

# **Joint Modeling of Detailed and Approximate Simulations**

## Overview

- Preliminary design of a complex system often involves exploring a broad design space. In addition to other techniques, such as screening (out) unimportant variables, this may require repeated use of computationally expensive simulations.
- To ease the computational burden of using detailed but expensive simulators, surrogate models (cheap-to-compute predictors) are built to provide rapid approximations of more expensive models. However, the surrogate models themselves are often expensive to build because they are based on repeated experiments with computationally expensive simulations.

- An alternative approach is to replace the detailed simulations with simplified approximate simulations, thereby sacrificing accuracy for reduced computational time. Thus methods are needed which improve the precision of surrogate models based on approximate simulations without significantly increasing computational time. Typically one will have available larger amounts of training data from the approximate simulator and lesser amounts of data from detailed simulations to obtain an accurate prediction model.

## Why do Different Simulations arise?

1. Different mathematical models for the same physical phenomenon (pendulum model for a rotor blade with and without friction)
2. Different numerical algorithms to solve the same mathematical model (Qian and Wu, 2006)
3. Different implementation choices for the same numerical method to solve a given mathematical model (mesh densities for a given FE)

## Setup

- Output from the fast (inferior, “bad”) code is denoted  $y(b, \mathbf{x})$
- Output from the slow (superior, “good”) code is  $y(g, \mathbf{x})$

- **Training Data**

$$y(b, \mathbf{x}_{1,1}^{tr}), \dots, y(b, \mathbf{x}_{1,n_1}^{tr}), y(g, \mathbf{x}_{2,1}^{tr}), \dots, y(g, \mathbf{x}_{2,n_2}^{tr})$$

- **Predict**  $y(g, \mathbf{x}_0)$  where  $\mathbf{x}_0$  is an untried new input

- **Every paper** on this problem is based on stochastic process models in which training data is viewed as a draw from a stochastic process  $Y(t, \mathbf{x})$ ,  $t \in \{b, g\}$ , which states that  $y(b, \mathbf{x})$  and  $y(g, \mathbf{x})$  have “similar” dependence on  $\mathbf{x}$ . Most  $Y(t, \mathbf{x})$  models use linear combinations of independent processes. Virtually all such models can be extended to an arbitrary number of codes of increasing fidelity (“reification”)

- **Kennedy and O’Hagan (2000, *Predicting the Output from a Complex Computer Code When Fast Approximations Are Available*)** use autoregressive model to describe  $y(b, \mathbf{x})$  and  $y(g, \mathbf{x})$  output.

## **Model**

$$Y(b, \mathbf{x}) = Z_b(\mathbf{x})$$

$$Y(g, \mathbf{x}) = \tau Y(b, \mathbf{x}) + \delta(\mathbf{x}) = \tau Z_b(\mathbf{x}) + \delta(\mathbf{x})$$

where  $Z_b(\cdot)$  and  $\delta(\cdot)$  are independent stationary GaSPs (each GaSP with its own mean, variance, and correlation parameters). KOH estimate correlation parameters by MLE and predict  $\hat{y}(\mathbf{x}_0)$  by

$$\hat{y}(\mathbf{x}_0) = E\{Y(\mathbf{x}_0) | \text{data, est corr. parameters}\}$$

## Potential Scaling Problem

$$Y(b, \mathbf{x}) = Z_b(\mathbf{x})$$

$$Y(g, \mathbf{x}) = \tau Z_b(\mathbf{x}) + \delta(\mathbf{x})$$

Suppose that both codes produce outputs on the same scale.

Then  $\sigma_{Z,b}^2$  and  $\tau^2 \sigma_{Z,b}^2 + \sigma_{\delta}^2$  should be on the same scale and there is typically non-identifiability between  $\tau^2$  and  $\sigma_{\delta}^2$

- **Qian, Wu, and Wu (2005, *Gaussian Process Models for Computer Experiments with Qualitative and Quantitative Factors*)** For the same setup as Kennedy and O’Hagan, and  $\mathbf{x} \in [0, 1]^d$

$$Y(b, \mathbf{x}) = \boldsymbol{\beta}_b^\top \mathbf{f}_b(\mathbf{x}) + Z_b(\mathbf{x})$$

$$Y(g, \mathbf{x}) = \boldsymbol{\beta}_g^\top \mathbf{f}_g(\mathbf{x}) + Z_g(\mathbf{x})$$

where  $Z_b(\mathbf{x})$  and  $Z_g(\mathbf{x})$  are zero mean GaSPs with common variance  $\sigma^2$  and

$$\text{Cov} (Z_i(\mathbf{x}^1), Z_j(\mathbf{x}^2)) = \begin{cases} \sigma^2 \prod_{\ell=1}^d \exp\{-\theta_\ell (x_\ell^1 - x_\ell^2)^2\}, & i = j \\ \delta \sigma^2 \prod_{\ell=1}^d \exp\{-\theta_\ell (x_\ell^1 - x_\ell^2)^2\}, & i \neq j \end{cases}$$

with  $\{\theta_\ell\}_\ell > 0$  and  $0 < \delta < 1$ .

$$\text{Cov} (Z_i(\mathbf{x}^1), Z_j(\mathbf{x}^2)) = \begin{cases} \sigma^2 \prod_{\ell=1}^d \exp\{-\theta_{\ell}(x_{\ell}^1 - x_{\ell}^2)^2\}, & i = j \\ \delta \sigma^2 \prod_{\ell=1}^d \exp\{-\theta_{\ell}(x_{\ell}^1 - x_{\ell}^2)^2\}, & i \neq j \end{cases}$$

with  $\{\theta_{\ell}\}_{\ell} > 0$  and  $0 < \delta < 1$ .

- QWW estimate correlation parameters by MLE and predict  $\hat{y}(\mathbf{x}_0)$  by

$$\hat{y}(\mathbf{x}_0) = E\{Y(\mathbf{x}_0) | \text{data, est corr. parameters}\}$$

- **Qian and Wu (2006, *Bayesian Hierarchical Modeling for Integrating Low-accuracy and High-accuracy Experiments*)**

$$Y(b, \mathbf{x}) = \boldsymbol{\beta}_1^\top \mathbf{f}_1(\mathbf{x}) + Z_b(\mathbf{x})$$

$$Y(g, \mathbf{x}) = \tau(\mathbf{x})Y(b, \mathbf{x}) + \delta(\mathbf{x})$$

where  $Z_b(\mathbf{x})$ ,  $\tau(\mathbf{x})$ , and  $\delta(\mathbf{x})$  are independent GaSPs. This model has “more flexible” scaling than the KOH model (and the same type of shift. QW analysis is hierarchical Bayesian.

**Gang and TJS (2007, *Prediction for Computer Experiments Having Quantitative and Qualitative Input Variables*)**

**A Hierarchical Model and Bayesian Analysis**

**Stage 1** Given  $(\boldsymbol{\beta}, \boldsymbol{\sigma}^2, \boldsymbol{\rho})$ ,

$$Y(t, \mathbf{x}) | (\boldsymbol{\beta}, \boldsymbol{\sigma}^2, \boldsymbol{\rho}) = \beta_t + Z_t(\mathbf{x}), \quad t \in \{b, g\}$$

where  $Z_b(\mathbf{x})$  and  $Z_g(\mathbf{x})$  are *independent* stationary GaSP with zero means, variances  $\sigma_b^2$  and  $\sigma_g^2$ , respectively,

$$\text{Cov}(Z_t(\mathbf{x}_1), Z_t(\mathbf{x}_2) | \boldsymbol{\rho}, \boldsymbol{\sigma}^2) = \sigma_t^2 R(\mathbf{x}_1 - \mathbf{x}_2 | \boldsymbol{\rho}_t),$$

and

$$R((z_1, \dots, z_d) | \boldsymbol{\rho}_t) = \prod_{j=1}^d \rho_{tj}^{z_j^2}, \quad t \in \{b, g\}$$

( $0 < \rho_{tj} < 1$ , i.e.,  $\rho_{tj} = \exp\{-\theta_{tj}\}$ ).

## Stage 2

1.  $\beta_b, \beta_g$  are *i.i.d.* non-informative prior
2.  $\sigma_b^2, \sigma_g^2$  are *i.i.d.* as  $\text{IG}(\alpha_\sigma, \gamma_\sigma)$

$$R((z_1, \dots, z_d) | \boldsymbol{\rho}_t) = \prod_{j=1}^d \rho_{tj}^{z_j^2}, \quad t \in \{b, g\}$$

3. Denote  $\boldsymbol{\rho}_t = (\rho_{t1}, \dots, \rho_{td})^\top$ , for  $t \in \{b, g\}$ . Then  $\rho_{bj}, \rho_{gj}$  are *i.i.d.*  $\text{Be}(\alpha_j, \gamma_j)$ , for each  $j = 1, \dots, d$ .

## Meaning

1.  $y(b, \boldsymbol{x})$  and  $y(g, \boldsymbol{x})$  can have “different” means (subject to the prior)
2.  $y(b, \boldsymbol{x})$  and  $y(g, \boldsymbol{x})$  have the “roughly” the same range

**Recall**  $\rho_{bj}$  and  $\rho_{gj}$  are *i.i.d.* for each  $j = 1, \dots, d$ .

$$\begin{aligned}\widehat{y}(t, \mathbf{x}_0) &= \widehat{\beta}_0 + \mathbf{r}^\top(\mathbf{x}_0) \mathbf{R}^{-1}(\mathbf{y}^n - \widehat{\beta} \mathbf{1}_n) \\ &= \widehat{\beta}_0 + \sum_{i=1}^n c_i R(\mathbf{x}_i^{tr} - \mathbf{x}_0 | \boldsymbol{\xi}) \\ &= \widehat{\beta}_0 + \sum_{i=1}^n c_i \exp\left(-\sum_{j=1}^d \theta_{tj} (x_{i,j}^{tr} - x_{0,j})^2\right)\end{aligned}$$

3.  $\implies y(b, \mathbf{x})$  and  $y(g, \mathbf{x})$  have roughly the same number of local maxima and minima in each dimension

## Implementation

$$\hat{y}(t_0, \mathbf{x}_0) = E(Y_0 | \mathbf{Y}^n) = E(E(Y_0 | \mathbf{Y}^n, \boldsymbol{\beta}, \boldsymbol{\sigma}^2, \boldsymbol{\rho}))$$

$$Var(Y_0 | \mathbf{Y}^n) = Var(E(Y_0 | \mathbf{Y}^n, \boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\sigma}^2)) + E(Var(Y_0 | \mathbf{Y}^n, \boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\sigma}^2)),$$

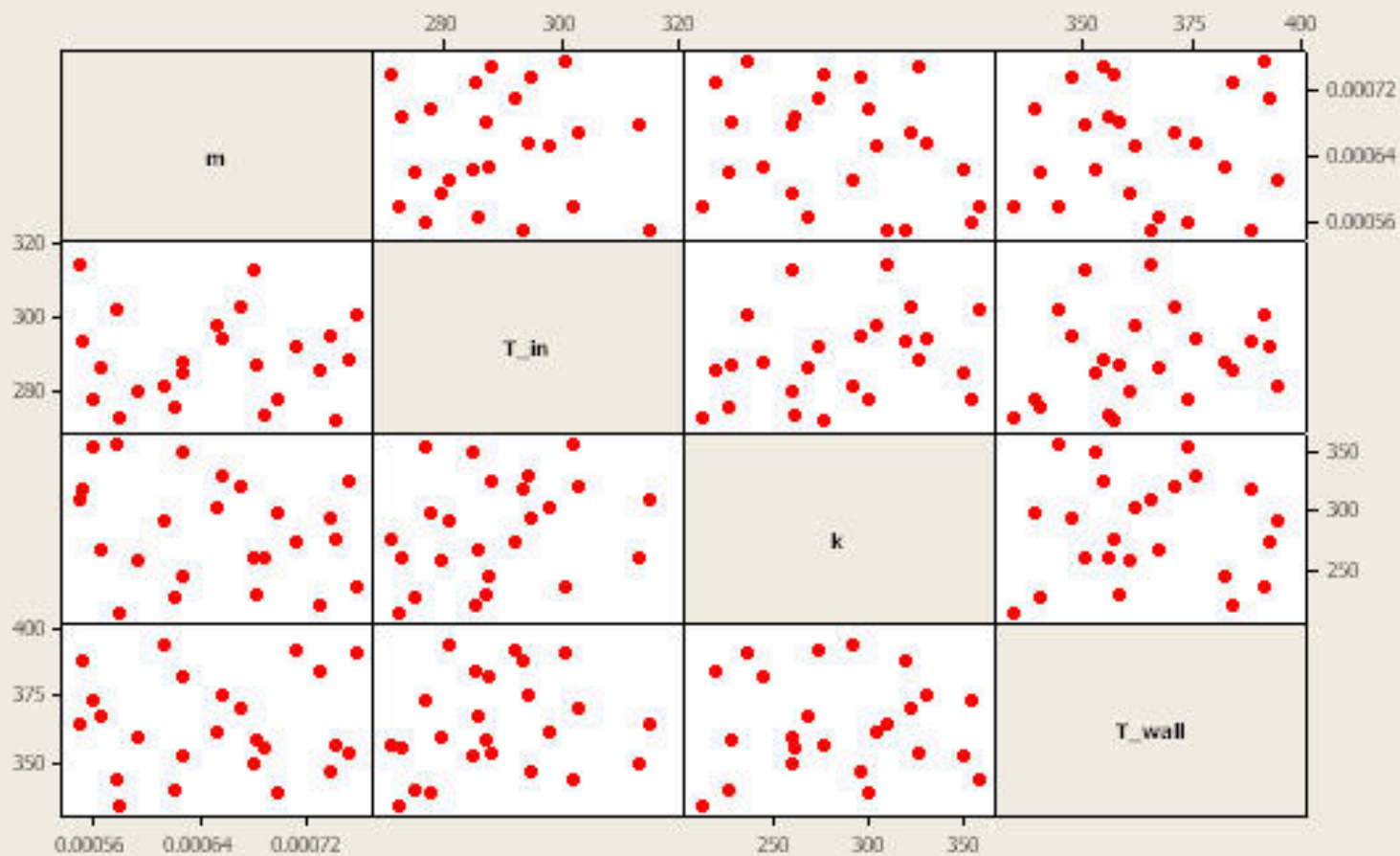
Need  $[\boldsymbol{\beta}, \boldsymbol{\sigma}^2, \boldsymbol{\rho} | \mathbf{Y}^n]$  (Sampling Importance Resampling aka SIR;

Importance Sampling; **MCMC** )

**Example from Qian and Wu** Design of **heat exchanger** used in an electronic cooling application

- $y(\cdot)$  = total rate of steady state heat transfer of the device
- $\boldsymbol{x} = (m \equiv \text{mass flow rate of entry air,}$   
 $T_{in} \equiv \text{temperature of entry air,}$   
 $T_{wall} \equiv \text{temperature of the heat source,}$   
 $k \equiv \text{solid material thermal conductivity})$
- $t \in \{b \equiv \text{fast simulation using finite difference method,}$   
 $g \equiv \text{finite element analysis}\}$
- $y(b, \boldsymbol{x})$  runs require 1-2 seconds while  $y(g, \boldsymbol{x})$  runs require 1-2 hours,
- Data  $\equiv$  24 runs from each code

Matrix Plot of  $m$ ,  $T_{in}$ ,  $k$ ,  $T_{wall}$



$m$	$T_{in}$	$k$	$T_{wall}$	$y(b, \boldsymbol{x})$	$y(g, \boldsymbol{x})$
0.00055	315	310	365	21.23	20.15
0.000552	293.53	318.63	388.29	11.44	10.17
0.00056	277.01	354.98	374	18.55	18.39
0.000566	285.77	266.71	367.27	20.74	20.52
0.000578	302.17	358.13	343.72	30.23	30.12
0.00058	272.26	211.71	333.65	18.13	18.18
0.000594	279.54	258.51	360.13	17.92	19.05
0.000612	280.83	291.53	394.72	17.47	16.95
0.00062	275	225	340	25.07	19.57
0.000626	284.89	350.46	352.29	18.93	23.33

Predict  $y(g, \boldsymbol{x})$  at 8 points

QW	REML-EBLUP	KOH	HQQV	$y(g, \boldsymbol{x})$
26.41	23.35	21.77	20.34	23.54
16.23	23.74	14.33	14.80	15.29
23.66	22.57	22.76	19.09	24.68
25.51	18.22	15.87	18.34	24.96
22.07	29.55	21.32	21.26	22.30
16.74	20.04	17.58	23.81	18.78
16.99	27.56	15.46	18.57	17.41
21.11	21.95	20.16	37.10	42.93
7.83	9.48	8.77	<b>4.29</b>	—

- In product development, approximate simulators may suffice to decrease some parts of the design space
- KOH can have **substantially smaller** prediction error than QW or GS if the approximate and detailed simulators have nearly equivalent behavior in  $x$
- KOH can be have **much greater prediction** error than QW or GS if the approximate and detailed simulators are “substantially” different in their  $x$  behavior
- The GS predictor effectively **combines** information from sets of curves in a wide variety of circumstances
- Most of the models above can be extended to allow for **measurement errors** and hence used to combine physical and computer experiment data into a predictor.