CLEMSON UNIVERSITY

COMPUTER MODEL CALIBRATION AS A METHOD OF DESIGN

Carl Ehrett

Comprehensive exam, Department of Mathematical Sciences

2018-08-20

Section 1

INTRODUCTION: COMPUTER MODEL CALIBRATION

Traditional computer model calibration

- In a problem of computer model calibration, we have:

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

  where $\mathrm{y}(\mathbf{x})$ is the observed value at $\mathbf{x}$, $\mathrm{f}(\mathbf{x})$ is the value of the true system at $\mathbf{x}$, $\epsilon(\mathbf{x})$ is the observation error (often iid Gaussian), $\eta(\mathbf{x}, \boldsymbol{\theta})$ is the value of the computer model and $\delta(\mathbf{x})$ captures the discrepancy between the computer model and the true system.

- Here, $\mathbf{x}$ gives the known and/or controllable input settings for the system, and $\boldsymbol{\theta}$ gives the unknown parameters which must be given as inputs to the computer model.

- The purpose of computer model calibration is to use a set $\mathbf{y}$ of observations to find the appropriate value of $\boldsymbol{\theta}$ so that $\mathrm{L}(\delta)$ is minimized for loss $\mathrm{L}(\cdot)$.

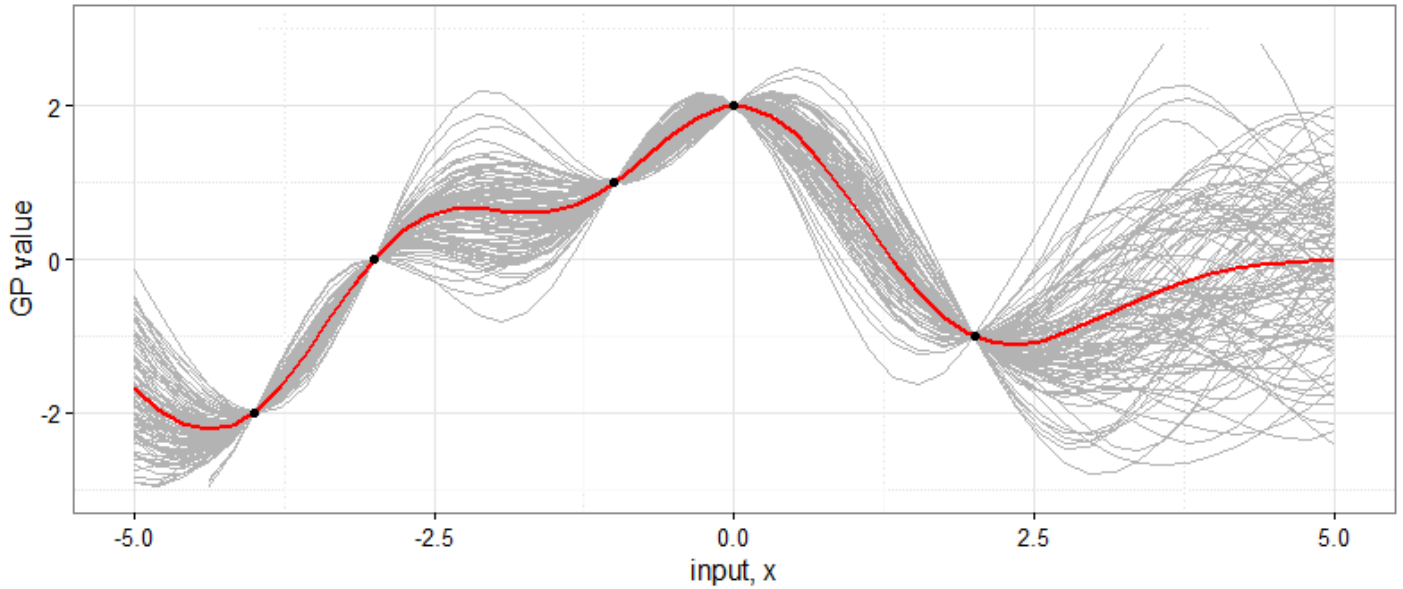
Calibration in the vein of Kennedy and O'Hagan (2001)

- Kennedy and O'Hagan propose a Bayesian method of calibration that results in a posterior distribution of $\boldsymbol{\theta}|\mathbf{y}$.
- A benefit of this approach is its inherent inclusion of uncertainty quantification.
- Despite identification difficulties between $\boldsymbol{\theta}$ and $\delta(\cdot)$, this framework has become very influential, and has been the locus of much research.

Kennedy, Marc C., and Anthony O'Hagan. 2001. "Bayesian Calibration of Computer Models." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63 (3). Blackwell Publishers Ltd.: 425–64.

Gaussian process emulators

- Kennedy and O'Hagan use Gaussian process (GP) emulators as code surrogates when calibrating computationally expensive models.



- For a prior GP with mean $\mu(\cdot)$ and covariance $C(\cdot,\cdot)$, given a set of observations $\boldsymbol{\eta}$ at points $\mathbf{X}$, the posterior mean and covariance at a new set of points $\mathbf{X}'$ are

$$\boldsymbol{\mu}^*_{\mathbf{X}'} = \boldsymbol{\mu}_{\mathbf{X}'} + \mathbf{C}_{\mathbf{X}',\mathbf{X}} \cdot \mathbf{C}^{-1}_{\mathbf{X},\mathbf{X}}(\boldsymbol{\eta} - \boldsymbol{\mu}_{\mathbf{X}}), \qquad \mathbf{C}^*_{\mathbf{X}',\mathbf{X}'} = \mathbf{C}_{\mathbf{X}',\mathbf{X}'} - \mathbf{C}_{\mathbf{X}',\mathbf{X}} \cdot \mathbf{C}^{-1}_{\mathbf{X},\mathbf{X}} \cdot \mathbf{C}_{\mathbf{X},\mathbf{X}'} \tag{2}$$

where $\boldsymbol{\mu}_{\mathbf{A}}$ is a vector giving $\mu(a)$ for each $a \in \mathbf{A}$, and $\mathbf{C}_{\mathbf{A},\mathbf{B}}$ is a matrix for which the i,j entry is $C(a_i, b_j)$ when $\mathbf{A} = (a_1, \cdots, a_n)^T, \mathbf{B} = (b_1, \cdots, b_m)^T$.

Benefits of GP emulators

Three benefits to GP priors in the context of computer model emulation:

1. They do not require detailed foreknowledge of the model's parametric form.
2. They easily interpolate the model output.
3. They facilitate uncertainty quantification.

The framework of Williams, Higdon, Gattiker et al. (2006)

- $\eta(\cdot,\cdot)$ is modeled with a mean-zero GP prior with covariance as the product power exponential function:

$$C\big((\mathbf{x},\mathbf{t}),(\mathbf{x}',\mathbf{t}')\big) = \frac{1}{\lambda_\eta}\prod_{k=1}^{p}\rho_k^{\eta\wedge(x_k-x_k')^2}\times\prod_{k=1}^{q}\rho_{p+k}^{\eta\wedge(t_k-t_k')^2} \qquad (3)$$

where $\mathbf{x}\in\mathcal{X}\subseteq\mathbb{R}^p$, $\mathbf{t}\in\mathcal{T}\subseteq\mathbb{R}^q$, with priors $\lambda_\eta\sim\text{Gamma}(5,5)$, $\rho_k^\eta\sim\text{Beta}(0.1,1)$.

- $\delta(\cdot)$ is similarly modeled with a mean-zero GP prior, with covariance

$$C_\delta(\mathbf{x},\mathbf{x}') = \frac{1}{\lambda_\delta}\prod_{k=1}^{p}\rho_k^{\delta\wedge(x_k-x_k')^2} \qquad (4)$$

with priors $\lambda_\delta\sim\text{Gamma}(5,5)$, $\rho_k^\delta\sim\text{Beta}(0.3,1)$.

Williams, Brian J., Dave Higdon, Jim Gattiker et al. 2006. “Combining Experimental Data and Computer Simulations, with an Application to Flyer Plate Experiments.” *Bayesian Analysis* 1 (4). International Society for Bayesian Analysis: 765–92.

The framework of Williams, Higdon, Gattiker et al. (2006)

- Let $\boldsymbol{\eta}$ be a vector of computer model outputs at inputs $\left((\mathbf{x_1}, \mathbf{t_1}), \cdots, (\mathbf{x_n}, \mathbf{t_n})\right)$ and $\mathbf{y}$ be a vector of real world observations at $(\mathbf{x_{n+1}}, \cdots, \mathbf{x_{n+m}})$ with $\mathbf{C_y}$ a matrix giving the (known) observation variance for $\mathbf{y}$. Then for $\boldsymbol{\mathcal{D}} = \left(\boldsymbol{\eta}^{\mathbf{T}}, \mathbf{y}^{\mathbf{T}}\right)^{\mathbf{T}}$ we have

$$\boldsymbol{\mathcal{D}}|\boldsymbol{\theta}, \lambda_\eta, \boldsymbol{\rho}^{\boldsymbol{\eta}}, \lambda_\delta, \boldsymbol{\rho}^{\boldsymbol{\delta}}, \mathbf{C_y} \sim \mathrm{N}(\mathbf{0}, \mathbf{C}_{\boldsymbol{\mathcal{D}}}) \tag{5}$$

  where $\mathbf{C}_{\boldsymbol{\mathcal{D}}}$ is the sum of a matrix giving $\mathrm{C}\left((\mathbf{x_i}, \mathbf{t_i}), (\mathbf{x_j}, \mathbf{t_j})\right)$ for the i, j entry with the matrix $\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C_y} + \mathbf{C_\delta} \end{bmatrix}$. The matrix $\mathbf{C_\delta}$ has, for its i, j entry, $\mathrm{C}_\delta(\mathrm{x_{n+i}}, \mathrm{x_{n+j}})$.

- The joint posterior under the model is

$$\pi\left(\boldsymbol{\theta}, \lambda_\eta, \boldsymbol{\rho}^{\boldsymbol{\eta}}, \lambda_\delta, \boldsymbol{\rho}^{\boldsymbol{\delta}}, \mathbf{C_y} \middle| \boldsymbol{\mathcal{D}}\right) \propto \pi\left(\boldsymbol{\mathcal{D}} \middle| \boldsymbol{\theta}, \lambda_\eta, \boldsymbol{\rho}^{\boldsymbol{\eta}}, \lambda_\delta, \boldsymbol{\rho}^{\boldsymbol{\delta}}, \mathbf{C_y}\right) \times \pi\left(\lambda_\eta\right) \times \pi(\boldsymbol{\rho}^{\boldsymbol{\eta}}) \times \pi(\lambda_\delta) \times \pi(\boldsymbol{\rho}^{\boldsymbol{\delta}}) \tag{6}$$

- This distribution can be explored via MCMC.

Williams, Brian J., Dave Higdon, Jim Gattiker et al. 2006. “Combining Experimental Data and Computer Simulations, with an Application to Flyer Plate Experiments.” *Bayesian Analysis* 1 (4). International Society for Bayesian Analysis: 765–92.

Section 2

ADAPTING THE CALIBRATION FRAMEWORK FOR OPTIMIZATION

Calibration to desired observations

- We can reconceptualize the Kennedy-O'Hagan calibration framework as a method for optimization.

- In the schema $\mathrm{y}(\mathbf{x}) = \mathrm{f}(\mathbf{x}) + \epsilon(\mathbf{x}) = \eta(\mathbf{x}, \boldsymbol{\theta}) + \delta(\mathbf{x}) + \epsilon(\mathbf{x})$, we now take $\boldsymbol{\theta}$ to be controllable inputs over which we wish to optimize model output.

- Instead of calibrating to a set of real-world observations $\mathbf{y}$, instead set $\mathbf{y}$ to be a set of performance targets. Call performance targets used as manufactured data in such calibration *desired observations*.

- The result of calibration to desired observations (CDO) is a posterior distribution $\boldsymbol{\theta}|\boldsymbol{y}$. As under Kennedy-O'Hagan calibration, uncertainty quantification is included.

Model shortcoming

- Given the schema $y(\mathbf{x}) = f(\mathbf{x}) + \epsilon(\mathbf{x}) = \eta(\mathbf{x}, \boldsymbol{\theta}) + \delta(\mathbf{x}) + \epsilon(\mathbf{x})$, “error” can be modeled as either $\delta(\mathbf{x}) + \epsilon(\mathbf{x})$ or just as $\epsilon(\mathbf{x})$ if a (nonzero) discrepancy is not included.

- Either works for achieving a distribution on $\boldsymbol{\theta}$ values which lead to output approximating desired observations $\mathbf{y}$.

- The benefit of using only $\epsilon(\cdot)$ is convenience in setting optimization priorities, and computational efficiency.

- The benefit of including $\delta(\cdot)$ is that the resulting error analysis is more accurate. The posterior distribution on $\delta(\cdot)$ describes the discrepancy between the desired observations and the optimal model output.

Hyperparameter estimation

- Rather than estimate the hyperparameters $\lambda_\eta, \boldsymbol{\rho}^{\boldsymbol{\eta}}$ as part of a full Bayesian analysis, we use their MLEs.

- This practice is common in calibration for convenience, but is crucial here for a different reason.

- These hyperparameters are part of our code surrogate. We want them to "learn" only from the code observations. We do **not** want them to "learn" from our (fake) desired observations $\mathbf{y}$.

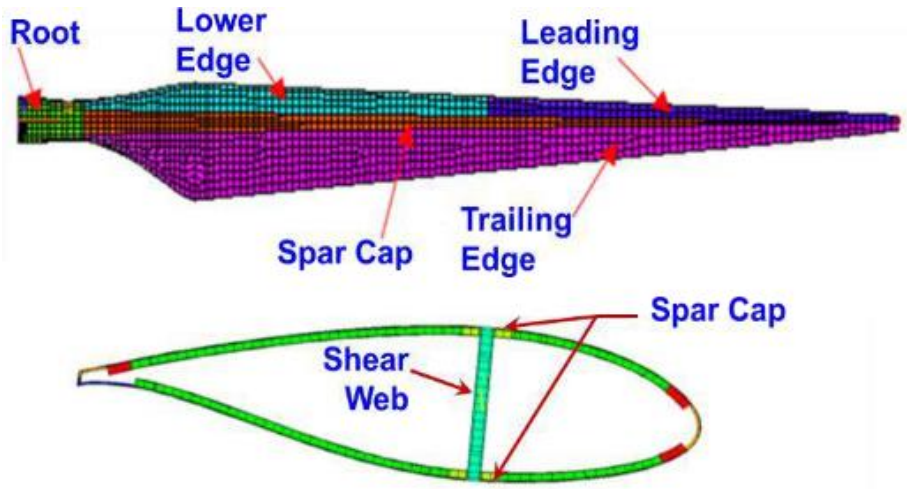
Section 3

APPLICATION: WIND TURBINE BLADE MATERIAL DESIGN

Wind turbine blade design problem

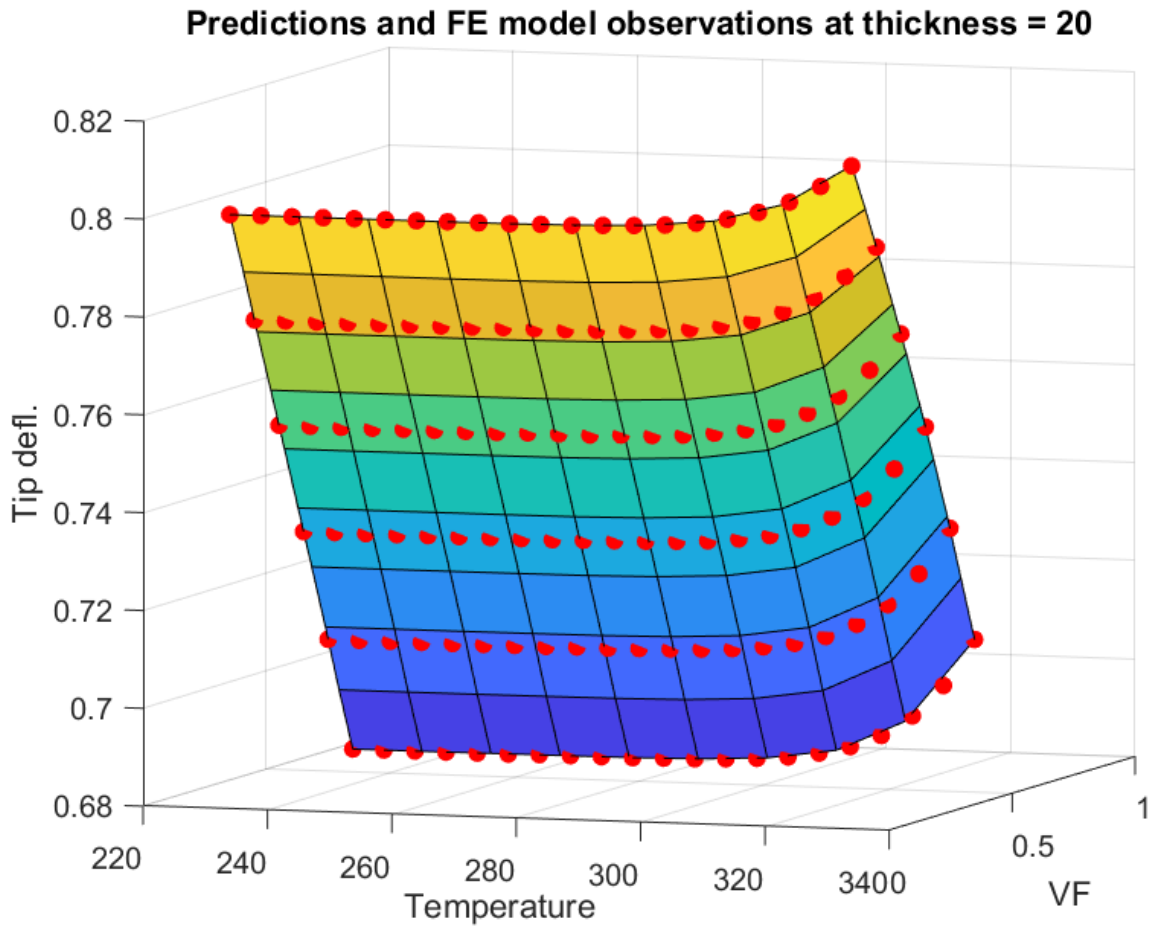
- Design problem: given a blade of fixed geometry, determine the optimal *volume fraction* $\mathrm{v} \in (0,1)$ and *thickness* in mm k, across a range of temperatures h.

- We want to minimize three model outputs: tip deflection in meters d, rotation in radians r, and cost in USD c.



Wind turbine blade computer model

- The computer model is a finite element model implemented in ANSYS. It is too computationally expensive to deploy in MCMC.
- We collected 504 observations in a latin hypercube on the inputs $(\mathrm{h}, \mathrm{v}, \mathrm{k})$.
- The output is trivariate. Dummy variables were added to convert the observations to 1512 univariate observations.
- Hyperparameters MLEs: $\widehat{\lambda_\eta} = 0.0152$, $\widehat{\boldsymbol{\rho}^{\boldsymbol{\eta}}} = (0.936, 0.651, 0.674, 0.480, 0.968)$



Model for wind turbine application

- Various options are available for the priors $\pi(\boldsymbol{\theta})$ and $\pi(\mathbf{C_y})$. But in full generality, the model is given by:

$$\boldsymbol{\pi}(\boldsymbol{\theta}, \mathbf{C_y}|\boldsymbol{\mathcal{D}}) \propto \boldsymbol{\pi}(\boldsymbol{\mathcal{D}}|\boldsymbol{\theta}, \mathbf{C_y}) \times \boldsymbol{\pi}(\boldsymbol{\theta}) \times \boldsymbol{\pi}(\mathbf{C_y})$$

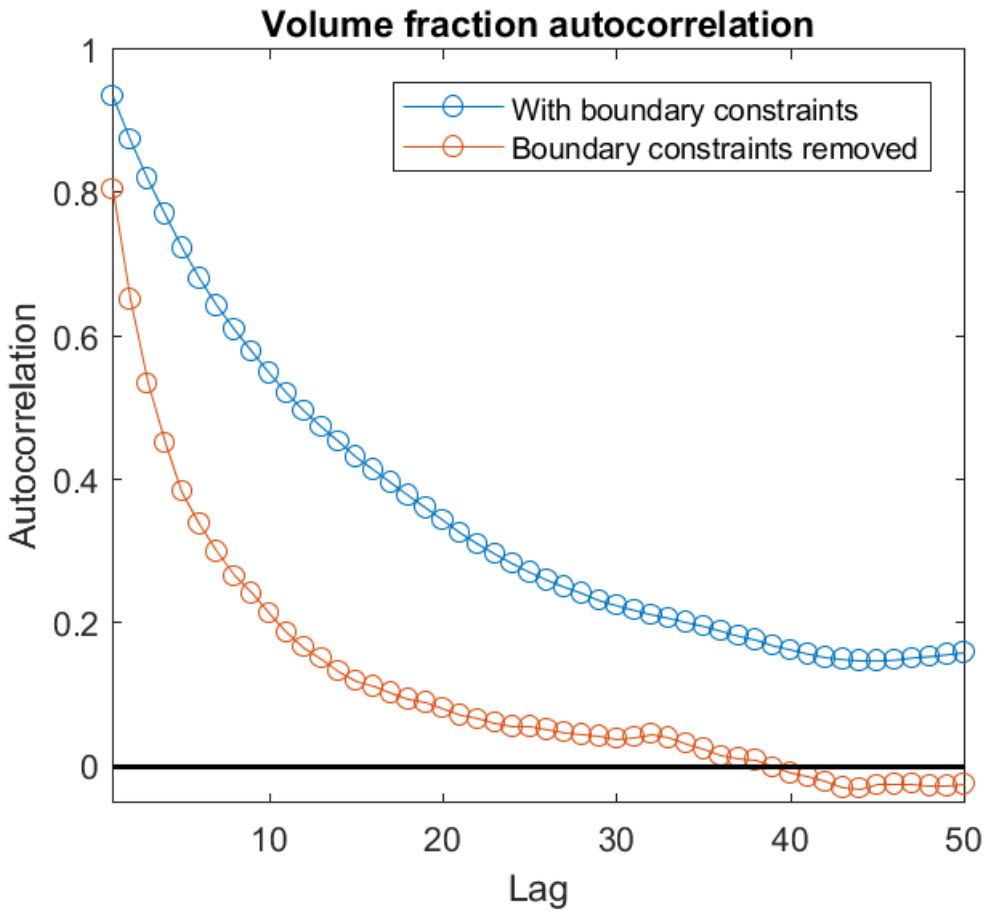
$$\propto |\mathbf{C}_{\boldsymbol{\mathcal{D}}}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\boldsymbol{\mathcal{D}}^{\mathbf{T}}\mathbf{C}_{\boldsymbol{\mathcal{D}}}^{-1}\boldsymbol{\mathcal{D}}\right) \times \pi(\boldsymbol{\theta}) \times \pi(\mathbf{C_y}) \quad (7)$$

- This constitutes an expansion of the framework of Williams et al. (2006), to include estimation of $\mathbf{C_y}$.

- Here, $\mathbf{C_y}$ is assumed to be diagonal. Options include homoscedasticity across model outputs ($\mathbf{C_y} = \boldsymbol{\sigma}^{\mathbf{2}}\mathbf{I}$), or allowing σ_i^2 to vary for $i = 1,2,3$ corresponding to the three model outputs.

MCMC details

- Boundary constraints: $\boldsymbol{\theta} \in [0.2,0.6] \times [10,25]$, $\sigma_i^2 > 0$.
- MCMC is improved by removing these boundary constraints by logit-transforming each element of $\boldsymbol{\theta}$, and taking the log of each σ_i^2.
- I used normal proposal densities with adaptive covariance matrices to promote acceptance rates ~23%.
- Adaptive algorithm: initialize $\tau = 1$. Every 100 draws during burn-in, let Σ be the sample covariance of all previous draws. If fewer than 20 of previous 100 draws were accepted, update $\tau = 0.75\tau$; if more than 25 accepted, update $\tau = 1.25\tau$. Set new proposal covariance to be $\tau\Sigma$.
- All computations are carried out on the log scale where possible, for computational efficiency.



Options for $\pi(\mathbf{C_y})$

- Two independent sets of options:
  1. Specify predetermined values for σ_i^2, $i = 1,2,3$, or else set a prior on them.
  2. Constrain $\sigma_1^2 = \sigma_2^2 = \sigma_3^2$, or else allow them to take different values.

- Specifying predetermined values is useful for implementing priorities. A prior is useful when one has insufficient knowledge of the Pareto front to implement priorities.

- With $\pi(\boldsymbol{\theta})$ uniform, when specifying values of $\boldsymbol{\sigma^2}$ the full model is:

$$\pi(\boldsymbol{\theta}, \mathbf{C_y}|\boldsymbol{\mathcal{D}}) \propto |\mathbf{C}_{\boldsymbol{\mathcal{D}}}|^{-\frac{1}{2}} \exp\left(\boldsymbol{\mathcal{D}}^{\mathbf{T}} \mathbf{C}_{\boldsymbol{\mathcal{D}}}^{-1} \boldsymbol{\mathcal{D}}\right) \tag{8}$$

and with the reference prior $\pi(\mathbf{C_y}) = \prod_{k=1}^{3} 1/\sigma_k^2$, the full model is:

$$\pi(\boldsymbol{\theta}, \mathbf{C_y}|\boldsymbol{\mathcal{D}}) \propto |\mathbf{C}_{\boldsymbol{\mathcal{D}}}|^{-\frac{1}{2}} \exp\left(\boldsymbol{\mathcal{D}}^{\mathbf{T}} \mathbf{C}_{\boldsymbol{\mathcal{D}}}^{-1} \boldsymbol{\mathcal{D}}\right) \times \prod_{k=1}^{3} \frac{1}{\sigma_k^2} \tag{9}$$

Section 4

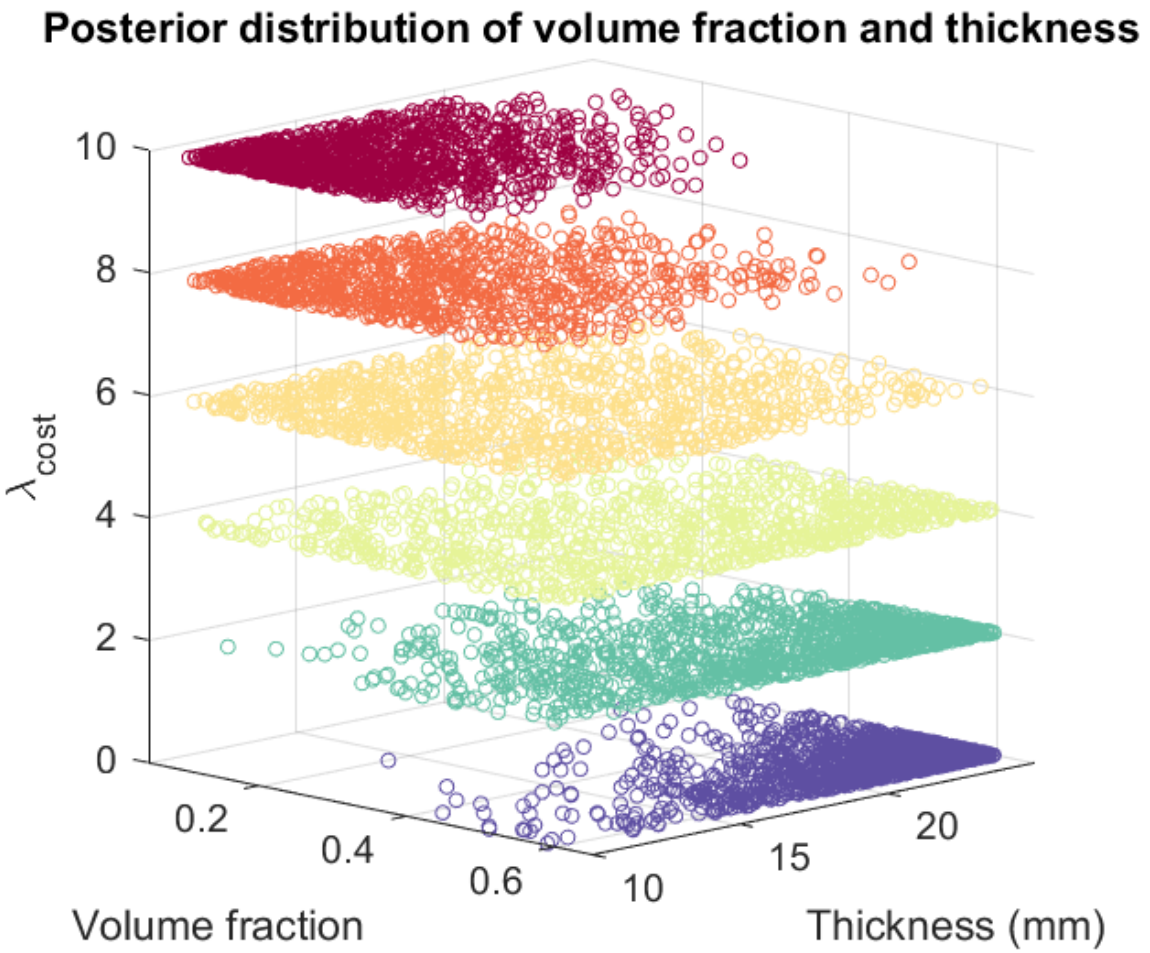
PARETO FRONT ESTIMATION

Exploring the Pareto front

- Sometimes priorities are not specific enough to be represented by a single set of desired observations.
- One might wish to explore optimal outputs over (e.g.) a range of costs.
- Two methods are described here for doing so: (1) exploiting known relationships in the model through a prior on $\boldsymbol{\theta}$, and (2) calibrating over a grid of “known” values of a subset of model outputs.

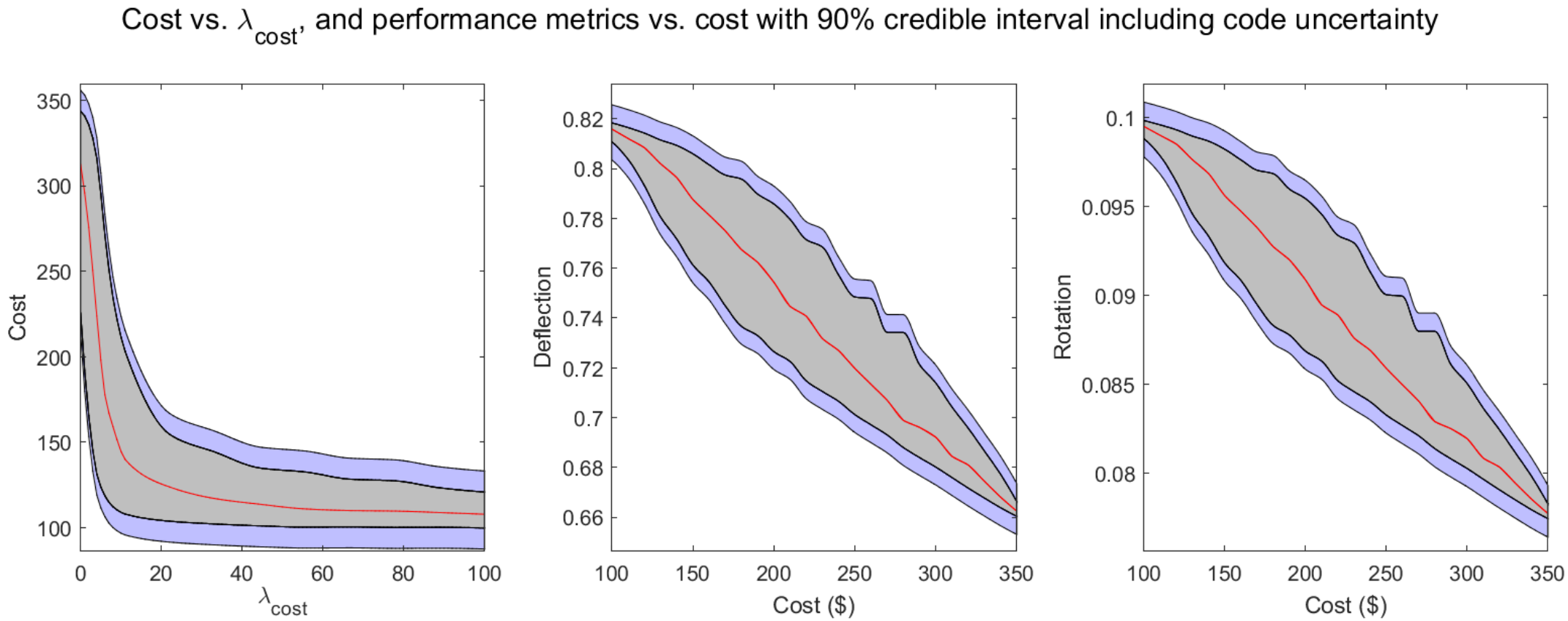
Exploiting known input/output relationships

- High volume fraction and high thickness are each known to correlate to high cost.
- Therefore, rather than include cost, we can use a prior $\pi(\boldsymbol{\theta}) = \exp -\lambda_c \|\boldsymbol{\theta}\|^2$ to penalize high-cost regions of the parameter space. The severity of the penalty depends upon $\lambda_c > 0$.
- Removing cost from the model brings computational benefits. We have now only 1008 observations rather than 1512.
- The effect of increasing the value of λ_c is to push $\boldsymbol{\theta}$ out of the upper (high-cost) region of its support toward the lower (low-cost) region.



Pareto bands using λ_c

Calibrating over each of a range of λ_c, we achieve a comprehensive picture of how optimal performance varies with cost, with included uncertainty bands.



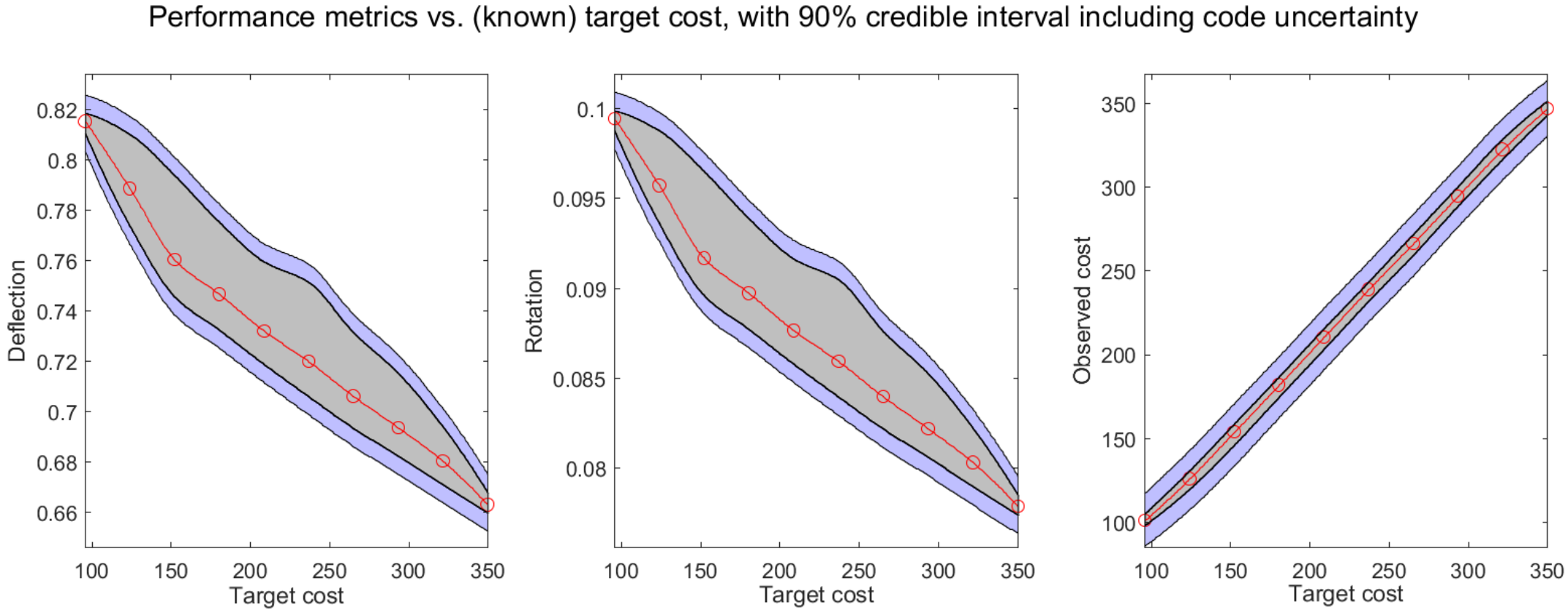
Specifying known cost

- Alternatively, one can achieve a similar result (even when no exploitable input/output relationship is known) by performing CDO over a grid of "known" costs.
- Set a grid $\{c_i\}_{i=1}^{p}$ of achievable costs, and for each grid point c_i, calibrate to desired observation (e.g.) $[0,0,c_i]$.
- In order to specify cost as known up to slight deviations, set σ_3^2 equal to a low value (here, 0.05 on standardized scale).
- When using the reference prior for the variance of deflection and rotation, and uniform prior on $\boldsymbol{\theta}$, the full posterior is:

$$\pi(\boldsymbol{\theta}, \mathbf{C_y} | \boldsymbol{\mathcal{D}}) \propto |\mathbf{C}_{\boldsymbol{\mathcal{D}}}|^{-\frac{1}{2}} \exp\left(\boldsymbol{\mathcal{D}}^{\mathbf{T}} \mathbf{C}_{\boldsymbol{\mathcal{D}}}^{-1} \boldsymbol{\mathcal{D}}^{\mathbf{T}}\right) \times \prod_{i=1}^{2} \frac{1}{\sigma_i^2} \qquad (10)$$

Pareto bands using cost grid

Calibrating over each of a range of values of cost, we achieve an estimate of optimal performance with uncertainty bands across costs.



Appendix

MCMC ALGORITHM

Algorithm 1:

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

Step B: Repeat M times.

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