Bayesian Functional Data Analysis for Computer Model Validation

Fei Liu

Institute of Statistics and Decision Sciences
Duke University

NCAR workshop III

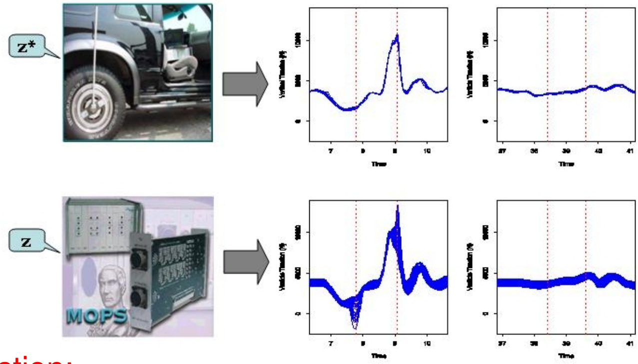
May 21, 2007

Joint work with J. O. Berger et al.

Outline

Outline

Computer model validation



Question:

Does the computer model adequately represent the reality?

Challenge 1 - Expensive-to-run

- Simulator $y^M(\boldsymbol{z})$ is exercised only at

$$y^M(\boldsymbol{z}_1), \ldots, y^M(\boldsymbol{z}_n) \,.$$

- *GaSP* (Sacks *et al.*, 1989) uses statistical models as fast surrogates,

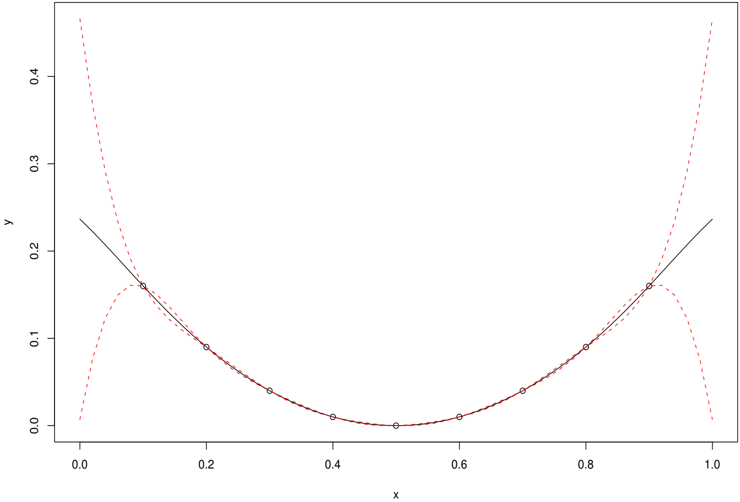
  - Prior:

$$y^M(\cdot) \sim \mathrm{GP}\left(\boldsymbol{\Psi}'(\cdot)\theta^L, \frac{1}{\lambda^M} c^M(\cdot,\cdot)\right) \,.$$

  - Emulator:

$$y^M(\boldsymbol{z}) \mid \mathrm{Data} \sim \mathrm{N}\left(\widehat{m}(\boldsymbol{z}), \widehat{V}(\boldsymbol{z})\right) \,.$$

Interpolator



Challenge 2 - Multiple sources of uncertainty

- Kennedy and O'Hagan (2001) give a broad discussion.
- Three major ones are:
  - Code uncertainty.
  - Calibration input $\boldsymbol{u}^*$.
  - Bias $b_{\boldsymbol{u}^*}$.
- *SAVE* (Bayarri *et al.*, 2005) models scalar outputs as

$$y_r^F(\boldsymbol{v}) = y^M(\boldsymbol{u}^*, \boldsymbol{v}) + b_{\boldsymbol{u}^*}(\boldsymbol{v}) + e_r(\boldsymbol{v}) \, .$$

  - Confounding between $\boldsymbol{u}^*$ and b.
  - Need informative priors.

More challenges

- Uncertain (field) inputs.

$$\boldsymbol{z} = (\boldsymbol{v}, \boldsymbol{u}, \delta) \,.$$

  - $\boldsymbol{v}$: configuration inputs.
  - $\boldsymbol{u}$: calibration inputs (true $\boldsymbol{u}^*$).
  - δ: unknown field inputs (true δ^*).
  - Let $(\boldsymbol{v}_i, \delta^*_{ij})$ denote those for the j^{th} field run in the i^{th} configuration.

- Functional outputs (over time).

  SAVE (Bayarri *et al.*, 2005) treats time t as another input.

Bias structure

$$
\begin{aligned}
y_r^F(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*; t) &= y^M(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*; t) + b(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*; t) + e_{ijr}(t)\,, \\
b(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*; t) &= b_{\boldsymbol{u}^*}(\boldsymbol{v}_i, t) + \epsilon_{ij}^b(t)\,.
\end{aligned}
$$

$$
\begin{aligned}
y^M &: \text{known}\,, \\
b(\cdot,\cdot) &\sim \mathrm{GP}\left(\mu_b, \tau^2 \mathbb{C}\mathrm{orr}_v(\cdot,\cdot)\mathbb{C}\mathrm{orr}_t(\cdot,\cdot)\right)\,, \\
\epsilon_{ij}^b(\cdot) &\sim \mathrm{GP}\left(0, \sigma_b^2 \mathbb{C}\mathrm{orr}_t(\cdot,\cdot)\right)\,, \\
e_{ijr}(\cdot) &\sim \mathrm{GP}\left(0, \sigma^2 \mathbb{C}\mathrm{orr}_t(\cdot,\cdot)\right)\,.
\end{aligned}
$$

Limitations

- Treating t as another input,
  - is only applicable to low dimensional problem.
  - need smoothness assumption.
- Need new methodology to deal with:
  - Complex computer models.
  - High dimensional irregular functional output.
  - Uncertain (field) inputs.

Outline

Notation

$$y_r^F(\boldsymbol{v}_i, \delta_{ij}^*; t) = y^M(\boldsymbol{v}_i, \delta_{ij}^*, \boldsymbol{u}^*; t) + y_{\boldsymbol{u}^*}^B(\boldsymbol{v}_i, \delta_{ij}^*; t) + e_{ijr}(t)$$

Notation

$$y^F_r(\boldsymbol{v}_i, \delta^*_{ij}; t) = y^M(\boldsymbol{v}_i, \delta^*_{ij}, \boldsymbol{u}^*; t) + y^B_{\boldsymbol{u}^*}(\boldsymbol{v}_i, \delta^*_{ij}; t) + e_{ijr}(t)$$

- Inputs to the code.
  - $\boldsymbol{v}_i$: configuration.
  - δ^*_{ij}: unknown inputs.
  - $\boldsymbol{u}^*$: calibration parameters.
- Simulator.
- Bias function.
- Reality y^R.
- Field runs.
- Measurement errors.

Notation

$$y_r^F(\boldsymbol{v}_i, \delta_{ij}^*; t) = y^M(\boldsymbol{v}_i, \delta_{ij}^*, \boldsymbol{u}^*; t) + y_{\boldsymbol{u}^*}^B(\boldsymbol{v}_i, \delta_{ij}^*; t) + e_{ijr}(t)$$

- Inputs to the code.
  - $\boldsymbol{v}_i$: configuration.
  - δ_{ij}^*: unknown inputs.
  - $\boldsymbol{u}^*$: calibration parameters.
- Simulator.
- Bias function.
- Reality y^R.
- Field runs.
- Measurement errors.

Notation

$$y_r^F(\boldsymbol{v}_i, \delta_{ij}^*; t) = y^M(\boldsymbol{v}_i, \delta_{ij}^*, \boldsymbol{u}^*; t) + y_{\boldsymbol{u}^*}^B(\boldsymbol{v}_i, \delta_{ij}^*; t) + e_{ijr}(t)$$

- Inputs to the code.
  - $\boldsymbol{v}_i$: configuration.
  - δ_{ij}^*: unknown inputs.
  - $\boldsymbol{u}^*$: calibration parameters.
- Simulator.
- Bias function.
- Reality y^R.
- Field runs.
- Measurement errors.

Notation

$$y_r^F(\boldsymbol{v}_i, \delta_{ij}^*; t) = y^M(\boldsymbol{v}_i, \delta_{ij}^*, \boldsymbol{u}^*; t) + y_{\boldsymbol{u}^*}^B(\boldsymbol{v}_i, \delta_{ij}^*; t) + e_{ijr}(t)$$

- Inputs to the code.
  - $\boldsymbol{v}_i$: configuration.
  - δ_{ij}^*: unknown inputs.
  - $\boldsymbol{u}^*$: calibration parameters.
- Simulator.
- Bias function.
- Reality y^R.
- Field runs.
- Measurement errors.

Notation

$$y_r^F(\boldsymbol{v}_i, \delta_{ij}^*; t) = y^M(\boldsymbol{v}_i, \delta_{ij}^*, \boldsymbol{u}^*; t) + y^B_{\boldsymbol{u}^*}(\boldsymbol{v}_i, \delta_{ij}^*; t) + e_{ijr}(t)$$

- Inputs to the code.
  - $\boldsymbol{v}_i$: configuration.
  - δ_{ij}^*: unknown inputs.
  - $\boldsymbol{u}^*$: calibration parameters.
- Simulator.
- Bias function.
- Reality y^R.
- Field runs.
- Measurement errors.

Notation

$$y_r^F(\boldsymbol{v}_i, \delta_{ij}^*; t) = y^M(\boldsymbol{v}_i, \delta_{ij}^*, \boldsymbol{u}^*; t) + y_{\boldsymbol{u}^*}^B(\boldsymbol{v}_i, \delta_{ij}^*; t) + e_{ijr}(t)$$

- Inputs to the code.
  - $\boldsymbol{v}_i$: configuration.
  - δ_{ij}^*: unknown inputs.
  - $\boldsymbol{u}^*$: calibration parameters.
- Simulator.
- Bias function.
- Reality y^R.
- Field runs.
- Measurement errors.

Basis expansion

Represent the functions as:

$$
\begin{aligned}
y^M(\boldsymbol{z}_j; t) &= \textstyle\sum_{k=1}^W w_k^M(\boldsymbol{z}_j)\psi_k(t), && j = 1, \ldots, n^M; \\
y^B_{\boldsymbol{u}^*}(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}; t) &= \textstyle\sum_{k=1}^W w_k^B(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij})\psi_k(t), && i = (1, \ldots, m), \\
& && j = (1, \ldots, n_i^f); \\
y^F_r(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}; t) &= \textstyle\sum_{k=1}^W w_{kr}^F(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij})\psi_k(t), && r = 1, \ldots, r_{ij}.
\end{aligned}
$$

Thresholding

Reduce computational expense,

$$
\begin{aligned}
y^M(\boldsymbol{z}_j;t) &\approx \textstyle\sum_{k=1}^{W_0} w_k^M(\boldsymbol{z}_j)\psi_k(t), && j=1,\ldots,n^M;\\
y^B_{\boldsymbol{u}^*}(\boldsymbol{v}_i,\boldsymbol{\delta}^*_{ij};t) &\approx \textstyle\sum_{k=1}^{W_0} w_k^B(\boldsymbol{v}_i,\boldsymbol{\delta}^*_{ij})\psi_k(t), && i=(1,\ldots,m)\,,\\
& && j=(1,\ldots,n_i^f)\,;\\
y^F_r(\boldsymbol{v}_i,\boldsymbol{\delta}^*_{ij};t) &\approx \textstyle\sum_{k=1}^{W_0} w_{kr}^F(\boldsymbol{v}_i,\boldsymbol{\delta}^*_{ij})\psi_k(t), && r=1,\ldots,r_{ij}\,.
\end{aligned}
$$

Emulators

For each $k = 1, \ldots, W_0$,

$$w_k^M(\cdot) \sim \text{GP}\left(\mu^{M_k}, \sigma^{2M_k}\mathbb{C}\text{orr}_k(\cdot,\cdot)\right) .$$

- $\mathbb{C}\text{orr}_k(\cdot,\cdot)$: power exponential family,

$$\mathbb{C}\text{orr}_k(\boldsymbol{z},\boldsymbol{z}') = \exp\left(-\sum_{d=1}^{p} \beta_d^{(M_k)} \mid \boldsymbol{z}_d - \boldsymbol{z}'_d \mid^{\alpha_d^{(M_k)}}\right) .$$

- Modular approach:

$$\left(w_k^M(\boldsymbol{z}) \mid \text{Data}, \hat{\mu}^{M_k}, \hat{\sigma}^{2M_k}, \hat{\boldsymbol{\alpha}}^{M_k}, \hat{\boldsymbol{\beta}}^{M_k}\right) \sim \text{N}\left(\widehat{m}_k^M(\boldsymbol{z}), \widehat{V}_k^M(\boldsymbol{z})\right) .$$

Bayesian analysis

For each $k = 1, \ldots, W_0$,

$$
\begin{aligned}
w^F_{kr}(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) &= w^R_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) + \epsilon_{ijkr}\,, \qquad \epsilon_{ijkr} \sim \mathrm{N}\left(0, \sigma^{2F}_k\right)\,, \\
w^R_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) &= w^M_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}, \boldsymbol{u}^*) + w^B_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij})\,. \\
w^B_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) &= b_k(\boldsymbol{v}_i) + \epsilon^b_{ijk}
\end{aligned}
$$

Bayesian analysis

For each $k = 1, \ldots, W_0$,

$$\begin{aligned} w_{kr}^F(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*) &= w_k^R(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*) + \epsilon_{ijkr}\,, \qquad \epsilon_{ijkr} \sim \mathrm{N}\left(0, \sigma_k^{2F}\right)\,, \\ w_k^R(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*) &= w_k^M(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*) + w_k^B(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*)\,. \\ w_k^B(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*) &= b_k(\boldsymbol{v}_i) + \epsilon_{ijk}^b \end{aligned}$$

- Data:
  - $\bar{w}_{ijk}^F = \frac{1}{r_{ij}} \sum_{r=1}^{r_{ij}} w_{kr}^F(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*)$.
  - $S_{ijk}^2 = \sum_{r=1}^{r_{ij}} (w_{kr}^F(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*) - \bar{w}_{ijk}^F)^2$.
  - $\widehat{m}_k(\cdot), \widehat{V}_k(\cdot)$.

Bayesian analysis

For each $k = 1, \ldots, W_0$,

$$\begin{aligned} w^F_{kr}(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) &= w^R_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) + \epsilon_{ijkr}\,, \qquad \epsilon_{ijkr} \sim \mathrm{N}\left(0, \sigma^{2F}_k\right)\,, \\ w^R_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) &= w^M_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}, \boldsymbol{u}^*) + w^B_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij})\,. \\ w^B_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) &= b_k(\boldsymbol{v}_i) + \epsilon^b_{ijk} \end{aligned}$$

- The bias,

$$b_k(\boldsymbol{v}_i) \sim \mathrm{GP}\left(\mu^B_k, \sigma^{2B}_k \mathbb{C}\mathrm{orr}^B_k(\cdot,\cdot)\right), \qquad \epsilon^b_{ijk} \sim \mathrm{N}(0, \sigma^{2B^\epsilon}_k)\,,$$

with correlation,

$$\mathbb{C}\mathrm{orr}^B_k(\boldsymbol{v}, \boldsymbol{v}') = \exp\left(-\sum_{d=1}^{p_v} \beta^{(B_k)}_d \mid \boldsymbol{v}_d - \boldsymbol{v}'_d \mid^{\alpha^{(B_k)}_d}\right).$$

Bayesian analysis

For each $k = 1, \ldots, W_0$,

$$
\begin{aligned}
w^F_{kr}(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) &= w^R_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) + \epsilon_{ijkr}\,, \qquad \epsilon_{ijkr} \sim \mathrm{N}\left(0, \sigma^{2F}_k\right)\,, \\
w^R_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) &= w^M_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}, \boldsymbol{u}^*) + w^B_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij})\,. \\
w^B_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) &= b_k(\boldsymbol{v}_i) + \epsilon^b_{ijk}
\end{aligned}
$$

- Unknowns:
  - Inputs: $\boldsymbol{u}^*$ and δ^*_{ij}.
  - Bias: $\{b_k(\cdot)\}$, $w^B_k(\boldsymbol{v}_i, \delta^*_{ij})$.
  - Model: $w^M_k(\boldsymbol{v}_i, \delta^*_{ij}, \boldsymbol{u}^*)$.
  - Reality: $\{w^R_k(\boldsymbol{v}_i, \delta^*_{ij})\}$
  - Hyper-parameters: σ^{2B}_k, $\sigma^{2B^\epsilon}_k$, σ^{2F}_k, μ^B_k.

Bayesian analysis

For each $k = 1, \ldots, W_0$,

$$w^F_{kr}(\mathbf{v}_i, \boldsymbol{\delta}^*_{ij}) = w^R_k(\mathbf{v}_i, \boldsymbol{\delta}^*_{ij}) + \epsilon_{ijkr}, \qquad \epsilon_{ijkr} \sim \mathrm{N}\left(0, \sigma^{2F}_k\right),$$

$$w^R_k(\mathbf{v}_i, \boldsymbol{\delta}^*_{ij}) = w^M_k(\mathbf{v}_i, \boldsymbol{\delta}^*_{ij}, \mathbf{u}^*) + w^B_k(\mathbf{v}_i, \boldsymbol{\delta}^*_{ij}).$$

$$w^B_k(\mathbf{v}_i, \boldsymbol{\delta}^*_{ij}) = b_k(\mathbf{v}_i) + \epsilon^b_{ijk}$$

- Unknowns:
  - Inputs: $\mathbf{u}^*$ and δ^*_{ij}.
  - Bias: $\{b_k(\cdot)\}$, $w^B_k(\mathbf{v}_i, \delta^*_{ij})$.
  - Model: $w^M_k(\mathbf{v}_i, \delta^*_{ij}, \mathbf{u}^*)$.
  - Reality: $\{w^R_k(\mathbf{v}_i, \delta^*_{ij})\}$
  - Hyper-parameters: σ^{2B}_k, $\sigma^{2B^\epsilon}_k$, σ^{2F}_k, μ^B_k.

Bayesian analysis

For each $k = 1, \ldots, W_0$,

$$w^F_{kr}(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) = w^R_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) + \epsilon_{ijkr}\,, \qquad \epsilon_{ijkr} \sim \mathrm{N}\left(0, \sigma^{2F}_k\right)\,,$$

$$w^R_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) = w^M_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}, \boldsymbol{u}^*) + w^B_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij})\,.$$

$$w^B_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) = b_k(\boldsymbol{v}_i) + \epsilon^b_{ijk}$$

- Unknowns:
  - Inputs: $\boldsymbol{u}^*$ and $\boldsymbol{\delta}^*_{ij}$.
  - Bias: $\{b_k(\cdot)\}$, $w^B_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij})$.
  - Model: $w^M_k(\boldsymbol{v}_i, \delta^*_{ij}, \boldsymbol{u}^*)$.
  - Reality: $\{w^R_k(\boldsymbol{v}_i, \delta^*_{ij})\}$
  - Hyper-parameters: σ^{2B}_k, $\sigma^{2B^\epsilon}_k$, σ^{2F}_k, μ^B_k.

Bayesian analysis

For each $k = 1, \ldots, W_0$,

$$\begin{aligned} w^F_{kr}(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) &= w^R_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) + \epsilon_{ijkr}\,, \qquad \epsilon_{ijkr} \sim \mathrm{N}\left(0, \sigma^{2F}_k\right)\,, \\ w^R_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) &= w^M_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}, \boldsymbol{u}^*) + w^B_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij})\,. \\ w^B_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) &= b_k(\boldsymbol{v}_i) + \epsilon^b_{ijk} \end{aligned}$$

- Unknowns:
  - Inputs: $\boldsymbol{u}^*$ and $\boldsymbol{\delta}^*_{ij}$.
  - Bias: $\{b_k(\cdot)\}$, $w^B_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij})$.
  - Model: $w^M_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}, \boldsymbol{u}^*)$.
  - Reality: $\{w^R_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij})\}$
  - Hyper-parameters: σ^{2B}_k, $\sigma^{2B^\epsilon}_k$, σ^{2F}_k, μ^B_k.

Bayesian analysis

For each $k = 1, \ldots, W_0$,

$$w_{kr}^F(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*) = w_k^R(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*) + \epsilon_{ijkr}\,, \qquad \epsilon_{ijkr} \sim \mathrm{N}\left(0, \sigma_k^{2F}\right)\,,$$

$$w_k^R(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*) = w_k^M(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*) + w_k^B(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*)\,.$$

$$w_k^B(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*) = b_k(\boldsymbol{v}_i) + \epsilon_{ijk}^b$$

- Unknowns:
  - Inputs: $\boldsymbol{u}^*$ and $\boldsymbol{\delta}_{ij}^*$.
  - Bias: $\{b_k(\cdot)\}$, $w_k^B(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*)$.
  - Model: $w_k^M(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*, \boldsymbol{u}^*)$.
  - Reality: $\{w_k^R(\boldsymbol{v}_i, \boldsymbol{\delta}_{ij}^*)\}$
  - Hyper-parameters: σ_k^{2B}, $\sigma_k^{2B^\epsilon}$, σ_k^{2F}, μ_k^B.

Bayesian analysis

For each $k = 1, \ldots, W_0$,

$$w^F_{kr}(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) = w^R_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) + \epsilon_{ijkr} , \qquad \epsilon_{ijkr} \sim \mathrm{N}\left(0, \sigma^{2F}_k\right) ,$$

$$w^R_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) = w^M_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}, \boldsymbol{u}^*) + w^B_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) .$$

$$w^B_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}) = b_k(\boldsymbol{v}_i) + \epsilon^b_{ijk}$$

- Unknowns:
  - Inputs: $\boldsymbol{u}^*$ and $\boldsymbol{\delta}^*_{ij}$.
  - Bias: $\{b_k(\cdot)\}$, $w^B_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij})$.
  - Model: $w^M_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij}, \boldsymbol{u}^*)$.
  - Reality: $\{w^R_k(\boldsymbol{v}_i, \boldsymbol{\delta}^*_{ij})\}$
  - Hyper-parameters: σ^{2B}_k, $\sigma^{2B^\epsilon}_k$, σ^{2F}_k,μ^B_k.

Outline

The road load data 1

- Time history data at $T = 90843$ time points.
- Inputs include:
  - One configuration $\boldsymbol{v} = \boldsymbol{x}_{nom}$.
  - Seven characteristics $\boldsymbol{x} = (x_1, x_2, \ldots, x_7)$, $\boldsymbol{x} = \boldsymbol{x}_{nom} + \delta$.
  - Two calibration parameters $\boldsymbol{u} = (u_1, u_2)$.

The road load data 2

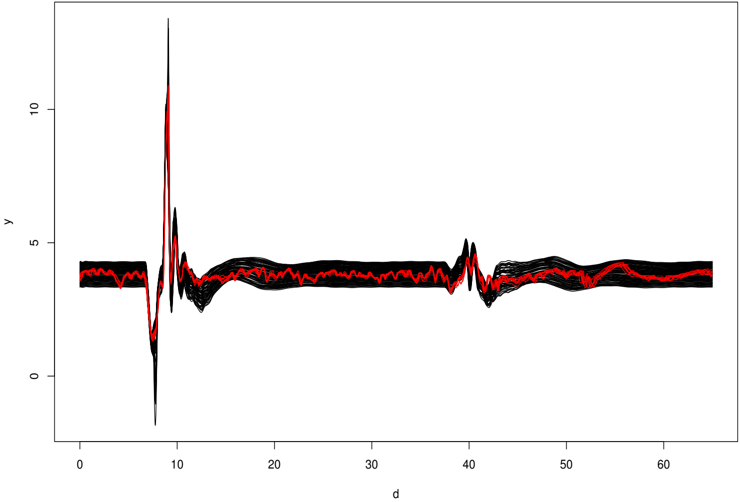
- 64 design points in 9-d space for computer model runs.

$$\{y^M(\boldsymbol{z}, t), \boldsymbol{z} \in \mathrm{D}^M, t \in 1, \ldots, T\}, \qquad \boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{u}).$$

- 7 field replicates.

$$\{y_r^F(\boldsymbol{x}^*, t), r \in (1, \ldots, 7)\}, \qquad \boldsymbol{x}^* = \boldsymbol{x}_{nom} + \delta^*.$$

The road load data 3



Wavelet representation

- Represent the time history data by wavelets:

$$y^M(\boldsymbol{z}_j; t) = \sum_{k=1}^{W} w_k^M(\boldsymbol{z}_j)\psi_k(t), \qquad j = 1, \ldots, 64;$$

$$y_r^F(\boldsymbol{x}^*; t) = \sum_{k=1}^{W} w_{kr}^F(\boldsymbol{x}^*)\psi_k(t), \qquad r = 1, \ldots, 7.$$

- Apply hard thresholding to reduce the computational expense.

  keep $W_0 = 289$ nonzero coefficients.

Emulators

For each wavelet coefficient,

$$w_k^M(\cdot) \sim \mathrm{GP}\left(\mu_k, \frac{1}{\lambda_k^M}\mathbb{C}\mathrm{orr}_k(\cdot,\cdot)\right).$$

- $\mathbb{C}\mathrm{orr}_k(\cdot,\cdot)$: power exponential family,

$$\mathbb{C}\mathrm{orr}_k(\boldsymbol{z},\boldsymbol{z}') = \exp\left(-\sum_{d=1}^{p} \beta_d^{(k)} \mid \boldsymbol{z}_d - \boldsymbol{z}_d' \mid^{\alpha_k^{(k)}}\right).$$

- Response Surface:

$$\left(w_k^M(\boldsymbol{z}) \mid \mathrm{Data}, \hat{\mu}_k, \hat{\lambda}_k^M, \hat{\boldsymbol{\alpha}}_k, \hat{\boldsymbol{\beta}}_k\right) \sim \mathrm{N}\left(\widehat{m}_k(\boldsymbol{z}), \widehat{V}_k(\boldsymbol{z})\right).$$

Bayesian analysis

- For each wavelet coefficient,

$$w_i^R(\boldsymbol{x}^*) = w_i^M(\boldsymbol{x}^*, \boldsymbol{u}^*) + b_i(\boldsymbol{x}^*), \qquad i = 1, \ldots, W,$$

$$w_{ir}^F(\boldsymbol{x}^*) = w_i^R(\boldsymbol{x}^*) + \epsilon_{ir}, \qquad r = 1, \ldots, f.$$

- The bias,

$$b_i(\boldsymbol{x}^*) \sim \mathrm{N}\left(0, \tau_j^2\right) \quad \text{or} \quad b_i(\boldsymbol{x}^*) \sim \mathrm{C}\left(0, \tau_j^2\right).$$

j: wavelet resolution level .

- The error,

$$\epsilon_{ir} \sim \mathrm{N}\left(0, \sigma_i^2\right).$$

Prior distributions

- The Input/Uncertainty Map,

Parameter	Type	Variability	Uncertainty
Damping_1	Calibration	Uncertain	15%
Damping_2	Calibration	Uncertain	15%
Stiffness_1	Manufacturing	Uncertain	10%
Stiffness_2	Manufacturing	Uncertain	10%
Front-rebound_1	Manufacturing	Uncertain	7%
Front-rebound_2	Manufacturing	Uncertain	8%
Sprung-Mass	Manufacturing	Uncertain	5%
Unsprung-Mass	Manufacturing	Uncertain	12%
Body-Pitch-Inertia	Manufacturing	Uncertain	8%

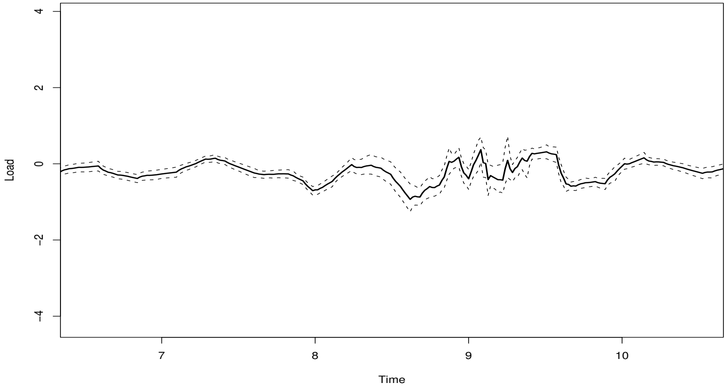
  Calibration: uniform over specified range.

  Manufacturing: truncated normals over the ranges.

- $\pi(\sigma_i^2) \propto 1/\sigma_i^2$, $\pi(\tau_j^2 \mid \{\sigma_i^2\}) \propto \frac{1}{\tau_j^2 + \frac{1}{7}\bar{\sigma}_j^2}$.

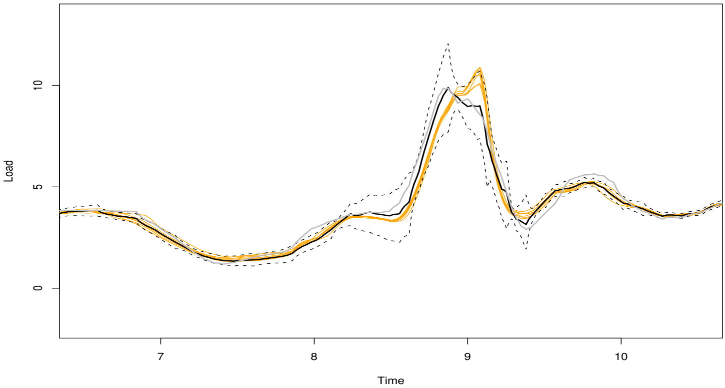
Bias under Gaussian assumption (Full Bayes)

$$b^h(t) = \sum_{i=1}^{W} b_i^h \psi_i(t), h = 1, \ldots, N$$



Reality under Gaussian assumption (Full Bayes)

$$y^{Rh}(\boldsymbol{u}^{*h}, \boldsymbol{x}^{*h}, t) = \sum_{i=1}^{W} \left(w_i^{Mh}(\boldsymbol{u}^{*h}, \boldsymbol{x}^{*h}) + b_i^h \right) \psi_i(t), h = 1, \ldots, N$$



Issue with the Gaussian bias

Confounding between $\boldsymbol{\sigma}^2$ and $\boldsymbol{\tau}^2$. We consider:

$$y_{ir} = \mu_i + b_i + \epsilon_{ir}, i = 1, \ldots, K; r = 1, \ldots, r_i$$

$$b_i \sim \mathrm{N}\left(0, \tau^2\right) \qquad \epsilon_{ir} \sim \mathrm{N}\left(0, \sigma_i^2\right) \ .$$

- The likelihood is

$$\prod_{i=1}^{K} \frac{\sigma_i^{1-n}}{\sqrt{\tau^2 + \frac{1}{n}\sigma_i^2}} \exp\left(-\frac{\left(\bar{y}_i - \mu_i\right)^2}{2\left(\tau^2 + \frac{1}{n}\sigma_i^2\right)} - \frac{s_i^2}{2\sigma_i^2}\right)$$

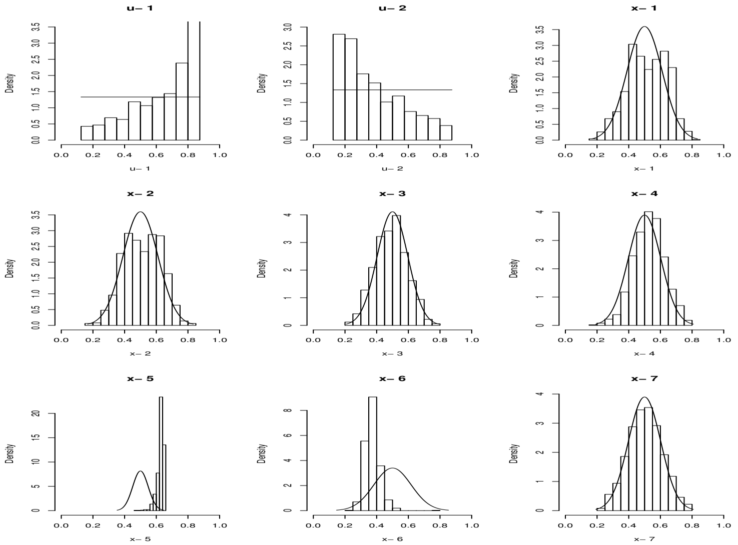
- For large K, the bias will be shrunk towards 0.
- *Modularization*: making inference about the $\{\sigma_i^2\}$ only from the replicate observations.

Analysis with Gaussian bias

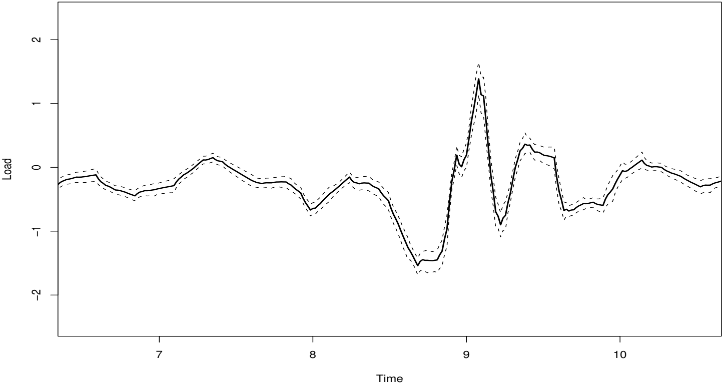
- Conditional on $\boldsymbol{\sigma}^2$, use Gibbs sampling for the rest parameters.
- Determine the posteriors for the σ_i^2 simply by the replicate observations.
- The posterior for $\boldsymbol{\sigma}^2$ is

$$\pi_{post}(\boldsymbol{\sigma}^2 \mid \boldsymbol{D}) \propto \left[\prod_{i \in I} \frac{1}{(\sigma_i^2)^3} \exp\left\{-\frac{s_i^2}{2\sigma_i^2}\right\}\right] \times \int L(\overline{\boldsymbol{w}}^F, \boldsymbol{s}^2 \mid \delta^*, \boldsymbol{u}^*, \boldsymbol{\sigma}^2, \boldsymbol{\tau}^2)\, d\delta^*\, d\boldsymbol{u}^*\, d\boldsymbol{\tau}^2 .$$

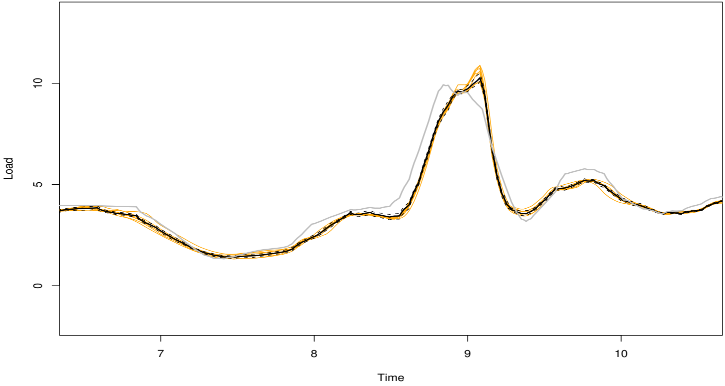
Marginal priors and marginal posteriors



Bias function (under Gaussian bias)



Reality (under Gaussian bias)



Analysis with Cauchy bias

Model the bias by robust distribution

$$\pi(w_i^b \mid \tau_{j(i)}^2) \sim \mathrm{Cauchy}\left(0, \tau_{j(i)}^2\right) .$$

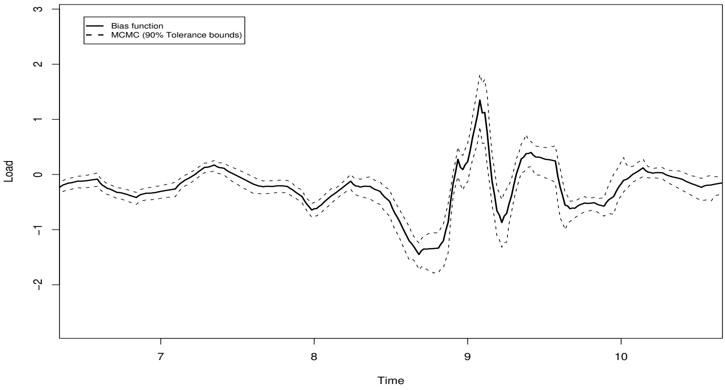
- Normal mixture

$$\pi(w_i^b \mid \tau_{j(i)}^2, \lambda_i) \sim \mathrm{N}\left(0, \tau_{j(i)}^2/\lambda_i\right) ,$$

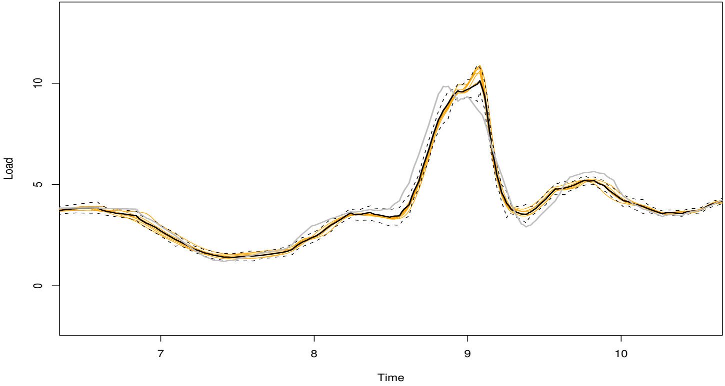
$$\lambda_i \sim \mathrm{Gamma}(\frac{1}{2}, 2) .$$

- Use the regular MCMC.

Bias function (under Cauchy bias)



Reality (under Cauchy bias)



Outline

Multiple computer codes

- Structural input $\boldsymbol{x} \in \boldsymbol{X}^M$.
- Build an emulator across all codes,

$$\begin{aligned} a_k^M(\boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{u}) \sim &\quad \mathrm{GP}\left(\mu_k, \sigma_k^{2M} \mathbb{C}\mathrm{orr}_k^{M_1}(\cdot,\cdot) \mathbb{C}\mathrm{orr}_k^{M_2}(\cdot,\cdot)\right) \,, \\ \mathbb{C}\mathrm{orr}_k^{M_1}(\cdot,\cdot) : &\quad \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R} \,, \\ \mathbb{C}\mathrm{orr}_k^{M_2}(\cdot,\cdot) : &\quad \mathbb{D}^M \times \mathbb{D}^M \rightarrow \mathbb{R} \,. \end{aligned}$$

Eigen-basis representation

- Represent the functional data by $\xi_k(t) = \sum_{i \in I} b_{ki} \Psi_i(t)$ (Ramsay, 1997),

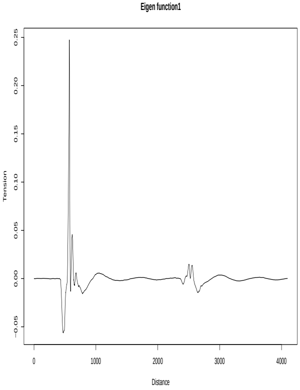
$$\sum_{i \in I} w_i^M(\boldsymbol{v}, \delta, \boldsymbol{u}) \Psi_i(t) \approx \sum_{k=1}^{p} a_k^M(\boldsymbol{v}, \delta, \boldsymbol{u}) \xi_k(t)$$

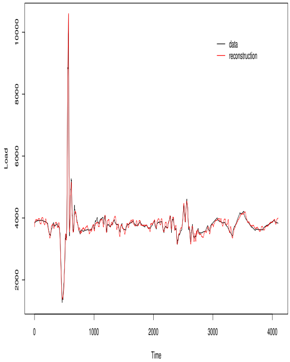
$$\sum_{i \in I} w_{ir}^F(\boldsymbol{v}, \delta) \Psi_i(t) \approx \sum_{k=1}^{p} a_{kr}^F(\boldsymbol{v}, \delta) \xi_k(t) \,.$$

- To obtain $\xi_k(t)$, we have

$$N^{-1} \boldsymbol{W}^t \boldsymbol{W} \boldsymbol{b}_k = \rho_k \boldsymbol{b}_k \,.$$

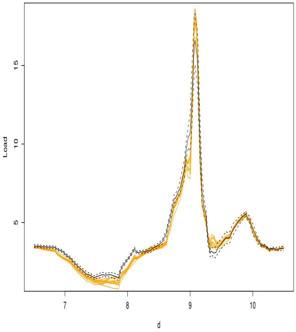
- Emulators for $\{a_k^M(\boldsymbol{v}, \delta, \boldsymbol{u})\}$.

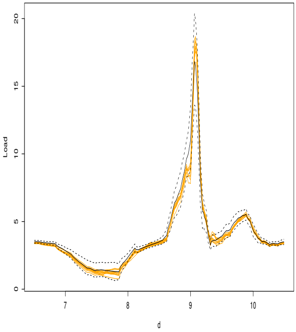

Eigen function1
Tension
Distance
0
1000
2000
3000
4000
-0.05
0.00
0.05
0.10
0.15
0.20
0.25


data
reconstruction
Load
Time
0
1000
2000
3000
4000
2000
4000
6000
8000
10000

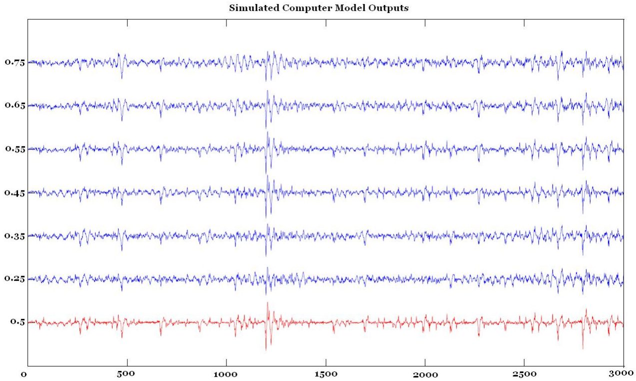
Result

Interpolate the bias/reality into new settings.





Dynamic Linear models

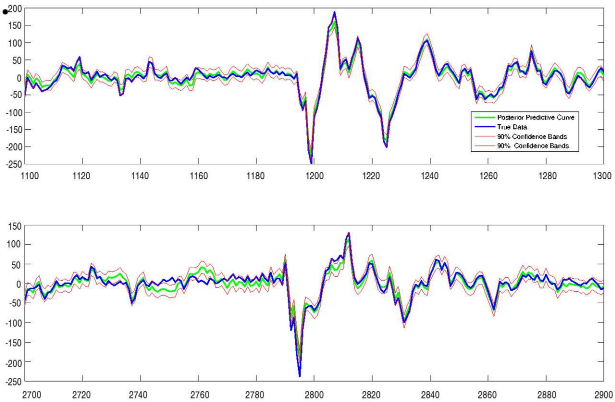


- Model the computer model run at $\boldsymbol{z}$ by multivariate TVAR (West and Harrison, 1997),

$$y^M(\boldsymbol{z}, t) = \sum_{j=1}^{p} \phi_{t,j} y^M(\boldsymbol{z}, t-j) + \underbrace{\epsilon_t^M(\boldsymbol{z})}_{\Downarrow \atop \mathrm{GP}(0, \sigma_t^2 \mathbb{C}\mathrm{orr}(\cdot,\cdot))},$$

- Interpolator with forecasting capability.
- Computation is done by Forward filtering backward sampling algorithm.
- Predict at untried input $\boldsymbol{z} = 0.5$.

Result


Posterior Predictive Curve
True Data
90% Confidence Bands
90% Confidence Bands
200
150
100
50
0
-50
-100
-150
-200
-250
1100
1120
1140
1160
1180
1200
1220
1240
1260
1280
1300
2700
2720
2740
2760
2780
2800
2820
2840
2860
2880
2900

Outline

Summary

- We developed various Bayesian functional data analysis approaches to computer model validation.
- The approaches automatically take into account model discrepancy and model calibration.
- Unknown (field) inputs can be incorporated.
- Bias functions are modeled using hierarchical structures when needed.

On-going work : Spatio-Temporal Outputs

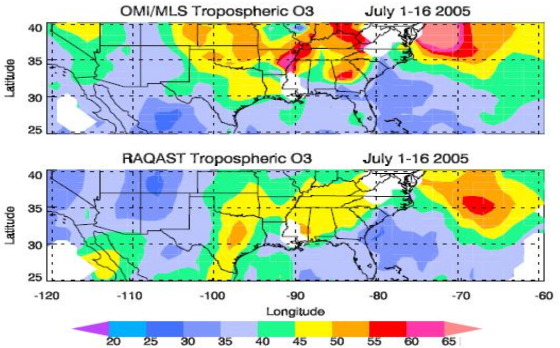


Figure: Regional Air Quality forecAST (RAQAST) Over the U.S. (S. Guillas et al., 2006

- The model,

$$y^F(s, \boldsymbol{\delta}^*; t) = y^M(s, \boldsymbol{\delta}^*, \boldsymbol{u}^*; t) + b(s, \boldsymbol{\delta}^*; t) + \epsilon_t^F .$$

  - s: location (configuration).
  - $\boldsymbol{u}$: calibration parameters.
  - $\boldsymbol{\delta}$: meteorology.
  - t: time.

- Spatial interpolation,

$$y^M(s, \boldsymbol{\delta}, \boldsymbol{u}; t) = \sum_{k \in \mathbb{K}} a_k(s, \boldsymbol{\delta}, \boldsymbol{u}) \psi_k(t) .$$

- Forecasting,

$$b(s, \boldsymbol{\delta}; t) = \sum_{j=1}^{p} \phi_{t,j} b(s, \boldsymbol{\delta}; t - j) + \epsilon_t^b(s, \boldsymbol{\delta}) .$$

On-going work: Time-dependent parameters

- Time-dependent parameters (Reichert, 2006)

$$y_t^F = \underbrace{y_t^M(\boldsymbol{z}^F + \phi_t^z)}_{\Downarrow} + \phi_t^F + \epsilon_t^F \, , \phi_t^z \sim \text{Stochastic process} \, ,$$

$$y_t^M(\boldsymbol{z}^F + \phi_t^z) \approx y_t^M(\boldsymbol{z}^F) + \nabla y_t^M(\boldsymbol{z}^F)\phi_t^z \, .$$

- Build emulators for $y_t^M(\boldsymbol{z}^F)$ and $\nabla y_t^M(\boldsymbol{z}^F)$.

Thank you!