \eta(\mathbf{x}_n))^T$. 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. 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.

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

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

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

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

where $\eta(\cdot,\cdot)$ is the computer model, $\boldsymbol{\theta}$ is the best[1] 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 correlation function:

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

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

The authors place the following priors on the hyperparameters:

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

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

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

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

with priors

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

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

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

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

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

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

Thus, the joint posterior density under the model is

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

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

[1] 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.

1.2.2 Computational difficulties

Consider a 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 $C(\mathbf{x}_i, \mathbf{x}_j)$. Training the GP on the n observations $\boldsymbol{\eta}$ at $\mathbf{X} \in \mathbb{R}^{n \times p}$, an updated mean and covariance matrix for the points $\mathbf{X}' = (\mathbf{x}'_1, \ldots, \mathbf{x}'_m)^T$ is

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

Poor conditioning of $\mathbf{C}_{\mathbf{X},\mathbf{X}}$ in (11) can make it difficult to invert and find the determinant of this matrix, as must be done in the course of the MCMC to find the relevant likelihoods. This problem can be alleviated by adding a small nugget to $\mathbf{C}_{\mathbf{X},\mathbf{X}}$. That is, we can set $\mathbf{C}^{\xi}_{\mathbf{X},\mathbf{X}} = \mathbf{C}_{\mathbf{X},\mathbf{X}} + \xi \cdot \mathbf{I}_{\dim(\mathbf{X})}$ 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. Furthermore, insofar as the effect is non-negligible, it is argued by Gramacy and Lee (2012) to be beneficial for emulating the computer code (so that for some applications one might prefer a larger nugget).

2 Calibration for design

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

the use of informative priors on the model discrepancy and observation error, the identifiability concerns of the Kennedy-O'Hagan approach (discussed by Bayarri et al. (2007b), **?** and others) are avoided.

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.

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 the wind turbine blade application considered in this work, the ideal material would (impossibly) not deform at all under load. In a different application, one might wish to design a material that deforms in a pre-specified (possible) way. In such a case, it would be appropriate to set desired observations that one indeed does hope to find as the posterior predictive mode after calibration. But in cases such as the wind turbine 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 (**??**) and (**??**) 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.

2.3 Setting the marginal variance for model discrepancy

Hereafter in this work, we will assume that model shortcoming will be captured via a discrepancy term $\delta(\cdot)$ modeled as a mean-zero, stationary GP. In order to successfully calibrate to the optimal region of the parameter space, it is necessary to either place an informative prior on the marginal precision λ_δ of the discrepancy, or else to specify that value outright. Otherwise, identifiability issues can cause the calibration to fail. This is a longstanding concern with the Kennedy-O'Hagan framework, raised in the discussion of the initial publication Kennedy and O'Hagan (2001) as well as in Bayarri et al. (2007b), Tuo and Jeff Wu (2015), and **?**.

2.4 Field observations and model discrepancy

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

2.5 Hyperparameter estimation

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

Therefore, care should be taken to prevent the desired observations from "infecting" the covariance hyperparameters, since we want the latter to reflect reality rather than our performance targets. The way that this is prevented in the application of Section 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 section 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[2] , 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.

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

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 (4) 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 **??**.

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 2.5. 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

$$\hat{\boldsymbol{\rho}}^\eta = (0.9358, 0.6509, 0.6736, 0.4797, 0.9673), \quad \hat{\lambda}_\eta = 0.0152 \tag{12}$$

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 **??**.

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

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

where $\mu^*(\mathbf{b}) = \mathbf{C}_{\mathbf{b},\mathbf{B}} \cdot \mathbf{C}_{\mathbf{B},\mathbf{B}}^{-1} \cdot \boldsymbol{\eta}$, $C^*(\mathbf{b}, \mathbf{b}') = \mathbf{C}_{(\mathbf{b}^T,\mathbf{b}'^T)^T,(\mathbf{b}^T,\mathbf{b}'^T)^T} - \mathbf{C}_{(\mathbf{b}^T,\mathbf{b}'^T)^T,\mathbf{B}} \cdot \mathbf{C}_{\mathbf{B},\mathbf{B}}^{-1} \cdot \mathbf{C}_{\mathbf{B},(\mathbf{b}^T,\mathbf{b}'^T)^T}$, $\mathbf{C}_{\Upsilon,\Gamma}$ is the matrix whose i, j element is equal to the covariance between the observation at the i^{th} row of Υ and at the j^{th} row of Γ, $\mathbf{b} = (\mathbf{x}, \mathbf{t})$ is a row vector of control and calibration inputs, $\mathbf{B} = (\mathbf{b}_1^T, \mathbf{b}_2^T, \cdots, \mathbf{b}_n^T)^T$ is the 1512×5 matrix of locations of the 1512 simulation observations, and $\boldsymbol{\eta}$ is a column vector of the 1512 simulation responses: $\eta_i = \eta(\mathbf{b}_i)$. All model inputs are normalized to [0,1] over their supports. All model outputs are standardized so that $\boldsymbol{\eta}$ has mean 0 and standard deviation 1. $C(\cdot, \cdot)$ is given by (4), where I plug in the MLEs given in (12).

4.2 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 (**??**) with a prior that penalizes high values of v and k, thereby penalizing high cost output:

$$\pi(\boldsymbol{\theta}) = \pi(v,k) \propto \exp(-\lambda_{cost}\|(v,k)\|^2) \tag{14}$$

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:

$$\begin{aligned}\pi(\boldsymbol{\theta}, \mathbf{C_y}|\mathcal{D}) &\propto \pi(\mathcal{D}|\boldsymbol{\theta}, \mathbf{C_y}) \times \pi(\boldsymbol{\theta}) \times \pi(\mathbf{C_y}) \\ &\propto |\mathbf{C}_\mathcal{D}|^{-1/2}\exp(\mathcal{D}^T\mathbf{C}_\mathcal{D}^{-1}\mathcal{D}) \times \exp(-\lambda_{cost}\|(v,k)\|^2) \times \prod_{i=1}^{3}\frac{1}{\sigma_i^2}\end{aligned} \tag{15}$$

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 "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, (13) 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 1 gives an example of this for the wind turbine application. Figure 2 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 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.

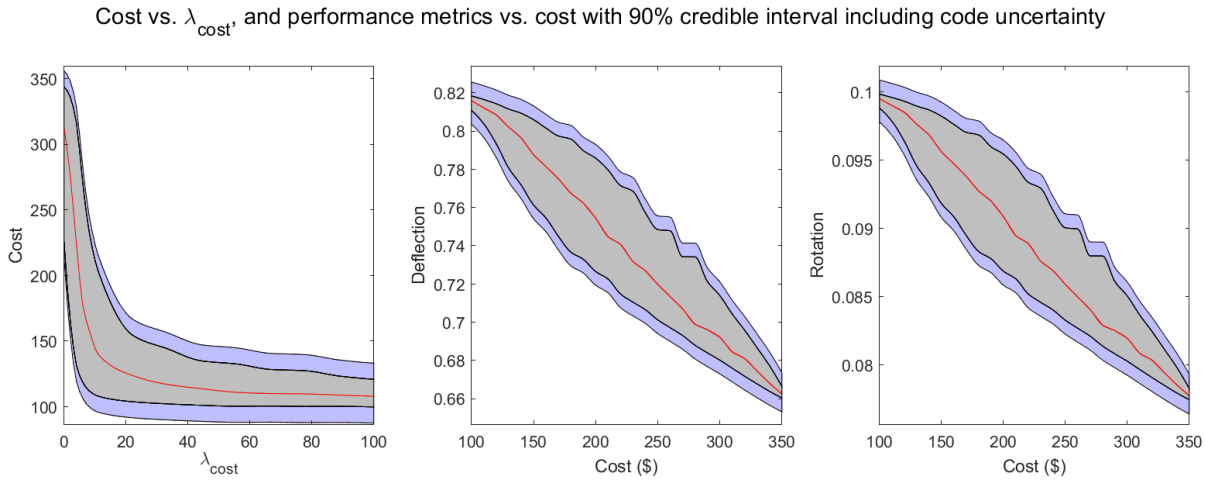

Figure 1: Resulting cost vs. λ_{cost}, and "Pareto bands" of wind turbine blade performance metric over a range of values of cost. Here, cost has been removed from the model and the prior $\exp(-\lambda_{cost}\|\theta\|^2)$ placed on the calibration parameters $\boldsymbol{\theta}$, with desired data 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 $\boldsymbol{\theta}$.

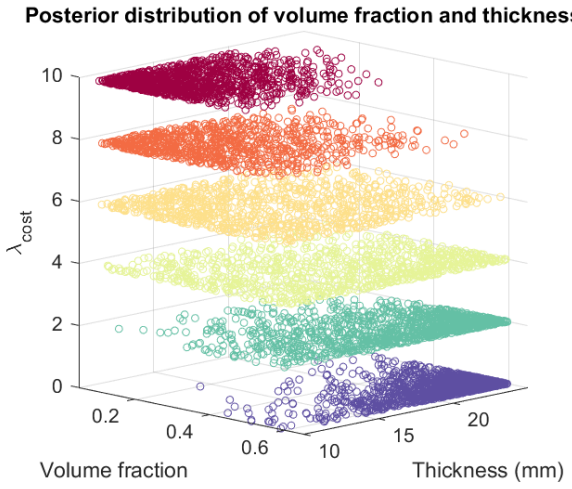

Figure 2: Posterior distributions of volume fraction and thickness for various values of λ_{cost}. Notice that higher values of λ_{cost} push the distribution away from the (high-cost) upper region of the supports of the two variables.

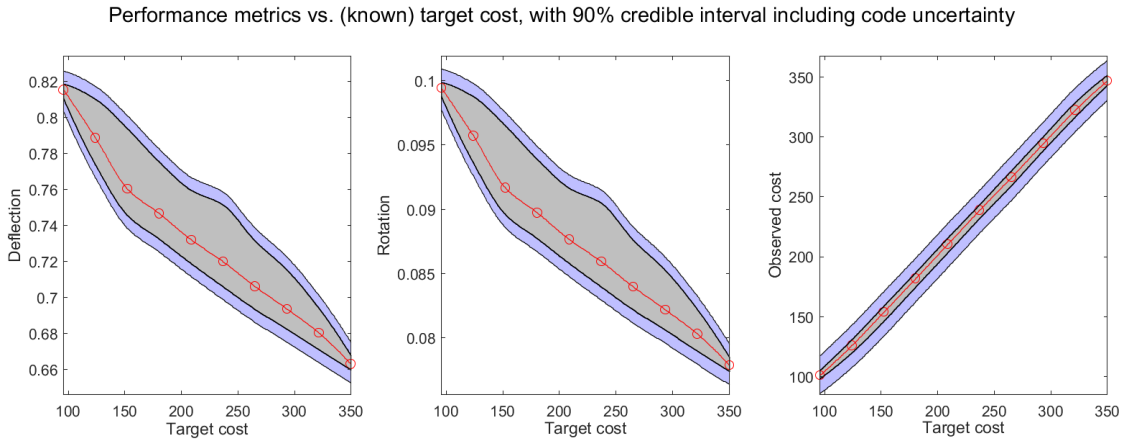

Figure 3: "Pareto bands" of wind turbine blade model outputs over a range of values of cost, where cost is treated as known up to slight deviations, with desired data 0 deflection and 0 rotation. The gray region gives a 90% credible interval only considering parameter uncertainty; the blue region extends this to include code uncertainty.

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 **??** 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

$$\begin{aligned}\pi(\boldsymbol{\theta}, \mathbf{C_y}|\mathcal{D}) &\propto \pi(\mathcal{D}|\boldsymbol{\theta}, \mathbf{C_y}) \times \pi(\boldsymbol{\theta}) \times \pi(\mathbf{C_y}) \\ &\propto \pi(\mathcal{D}|\boldsymbol{\theta}, \mathbf{C_y}) \times \pi(\mathbf{C_y}) \\ &\propto |\mathbf{C}_\mathcal{D}|^{-1/2} \exp(\mathcal{D}^T \mathbf{C}_\mathcal{D}^{-1} \mathcal{D}) \times \prod_{i=1}^{2} \frac{1}{\sigma_i^2}\end{aligned} \tag{16}$$

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 into specific choices of desired observations and observation variance schemes.

The same sort of uncertainty analysis described in Section 4.2 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 3, 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.

References

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

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

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

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

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

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

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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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.

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

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.