Published in final edited form as:

Technometrics. 2009 November 1; 51(4): 464–474. doi:10.1198/TECH.2009.08126.

Simultaneous Determination of Tuning and Calibration Parameters for Computer Experiments

Gang Han,

H. Lee Moffitt Cancer Center & Research Institute, MRC/BIOSAT, 12902 Magnolia Drive, Tampa, FL 33612, (gang.han@moffitt.org)

Thomas J. Santner, and

Department of Statistics, The Ohio State University, 415 Cockins Hall, 1958 Neil Avenue, Columbus, OH 43210, (tjs@stat.osu.edu)

Jeremy J. Rawlinson

C2-527 Clinical Programs Center, Department of Clinical Sciences, College of Veterinary Medicine, Cornell University, Ithaca, NY 14853, (jjr4@cornell.edu)

Abstract

Tuning and calibration are processes for improving the representativeness of a computer simulation code to a physical phenomenon. This article introduces a statistical methodology for simultaneously determining tuning and calibration parameters in settings where data are available from a computer code and the associated physical experiment. Tuning parameters are set by minimizing a discrepancy measure while the distribution of the calibration parameters are determined based on a hierarchical Bayesian model. The proposed Bayesian model views the output as a realization of a Gaussian stochastic process with hyperpriors. Draws from the resulting posterior distribution are obtained by the Markov chain Monte Carlo simulation. Our methodology is compared with an alternative approach in examples and is illustrated in a biomechanical engineering application. Supplemental materials, including the software and a user manual, are available online and can be requested from the first author.

Keywords

Gaussian stochastic process model; Hierarchical Bayesian model; Kriging

1. INTRODUCTION

Complex computer simulation codes (computer experiments) are used extensively in scientific and engineering studies (see Santner, Williams, and Notz 2003; Fang, Li, and Sudjianto 2005; and Forrester, Sobester, and Keane 2008). Many computer codes were developed for applications where the results of physical experiments are also available to describe the true input/output relationship of interest. In this article we are interested in settings where both types of data are available, but the number of computer code runs is limited because they have lengthy run times. In such a case, statistical models are needed to extend the set of inputs at which

© 2009 American Statistical Association and the American Society for Quality

SUPPLEMENTAL MATERIALS

A MATLAB Routine: The supplemental materials include a user guide to this MATLAB package (STaCguide.pdf), the software package (directory STaCprogram), and an example subdirectory having the driver file for running the example in Section 5.3. (STaC_software_version1.0.zip, zip archive)

output is available by predicting the response from the computer and physical experiments based on a limited number of training runs.

Such statistical models must incorporate different types of input variables. This article focuses on computer experiments having three types of inputs: control variables, tuning parameters, and calibration parameters. Control variables are (design) inputs to both the simulation code and the corresponding physical experiment (Santner, Williams, and Notz 2003, p. 15). A primary goal of the physical experiment is to study the relationship between the response and control variables.

Tuning parameters are inputs that are present only in the simulation code. Tuning parameters are typically numerical quantities that control the solution of a numerical algorithm implemented by a computer experiment. Thus, tuning parameters have *no* meaning in the physical experiment. For example, Rawlinson et al. (2006) described a finite element analysis (FEA) computer code simulating the forces and movements of a knee prosthesis under a given loading regimen. The two tuning parameters for their model were the mesh density of the tibial tray and the amount of discretization in the curve used to describe the knee loading. Tuning is the process of determining the values of the tuning parameters so that a computer simulation best represents a physical experiment.

Calibration parameters are also present only in the simulation code. Unlike tuning parameters, calibration parameters have meaning in the physical experiment, but their values are either unknown or unmeasured during the running of the physical experiment. For example, Kennedy and O'Hagan (2001) described a computer code for simulating the deposition of ruthenium 106 in the Tomsk-7 chemical plant, which experienced an accident in 1993. In their code, one calibration parameter was the deposition velocity. Calibration is the process of determining plausible values of the calibration parameters to match the results of the physical experiment.

Some computer experiments have both tuning and calibration parameters. For example, in the FEA code of Rawlinson et al. (2006) two calibration parameters were position of the femur relative to the tibia in the initial gait cycle and friction between the bone and the prosthesis.

Although considerable research has been dedicated to setting either tuning or calibration parameters, there has been little study on the problem of setting both simultaneously. In this article, we introduce a methodology for the simultaneous determination of tuning and calibration parameters. Section 2 reviews current methods for tuning and calibration and discusses their limitations. Section 3 introduces a hierarchical Bayesian model describing the response from a physical experiment together with the output from the computer simulation code having both tuning and calibration parameters. Section 4 proposes a methodology for simultaneously setting these parameters. Section 5 compares the proposed methodology with an approach that treats all parameters as calibration parameters.

2. CURRENT METHODS

Previous proposals for calibration have been primarily Bayesian. Their goal was to simulate the posterior distributions of the calibration parameters and to use this posterior distribution to better predict unknown responses and assess the uncertainty in the prediction. Craig et al. (1996, 2001) developed a *Bayesian linear forecasting method*, which can be seen as a pragmatic approximation to a fully Bayesian analysis. Kennedy and O'Hagan (2001) described a Bayesian calibration framework based on a model having a bias function and a (modular) Bayesian analysis. Higdon et al. (2004) proposed a fully Bayesian implementation for the model in Kennedy and O'Hagan (2001). Higdon et al. (2008) developed a methodology for calibration when the outputs are multivariate. Gattiker (2005) implemented the models in Higdon et al. (2004, 2008) in MATLAB. Loeppky, Bingham, and Welch (2006) proved that

the previous procedures would lead to correct estimation of the calibration parameters asymptotically, provided the true values of the calibration parameters could reduce the bias of the computer simulation to zero. We discuss a lemma in Loeppky, Bingham, and Welch (2006) in Section 5.

Classical methods used to set tuning parameters, such as Park (1991) and Cox, Park, and Singer (1996), select these inputs to make the computer code output fit the physical observations as closely as possible in an integrated prediction error sense. These proposals for selecting tuning parameters do not provide estimated uncertainties associated with the selected values. However, several recent proposals for “validating” computer experiments are similar to the methodology for Bayesian calibration. For example, Bayarri et al. (2007b) presented a framework with a Bayesian model for validation and Bayarri et al. (2007a) used wavelet decomposition based on the same Bayesian model to validate computer experiments having multivariate outputs.

Because the methods above focus on either tuning or calibration, they may suffer limitations in applications where both tuning and calibration parameters are present. Specifically, if one applies classical tuning methodology to a problem involving both tuning and calibration parameters, one cannot quantify the uncertainties of the predicted calibration parameters. If one applies calibration methodology to such a problem and treats all parameters as calibration parameters, one can obtain biased parameter predictions and/or undesirable estimated uncertainties of the tuning and calibration parameters as we will show below. These misleading predictions and estimated uncertainties can lead to large prediction errors.

To demonstrate the limitations alluded to above, we used the Bayesian calibration program (gpmsa) described in Gattiker (2005) to set tuning and calibration parameters for an application in Rawlinson et al. (2006). Originally, the goal of this project was to compare the damage in two knee implants [the Install-Burstein (IB) manufactured by Zimmer, Inc. and the Optetrak produced by Exactech, Inc.] under certain loading regimens.

Here, we consider only the IB knee implant and a physical experiment in which the measurements consist of kinematics and kinetics of knees tested in an Instron-Stanmore KC1 testing device, a “knee simulator.” The loading (magnitude, angle, and rate), joint stiffness, and knee design were modeled in a finite element analysis (FEA) computer code whose output included the kinematics, kinetics, and stresses experienced by the knee component. The output we considered was the anterior–posterior displacement (APD) of the femoral component with respect to the tibial tray, which is an important kinematic output of fore-aft motion during knee function. The APD was measured in both the knee simulator and the computer simulation; it is roughly proportional to the anterior–posterior force that, coupled with the vertical joint load, contributed to damage of the prosthesis. Thus this computer experiment had one control variable (the percentile in gait cycle), two tuning parameters (finite element mesh density and load discretization), and two calibration parameters (friction and initial position).

The design of the computer and the physical experiments were roughly maximin Latin hypercube designs (McKay, Beckman, and Conover 1979). A pilot study of the APD for this design acquired 439 observations from the computer code and 36 observations from the physical experiment for the purpose of tuning and calibration. Figure 1 depicts the training data. A previous sensitivity analysis (Saltelli, Chan, and Scott 2000) found that APD was neither sensitive to friction nor mesh density. Therefore, we study load discretization (a tuning parameter) and initial position (a calibration parameter) regarding the percentage of the gait cycle as a design input to both physical and computer experiments.

Using gpmsa (Gattiker 2005), we treated both load discretization and initial position as calibration parameters whose prior distributions were near uniform distribution. We ran gpmsa

with 8000 burn-in iterations and 2000 production runs, and based our inferences on 100 equally spaced values of each parameter taken from the production runs. Figure 2 shows the simulated posterior distributions of these two parameters. The posterior distribution of load discretization is bimodal and that of initial position has a lower value in its range as mode, but has large variation. Thus, plausible values of the tuning and calibration parameters remain unclear. We demonstrate in Section 5.2 that our methodology can significantly improve the selection of the tuning and calibration parameters in this application.

3. A HIERARCHICAL BAYESIAN MODEL FOR TUNING AND CALIBRATION

We let $\mathbf{x}$ denote (the vector of) control inputs, $\mathbf{t}$ denote the vector of tuning parameters, and $\mathbf{t}^\star$ denote the “best” values of the tuning parameters, where $\mathbf{t}^\star$ will be defined in Section 4. We let $\mathbf{c}$ denote the vector of calibration parameters and $\mathbf{c}^\star$ denote the *unknown* true values of the calibration parameters. We assume $\mathbf{x} \in [0, 1]^{p_x}$, $\mathbf{t} \in [0, 1]^{p_t}$, and $\mathbf{c} \in [0, 1]^{p_c}$ or they can be so scaled. Information about the value of the unknown $\mathbf{c}^\star$ will be contained in its (marginal) prior, which will be denoted by $[\mathbf{c}^\star]$.

We let $\{(\mathbf{x}_i^s, \mathbf{c}_i, \mathbf{t}_i), y^s(\mathbf{x}_i^s, \mathbf{c}_i, \mathbf{t}_i); i=1, 2, \dots, n_s\}$ denote the training data from the computer experiment and $\{(\mathbf{x}_j^p, y^p(\mathbf{x}_j^p)); j=1, 2, \dots, n_p\}$ denote the training data from the physical experiment. Here n_s and n_p are the numbers of runs for each type of data. A more complete notation for the physical experiment response is $y^p(\mathbf{x}, \mathbf{c}^\star)$; throughout this article, $y^p(\mathbf{x})$ and $y^p(\mathbf{x}, \mathbf{c}^\star)$ are equivalent. We let $\mathbf{y}^s$ be the $n_s \times 1$ vector with i th element $y^s(\mathbf{x}_i^s, \mathbf{c}_i, \mathbf{t}_i)$ and $\mathbf{y}^p$ be the $n_p \times 1$ vector with j th element $y^p(\mathbf{x}_j^p)$.

The model proposed here is similar in spirit to the ones in Kennedy and O’Hagan (2001) and Higdon et al. (2005) except that it explicitly incorporates both the tuning and calibration parameters. In other words, the response from the physical experiment is the sum of the “true response” and a random noise; the output from the computer experiment is the difference between the true response and the code bias.

Formally, we assume that the code output at $(\mathbf{x}, \mathbf{c}, \mathbf{t})$, $y^s(\mathbf{x}, \mathbf{c}, \mathbf{t})$, is a realization of the Gaussian stochastic process

$$Y^s(\mathbf{x}, \mathbf{c}, \mathbf{t}) = \mathbf{f}_Z^\top(\mathbf{x}, \mathbf{c}, \mathbf{t})\boldsymbol{\beta}_Z + Z(\mathbf{x}, \mathbf{c}, \mathbf{t}), \quad (1)$$

where $\mathbf{f}_Z^\top(\mathbf{x}, \mathbf{c}, \mathbf{t})$ is a vector of *known* regression functions and $\boldsymbol{\beta}_Z$ is a vector of *unknown* regression parameters. If $Y^s(\bullet)$ is assumed to have a constant mean, then $\mathbf{f}_Z^\top(\mathbf{x}, \mathbf{c}, \mathbf{t})\boldsymbol{\beta}_Z = \beta_{Z,0}$. In Equation (1), $Z(\bullet)$ is a stationary Gaussian stochastic process with mean 0, variance σ_Z^2 , and *product Gaussian correlation*

$$\begin{aligned} & \text{Cor}(Z(\mathbf{x}_1, \mathbf{c}_1, \mathbf{t}_1), Z(\mathbf{x}_2, \mathbf{c}_2, \mathbf{t}_2) | \rho_Z) \\ &= \prod_{i=1}^{p_x} \rho_{Z,x,i}^{4(x_{1,i}-x_{2,i})^2} \times \prod_{j=1}^{p_c} \rho_{Z,c,j}^{4(c_{1,j}-c_{2,j})^2} \times \prod_{k=1}^{p_t} \rho_{Z,t,k}^{4(t_{1,k}-t_{2,k})^2}, \end{aligned} \quad (2)$$

with correlation parameters $\rho_Z = (\rho_{Z,x,1}, \dots, \rho_{Z,x,p_x}, \rho_{Z,c,1}, \dots, \rho_{Z,c,p_c}, \rho_{Z,t,1}, \dots, \rho_{Z,t,p_t})^\top$. Thus $\mathbf{y}^s$ is a realization of the random vector $\mathbf{Y}^s$ whose i th element is $Y^s(\mathbf{x}_i^s, \mathbf{c}_i, \mathbf{t}_i)$.

We regard the output of the physical experiment at $\mathbf{x}$, $y^p(\mathbf{x})$, as a realization of

$$Y^p(\mathbf{x}) = \eta(\mathbf{x}, \mathbf{c}^*) + \varepsilon, \quad (3)$$

where $\eta(\mathbf{x}, \mathbf{c}^*) = E(Y^p(\mathbf{x}))$ is the expected response at $\mathbf{x}$ and ε is a white noise (Gaussian) process with mean 0 and variance σ_ε^2 . Thus $\mathbf{y}^p$ is a realization of the random vector $\mathbf{Y}^p$ whose i th element is $Y^p(\mathbf{x}_i^p)$.

We define

$$\delta(\mathbf{x}, \mathbf{c}, \mathbf{t}) = \eta(\mathbf{x}, \mathbf{c}^*) - y^s(\mathbf{x}, \mathbf{c}, \mathbf{t}) \quad (4)$$

to be the code bias at $(\mathbf{x}, \mathbf{c}, \mathbf{t})$. Our model assumes $\delta(\bullet)$ is a draw from the Gaussian stochastic process

$$\Delta(\mathbf{x}, \mathbf{c}, \mathbf{t}) = \mathbf{f}_D^\top(\mathbf{x}, \mathbf{c}, \mathbf{t})\beta_D + D(\mathbf{x}, \mathbf{c}, \mathbf{t}), \quad (5)$$

where $\mathbf{f}_D(\mathbf{x}, \mathbf{c}, \mathbf{t})$ is a vector of known regression functions and β_D is a vector of unknown regression parameters; if $\Delta(\bullet)$ is assumed to have a constant mean, then $\mathbf{f}_D^\top(\mathbf{x}, \mathbf{c}, \mathbf{t})\beta_D = \beta_{D,0}$. Similar to $Z(\bullet)$, $D(\bullet)$ is defined to be a stationary Gaussian stochastic process with mean 0, variance σ_D^2 , and correlation function

$$\begin{aligned} \text{Cor}(D(\mathbf{x}_1, \mathbf{c}_1, \mathbf{t}_1), D(\mathbf{x}_2, \mathbf{c}_2, \mathbf{t}_2) | \rho_D) \\ = \prod_{i=1}^{p_x} \rho_{D,x,i}^{4(x_{1,i} - x_{2,i})^2} \times \prod_{j=1}^{p_c} \rho_{D,c,j}^{4(c_{1,j} - c_{2,j})^2} \times \prod_{k=1}^{p_t} \rho_{D,t,k}^{4(t_{1,k} - t_{2,k})^2}, \end{aligned} \quad (6)$$

where $\rho_D = (\rho_{D,x,1}, \dots, \rho_{D,x,p_x}, \rho_{D,c,1}, \dots, \rho_{D,c,p_c}, \rho_{D,t,1}, \dots, \rho_{D,t,p_t})^\top$. Finally we assume that, given their parameters, $Z(\bullet)$ and $D(\bullet)$ [hence $Y^s(\bullet)$ and $\Delta(\bullet)$] are mutually *independent* processes. Some authors call $\delta(\bullet)$ the “code discrepancy” or “code inadequacy.” Throughout this article, $\delta(\bullet)$ is referred to as code bias and the word “discrepancy” is used in Section 4.1 to designate an objective function used in the determination of the tuning parameters.

In conclusion, conditional on the parameters $(\beta_z, \sigma_z^2, \rho_z)$, our model assumes that for *given* inputs $(\mathbf{x}, \mathbf{c}, \mathbf{t})$, the simulator output $y^s(\mathbf{x}, \mathbf{c}, \mathbf{t})$ is a draw from

$$Y^s(\mathbf{x}, \mathbf{c}, \mathbf{t}) = \mathbf{f}_z^\top(\mathbf{x}, \mathbf{c}, \mathbf{t})\beta_z + Z(\mathbf{x}, \mathbf{c}, \mathbf{t}). \quad (7)$$

Similarly conditional on unknown parameters, the physical experiment output at control input $\mathbf{x}$, $y^p(\mathbf{x})$, can be viewed as a draw from

$$\begin{aligned} Y^p(\mathbf{x}) &= Y^s(\mathbf{x}, \mathbf{c}^*, \mathbf{t}) = \Delta(\mathbf{x}, \mathbf{c}^*, \mathbf{t}) + \varepsilon \\ &= (\mathbf{f}_z^\top(\mathbf{x}, \mathbf{c}^*, \mathbf{t})\beta_z + Z(\mathbf{x}, \mathbf{c}^*, \mathbf{t})) + (\mathbf{f}_D^\top(\mathbf{x}, \mathbf{c}^*, \mathbf{t})\beta_D + D(\mathbf{x}, \mathbf{c}^*, \mathbf{t}) + \varepsilon), \end{aligned} \quad (8)$$

for any given vector $\mathbf{t}$ of tuning parameters. The unknown parameters being conditioned in Equation (8) are those of $Y^s(\mathbf{x}, \mathbf{c}^*, \mathbf{t})$, $\Delta(\mathbf{x}, \mathbf{c}^*, \mathbf{t})$, and ε , which are $(\mathbf{c}^*, \beta_z, \sigma_z^2, \rho_z)$, $(\mathbf{c}^*, \beta_D, \sigma_D^2, \rho_D)$, and σ_ε^2 , respectively. Thus all conditional statements given

$\mathbf{Y}^s = \mathbf{y}^s$ and $\mathbf{Y}^p = \mathbf{y}^p$ are computed with respect to Equations (7) and (8). Appendix B lists the variance–covariance matrix for the vectors $\mathbf{Y}^s$ and $\mathbf{Y}^p$ and their cross-covariance matrix.

Next we introduce our prior model for the unknown model parameters and derive the posteriors given the tuning parameters $\mathbf{t}$. The proposed priors are similar to the ones in Higdon et al. (2004, 2005), and Gattiker (2005). For a generic parameter (or a vector of parameters) $\mathbf{v}$, we let $[\mathbf{v}]$ denote the (joint) prior distribution.

We assume that the priors for the calibration parameters, $\mathbf{c}^\star$, the process means,

$\beta = (\beta_z^\top, \beta_d^\top)^\top$, the process variances, $\sigma = (\sigma_z^2, \sigma_d^2, \sigma_\varepsilon^2)^\top$, and process correlations, $\rho = (\rho_z^\top, \rho_d^\top)^\top$, are independent so that

$$[\mathbf{c}^\star, \beta, \sigma, \rho] = [\mathbf{c}^\star] \times [\beta] \times [\sigma] \times [\rho]. \quad (9)$$

If expert knowledge about the priors is available it should be used. For example, if the researcher knows (roughly) the magnitude of the noise term, it should be used to construct the prior for σ_ε^2 . Lacking such knowledge, our methodology is based on the following priors.

The examples and software take $\mathbf{f}_d^\top(\bullet)\beta_z = \beta_{z,0}$ and $\mathbf{f}_d^\top(\bullet)\beta_d = \beta_{d,0}$. Our methodology for selecting $(\beta_{z,0}, \beta_{d,0})$ is modular in the spirit of Bayarri et al. (2007a) in that these mean parameters are determined by non-Bayesian arguments in Section 4.2. All other parameters are determined from draws of the posterior.

Because the calibration parameters are scaled to $[0, 1]^{p_c}$, all the elements of $\mathbf{c}^\star$ are taken to have independent and identical truncated normal distributions with mean 0.5, standard deviation 2, and support $[0, 1]$. This prior is close to a uniform distribution on $[0, 1]^{p_c}$.

The variance parameters are taken to have independent inverse Gamma prior distributions. Letting $\text{IG}(\alpha, \gamma)$ denote the inverse Gamma distribution with mean $1/[\gamma(\alpha - 1)]$ and variance $1/[\gamma^2(\alpha - 1)^2(\alpha - 2)]$, we take $\sigma_z^2 \sim \text{IG}(\alpha_z, \gamma_z)$, $\sigma_\varepsilon^2 \sim \text{IG}(\alpha_\varepsilon, \gamma_\varepsilon)$, and $\sigma_d^2 \sim \text{IG}(\alpha_d, \gamma_d)$ where the parameters are set data-adaptively, as follows. We first compute the sample variances of $\mathbf{y}^s$ and $\mathbf{y}^p$, which we denote by $\widehat{\sigma}_s^2$ and $\widehat{\sigma}_p^2$ respectively. We take σ_z^2 to have a (mildly) informative prior distribution by setting $(\alpha_z, \gamma_z) = (10, 0.1/\widehat{\sigma}_s^2)$. We select the σ_ε^2 prior to have a small magnitude and variation by taking $(\alpha_\varepsilon, \gamma_\varepsilon) = (1, 1000/\widehat{\sigma}_p^2)$ as the default. The supplemental software package allows the user to specify a multiplier for σ_ε^2 so that the program can suit physical experiments with random noise of different magnitudes. We use the value of $(\widehat{\sigma}_p^2 - \widehat{\sigma}_s^2)$ to determine the prior for σ_d^2 : if $\widehat{\sigma}_p^2 > \widehat{\sigma}_s^2$, σ_d^2 is set to have a mildly informative prior by taking $(\alpha_d, \gamma_d) = (10, 0.1/(\widehat{\sigma}_p^2 - \widehat{\sigma}_s^2))$ while $\widehat{\sigma}_p^2 \leq \widehat{\sigma}_s^2$, σ_d^2 is set to have a prior with small magnitude and variation by selecting $(\alpha_d, \gamma_d) = (1, 100/\widehat{\sigma}_s^2)$.

Finally, we take the priors of the correlation parameters to be independently and identically distributed Beta(1, 0.5) distributions. This Beta distribution has mean 2/3, mode close to 1, and a rather diffuse support over (0, 1); it encourages dependence of the output on each input, but the broad support allows the data a strong voice in determining the posteriors (Linkletter et al. 2006 and Higdon et al. 2008).

Let $\phi = (\beta, \sigma, \rho)$ [or $\phi = (\sigma, \rho)$ when β is fixed as below] and $[\mathbf{c}^\star, \phi]$ denote the joint prior of $\mathbf{c}^\star$ and ϕ . For a fixed $\mathbf{t}$ and given the data $(\mathbf{y}^s, \mathbf{y}^p)$, the joint posterior distribution can be written as

$$[\mathbf{c}^*, \phi | \mathbf{y}^s, \mathbf{y}^p, \mathbf{t}] \propto [\mathbf{y}^p, \mathbf{y}^s | \mathbf{c}^*, \phi, \mathbf{t}] \times [\mathbf{c}^*, \phi, \mathbf{t}] \quad (10)$$

$$\propto [\mathbf{y}^p, \mathbf{y}^s | \mathbf{c}^*, \phi, \mathbf{t}] \times [\mathbf{c}^*, \phi]. \quad (11)$$

(See Appendix A for details of the posterior distribution.) We simulate draws from $[\mathbf{c}^*, \phi | \mathbf{y}^s, \mathbf{y}^p, \mathbf{t}]$ by implementing a Metropolis–Hastings (MH) algorithm. Appendix A sketches this algorithm (see Higdon et al. 2004, 2005 for details about related MH algorithms). Using this model and the relationship in Equations (10) and (11), draws can be made from the joint posterior distribution $[\mathbf{c}^*, \phi | \mathbf{y}^s, \mathbf{y}^p, \mathbf{t}]$ which forms the basis for our proposed methodology for simultaneous tuning and calibration in Section 4.

4. METHODOLOGY FOR SIMULTANEOUS TUNING AND CALIBRATION

4.1 The Discrepancy Function

Our definition of the ideal tuning parameters is based on a function that measures the discrepancy between the physical experiment response and the computer code output. The discrepancy is a function of $(\mathbf{c}, \mathbf{t})$, but can be viewed as a function of $\mathbf{t}$ for given $\mathbf{c} = \mathbf{c}^*$. Three of the many possible discrepancy functions are L_2 discrepancy, $\int (\eta(\mathbf{x}, \mathbf{c}^*) - y^s(\mathbf{x}, \mathbf{c}, \mathbf{t}))^2 d\mathbf{x}$, L_1 discrepancy, $\int |\eta(\mathbf{x}, \mathbf{c}^*) - y^s(\mathbf{x}, \mathbf{c}, \mathbf{t})| d\mathbf{x}$, and L_∞ discrepancy, $\max_{\mathbf{x}} |\eta(\mathbf{x}, \mathbf{c}^*) - y^s(\mathbf{x}, \mathbf{c}, \mathbf{t})|$; here (and below) the integrals are over $[0, 1]$ raised to an appropriate power. In this article, we use L_2 discrepancy, but our methodology can be applied with any other discrepancy function.

If $\mathbf{c}^*$ is known, the L_2 discrepancy is defined by

$$\int (\eta(\mathbf{x}, \mathbf{c}^*) - y^s(\mathbf{x}, \mathbf{c}^*, \mathbf{t}))^2 d\mathbf{x} = \int \delta^2(\mathbf{x}, \mathbf{c}^*, \mathbf{t}) d\mathbf{x}, \quad (12)$$

regarded as a function of $\mathbf{t}$. If $\mathbf{c}^*$ is unknown, but has a posterior distribution $[\mathbf{c}^* | \mathbf{y}^s, \mathbf{y}^p, \mathbf{t}]$, the L_2 discrepancy becomes

$$\int \int \delta^2(\mathbf{x}, \mathbf{c}^*, \mathbf{t}) [\mathbf{c}^* | \mathbf{y}^s, \mathbf{y}^p, \mathbf{t}] d\mathbf{c}^* d\mathbf{x}.$$

Our objective in tuning is to estimate

$$\mathbf{t}^* = \underset{\mathbf{t}}{\operatorname{argmin}} \int \int \delta^2(\mathbf{x}, \mathbf{c}^*, \mathbf{t}) [\mathbf{c}^* | \mathbf{y}^s, \mathbf{y}^p, \mathbf{t}] d\mathbf{c}^* d\mathbf{x}. \quad (13)$$

Our objective in calibration is to compute the posterior distribution $[\mathbf{c}^* | \mathbf{y}^s, \mathbf{y}^p, \mathbf{t}^*]$. We achieve the two objectives by minimizing an estimated squared discrepancy (defined in Section 4.2) and simulating the posterior distribution of $\mathbf{c}^*$ given the minimizer.

4.2 Simultaneous Tuning and Calibration

To review, the main idea is to choose the most plausible values of the tuning parameters so that the computer code can best represent the physical experiment and use these tuning parameters in the inferences about calibration parameters. Next, we describe our method for estimating $\delta^2(\mathbf{x}, \mathbf{c}^*, \mathbf{t})$, $\mathbf{t}^*$, and $[\mathbf{c}^* | \mathbf{y}^s, \mathbf{y}^p, \mathbf{t}^*]$ which we call the STaC procedure, which stands

for simultaneous tuning and calibration. Then we summarize the methodology in a three step procedure.

Our first of the three steps is to estimate $\delta^2(\mathbf{x}, \mathbf{c}^*, \mathbf{t})$ by the posterior mean $E(\Delta^2(\mathbf{x}, \mathbf{c}^*, \mathbf{t})|\mathbf{y}^s, \mathbf{y}^p, \mathbf{t})$, which is the Bayes predictor of $\delta^2(\mathbf{x}, \mathbf{c}^*, \mathbf{t})$ under *squared error loss* (Santner, Williams, and Notz 2003). Notice that $E(\Delta^2(\mathbf{x}, \mathbf{c}^*, \mathbf{t})|\mathbf{c}^*, \phi, \mathbf{y}^s, \mathbf{y}^p, \mathbf{t})$ is equal to

$$[E(\Delta(\mathbf{x}, \mathbf{c}^*, \mathbf{t})|\mathbf{c}^*, \phi, \mathbf{y}^s, \mathbf{y}^p, \mathbf{t})]^2 + \text{Var}(\Delta(\mathbf{x}, \mathbf{c}^*, \mathbf{t})|\mathbf{c}^*, \phi, \mathbf{y}^s, \mathbf{y}^p, \mathbf{t}). \quad (14)$$

Closed form expressions for both $E(\Delta(\mathbf{x}, \mathbf{c}^*, \mathbf{t})|\mathbf{c}^*, \phi, \mathbf{y}^s, \mathbf{y}^p, \mathbf{t})$ and $\text{Var}(\Delta(\mathbf{x}, \mathbf{c}^*, \mathbf{t})|\mathbf{c}^*, \phi, \mathbf{y}^s, \mathbf{y}^p, \mathbf{t})$ can be derived analytically based on the probability density of the multivariate normal distribution (see Appendix B).

The two terms in Equation (14) can be interpreted intuitively, as follows. Large values of $[E(\Delta(\mathbf{x}, \mathbf{c}^*, \mathbf{t})|\mathbf{c}^*, \phi, \mathbf{y}^s, \mathbf{y}^p, \mathbf{t})]^2$ imply that there is a large discrepancy between the computer and the physical experiments; thus $[E(\Delta(\mathbf{x}, \mathbf{c}^*, \mathbf{t})|\mathbf{c}^*, \phi, \mathbf{y}^s, \mathbf{y}^p, \mathbf{t})]^2$ can be interpreted as *squared prediction bias*. The term $\text{Var}(\Delta(\mathbf{x}, \mathbf{c}^*, \mathbf{t})|\mathbf{c}^*, \phi, \mathbf{y}^s, \mathbf{y}^p, \mathbf{t})$ is a measure of the uncertainty of the predicted bias. Thus Equation (14) favors a $\mathbf{t}$ that minimizes the sum of the squared prediction bias and its predicted uncertainty.

The right-hand side of

$$\begin{aligned} & E(\Delta^2(\mathbf{x}, \mathbf{c}^*, \mathbf{t})|\mathbf{y}^s, \mathbf{y}^p, \mathbf{t}) \\ &= E_{[\mathbf{c}^*, \phi|\mathbf{y}^s, \mathbf{y}^p, \mathbf{t}]}[E(\Delta^2(\mathbf{x}, \mathbf{c}^*, \mathbf{t})|\mathbf{c}^*, \phi, \mathbf{y}^s, \mathbf{y}^p, \mathbf{t})], \end{aligned}$$

can be approximated using Equation (14), and based on draws $(\widehat{\mathbf{c}}_\ell^*, \widehat{\phi}_\ell)$ from $[\mathbf{c}^*, \phi|\mathbf{y}^s, \mathbf{y}^p, \mathbf{t}]$, which can be obtained using the MH algorithm. The law of large numbers guarantees that

$$\begin{aligned} & E(\Delta^2(\mathbf{x}, \mathbf{c}^*, \mathbf{t})|\mathbf{y}^s, \mathbf{y}^p, \mathbf{t}) \\ & \approx \frac{1}{N_{MC}} \sum_{\ell=1}^{N_{MC}} E(\Delta^2(\mathbf{x}, \widehat{\mathbf{c}}_\ell^*, \mathbf{t})|\widehat{\mathbf{c}}_\ell^*, \widehat{\phi}_\ell, \mathbf{y}^s, \mathbf{y}^p, \mathbf{t}), \end{aligned} \quad (15)$$

for a sufficiently large N_{MC} . Thus, we can approximate $E(\Delta^2(\mathbf{x}, \mathbf{c}^*, \mathbf{t})|\mathbf{y}^s, \mathbf{y}^p, \mathbf{t})$ asymptotically.

The estimation of $\delta^2(\mathbf{x}, \mathbf{c}^*, \mathbf{t})$ using Equation (15) requires knowledge of $(\beta_{Z,0}, \beta_{D,0})$; these parameters are set as follows. The parameter $\beta_{D,0}$ can be interpreted as the difference between the average of the physical experiment and the average of the computer experiment outputs when the calibration and tuning parameters are equal to $(\mathbf{c}^*, \mathbf{t})$. The value of $\beta_{D,0}$ can impact both calibration and tuning and thus must be carefully chosen. Assuming expert knowledge about the difference $\beta_{D,0}$ is not available, we suggest setting $\beta_{D,0}$ based on the assumption that, for any given $\mathbf{t}$, the code inadequacy is decreased as $\mathbf{c}$ gets closer to $\mathbf{c}^*$ so that $\beta_{D,0}$ becomes smaller. Thus $\beta_{D,0}$ is minimized when $\mathbf{c} = \mathbf{c}^*$. Hence we take $\beta_{D,0}$ to be the minimized difference, over $\mathbf{c}$ values, between the sample mean of the n_p physical experiment responses and the sample mean of n_p predicted code outputs whose corresponding inputs are

$\{(\mathbf{x}_j^p, \mathbf{c}, \mathbf{t})\}_{j=1}^{n_p}$, where $\{\mathbf{x}_j^p\}_{j=1}^{n_p}$ are the inputs to the physical experiment and $\mathbf{t}$ is given. The $\mathbf{c}$ minimization must be over the calibration parameter input space. Operationally, this is accomplished by using a space-filling maximin Latin hypercube design to select a set of n_s $\mathbf{c}$ values over which the minimization is carried out. For each choice of $\mathbf{c}$, we predict n_p outputs

$\{y^s(\mathbf{x}_j^p, \mathbf{c}, \mathbf{t})\}_{j=1}^{n_p}$ based on the training data from the computer experiment using a MATLAB version of the parametric empirical kriging (PERK) program in Santner, Williams, and Notz (2003) and Appendix C. We then compute the difference between the sample mean of $\{y^p(\mathbf{x}_j^p)\}_{j=1}^{n_p}$ and the sample mean of the predicted $\{y^s(\mathbf{x}_j^p, \mathbf{c}, \mathbf{t})\}_{j=1}^{n_p}$. Repeating this operation with each choice of $\mathbf{c}$ yields n_s differences. We set $\beta_{D,0}$ to the smallest difference and then set $\beta_{Z,0}$ so that $\beta_{Z,0} + \beta_{D,0}$ is the average of the n_p physical experiment responses.

In the second step, we estimate $\mathbf{t}^*$ by

$$\begin{aligned} \widehat{\mathbf{t}}^* &= \underset{\mathbf{t}}{\operatorname{argmin}} \int \int E[\Delta^2(\mathbf{x}, \mathbf{c}^*, \mathbf{t}) | \mathbf{y}^s, \mathbf{y}^p, \mathbf{t}] \times [\mathbf{c}^*, \boldsymbol{\phi} | \mathbf{y}^s, \mathbf{y}^p, \mathbf{t}] d\mathbf{c}^* d\mathbf{x} \\ &\approx \underset{\mathbf{t}}{\operatorname{argmin}} \int \frac{1}{N_{MC}} \times \sum_{\ell=1}^{N_{MC}} E[\Delta^2(\mathbf{x}, \widehat{\mathbf{c}}_\ell^*, \mathbf{t}) | \widehat{\mathbf{c}}_\ell^*, \widehat{\boldsymbol{\phi}}_\ell, \mathbf{y}^s, \mathbf{y}^p, \mathbf{t}] d\mathbf{x}. \end{aligned} \quad (16)$$

The integral over $\mathbf{x}$ in Equation (16) can be approximated by a further Monte Carlo integration that averages Equation (15) over a grid of $\mathbf{x}$ inputs. In our examples, we took

$$\widehat{\mathbf{t}}^* \approx \underset{\mathbf{t}}{\operatorname{argmin}} \frac{1}{N_x} \frac{1}{N_{MC}} \times \sum_{i=1}^{N_x} \sum_{\ell=1}^{N_{MC}} E[\Delta^2(\mathbf{x}_i, \widehat{\mathbf{c}}_\ell^*, \mathbf{t}) | \widehat{\mathbf{c}}_\ell^*, \widehat{\boldsymbol{\phi}}_\ell, \mathbf{y}^s, \mathbf{y}^p, \mathbf{t}], \quad (17)$$

with $N_x = 101$ and an equally spaced grid of values of $\mathbf{x}$. By the law of large numbers, $\widehat{\mathbf{t}}^*$ converges to $\mathbf{t}^*$ almost surely as N_x and N_{MC} go to infinity. We regard $\widehat{\mathbf{t}}^*$ in Equation (17) as the estimator of $\mathbf{t}^*$ and the quantity $\sum_{i=1}^{N_x} \sum_{\ell=1}^{N_{MC}} E[\Delta^2(\mathbf{x}_i, \widehat{\mathbf{c}}_\ell^*, \mathbf{t}) | \widehat{\mathbf{c}}_\ell^*, \widehat{\boldsymbol{\phi}}_\ell, \mathbf{y}^s, \mathbf{y}^p, \mathbf{t}] / (N_x N_{MC})$ as the *estimated squared discrepancy*.

In the third step, we estimate the posterior distribution $[\mathbf{c}^* | \mathbf{y}^s, \mathbf{y}^p, \mathbf{t}^*]$. Because $\widehat{\mathbf{t}}^*$ converges to $\mathbf{t}^*$ almost surely, $[\mathbf{c}^* | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*]$ converges to $[\mathbf{c}^* | \mathbf{y}^s, \mathbf{y}^p, \mathbf{t}^*]$ in distribution. We therefore estimate the distribution $[\mathbf{c}^* | \mathbf{y}^s, \mathbf{y}^p, \mathbf{t}^*]$ using the draws of $\mathbf{c}^*$ from $[\mathbf{c}^*, \boldsymbol{\phi} | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*]$.

In summary, we conduct simultaneous tuning and calibration as follows:

Step 1. For each possible $\mathbf{t}$ in a grid of the tuning parameter vectors, set the values of $(\beta_{Z,0}, \beta_{D,0})$, make draws $\{(\widehat{\mathbf{c}}_\ell^*, \widehat{\boldsymbol{\phi}}_\ell); \ell=1, \dots, N_{MC}\}$ from $[\mathbf{c}^*, \boldsymbol{\phi} | \mathbf{y}^s, \mathbf{y}^p, \mathbf{t}]$, and compute the estimated squared discrepancy at a set of inputs $\{\mathbf{x}_i; i=1, \dots, N_x\}$.

Step 2. Compute $\widehat{\mathbf{t}}^* = \underset{\mathbf{t}}{\operatorname{argmin}} \frac{1}{N_x} \frac{1}{N_{MC}} \sum_{i=1}^{N_x} \sum_{\ell=1}^{N_{MC}} E(\Delta^2(\mathbf{x}_i, \widehat{\mathbf{c}}_\ell^*, \mathbf{t}) | \widehat{\mathbf{c}}_\ell^*, \widehat{\boldsymbol{\phi}}_\ell, \mathbf{y}^s, \mathbf{y}^p, \mathbf{t})$

Step 3. Estimate $[\mathbf{c}^* | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*]$.

4.3 Prediction

In addition to tuning and calibration, we can use $\widehat{\mathbf{t}}^*$ and the draws $\{(\widehat{\mathbf{c}}_\ell^*, \widehat{\boldsymbol{\phi}}_\ell)\}_\ell$ from $[\mathbf{c}^*, \boldsymbol{\phi} | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*]$ to predict $y^s(\bullet)$ and $\eta(\bullet)$ at arbitrary inputs and to construct predictive intervals for these quantities. Given $(\mathbf{x}, \mathbf{c}, \mathbf{t})$, the best linear unbiased prediction (BLUP) of $y^s(\mathbf{x}, \mathbf{c}, \mathbf{t})$ is $E(Y^s(\mathbf{x}, \mathbf{c}, \mathbf{t}) | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*) = E_{[\mathbf{c}^*, \boldsymbol{\phi} | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*]}[E(Y^s(\mathbf{x}, \mathbf{c}, \mathbf{t}) | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*, \mathbf{c}^*, \boldsymbol{\phi})]$. Thus, by the law of large numbers,

$$\widehat{y^s}(\mathbf{x}, \mathbf{c}, \mathbf{t}) = \frac{1}{N_{MC}} \sum_{\ell=1}^{N_{MC}} E[Y^s(\mathbf{x}, \mathbf{c}, \mathbf{t}) | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*, \widehat{\mathbf{c}}_\ell^*, \widehat{\boldsymbol{\phi}}_\ell^*] \quad (18)$$

is an approximation to the BLUP of $y^s(\mathbf{x}, \mathbf{c}, \mathbf{t})$ for large N_{MC} . Similarly, given $\mathbf{x}$,

$$\widehat{\eta}(\mathbf{x}, \mathbf{c}^*) = \frac{1}{N_{MC}} \sum_{\ell=1}^{N_{MC}} E(Y^s(\mathbf{x}, \widehat{\mathbf{c}}_\ell^*, \widehat{\mathbf{t}}^*) + \Delta(\mathbf{x}, \widehat{\mathbf{c}}_\ell^*, \widehat{\mathbf{t}}^*) | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*, \widehat{\mathbf{c}}_\ell^*, \widehat{\boldsymbol{\phi}}_\ell^*) \quad (19)$$

is an approximation to the BLUP of $\eta(\mathbf{x}, \mathbf{c}^*)$ by Equations (3) through (5).

We quantify the uncertainty in the $\widehat{y^s}(\mathbf{x}, \mathbf{c}, \mathbf{t})$ predictor using

$$\begin{aligned} & \text{Var}(Y^s(\mathbf{x}, \mathbf{c}, \mathbf{t}) | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*) \\ &= E_{[\mathbf{c}^*, \boldsymbol{\phi} | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*]} [\text{Var}(Y^s(\mathbf{x}, \mathbf{c}, \mathbf{t}) | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*, \mathbf{c}^*, \boldsymbol{\phi})] + \text{Var}_{[\mathbf{c}^*, \boldsymbol{\phi} | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*]} [E(Y^s(\mathbf{x}, \mathbf{c}, \mathbf{t}) | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*, \mathbf{c}^*, \boldsymbol{\phi})], \end{aligned} \quad (20)$$

which can be approximated using draws from the MH algorithm. We provide details in Appendix C. Similarly, we quantify the uncertainty in the $\widehat{\eta}(\mathbf{x}, \mathbf{c}^*)$ predictor using $\text{Var}(Y^s(\mathbf{x}, \mathbf{c}^*, \widehat{\mathbf{t}}^*) + \Delta(\mathbf{x}, \mathbf{c}^*, \widehat{\mathbf{t}}^*) | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*)$. Details of this computation are also given in Appendix C.

Based on Equations (18), (19), (20), and the results in Appendix C, we can compute point-wise $100(1 - \alpha)\%$ prediction bands for $y^s(\mathbf{x}, \mathbf{c}, \mathbf{t})$ and $\eta(\mathbf{x}, \mathbf{c}^*)$ as

$$\widehat{y^s}(\mathbf{x}, \mathbf{c}, \mathbf{t}) \pm z^{\alpha/2} \sqrt{\text{Var}(Y^s(\mathbf{x}, \mathbf{c}, \mathbf{t}) | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*)}$$

and

$$\widehat{\eta}(\mathbf{x}, \mathbf{c}^*) \pm z^{\alpha/2} \sqrt{\text{Var}(Y^s(\mathbf{x}, \mathbf{c}^*, \widehat{\mathbf{t}}^*) + \Delta(\mathbf{x}, \mathbf{c}^*, \widehat{\mathbf{t}}^*) | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*)},$$

where $z^{\alpha/2}$ is the upper $\alpha/2$ critical point of the standard normal distribution.

5. EXAMPLES AND COMPARISON

5.1 Introduction

In this section, we compare Bayesian calibration as implemented by the gpmsa software with STaC. Recall that the former performs a fully Bayesian calibration for the tuning as well as the calibration parameters, whereas STaC sets the tuning parameters to minimize a predicted L_2 discrepancy between the computer and the physical experiment outputs. Lemma 1 in Loepky, Bingham, and Welch (2006) shows that, roughly, for their Gaussian stochastic process model, under the conditions (a) the discrepancy between the computer and the physical experiments can be minimized to be zero and (b) infinitely many runs from the computer simulation and the physical experiment are available, the maximum likelihood estimator (MLE) of the discrepancy is zero and the MLE of the calibration parameters minimizes the discrepancy asymptotically. Based on this lemma, Bayesian calibration and STaC can give

comparable estimates of $t^\star$ in our model if (A) the discrepancy $\delta(\bullet)$ can be made sufficiently close to zero, (B) the number of observations from the computer experiment is reasonably large, and (C) the posterior mode of every calibration parameter is close to the MLE of the parameter.

However, (A), (B), and (C) can be hard or impossible to verify/implement in many applications for at least three reasons. It may not be possible to drive the discrepancy $\delta(\bullet)$ to zero because of the simplified physics or biology used to construct the simulation code. The number of computer and physical experiment runs are typically limited. The posterior mode of a calibration parameter can differ from the MLE if either the prior for $c^\star$ or $\Delta(\bullet)$ is informative. Therefore, the Bayesian calibration approach may fail to minimize the difference between the computer and the physical experiments. Next we compare STaC with gpmsa in a numerical example where the bias can be minimized to zero, yet the training data are scarce.

5.2 An Illustrative Example

This example has three real-valued inputs to a computer code: one control variable x , one tuning parameter t , and one calibration parameter c . All three inputs have the support $[0, 1]$. The output from the computer code was generated as

$$y^s(x, c, t) = x^2 + c \times (x+1) + 0.5 \times t,$$

and the response from the physical experiment as

$$y^p(x) = x^2 + 0.1 \times x + 0.5 \times \varepsilon,$$

where ε denotes noise generated from normal distribution having mean zero and standard deviation 0.01. In this example the bias term can be driven to zero by setting $c = 0.1$ and $t = 0.8$. Thus any approach that suggests $t^\star = 0.8$ and $c^\star = 0.1$ is favorable.

We generated 30 inputs for the computer experiment and 5 for the physical experiment with both designs being maximin Latin hypercube designs. The Bayesian calibration program gpmsa was run treating c and t as calibration parameters. Following 8000 burn-in iterations and 2000 production ones, we regarded 100 equally spaced samples drawn from the 2000 production runs as draws from the joint posterior distribution of $(t^\star, c^\star)$. The estimated posterior distributions of $t^\star$ and $c^\star$ are shown in Figure 3. The estimated $[t^\star | y^p, y^s]$ is not able to pinpoint that $t^\star = 0.8$ and the estimated $[c^\star | y^p, y^s]$ fails to suggest values close to 0.1. We next used gpmsa to predict $\eta(x, c^\star)$ over the grid, $x = 0, 0.02, \dots, 1$, of equally spaced inputs. Figure 4 depicts the training data, the true response curve, and the predictions. We see that the predictions are biased when x is near 0. The root mean squared prediction error (RMSPE) was computed as a measure of the predictive accuracy; the RMSPE of the generic predictor $\hat{\eta}(\bullet)$

with respect to the x grid above is defined to be $\sqrt{\frac{1}{51} \sum_{i=1}^{51} (\hat{\eta}(x_i, c^\star) - \eta(x_i, c^\star))^2}$. The RMSPE of the predictor obtained by gpmsa is 0.0693.

We applied STaC to this example with $N_{MC} = 100$ and $N_x = 101$. The number of iterations in the burn-in period, the number of production runs, the sampling space, and the prediction inputs are the same as used for gpmsa. The left panel in Figure 5 plots the L_2 discrepancy against t , for $t = 0, 0.1, \dots, 0.9, 1$; thus, STaC picks $\hat{t}^\star = 0.8$. The right panel in Figure 5 shows the histogram of the simulated posterior distributions of $c^\star$ when $t^\star = 0.8$. The posterior distribution has a center near 0.1. By running the program multiple times, we found that the smallest discrepancy occurred when t was between $[0.7, 0.9]$ and the posterior distribution of c was centered at 0.1.

Figure 6 plots the predictions and the 99% two-sided prediction bands for true input-output function $\eta(x, c^*)$ using STaC. We see that the predictions are close to the true responses and the 99% prediction band contains the true curve for all $x \in [0, 1]$. The RMSPE of the predictor obtained by the STaC program is 0.0382. The relative improvement of STaC compared with the gpmsa-based predictor is 44.8% $[(0.0693 - 0.0382)/0.0693 \times 100\%]$.

Thus, for this example, the inferences about the tuning and calibration parameters and the predictions with STaC are more informative and more accurate than using a fully Bayesian procedure.

5.3 A Biomechanics Example

We apply STaC to the biomechanics application described in Section 2. The number of burn-in and production runs are identical to those used in our first example. The left panel of Figure 7 plots the estimated squared discrepancies for the tuning parameter equal to 15, 16, ..., 30 (with $N_{MC} = 100$ and $N_x = 101$); this figure shows that the computer simulation best matches the knee simulator when the load discretization is set to $\hat{t}^* = 19$. The right panel shows the histogram of the simulated posterior distribution $[c^* | y^p, y^s, \hat{t}^* = 19]$. The simulated posterior distribution of the initial position has mode -1.7 with substantially smaller uncertainty than does the Bayesian calibration in Figure 2 (right panel). Thus STaC indicates that the true initial position is on or below the lower bound of the current design space. These results can be used for future runs of the FEA code to study the anterior-posterior displacement.

Figure 8 depicts the predictions and 99% prediction bands for the anterior-posterior displacement in the physical experiment using the current data. Once the initial position and load discretization are set using STaC, the predictions clearly match those provided by the knee simulator.

6. SUMMARY AND DISCUSSION

This article introduces a modular Bayesian methodology for simultaneous tuning and calibration (STaC). It demonstrates, with examples, that STaC performs well for tuning, calibration, and prediction of the true input-output relationship. The examples also show that treating tuning and calibration parameters differentially is beneficial whether or not the code bias can be reduced to zero.

There are several areas of continuing research interest concerning STaC. Currently, determining the tuning parameters in the first step of STaC can be time consuming; additional work on speeding up the computation is needed. Second, a measure of the uncertainty in the estimated tuning parameter $\hat{t}^*$, perhaps based on the curvature of the integral

$\int_0^1 \int_0^1 \delta^2(x, c^*, t) [c^* | y^s, y^p, t] dc^* dx$, is needed. Third, it is of interest to extend STaC to other settings, for example, to cases where outputs are multivariate and to applications where both the computer code and the physical experiment have mixed quantitative and qualitative inputs. Fourth, there are other estimates of δ that can produce superior results for tuning and thus deserve further investigation. For example, another estimate of $\delta(x, c^*, t)$ is $E(Y^p(x) | c^*, \phi, y^s, y^p, t) - E(Y^s(x, c^*, t) | c^*, \phi, y^s, t)$, where the second expectation is not conditioning on y^p . If some of the tuning parameters are continuous, another extension of the current approach is to model the response surface of the estimated squared discrepancy and identify $\hat{t}^*$ using the response surface.

Lastly, we sketch two research areas related to calibration and tuning. One is to determine better experimental designs for the inputs to the computer and physical experiments when STaC is the objective, i.e., inputs $\{(x_i^s, c_i, t_i); i=1, 2, \dots, n_s\}$. We use two separate Latin hypercube

designs for the physical and computer experiments in our examples. We believe nested Latin hypercube designs, for example, orthogonal array based nested Latin hypercube designs proposed by Qian, Ai, and Wu (2009), Qian (2009), and Qian and Wu (2009), may better determine tuning and calibration parameters. The second research area is to solve the potential issue of nonidentifiability. Note that tuning parameters, calibration parameters, and the code bias δ may be correlated. For example, if $y^p(x) = x^2 + 1 + \varepsilon$ and $y^s(x, c, t) = x^2 + c + t$, then c , t , and δ can be three arbitrary real values whose sum is 1. The detection and elimination of the nonidentifiability will be important for calibration, tuning, and validation for complex computer models.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

APPENDIX A: IMPLEMENTATION OF THE BAYESIAN MODEL

The posterior density is proportional to the product of the joint prior density $[\mathbf{c}^*, \boldsymbol{\phi} | \mathbf{t}]$ and the likelihood $[\mathbf{y}^s, \mathbf{y}^p | (\mathbf{c}^*, \boldsymbol{\phi}, \mathbf{t})]$. We can determine $[\mathbf{c}^*, \boldsymbol{\phi} | \mathbf{t}]$ by computing the prior densities of $[\mathbf{c}^*]$, $[\boldsymbol{\beta}]$, $[\boldsymbol{\sigma}]$, and $[\boldsymbol{\rho}]$ and applying Equation (9). The density $[\mathbf{y}^s, \mathbf{y}^p | (\mathbf{c}^*, \boldsymbol{\phi}, \mathbf{t})]$ is analytically derivable because $\mathbf{y}^s$ and $\mathbf{y}^p$ are viewed as observations from Gaussian stochastic processes.

Specifically, our model is $Y^s(\mathbf{x}_i^s, \mathbf{c}_i, \mathbf{t}_i) = \mathbf{f}_z^T(\mathbf{x}_i^s, \mathbf{c}_i, \mathbf{t}_i)\boldsymbol{\beta}_z + Z(\mathbf{x}_i^s, \mathbf{c}_i, \mathbf{t}_i)$ and $Y^p(\mathbf{x}_j^p) = Y^s(\mathbf{x}_j^p, \mathbf{c}^*, \mathbf{t}) + \Delta(\mathbf{x}_j^p, \mathbf{c}^*, \mathbf{t}) + \varepsilon$ for all $i = 1, \dots, n_s$ and $j = 1, \dots, n_p$. The mean values of $Y^p(\mathbf{x}_j^p)$ and $Y^s(\mathbf{x}_i^s, \mathbf{c}_i, \mathbf{t}_i)$ are $\mathbf{f}_z^T(\mathbf{x}_i^s, \mathbf{c}_i, \mathbf{t}_i)\boldsymbol{\beta}_z + \mathbf{f}_z^T(\mathbf{x}_i^s, \mathbf{c}_i, \mathbf{t}_i)\boldsymbol{\beta}_D$ and $\mathbf{f}_z^T(\mathbf{x}_j^p, \mathbf{c}^*, \mathbf{t})\boldsymbol{\beta}_z$ respectively. The covariance matrix between $\mathbf{Y}^s$ and $\mathbf{Y}^p$ given $(\mathbf{c}^*, \boldsymbol{\phi}, \mathbf{t})$ is an $n_s \times n_p$ matrix with (i, j) th element

$$\begin{aligned} & \text{Cov}[Y^s(\mathbf{x}_i^s, \mathbf{c}_i, \mathbf{t}_i), Y^p(\mathbf{x}_j^p) | (\mathbf{c}^*, \boldsymbol{\phi}, \mathbf{t})] \\ &= \text{Cov}[Z(\mathbf{x}_i^s, \mathbf{c}_i, \mathbf{t}_i), Z(\mathbf{x}_j^p, \mathbf{c}^*, \mathbf{t}) | (\mathbf{c}^*, \boldsymbol{\phi}, \mathbf{t})], \end{aligned}$$

which can be computed using Equation (2).

With the joint prior density and the likelihood, the Metropolis–Hastings algorithm updates each of the parameters using the specifications listed in Table 1. For all the parameters, the proposal distributions are uniform distributions with centers set to be the previous draws and widths specified by the last column of Table 1. However, for all correlation parameters ρ , the proposal draws are made on $\xi = -4 \log(\rho)$.

APPENDIX B: DERIVATION OF FORMULA (14)

Using Equations (5) and (6) we compute

$$[E(\Delta(\mathbf{x}, \mathbf{c}^*, \mathbf{t}) | \mathbf{c}^*, \boldsymbol{\phi}, \mathbf{y}^s, \mathbf{y}^p, \mathbf{t})]^2 = [\mathbf{f}_z^T(\mathbf{x}, \mathbf{c}^*, \mathbf{t})\boldsymbol{\beta}_D + \sum_{DZ}^T \sum_{ZZ}^{-1} \mathbf{D}_t]^2 \text{ and}$$

$\text{Var}(\Delta(\mathbf{x}, \mathbf{c}^*, \mathbf{t}) | \mathbf{c}^*, \boldsymbol{\phi}, \mathbf{y}^s, \mathbf{y}^p, \mathbf{t}) = \sigma_D^2 - \sum_{DZ}^T \sum_{ZZ}^{-1} \Sigma_{DZ}$ where Σ_{DZ} is the $(n_p + n_s) \times 1$ vector of covariance between $\Delta(\mathbf{x}, \mathbf{c}^*, \mathbf{t})$ and $(\mathbf{Y}^s, \mathbf{Y}^p)^T$, Σ_{ZZ} is the $(n_p + n_s) \times (n_p + n_s)$ variance–covariance matrix of $(\mathbf{Y}^s, \mathbf{Y}^p)^T$, and $\mathbf{D}_t$ is the $(n_p + n_s) \times 1$ vector of differences between the observed data and its expectation; i.e., $\mathbf{D}_t = (\mathbf{Y}^s, \mathbf{Y}^p)^T - (E(\mathbf{Y}^p | \mathbf{t})^T E(\mathbf{Y}^s | \mathbf{t})^T)^T$. We list the elements of Σ_{DZ} , Σ_{ZZ} , and $\mathbf{D}_t$ below.

(B.1) For $i = 1, 2, \dots, n_p$, the i th element of Σ_{DZ} is

$$\text{Cov}(\Delta(\mathbf{x}, \mathbf{c}^*, \mathbf{t}), Y^p(\mathbf{x}_i^p) | (\mathbf{c}^*, \boldsymbol{\phi}, \mathbf{t})) = \text{Cov}(\Delta(\mathbf{x}, \mathbf{t}, \mathbf{c}^*), \Delta(\mathbf{x}_i^p, \mathbf{c}^*, \mathbf{t}) | (\mathbf{c}^*, \boldsymbol{\phi})). \text{ For } i = n_p +$$

$1, n_p + 2, \dots, n_p + n_s$, the i th element of Σ_{DZ} is
 $\text{Cov}(\Delta(\mathbf{x}, \mathbf{c}^*, \mathbf{t}), Y^s(\mathbf{x}_{i-n_p}^s, \mathbf{c}_{i-n_p}, \mathbf{t}_{i-n_p}) | (\mathbf{c}^*, \boldsymbol{\phi}, \mathbf{t})) = 0$.

(B.2)

$$\Sigma_{ZZ} = \begin{pmatrix} \Sigma_{PP} & \Sigma_{PS} \\ \Sigma_{PS}^\top & \Sigma_{SS} \end{pmatrix}, \quad (\text{B.1})$$

where

- $-\Sigma_{PP}$ is an $n_p \times n_p$ matrix whose (i_1, i_2) th entry is
 $\text{Cov}(Y^p(\mathbf{x}_{i_1}^p), Y^p(\mathbf{x}_{i_2}^p) | (\mathbf{c}^*, \boldsymbol{\phi}, \mathbf{t}))$, for $i, i_1, i_2 = 1, 2, \dots, n_p$;
- $-\Sigma_{PS}$ is an $n_p \times n_s$ matrix whose (i, j) th entry is
 $\text{Cov}(Y^p(\mathbf{x}_i^p), Y^s(\mathbf{x}_j^s, \mathbf{c}_j, \mathbf{t}_j) | (\mathbf{c}^*, \boldsymbol{\phi}, \mathbf{t}))$, for $i = 1, 2, \dots, n_p$ and $j = 1, 2, \dots, n_s$;
- $-\Sigma_{SS}$ is an $n_s \times n_s$ matrix whose (j_1, j_2) th entry is
 $\text{Cov}(Y^s(\mathbf{x}_{j_1}^s, \mathbf{c}_{j_1}, \mathbf{t}_{j_1}), Y^s(\mathbf{x}_{j_2}^s, \mathbf{c}_{j_2}, \mathbf{t}_{j_2}) | (\mathbf{c}^*, \boldsymbol{\phi}, \mathbf{t}))$, for $j_1, j_2 = 1, 2, \dots, n_s$.

(B.3) For $i = 1, 2, \dots, n_p$, the i th element of $\mathbf{D}_t$ is

$y^p(\mathbf{x}_i^p) - \mathbf{f}_z^\top(\mathbf{x}_i^p, \mathbf{c}^*, \mathbf{t})\beta_z - \mathbf{f}_D^\top(\mathbf{x}_i^p, \mathbf{c}^*, \mathbf{t})\beta_D$. For $i = n_p + 1, n_p + 2, \dots, n_p + n_s$, the i th element of $\mathbf{D}_t$ is $y^s(\mathbf{x}_{i-n_p}^s, \mathbf{c}_{i-n_p}, \mathbf{t}_{i-n_p}) - \mathbf{f}_z^\top(\mathbf{x}_{i-n_p}^s, \mathbf{c}_{i-n_p}, \mathbf{t}_{i-n_p})\beta_z$

APPENDIX C: DETAILS OF THE PREDICTION INTERVAL

By conditioning on $(\widehat{\mathbf{c}}_\ell^*, \widehat{\boldsymbol{\phi}}_\ell)$ and the law of large numbers,

$$\begin{aligned} & \text{Var}(Y^s(\mathbf{x}, \mathbf{c}, \mathbf{t}) | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*) \\ & \approx \frac{1}{N_{MC}} \sum_{\ell=1}^{N_{MC}} \text{Var}(Y^s(\mathbf{x}, \mathbf{c}, \mathbf{t}) | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*, \widehat{\mathbf{c}}_\ell^*, \widehat{\boldsymbol{\phi}}_\ell) + \frac{1}{N_{MC}} \sum_{\ell=1}^{N_{MC}} [E(Y^s(\mathbf{x}, \mathbf{c}, \mathbf{t}) | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*, \widehat{\mathbf{c}}_\ell^*, \widehat{\boldsymbol{\phi}}_\ell)]^2 - \left[\frac{1}{N_{MC}} \sum_{\ell=1}^{N_{MC}} E(Y^s(\mathbf{x}, \mathbf{c}, \mathbf{t}) | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*, \widehat{\mathbf{c}}_\ell^*, \widehat{\boldsymbol{\phi}}_\ell) \right]^2 \end{aligned} \quad (\text{C.1})$$

and

$$\begin{aligned} & \text{Var}(Y^s(\mathbf{x}, \mathbf{c}^*, \widehat{\mathbf{t}}^* + \Delta(\mathbf{x}, \mathbf{c}^*, \widehat{\mathbf{t}}^*) | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*) \\ & \approx \frac{1}{N_{MC}} \sum_{\ell=1}^{N_{MC}} \text{Var}(Y^s(\mathbf{x}, \widehat{\mathbf{c}}_\ell^*, \widehat{\mathbf{t}}^*) + \Delta(\mathbf{x}, \widehat{\mathbf{c}}_\ell^*, \widehat{\mathbf{t}}^*) | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*, \widehat{\mathbf{c}}_\ell^*, \widehat{\boldsymbol{\phi}}_\ell) + \frac{1}{N_{MC}} \sum_{\ell=1}^{N_{MC}} [E(Y^s(\mathbf{x}, \widehat{\mathbf{c}}_\ell^*, \widehat{\mathbf{t}}^*) + \Delta(\mathbf{x}, \widehat{\mathbf{c}}_\ell^*, \widehat{\mathbf{t}}^*) | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*, \widehat{\mathbf{c}}_\ell^*, \widehat{\boldsymbol{\phi}}_\ell)]^2 - \left[\frac{1}{N_{MC}} \sum_{\ell=1}^{N_{MC}} E(Y^s(\mathbf{x}, \widehat{\mathbf{c}}_\ell^*, \widehat{\mathbf{t}}^*) + \Delta(\mathbf{x}, \widehat{\mathbf{c}}_\ell^*, \widehat{\mathbf{t}}^*) | \mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}^*, \widehat{\mathbf{c}}_\ell^*, \widehat{\boldsymbol{\phi}}_\ell) \right]^2 \end{aligned} \quad (\text{C.2})$$

for large N_{MC} . Using Equations (1), (2), (4), (5), (6), and the following formulas for $E(Y^s(\mathbf{x}, \mathbf{c}, \mathbf{t})|\mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}, \widehat{\mathbf{c}^*}, \widehat{\boldsymbol{\phi}})$, $E(Y^s(\mathbf{x}, \mathbf{c}^*, \mathbf{t}) + \Delta(\mathbf{x}, \mathbf{c}^*, \mathbf{t})|\mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}, \widehat{\mathbf{c}^*}, \widehat{\boldsymbol{\phi}})$, $\text{Var}(Y^s(\mathbf{x}, \mathbf{c}, \mathbf{t})|\mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}, \widehat{\mathbf{c}^*}, \widehat{\boldsymbol{\phi}})$, and $\text{Var}(Y^s(\mathbf{x}, \mathbf{c}^*, \mathbf{t}) + \Delta(\mathbf{x}, \mathbf{c}^*, \mathbf{t})|\mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}, \widehat{\mathbf{c}^*}, \widehat{\boldsymbol{\phi}})$ one can compute Equations (18), ((19)), ((C.1)), and ((C.2)) by taking $(\widehat{\mathbf{t}}, \widehat{\mathbf{c}^*}, \widehat{\boldsymbol{\phi}})$ to be $(\widehat{\mathbf{t}^*}, \widehat{\mathbf{c}^*}, \widehat{\boldsymbol{\phi}})$.

- (C.1) The conditional expectation $E(Y^s(\mathbf{x}, \mathbf{c}, \mathbf{t})|\mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}, \widehat{\mathbf{c}^*}, \widehat{\boldsymbol{\phi}})$, is $\mathbf{f}_z^T(\mathbf{x}, \mathbf{c}, \mathbf{t})\beta_z + \widehat{\Sigma}_{0z}^T \widehat{\Sigma}_{zz}^{-1} \widehat{\mathbf{D}}_t$ where $\widehat{\Sigma}_{0z}$ is the $(n_p + n_s) \times 1$ vector of covariances between $Z(\mathbf{x}, \mathbf{c}, \mathbf{t})$ and the entries of $(\mathbf{Y}^p)^T, (\mathbf{Y}^s)^T$ given $(\widehat{\mathbf{t}}, \widehat{\mathbf{c}^*}, \widehat{\boldsymbol{\phi}})$. For $i = 1, \dots, n_p$, the i th element of $\widehat{\Sigma}_{0z}$ is $\text{Cov}(Z(\mathbf{x}, \mathbf{c}, \mathbf{t}), Z(\mathbf{x}_i^p, \mathbf{c}^*, \mathbf{t})|\widehat{\mathbf{t}}, \widehat{\mathbf{c}^*}, \widehat{\boldsymbol{\phi}})$ while for $i = n_p + 1, \dots, n_p + n_s$ the i th element is $\text{Cov}(Z(\mathbf{x}, \mathbf{c}, \mathbf{t}), Z(\mathbf{x}_{i-n_p}^s, \mathbf{c}_{i-n_p}^*, \mathbf{t}_{i-n_p})|\widehat{\mathbf{t}}, \widehat{\mathbf{c}^*}, \widehat{\boldsymbol{\phi}})$. Similarly, $\widehat{\Sigma}_{zz}$ is Equation (B.1) and $\widehat{\mathbf{D}}_t$ is given by (B.3), both conditional on $(\widehat{\mathbf{t}}, \widehat{\mathbf{c}^*}, \widehat{\boldsymbol{\phi}})$.
- (C.2) The conditional variance $\text{Var}(Y^s(\mathbf{x}, \mathbf{c}, \mathbf{t})|\mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}, \widehat{\mathbf{c}^*}, \widehat{\boldsymbol{\phi}}) = \widehat{\sigma}_z^2 - \widehat{\Sigma}_{0z}^T \widehat{\Sigma}_{zz}^{-1} \widehat{\Sigma}_{0z}$ where $\widehat{\Sigma}_{0z}$, $\widehat{\Sigma}_{zz}$ are defined in (C.1) while $\widehat{\sigma}_z^2$ is taken from $\widehat{\boldsymbol{\phi}}$.
- (C.3) The conditional mean of $\eta(\mathbf{x}, \mathbf{c}^*)$ is $E(Y^s(\mathbf{x}, \mathbf{c}, \mathbf{t})|\mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}, \widehat{\mathbf{c}^*}, \widehat{\boldsymbol{\phi}}) + E(\Delta(\mathbf{x}, \mathbf{c}, \mathbf{t})|\mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}, \widehat{\mathbf{c}^*}, \widehat{\boldsymbol{\phi}})$, where $E(Y^s(\mathbf{x}, \mathbf{c}, \mathbf{t})|\mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}, \widehat{\mathbf{c}^*}, \widehat{\boldsymbol{\phi}})$ and $E(\Delta(\mathbf{x}, \mathbf{c}, \mathbf{t})|\mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}, \widehat{\mathbf{c}^*}, \widehat{\boldsymbol{\phi}})$ is derived in (C.1) and Appendix B, respectively.
- (C.4) The conditional variance $\text{Var}(Y^s(\mathbf{x}, \mathbf{c}, \mathbf{t}) + \Delta(\mathbf{x}, \mathbf{c}, \mathbf{t})|\mathbf{y}^s, \mathbf{y}^p, \widehat{\mathbf{t}}, \widehat{\mathbf{c}^*}, \widehat{\boldsymbol{\phi}}) = \widehat{\sigma}_z^2 + \widehat{\sigma}_d^2 - (\widehat{\Sigma}_{0z}^T + \widehat{\Sigma}_{dz}^T) \widehat{\Sigma}_{zz}^{-1} (\widehat{\Sigma}_{0z} + \widehat{\Sigma}_{dz}^T)$, where $\widehat{\sigma}_z^2$ and $\widehat{\sigma}_d^2$ are taken from $\widehat{\boldsymbol{\phi}}$, $\widehat{\Sigma}_{0z}$ and $\widehat{\Sigma}_{zz}$ given in (C.1), and $\widehat{\Sigma}_{dz}$ is given by (B.1) where conditioning is given $(\widehat{\mathbf{t}}, \widehat{\mathbf{c}^*}, \widehat{\boldsymbol{\phi}})$.

Acknowledgments

This research was sponsored, in part, by the National Science Foundation under Agreements DMS-0112069 (SAMS/Duke University) and DMS-0406026 (The Ohio State University). Research support for the development of the biomechanics code and knee simulator used to produce the data for the example in Sections 2 and 5.3 was provided by NIH grant AR049793, Exactech, Inc., and the Clark and Kirby Foundations. The authors thank the editor, associate editor, and two reviewers for their constructive suggestions and valuable comments.

REFERENCES

- Bayarri MJ, Berger JO, Cafeo J, Garcia-Donato G, Liu F, Palomo J, Parthasarathy RJ, Paulo R, Sacks J, Walsh D. Computer Model Validation With Functional Output. *The Annals of Statistics* 2007a;35(5): 1874–1906.
- Bayarri MJ, Berger JO, Paulo R, Sacks J, Cafeo JA, Cavendish J, Lin C-H, Tu J. A Framework for Validation of Computer Models. *Technometrics* 2007b;49(2):138–154.
- Cox, DD.; Park, JS.; Singer, CE. A Statistical Method for Tuning a Computer Code to a Database. Rice University, Dept. of Statistics; 1996. Technical Report 96-3
- Craig PC, Goldstein M, Rougier JC, Seheult AH. Bayesian Forecasting for Complex Systems Using Computer Simulators. *Journal of the American Statistical Association* 2001;96:717–729.
- Craig, PS.; Goldstein, M.; Seheult, AH.; Smith, JA. Bayes Linear Strategies for History Matching of Hydrocarbon Reservoirs. In: Bernardo, JM.; Berger, JO.; Dawid, AP.; Smith, AFM., editors. *Bayesian Statistics*. Vol. 5. Oxford: Oxford University Press; 1996. p. 69-95.
- Fang, KT.; Li, R.; Sudjianto, A. *Design and Modeling for Computer Experiments*. London: Chapman & Hall; 2005.
- Forrester, A.; Sobester, A.; Keane, A. *Engineering Design via Surrogate Modelling: A Practical Guide*. Chichester, U.K: Wiley; 2008.

- Gattiker, J. Using the Gaussian Process Model for Simulation Analysis (GPM/SA) Code. Los Alamos National Laboratory; 2005. technical report
- Higdon D, Gattiker J, Williams B, Rightley M. Computer Model Calibration Using High Dimensional Output. *Journal of the American Statistical Association* 2008;103:570–583.
- Higdon D, Kennedy M, Cavendish J, Cafeo J, Ryne R. Combining Field Data and Computer Simulations for Calibration and Prediction. *SIAM Journal of Scientific Computing* 2004;26:448–466.
- Higdon, D.; Williams, B.; Moore, L.; McKay, M.; Keller-McNulty, S. Uncertainty Quantification for Combining Experimental Data and Computer Simulations. Los Alamos National Laboratory; 2005. Technical Report LA-UR 05-4951
- Kennedy MC, O'Hagan A. Bayesian Calibration of Computer Models. *Journal of the Royal Statistical Society, Ser. B* 2001;63:425–464. (with discussion).
- Linkletter C, Bingham D, Hengartner N, Higdon D, Ye KQ. Variable Selection for Gaussian Process Models in Computer Experiments. *Technometrics* 2006;48:478–490.
- Loeppky, J.; Bingham, D.; Welch, W. Computer Model Calibration or Tuning in Practice. University of British Columbia; 2006. technical report
- McKay MD, Beckman RJ, Conover WJ. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code. *Technometrics* 1979;21:239–245.
- Park, JS. Ph.D. thesis. Champaign/Urbana, IL: University of Illinois; 1991. Tuning Complex Computer Codes to Data and Optimal Designs.
- Qian PZG. Nested Latin Hypercube Designs. *Biometrika*. 2009 to appear, doi: [10.1093/biomet/asp045](https://doi.org/10.1093/biomet/asp045).
- Qian PZG, Wu CFJ. Sliced Space-Filling Designs. *Biometrika*. 2009 to appear, doi: [10.1093/biomet/asp044](https://doi.org/10.1093/biomet/asp044).
- Qian PZG, Ai M, Wu CFJ. Construction of Nested Space-Filling Designs. *The Annals of Statistics* 2009;37:3616–3643.
- Rawlinson JJ, Furman BD, Li S, Wright TM, Bartel DL. Retrieval, Experimental, and Computational Assessment of the Performance of Total Knee Replacements. *Journal of Orthopaedic Research* 2006;24(7):1384–1394. [PubMed: 16705715]
- Saltelli, A.; Chan, K.; Scott, E. Sensitivity Analysis. Chichester: Wiley; 2000.
- Santner, TJ.; Williams, BJ.; Notz, WI. The Design and Analysis of Computer Experiments. New York: Springer Verlag; 2003.

Figure 1.
Scatterplot of the measured APD against percentile in the gait cycle, from the knee simulator (triangles) and the FEA computer code (dots).

Figure 2. Simulated posterior distributions of load discretization (the left panel) and initial position (the right panel).

Figure 3. Simulated posterior distributions of t^* (the left panel) and c^* (the right panel) using the Bayesian calibration program.

Figure 4.
The training data (solid circles), the true response curve (the solid line), and the predictions (pluses) obtained by gpmsa.

Figure 5.

Results of tuning and calibration from the STaC program: The left panel plots of the estimated L_2 discrepancy against t where $t = 0, 0.1, \dots, 0.9, 1$; the right panel is a histogram of the draws from the posterior distribution of c^* .

Figure 6.
The training data (solid circles), true response curve (the solid line), predictions (pluses), and 99% prediction bands (dashes) using the STaC program.

Figure 7. Scatterplot of the estimated squared discrepancy versus the value of the tuning parameter (left panel); histogram of the simulated posterior distribution of the calibration parameter when $\hat{l}^* = 19$ (right panel).

Figure 8.
The training data (triangles), predictions (pluses), and 99% prediction bands (dashes) using the STaC program.

Table 1

Specification of the Metropolis–Hastings algorithm for each parameter. Columns 2–5, from left to right, correspond to the prior distributions, the lower and upper bounds of the parameters as the program iterates, the initial values of the parameters, and the widths of the uniform distributions. The notation $TN(\mu, \sigma^2)$ on $[a, b]$ denotes the truncated normal distribution with mean μ , variance σ^2 , and support $[a, b]$; $\Gamma(\alpha, \beta)$ denotes the Gamma distribution having mean $\alpha\beta$ and variance $\alpha\beta^2$

Model parameter	Prior distribution	Support	Initial value	Width
All elements of $\mathbf{c}^*$	$TN(0.5, 2^2)$	$[0, 1]$	0.5	0.1
All elements of $\mathbf{p}_z$ and $\mathbf{p}_\delta$	$\text{Beta}(1, 0.5)$	$(0, 1)$	$2/3$	0.3
$1/\sigma_z^2$	$\Gamma(10, 0.1/\hat{\sigma}_s^2)$	$(0, +\infty)$	$1/\hat{\sigma}_s^2$	$0.2/\hat{\sigma}_s^2$
$1/\sigma_D^2$, if $\hat{\sigma}_s^2 > \hat{\sigma}_p^2$	$\Gamma(1, 100/\hat{\sigma}_s^2)$	$(0, +\infty)$	$100/\hat{\sigma}_s^2$	$10/\hat{\sigma}_s^2$
$1/\sigma_D^2$, if $\hat{\sigma}_s^2 < \hat{\sigma}_p^2$	$\Gamma(10, 0.1/(\hat{\sigma}_p^2 - \hat{\sigma}_s^2))$	$(0, +\infty)$	$1/(\hat{\sigma}_p^2 - \hat{\sigma}_s^2)$	$0.1/(\hat{\sigma}_p^2 - \hat{\sigma}_s^2)$
$1/\sigma_\varepsilon^2$	$\Gamma(1, 1000/\hat{\sigma}_p^2)$	$(0, +\infty)$	$1000/\hat{\sigma}_p^2$	$2/\hat{\sigma}_s^2$