Calibration and uncertainty quantification using multivariate simulator output

LA-UR 04-7145

Dave Higdon, Statistical Sciences Group, LANL
Jim Gattiker, Statistical Sciences Group, LANL
Brian Williams, Statistical Sciences Group, LANL
Inference combining a physics model with experimental data

Data generated from model:
\[ \frac{d^2z}{dt^2} = -1 - 0.3 \frac{dz}{dt} + \epsilon \]

Simulation model:
\[ \frac{d^2z}{dt^2} = -1 \]

Statistical model:
\[ y(z) = \eta(z) + \delta(z) + \epsilon \]

Improved physics model:
\[ \frac{d^2z}{dt^2} = -1 - \theta \frac{dz}{dt} + \epsilon \]

Statistical model:
\[ y(z) = \eta(z, \theta) + \delta(z) + \epsilon \]
Basic formulation – borrows from Kennedy and O’Hagan (2001)

\[ x \quad \text{experimental conditions} \]
\[ \theta \quad \text{calibration parameters} \]
\[ \zeta(x) \quad \text{true physical system response given inputs } x \]
\[ \eta(x, \theta) \quad \text{simulator response at } x \text{ and } \theta. \]
\[ y(x) \quad \text{experimental observation of the physical system} \]
\[ \delta(x) \quad \text{discrepancy between } \zeta(x) \text{ and } \eta(x, \theta) \]
\[ e(x) \quad \text{observation error of the experimental data} \]

\[ y(x) = \zeta(x) + e(x) \]
\[ y(x) = \eta(x, \theta) + \delta(x) + e(x) \]
A Bayesian approach for combining simulations and experimental data for forecasting, calibration and uncertainty quantification

- A simple example...
  
  \( x \) model or system inputs
  
  \( \theta \) model calibration parameters
  
  \( \zeta(x) \) true physical system response given inputs \( x \)
  
  \( \eta(x, \theta) \) simulator response at \( x \) and \( \theta \).
  
  \( y(x) \) experimental observation of the physical system
  
  \( e(x) \) observation error of the experimental data

Assume: 
\[
y(x) = \zeta(x) + e(x)
\]
  
\[
= \eta(x, \theta) + e(x)
\]
  
\( \theta \) unknown.

Standard Bayesian estimation gives: 
\[ \pi(\theta|y(x)) \propto L(y(x)|\eta(x, \theta)) \times \pi(\theta) \]
Accounting for limited simulator runs


\[ x \] model or system inputs
\[ \theta \] calibration parameters
\[ \zeta(x) \] true physical system response given inputs \( x \)
\[ \eta(x, \theta) \] simulator response at \( x \) and \( \theta \).

simulator run at limited input settings
\[ \eta = (\eta(x_1^*, \theta_1^*), \ldots, \eta(x_m^*, \theta_m^*))^T \]

treat \( \eta(\cdot, \cdot) \) as a random function
use GP prior for \( \eta(\cdot, \cdot) \)

\[ y(x) \] experimental observation of the physical system
\[ e(x) \] observation error of the experimental data

\[
y(x) = \zeta(x) + e(x) \\
y(x) = \eta(x, \theta) + e(x)
\]
Accounting for limited simulation runs

Again, standard Bayesian estimation gives:

\[ \pi(\theta, \eta(\cdot, \cdot)|y(x)) \propto L(y(x)|\eta(x, \theta)) \times \pi(\theta) \times \pi(\eta(\cdot, \cdot)) \]

- Posterior means and quantiles shown.
- Uncertainty in \( \theta \) and \( \eta(x, \theta) \) are incorporated into the forecast.
- Gaussian process models for \( \eta(\cdot, \cdot) \).
Accounting for model discrepancy


\[
x \quad \text{model or system inputs}
\]

\[
\theta \quad \text{model or system inputs}
\]

\[
\zeta(x) \quad \text{true physical system response given inputs } x
\]

\[
\eta(x, \theta) \quad \text{simulator response at } x \text{ and } \theta.
\]

\[
y(x) \quad \text{experimental observation of the physical system}
\]

\[
\delta(x) \quad \text{discrepancy between } \zeta(x) \text{ and } \eta(x, \theta)
\]

\[
e(x) \quad \text{observation error of the experimental data}
\]

\[
y(x) = \zeta(x) + e(x)
\]

\[
y(x) = \eta(x, \theta) + \delta(x) + e(x)
\]

\[
y(x) = \eta(x, \theta) + \delta_n(x) + \delta_b(x) + e(x)
\]
Again, standard Bayesian estimation gives:

\[ \pi(\theta, \delta_n, \delta_b | y(x)) \propto L(y(x) | \eta(x, \theta), \delta(x)) \times \pi(\theta) \times \pi(\eta) \times \pi(\delta) \]

- Posterior means and 90% CI's shown.
- Posterior prediction for \( \zeta(x) \) is obtained by computing the posterior distribution for \( \eta(x, \theta) + \delta(x) \)
- Uncertainty in \( \theta, \eta(x, t), \) and \( \delta(x) \) are incorporated into the forecast.
- Gaussian process models for \( \eta(x, t) \) and \( \delta(x) \)
• Initial work on implosion for fat man.
• Use high explosive (HE) to crush steel cylindrical shells
• Investigate the feasibility of a controlled implosion
Some History

Early work on cylinders called “beer can experiments.”

- Early work not encouraging:
  - “…I question Dr. Neddermeyer’s seriousness…” – Deke Parsons.
  - “It stinks.” – R. Feynman
  - Teller and VonNeumann were quite supportive of the implosion idea

Data on collapsing cylinder from high speed photography.

Symmetrical implosion eventually accomplished using HE lenses by Kistiakowsky.

Implosion played a key role in early computer experiments.

Feynman worked on implosion calculations with IBM accounting machines.

Eventually first computer with addressable memory was developed (MANIAC 1).
The Experiments

Fig. 10
SECTION OF TYPICAL ASSEMBLY
DRAWN TO SCALE OF EXPERIMENT # 26

Fig. 11: Exp. 11: 3" CO, 1/2" wall, 6" long
Wood, 1/3" thick, 1/3" long

Both detonated from 3 points at lower end in photograph

Fig. 12: Exp. 12: 3" CO, 1/2" wall, 6" long
Comp. 6, 1/3" thick, 1/3" long

Exp. 13: 3" CO, 1/2" wall, 6" long
Exp. 14: 3" CO, 1/2" wall, 6" long
Comp. 6, 1/3" thick, 1/3" long

Plastic flow can be seen through end of cylinder
Energy from HE imparts an initial inward velocity to the cylinder

$$v_0 = \frac{m_e}{m} \sqrt{\frac{2u_0}{1 + m_e/m}}$$

mass ratio $m_e/m$ of HE to steel; $u_0$ energy per unit mass from HE.

Energy converts to work done on the cylinder:

work per unit mass $= w = \frac{s}{2\rho(1 - \lambda)} \left\{ r_i^2 \log r_i^2 - r_o^2 \log r_o^2 + \lambda^2 \log \lambda^2 \right\}$

$r_i =$ scaled inner radius; $r_o =$ scaled outer radius; $\lambda =$ initial $r_i/r_o$; $s =$ steel yielding stress; $\rho =$ density of steel.
Neddermeyer’s Model

ODE: \[ \frac{dr}{dt} = \left[ \frac{1}{R_1^2 f(r)} \left\{ v_0^2 - \frac{s}{\rho g(r)} \right\} \right]^{1/2} \]

where

\[ r = \text{inner radius of cylinder – varies with time} \]
\[ R_1 = \text{initial outer radius of cylinder} \]
\[ f(r) = \frac{r^2}{1 - \lambda^2} \ln \left( \frac{r^2 + 1 - \lambda^2}{r^2} \right) \]
\[ g(r) = (1 - \lambda^2)^{-1} [r^2 \ln r^2 - (r^2 + 1 - \lambda^2) \ln(r^2 + 1 - \lambda^2) - \lambda^2 \ln \lambda^2] \]
\[ \lambda = \text{initial ratio of cylinder } r(t = 0)/R_1 \]

constant volume condition: \( r_{\text{outer}}^2 - r^2 = 1 - \lambda^2 \)
Goal: use experimental data to calibrate $s$ and $u_0$; obtain prediction uncertainty for new experiment

Hypothetical data obtained from photos at different times during the 3 experimental implosions. All cylinders had a 1.5in outer and a 1.0in inner radius. ($\lambda = \frac{2}{3}$).

$$m_e/m \approx 0.32 \quad m_e/m \approx 0.17 \quad m_e/m \approx 0.36$$
Carry out simulated implosions using Neddermeyer’s model

Sequence of runs carried at $m$ input settings $(x^*, \theta_1^*, \theta_2^*) = (m_e/m, s, u_0)$ varying over predefined ranges using an OA$(32, 4^3)$-based LH design.

$\begin{pmatrix} x_1^* & \theta_{11}^* & \theta_{12}^* \\ \vdots & \vdots & \vdots \\ x_m^* & \theta_{m1}^* & \theta_{m2}^* \end{pmatrix}$

Each simulation produces a $n_\eta = 22 \cdot 26$ vector of radii for 22 times $\times$ 26 angles.
Generating OA-based LH designs

Example: \( N = 16 \), 3 factors each at 4 levels

\[ \text{OA}(16, 4^3) \] design

induced LH design

Ensures some higher dimensional filling relative to standard LH designs.
Example: NOA(48, 4^8) 
N = 48, 8 factors each at 4 levels 
columns of NOA design matrix \( X \) are not exactly orthogonal \( \Rightarrow \) allows more factors with good higher dimensional properties.
PC representation of simulation output

Ξ = [η₁; · · · ; ηₘ] – a \( n_η \times m \) matrix that holds output of \( m \) simulations

SVD decomposition: \( \Xi = UDV^T \)

\( K_η \) is 1st \( p_η \) columns of \( \left[ \frac{1}{\sqrt{m}} UD \right] \) – columns of \( \left[ \sqrt{m} V^T \right] \) have variance 1

Cylinder example:

\( p_η = 3 \) PC’s: \( K_η = [k_1; k_2; k_3] \) – each vector \( k_i \) holds trace of PC \( i \).

\( k_i \)'s do not change with \( \phi \) – from symmetry of Neddermeyer’s model.

Simulated trace \( \eta(x_1^*, \theta_{i1}^*, \theta_{i2}^*) = K_η w(x_1^*, \theta_{i1}^*, \theta_{i2}^*) + \epsilon_i, \epsilon_i \text{'s} \sim iid N(0, \lambda_η^{-1}) \), for any set of tried simulation inputs \((x_1^*, \theta_{i1}^*, \theta_{i2}^*)\).
Gaussian process models for PC weights

Want to evaluate \( \eta(x, \theta_1, \theta_2) \) at arbitrary input setting \((x, \theta_1, \theta_2)\).

Also want analysis to account for uncertainty here.

Approach: model each PC weight as a Gaussian process:

\[
w_i(x, \theta_1, \theta_2) \sim \text{GP}(0, \lambda_{wi}^{-1} R((x, \theta), (x', \theta'); \rho_{wi}))
\]

where

\[
R((x, \theta), (x', \theta'); \rho_{wi}) = \prod_{k=1}^{p_x} \rho_{wik}^{-4(x_k - x'_k)^2} \times \prod_{k=1}^{p_\theta} \rho_{wip_{x+k}}^{-4(\theta_k - \theta'_k)^2}
\]  

(1)

Restricting to the design settings \[
\begin{pmatrix}
x_1^* & \theta_{11}^* & \theta_{12}^* \\
\vdots & \vdots & \vdots \\
x_m^* & \theta_{m1}^* & \theta_{m2}^*
\end{pmatrix}
\]

and specifying

\[
w_i = (w_i(x_1^*, \theta_{11}^*, \theta_{12}^*), \ldots, w_i(x_m^*, \theta_{m1}^*, \theta_{m2}^*))^T
\]

gives

\[
w_i \overset{iid}{\sim} N \left(0, \lambda_{wi}^{-1} R((x^*, \theta^*); \rho_{wi})\right), \quad i = 1, \ldots, p_\eta
\]

where \( R((x^*, \theta^*); \rho_{wi})_{m \times m} \) is given by (1).

*note: additional nugget term \( w_i \overset{iid}{\sim} N \left(0, \lambda_{wi}^{-1} R((x^*, \theta^*); \rho_{wi}) + \lambda_{ci}^{-1} I_m\right), \quad i = 1, \ldots, p_\eta \), may be useful.
Gaussian process models for PC weights

At the $m$ simulation input settings the $mp_\eta$-vector $w$ has prior distribution

$$w = \begin{pmatrix} w_1 \\ \vdots \\ w_{p_\eta} \end{pmatrix} \sim N \left( \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} \lambda_{w1}^{-1} R((x^*, \theta^*); \rho_{w1}) & 0 & 0 \\ 0 & \ddots & \vdots \\ 0 & 0 & \lambda_{wp_\eta}^{-1} R((x^*, \theta^*); \rho_{wp_\eta}) \end{pmatrix} \right)$$

$$\Rightarrow w \sim N(0, \Sigma_w);$$

note $\Sigma_w = I_{p_\eta} \otimes \lambda_{w}^{-1} R((x^*, \theta^*); \rho_w)$ can break down.

Emulator likelihood: $\eta = \text{vec}([\eta(x^*_1, \theta^*_{11}, \theta^*_{12}); \cdots; \eta(x^*_m, \theta^*_{m1}, \theta^*_{m2})])$

$$L(\eta|w, \lambda_\eta) \propto \lambda_\eta^{\frac{mn_\eta}{2}} \exp \left\{ -\frac{1}{2} \lambda_\eta (\eta - Kw)^T (\eta - Kw) \right\}, \quad \lambda_\eta \sim \Gamma(a_\eta, b_\eta)$$

where $n_\eta$ is the number of observations in a simulated trace and

Equivalently

$$K = [I_m \otimes k_1; \cdots; I_m \otimes k_{p_\eta}].$$

$$L(\eta|w, \lambda_\eta) \propto \lambda_\eta^{\frac{mp_\eta}{2}} \exp \left\{ -\frac{1}{2} \lambda_\eta (w - \hat{w})^T (K^T K) (w - \hat{w}) \right\} \times \lambda_\eta^{\frac{m(n_\eta - p_\eta)}{2}} \exp \left\{ -\frac{1}{2} \lambda_\eta \eta^T (I - K (K^T K)^{-1} K^T) \eta \right\} \times \lambda_\eta^{\frac{mp_\eta}{2}} \exp \left\{ -\frac{1}{2} \lambda_\eta (w - \hat{w})^T (K^T K) (w - \hat{w}) \right\}, \quad \lambda_\eta \sim \Gamma(a'_\eta, b'_\eta)$$

$$a'_\eta = a_\eta + \frac{m(n_\eta - p_\eta)}{2}, \quad b'_\eta = b_\eta + \frac{1}{2} \eta^T (I - K (K^T K)^{-1} K^T) \eta, \quad \hat{w} = (K^T K)^{-1} K^T \eta.$$
Gaussian process models for PC weights

Resulting posterior can then be based on computed PC weights $\hat{w}$:

\[
\hat{w} | w, \lambda_\eta \sim N(w, (\lambda_\eta K^T K)^{-1}) \\
w | \lambda_w, \rho_w \sim N(0, \Sigma_w) \\
\Rightarrow \hat{w} | \lambda_\eta, \lambda_w, \rho_w \sim N(0, (\lambda_\eta K^T K)^{-1} + \Sigma_w)
\]

Resulting posterior is then:

\[
\pi(\lambda_\eta, \lambda_w, \rho_w | \hat{w}) \propto \left|(\lambda_\eta K^T K)^{-1} + \Sigma_w\right|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} \hat{w}^T [\lambda_\eta K^T K]^{-1} + \Sigma_w \right\}^{-1} \hat{w} \times \\
\lambda_\eta^{d_\eta - 1} e^{-b_\eta \lambda_\eta} \times \prod_{i=1}^{p_\eta} \lambda_{wi}^{a_w - 1} e^{-b_w \lambda_{wi}} \times \\
\prod_{i=1}^{p_\eta} \left\{ \prod_{j=1}^{p_x} (1 - \rho_{w_{ij}})^{b_\rho - 1} \prod_{j=1}^{p_\theta} (1 - \rho_{w_{ij}(j+p_x)})^{b_\rho - 1} \right\}
\]

MCMC via Metropolis works fine here.

Bounded range of $\rho_{wij}$'s facilitates MCMC.
Posterior distribution of $\rho_w$

Separate models by PC

More opportunity to take advantage of effect sparsity
Predicting simulator output at untried \((x^*, \theta_1^*, \theta_2^*)\)

Want \(\eta(x^*, \theta_1^*, \theta_2^*) = Kw(x^*, \theta_1^*, \theta_2^*)\)

For a given draw \((\lambda_\eta, \lambda_w, \rho_w)\) a draw of \(w^*\) can be produced:

\[
\begin{pmatrix}
\hat{w} \\
w^*
\end{pmatrix} \sim N \left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} (\lambda_\eta K^T K)^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \Sigma_{w,w^*}(\lambda_w, \rho_w) \right)
\]

Define

\[
V = \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix} = \begin{pmatrix} (\lambda_\eta K^T K)^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \Sigma_{w,w^*}(\lambda_w, \rho_w)
\]

Then

\[
w^*|\hat{w} \sim N(V_{21}V_{11}^{-1}\hat{w}, V_{22} - V_{21}V_{11}^{-1}V_{12})
\]

Realizations can be generated from sample of MCMC output.

Lots of info (data?) makes conditioning on point estimate \((\hat{\lambda}_\eta, \hat{\lambda}_w, \hat{\rho}_w)\) a good approximation to the posterior.

Posterior mean or median work well for \((\hat{\lambda}_\eta, \hat{\lambda}_w, \hat{\rho}_w)\)
Exploring sensitivity of simulator output to model inputs
Simulator predictions varying 1 input, holding others at nominal
Basic formulation – borrows from Kennedy and O’Hagan (2001)

Let $(t, \phi)$ be the simulation output space.

- **$x$** are experimental conditions.
- **$\theta$** are calibration parameters.
- **$\zeta(x)$** is the true physical system response given conditions $x$.
- **$\eta(x, \theta)$** is the simulator response at $x$ and $\theta$.
- **$y(x)$** is the experimental observation of the physical system.
- **$\delta(x)$** is the discrepancy between $\zeta(x)$ and $\eta(x, \theta)$.
- This discrepancy may be decomposed into numerical error and bias:
  - **$e(x)$** is the observation error of the experimental data.

The equations are:

$$y(x) = \zeta(x) + e(x)$$

$$y(x) = \eta(x, \theta) + \delta(x) + e(x)$$

**Parameters**:

- $x = \frac{m_e}{m} \approx 0.32$
- $\theta_1 = s \approx ?$
- $\theta_2 = u_0 \approx ?$
Kernel basis representation for spatial processes $\delta(s)$

Define $p_\delta$ basis functions $d_1(s), \ldots, d_{p_\delta}(s)$.

-2 0 2 4 6 8 10 12

Here $d_j(s)$ is normal density centered at spatial location $\omega_j$:

$$d_j(s) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(s - \omega_j)^2\right\}$$

Set $\delta(s) = \sum_{j=1}^{p_\delta} d_j(s) v_j$ where $v \sim N(0, \lambda_v^{-1} I_{p_\delta})$.

Can represent $\delta = (\delta(s_1), \ldots, \delta(s_n))^T$ as $\delta = Dv$ where

$$D_{ij} = d_j(s_i)$$
\( v \) and \( d(s) \) determine spatial processes \( \delta(s) \)

\[
d_j(s)v_j
\]

\[
\delta(s) = \sum_{j=1}^{p_\delta} d_j(s)v_j \text{ where } v \sim N(0, \lambda_v^{-1}I_{p_\delta}).
\]

Continuous representation:

Discrete representation: For \( \delta = (\delta(s_1), \ldots, \delta(s_n))^T \), \( \delta = Dv \) where \( D_{ij} = d_j(s_i) \)
Basis representation of discrepancy

Represent discrepancy $\delta(x)$ using basis functions and weights $p_\delta = 24$ basis functions over $(t, \phi)$; $D = [d_1; \cdots; d_{p_\delta}]$; $d_k$’s hold basis.

$$\delta(x) = Dv(x) \text{ where } v(x) \sim \text{GP} \left(0, \lambda_v^{-1} I_{p_\delta} \otimes R(x, x'; \rho_v)\right)$$

with

$$R(x, x'; \rho_v) = \prod_{k=1}^{p_x} \rho_{v_k}^{-4(x_k-x'_k)^2} \tag{2}$$

$$p_\delta = 24$$

$$D = [d_1; \cdots; d_{p_\delta}]$$

$$d_k$$’s hold basis.
Integrated model formulation

Data \( y(x_1), \ldots, y(x_n) \) collected for \( n \) experiments at input conditions \( x_1, \ldots, x_n \).

Each \( y(x_i) \) is a collection of \( n_{y_i} \) measurements over points indexed by \((t, \phi)\).

\[
y(x_i) = \eta(x_i, \theta) + \delta(x_i) + e_i \\
= K_i w(x_i, \theta) + D_i v(x_i) + e_i
\]

\[
y(x_i) | w(x_i, \theta), v(x_i), \lambda_y \sim N \left( \begin{bmatrix} D_i; K_i \end{bmatrix} \begin{bmatrix} v(x_i) \\ w(x_i, \theta) \end{bmatrix}, (\lambda_y W_i)^{-1} \right)
\]

Since support of each \( y(x_i) \) varies and doesn’t match that of sims, the basis vectors in \( K_i \) must be interpolated from \( K_\eta \); similarly, \( D_i \) must be computed from the support of \( y(x_i) \):

*note: cubic spline interpolation over \((\text{time}, \phi)\) used here.
Integrated model formulation

Define

\[ n_y = n_{y_1} + \cdots + n_{y_n}, \] the total number of experimental data points,

\[ y \] to be the \( n_y \)-vector from concatenation of the \( y(x_i) \)'s,

\[ v = \text{vec}([v(x_1); \cdots; v(x_n)]^T) \] and

\[ u(\theta) = \text{vec}([w(x_1, \theta_1, \theta_2); \cdots; w(x_n, \theta_1, \theta_2)]^T) \]

\[ y \mid v, u(\theta), \lambda_y \sim \mathcal{N}\left(B \begin{pmatrix} v \\ u(\theta) \end{pmatrix}, (\lambda_y W_y)^{-1}\right), \lambda_y \sim \Gamma(a_y, b_y) \quad (3) \]

where

\[ W_y = \text{diag}(W_1, \ldots, W_n) \] and

\[ B = \text{diag}(D_1, \ldots, D_n, K_1, \ldots, K_n) \begin{pmatrix} P_D^T & 0 \\ 0 & P_K^T \end{pmatrix} \]

\( P_D \) and \( P_K \) are permutation matrices whose rows are given by:

\[ P_D(j + n(i - 1); \cdot) = e_{(j-1)p_{\delta} + i}^T, \quad i = 1, \ldots, p_{\delta}; \quad j = 1, \ldots, n \]

\[ P_K(j + n(i - 1); \cdot) = e_{(j-1)p_{\eta} + i}^T, \quad i = 1, \ldots, p_{\eta}; \quad j = 1, \ldots, n \]
Integrated model formulation (continued)

Equivalently (3) can be represented

\[
\begin{pmatrix}
\hat{v} \\
\hat{u}
\end{pmatrix} \sim \mathcal{N}
\begin{pmatrix}
v \\
u(\theta)
\end{pmatrix}, \lambda_y \sim \mathcal{N}
\begin{pmatrix}
\hat{v} \\
u(\theta)
\end{pmatrix}, (\lambda_y B^T W_y B)^{-1}, \lambda_y \sim \Gamma(a'_y, b'_y)
\]

with

\[
n_y = n_{y1} + \cdots + n_{yn}, \text{ the total number of experimental data points}
\]

\[
\begin{pmatrix}
\hat{v} \\
\hat{u}
\end{pmatrix} = (B^T W_y B)^{-1} B^T W_y y
\]

\[
a'_y = a_y + \frac{1}{2}[n_y - n(p_\delta + p_\eta)]
\]

\[
b'_y = b_y + \frac{1}{2} \left[ (y - B \begin{pmatrix}
\hat{v} \\
\hat{u}
\end{pmatrix})^T W_y (y - B \begin{pmatrix}
\hat{v} \\
\hat{u}
\end{pmatrix}) \right]
\]

**dimension reduction**

<table>
<thead>
<tr>
<th>model simulator data and discrep standard basis</th>
<th>( n_\eta \cdot m )</th>
<th>( n_y )</th>
<th>( p_\eta \cdot m )</th>
<th>( n \cdot (p_\delta + p_\eta) )</th>
</tr>
</thead>
</table>

Basis approach particularly efficient when \( n_\eta \) and \( n_y \) are large.
Marginal likelihood

The (marginal) likelihood \( L(\hat{v}, \hat{u}, \hat{w} | \lambda_\eta, \lambda_w, \rho_w, \lambda_y, \lambda_v, \rho_v, \theta) \) has the form

\[
\begin{pmatrix}
\hat{v} \\
\hat{u} \\
\hat{w}
\end{pmatrix}
\sim
\mathcal{N}
\begin{pmatrix}
0 \\
0 \\
0
\end{pmatrix},
\begin{pmatrix}
\Lambda_y^{-1} & 0 & 0 \\
0 & \Lambda_\eta^{-1} & 0 \\
0 & 0 & \Lambda_w^{-1}
\end{pmatrix}
\begin{pmatrix}
\Sigma_v & 0 & 0 \\
0 & \Sigma_{uw}
\end{pmatrix}
\]

where

\[
\Lambda_y = \lambda_y B^T W_y B
\]
\[
\Lambda_\eta = \lambda_\eta K^T K
\]
\[
\Sigma_v = \lambda_v^{-1} I_{p_\eta} \otimes R(x, x; \rho_v)
\]
\[
R(x, x; \rho_v) = n \times n \text{ correlation matrix from applying (2) to the conditions } x_1, \ldots, x_n \text{ corresponding the the } n \text{ experiments.}
\]
\[
\Sigma_{uw} =
\begin{pmatrix}
\lambda_w^{-1} R((x, \theta), (x, \theta); \rho_w) & 0 & 0 & \lambda_w^{-1} R((x, \theta), (x^*, \theta^*); \rho_w) & 0 & 0 \\
0 & \ddots & 0 & 0 & \ddots & 0 \\
0 & 0 & \lambda_w^{-1} R((x, \theta), (x, \theta); \rho_w) & 0 & 0 & \lambda_w^{-1} R((x, \theta), (x^*, \theta^*); \rho_w) \\
\lambda_{wp}^{-1} R((x^*, \theta^*), (x, \theta); \rho_{wp}) & 0 & 0 & \lambda_w^{-1} R((x^*, \theta^*), (x^*, \theta^*); \rho_w) & 0 & 0 \\
0 & \ddots & 0 & 0 & \ddots & 0 \\
0 & 0 & \lambda_w^{-1} R((x^*, \theta^*), (x, \theta); \rho_{wp}) & 0 & 0 & \lambda_w^{-1} R((x^*, \theta^*), (x^*, \theta^*); \rho_{wp})
\end{pmatrix}
\]

Permutation of \( \Sigma_{uw} \) is block diagonal \( \Rightarrow \) can speed up computations.

Only off diagonal blocks of \( \Sigma_{uw} \) depend on \( \theta \).
Posterior distribution

Likelihood: \( L(\hat{v}, \hat{u}, \hat{w} | \lambda_\eta, \lambda_w, \rho_w, \lambda_y, \lambda_v, \rho_v, \theta) \)

Prior: \( \pi(\lambda_\eta, \lambda_w, \rho_w, \lambda_y, \lambda_v, \rho_v, \theta) \)

\[ \Rightarrow \text{Posterior:} \]
\[ \pi(\lambda_\eta, \lambda_w, \rho_w, \lambda_y, \lambda_v, \rho_v, \theta | \hat{v}, \hat{u}, \hat{w}) \propto L(\hat{v}, \hat{u}, \hat{w} | \lambda_\eta, \lambda_w, \rho_w, \lambda_y, \lambda_v, \rho_v, \theta) \times \pi(\lambda_\eta, \lambda_w, \rho_w, \lambda_y, \lambda_v, \rho_v, \theta) \]

Posterior exploration via MCMC

Can take advantage of structure and sparsity to speed up sampling.

A useful approximation to speed up posterior evaluation:
\[ \pi(\lambda_\eta, \lambda_w, \rho_w, \lambda_y, \lambda_v, \rho_v, \theta | \hat{v}, \hat{u}, \hat{w}) \propto L(\hat{w} | \lambda_\eta, \lambda_w, \rho_w) \times \pi(\lambda_\eta, \lambda_w, \rho_w) \times L(\hat{v}, \hat{u} | \lambda_\eta, \lambda_w, \rho_w, \lambda_y, \lambda_v, \rho_v, \theta) \times \pi(\lambda_y, \lambda_v, \rho_v, \theta) \]

In this approximation, experimental data is not used to inform about parameters \( \lambda_\eta, \lambda_w, \rho_w \) which govern the simulator process \( \eta(x, \theta) \).
Posterior distribution of model parameters \((\theta_1, \theta_2)\)
Posterior mean decomposition for each experiment

Experiment 1

Experiment 2

Experiment 3
Posterior prediction for implosion in each experiment
90% prediction intervals for implosions at exposure times

Predictions from separate analyses which hold data from the experiment being predicted.
Quantifying uncertainty for simulation-based forecasts

- Simulation-based predictions accumulate uncertainty due to:
  - parameter/calibration uncertainty;
  - simulator discrepancy/inadequacy;
  - observation error in data;
  - sparseness in data

- Limits on dimensionality? Have dealt with up to 20-dimensional $\theta$.

- Statistics typically uses the wrong model (eg. a regression line) to explain data. So the framework is nothing new, in principle.

- Calibrating a model with substantial inadequacy? only slight inadequacy?

- The slowness of the simulator and high dimensionality make things difficult.

- Extrapolation is often a goal in such investigations. Generally, the closer to reality the simulator is, the more it can be trusted for extrapolation. Can this be more rigorously formalized?
Application: Ta Flyerplate Experiment

- PTW model governs features on the visar velocity profile.
- Use principal components (EOF’s) to deal with high dimensionality.
### PTW calibration parameters with input domains

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_0$</td>
<td>Initial strain hardening rate</td>
<td>0.00627, 0.00973</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>Material constant in thermal activation energy term — relates to the temperature dependence</td>
<td>0.39316, 0.96899</td>
</tr>
<tr>
<td>$-\log(\gamma)$</td>
<td>Material constant in thermal activation energy term — relates to the strain rate dependence</td>
<td>7.691, 15.355</td>
</tr>
<tr>
<td>$y_0$</td>
<td>Maximum yield stress (at 0 K)</td>
<td>0.00689, 0.01147</td>
</tr>
<tr>
<td>$y_\infty$</td>
<td>Minimum yield stress (∼ melting)</td>
<td>0.00112, 0.00182</td>
</tr>
<tr>
<td>$s_0$</td>
<td>Maximum saturation stress (at 0 K)</td>
<td>0.00455, 0.03061</td>
</tr>
<tr>
<td>$s_\infty$</td>
<td>Minimum saturation stress (∼ melting)</td>
<td>0.00281, 0.00435</td>
</tr>
</tbody>
</table>
PC representation of simulation output

fit using $p_\eta = 5 - 75\%$ of variation
Marginal Posteriors for spatial correlation parameters $\rho_{wij}$
PC-based sensitivities
Represent discrepancy $\delta(x)$ using basis functions and weights. Here $d_j(s)$ is normal density centered at spatial location $\omega_j$:

$$d_j(s) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(s - \omega_j)^2\right\}$$

Set $\delta(s) = \sum_{j=1}^{p_\delta} d_j(s) v_j$ where $v \sim N(0, \lambda_v^{-1} I_{p_\delta})$. Can represent $\delta = (\delta(s_1), \ldots, \delta(s_n))^T$ as $\delta = Dv$ where

$$D_{ij} = d_j(s_i)$$

$p_\delta = 10$ basis functions over $t$. 
posterior distribution for PTW parameters
posterior predictive distribution for trace

- η
- Y
- δ
posterior predictive distribution for trace (original scale)