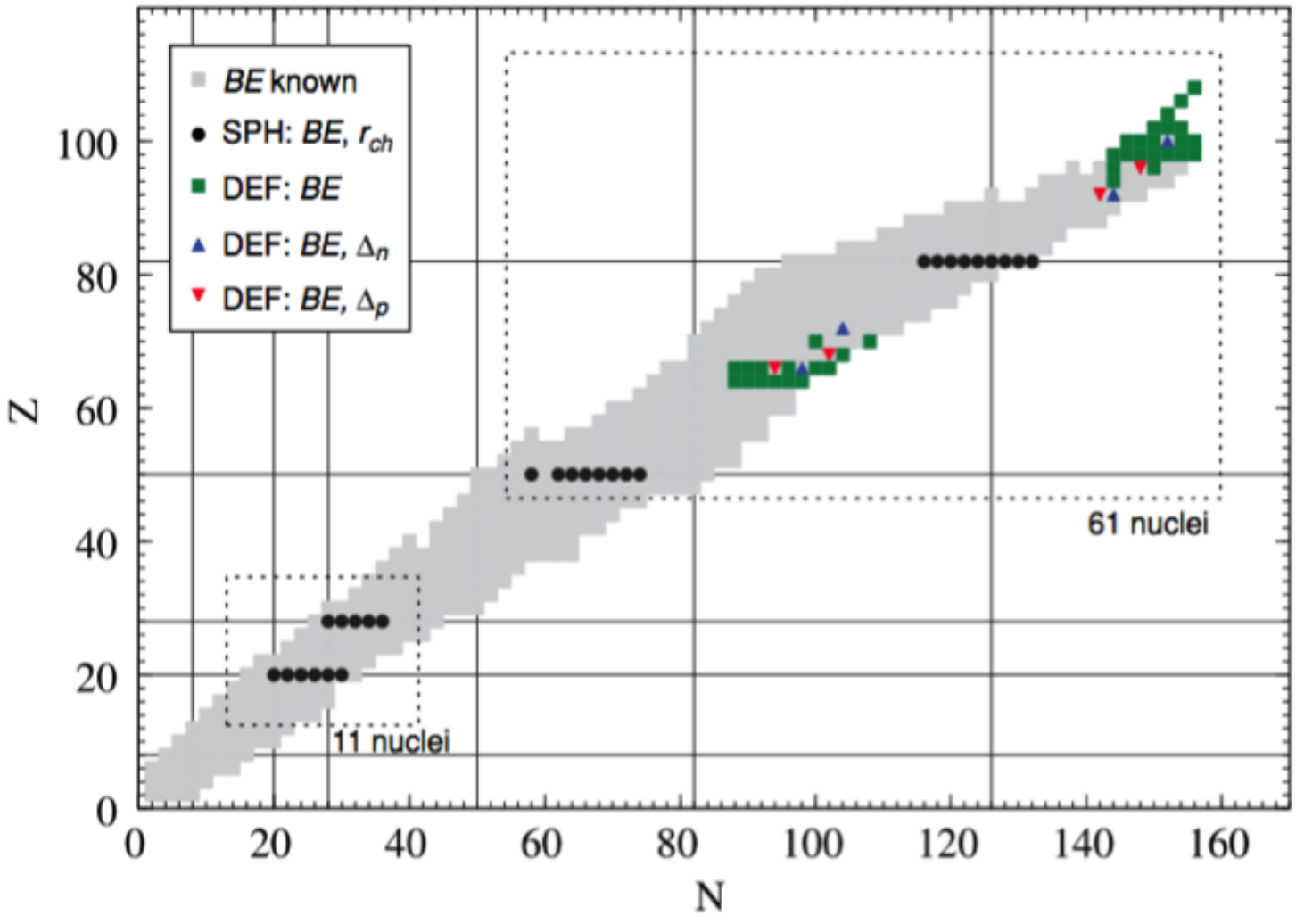
Bayesian uncertainty quantification for nuclear density functional theory

Earl Lawrence

LANL

Los Alamos NATIONAL LABORATORY EST.1943

Bayesian Inference for Nuclear Density Functional Theory



Los Alamos
NATIONAL LABORATORY
EST. 1943

Bayesian Inference for Nuclear Density Functional Theory

$$\{BindingEnergies, Radii, Odd\ Even\ Scattering, FIEs, \cdots\} =$$
$$\eta(\rho, E/A, K, a_{sym}, L, 1/M_s^*, C_0^{\rho\Delta\rho}, C_1^{\rho\Delta\rho},$$
$$V_0^n, V_0^p, C_0^{\rho\nabla J}, C_1^{\rho\nabla J}, C_0^{JJ}, C_1^{JJ}, 1/M_v^*)$$

How do we learn inputs for UNEDF2 ("a damn good functional") from the data while accounting for uncertainty, when UNEDF2 is sort of slow.

Bayesian Thinking

Uncertainty is quantified using probability.

Everything is basically a random variable.

Building a Bayesian Model

1. Write down the conditional distribution of things you observe (data) given things you want to know (models, parameters).
2. Multiply this by the marginal of the unknowns to get the joint probability of everything.

$$p(\theta|y) \propto \pi(\theta) f(y|\theta)$$

Simple Gaussian Example

- $y = \theta + \epsilon,\ \epsilon \sim N(0, \sigma^2)$
- $\theta \sim N(\mu, \delta^2)$
- Observe $y_1, \cdots, y_n$

$$
\begin{aligned}
p(\theta|y) &\propto \exp\left\{-\frac{1}{2\delta^2}(\theta-\mu)^2\right\} \exp\left\{-\frac{1}{2\sigma^2}\sum_i (y_i-\theta)^2\right\} \\
&\vdots \\
\theta|y &\sim N\left(\nu, \gamma^2\right) \\
\nu &= \frac{\frac{n}{\sigma^2}\bar{Y} + \frac{1}{\delta^2}\mu}{\frac{n}{\sigma^2}+\frac{1}{\delta^2}} \\
\gamma^2 &= \left(\frac{n}{\sigma^2}+\frac{1}{\delta^2}\right)^{-1}
\end{aligned}
$$

Los Alamos
NATIONAL LABORATORY
EST. 1943

Markov Chain Monte Carlo

Method of drawing sequence of correlated samples from a distribution by constructing a Markov chain whose stationary distribution is the one of interest. This is useful when the distribution is not otherwise tractable and only requires knowing the distribution up to a constant. The samples can be used for inference (e.g. means, variances quantiles).

Markov Chain Monte Carlo: Metropolis-Hastings

Assume x follows some distribution with density p and that we have x_k with $p(x_k) > 0$.

1. Draw a candidate x' from $q(x'|x_k)$.
2. Compute $\alpha = \frac{p(x')q(x_k|x')}{p(x_k)q(x'|x_k)}$
3. Draw $u \sim \mathit{Unif}(0, 1)$.
4. If $u \leq \alpha$, set $x_{k+1} = x'$, else set $x_{k+1} = x_k$.

Often, q is a random walk so $q(x'|x_k) = q(x_k|x')$ and α simplifies (original Metropolis). Sometimes, q is p (Gibbs sampling). Good results often require some tuning of q (e.g. the step size of the random walk).

Simple Gaussian Example with MCMC

```
spec <- read.table('smoothspectra.txt')
logk <- spec[,1]
spec <- spec[,seq(2,223,6)]
spec <- as.numeric(spec[42,])
M000 <- read.table('smoothM000.txt')
M000 <- M000[42,2]

# Make some data
obs.mean <- M000
obs.var <- .0009
y <- rnorm(n=15, mean=obs.mean, sd=sqrt(obs.var))

# Prior
mu.prior.mean <- mean(spec)
mu.prior.var <- var(spec)
```

Simple Gaussian Example with MCMC

```
# Posterior for mu
log.post <- function(mu, y) {
  lp <- -.5*sum((y-mu)^2)/obs.var
  lp <- lp - .5*((mu-mu.prior.mean)^2)/mu.prior.var
}

# MCMC
M <- 1000
ss <- 4
mu <- numeric(M)
mu[1] <- rnorm(n=1, mean=mu.prior.mean, sd=sqrt(mu.prior.var))
for(i in 2:M) {
  mu.cand <- mu[i-1] + rnorm(n=1,sd=ss)
  acc.ratio <- log.post(mu.cand, y) - log.post(mu[i-1], y)
  if(log(runif(n=1)) <= acc.ratio) {
    mu[i] <- mu.cand
  } else {
    mu[i] <- mu[i-1]
  }
}
```

Los Alamos
NATIONAL LABORATORY
EST.1943

Simple Gaussian Example with MCMC

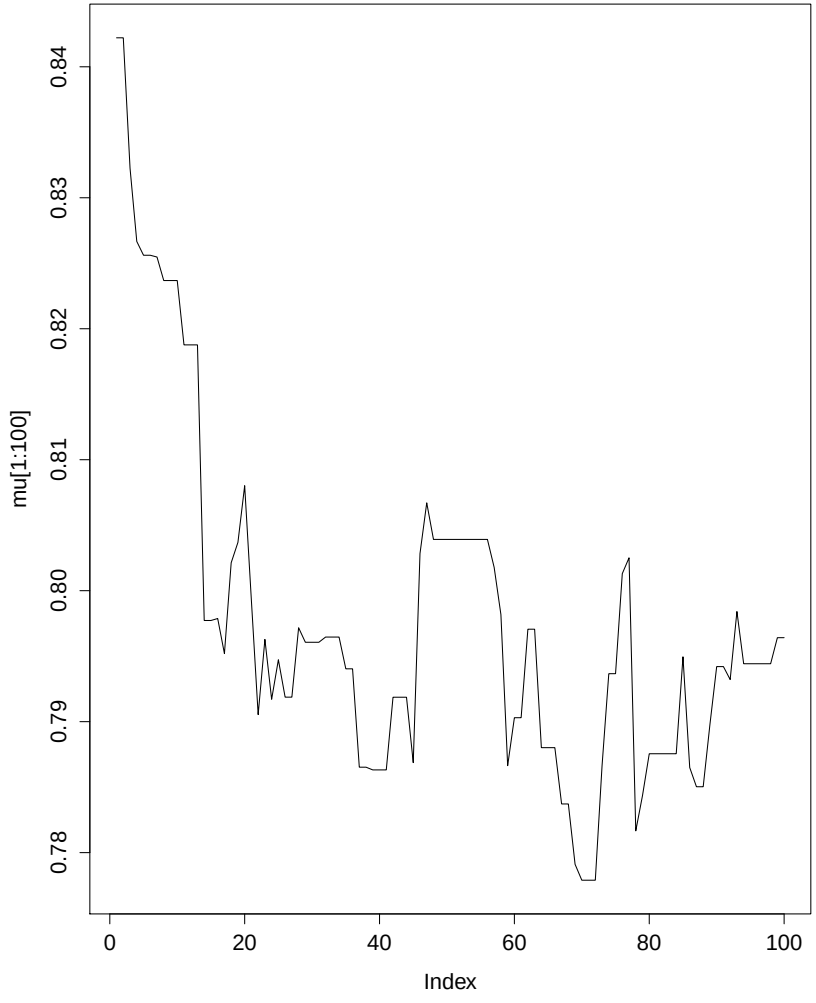


Figure: First 100 draws μ.

Los Alamos
NATIONAL LABORATORY
EST.1943

Simple Gaussian Example with MCMC

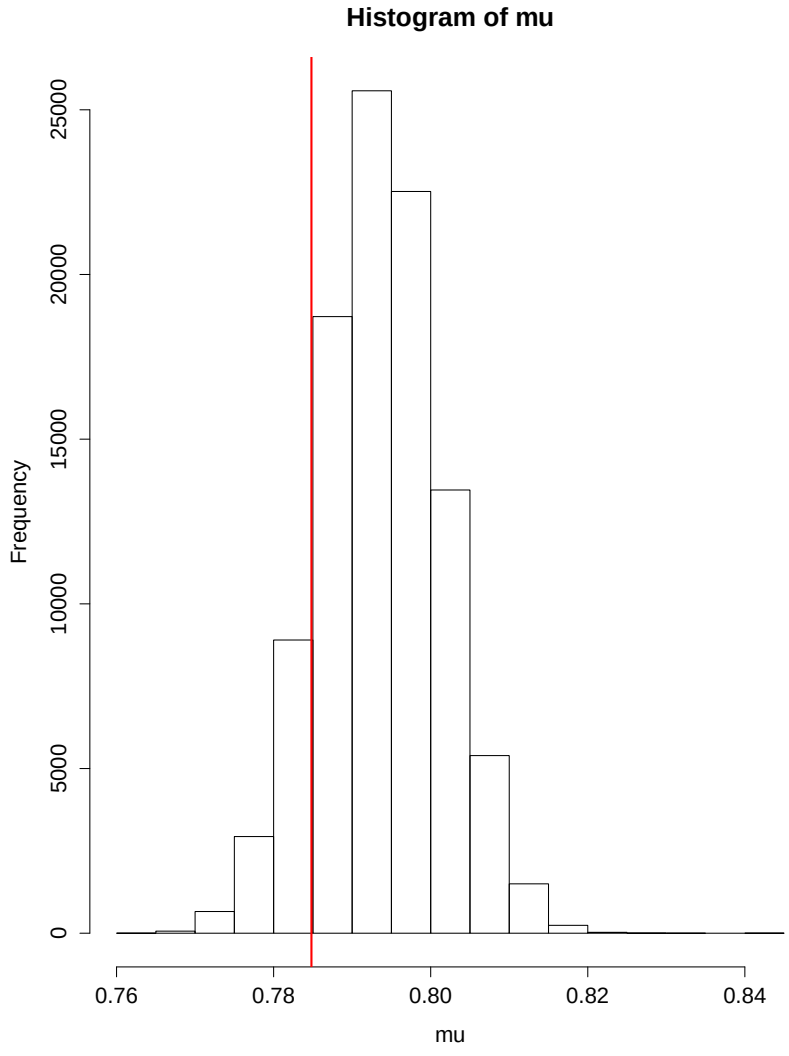


Figure: Histogram of μ with "true" value.

Los Alamos
NATIONAL LABORATORY
EST.1943

Gaussian Example with Unknown Mean and Variance

- $y|\theta \sim N(\theta, \sigma^2)$
- $\theta \sim N(\mu, \delta^2)$
- $\sigma^2 \sim Unif(0, U)$
- Observe $y_1, \cdots, y_n$

Sample from $p(\mu, \sigma^2|y)$ by sampling sequentially from the full conditional posteriors: $p(\mu|\sigma, y)$ and $p(\sigma^2|\mu, y)$ which are simply proportional to their joint density.

Los Alamos
NATIONAL LABORATORY
EST. 1943

Gaussian Example with Unknown Mean and Variance

```
spec <- read.table('smoothspectra.txt')
logk <- spec[,1]
spec <- spec[,seq(2,223,6)]
spec <- as.numeric(spec[42,])
M000 <- read.table('smoothM000.txt')
M000 <- M000[42,2]

# Make some data
obs.mean <- M000
obs.var <- .0009
y <- rnorm(n=15, mean=obs.mean, sd=sqrt(obs.var))

# Prior
# Gaussian for mu
mu.prior.mean <- mean(spec)
mu.prior.var <- var(spec)
# Uniform for variance
sigma2.prior.upper <- 0.01
```

Los Alamos
NATIONAL LABORATORY
EST.1943

Gaussian Example with Unknown Mean and Variance

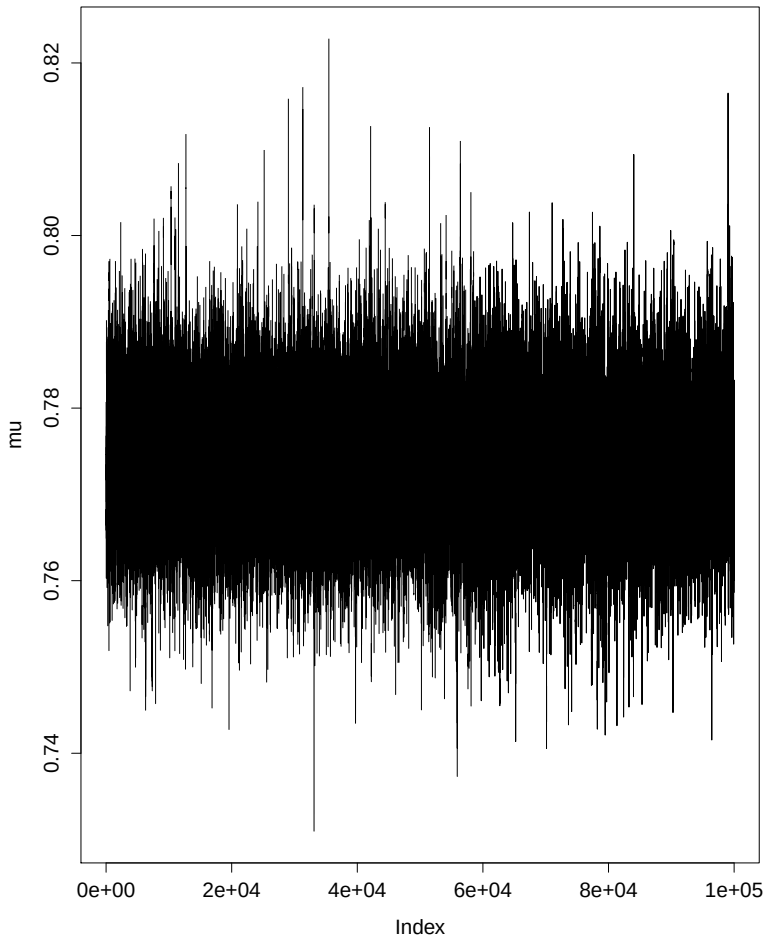
```
# Conditional posterior for mu
log.post.mu <- function(mu, sigma2, y) {
  lp <- -.5*sum((y-mu)^2)/sigma2
  lp <- lp - .5*((mu-mu.prior.mean)^2)/mu.prior.var
}

# Conditional posterior for sigma2
log.post.sigma2 <- function(sigma2, mu, y) {
  if((sigma2 <= 0) || (sigma2 > sigma2.prior.upper)) {
    lp <- -Inf
  } else {
    lp <- -(length(y)/2)*log(sigma2) -.5*sum((y-mu)^2)/sigma2
  }
  return(lp)
}
```

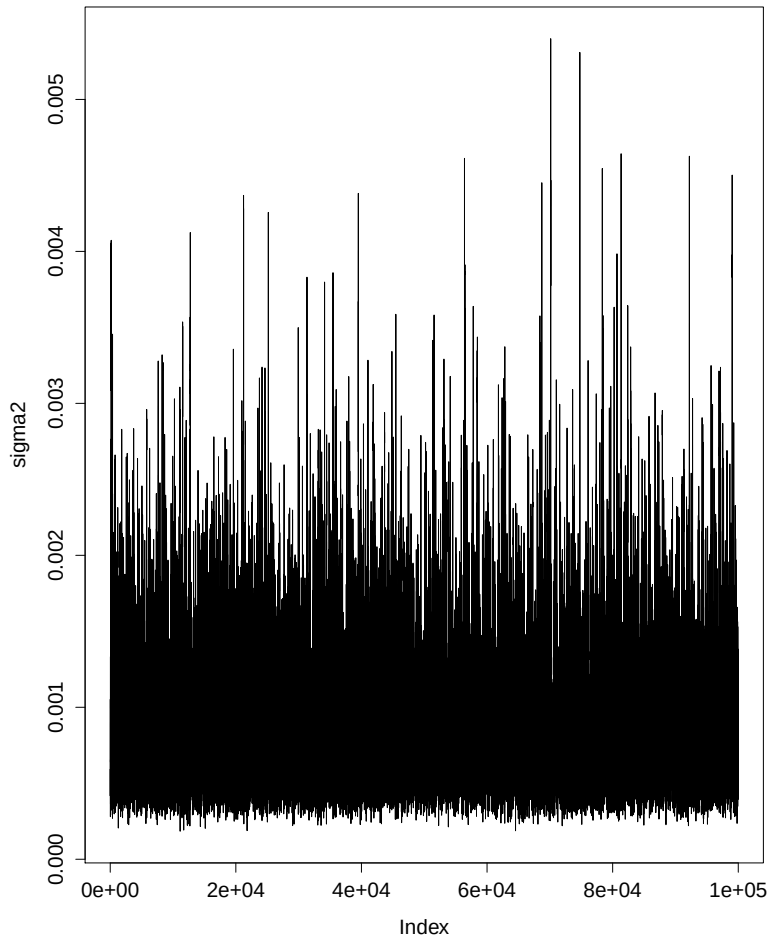
Gaussian Example with Unknown Mean and Variance

```
# MCMC
M <- 1000
ss.mu <- .01
ss.sigma2 <- .001
mu <- numeric(M)
mu[1] <- mean(y)
sigma2 <- numeric(M)
sigma2[1] <- var(y)
for(i in 2:M) {

  # Draw sigma2
  sigma2.cand <- sigma2[i-1] + rnorm(n=1,sd=ss.sigma2)
  acc.ratio <- log.post.sigma2(sigma2.cand, mu[i-1], y)
  acc.ratio <- acc.ratio - log.post.sigma2(sigma2[i-1], mu[i-1], y)
  if(log(runif(n=1)) <= acc.ratio) {
    sigma2[i] <- sigma2.cand
  } else {
    sigma2[i] <- sigma2[i-1]
  }

  # Draw mu
  mu.cand <- mu[i-1] + rnorm(n=1, sd=ss.mu)
  acc.ratio <- log.post.mu(mu.cand, sigma2[i], y)
  acc.ratio <- acc.ratio - log.post.mu(mu[i-1], sigma2[i], y)
  if(log(runif(n=1)) <= acc.ratio) {
    mu[i] <- mu.cand
  } else {
    mu[i] <- mu[i-1]
  }
}
```

Los Alamos
NATIONAL LABORATORY
EST.1943

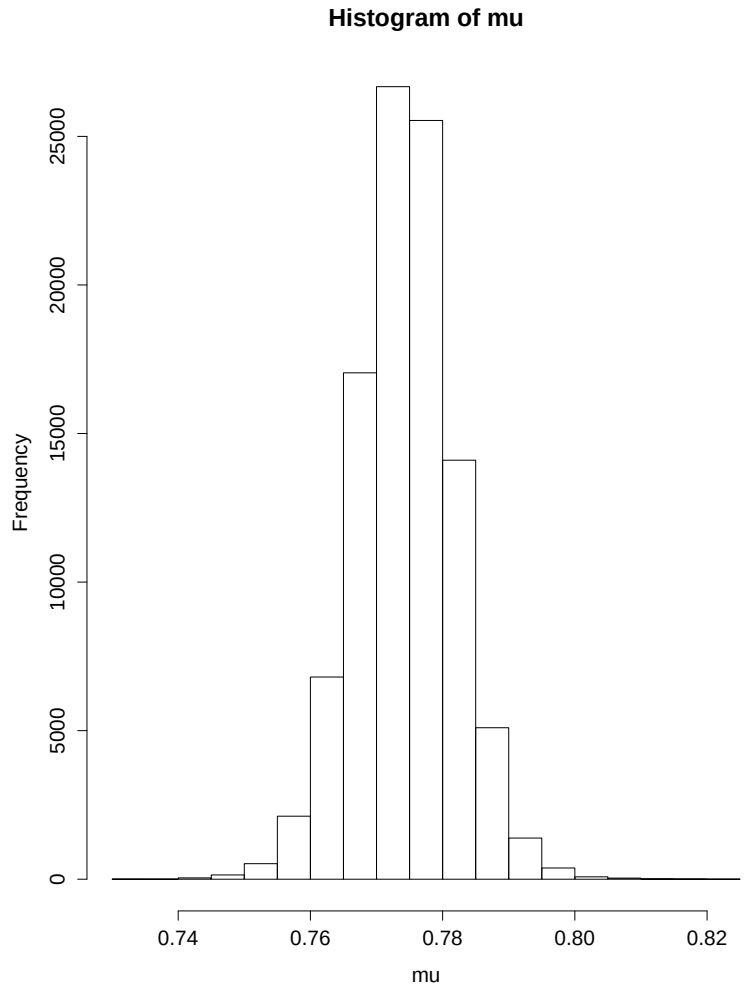
Gaussian Example with Unknown Mean and Variance

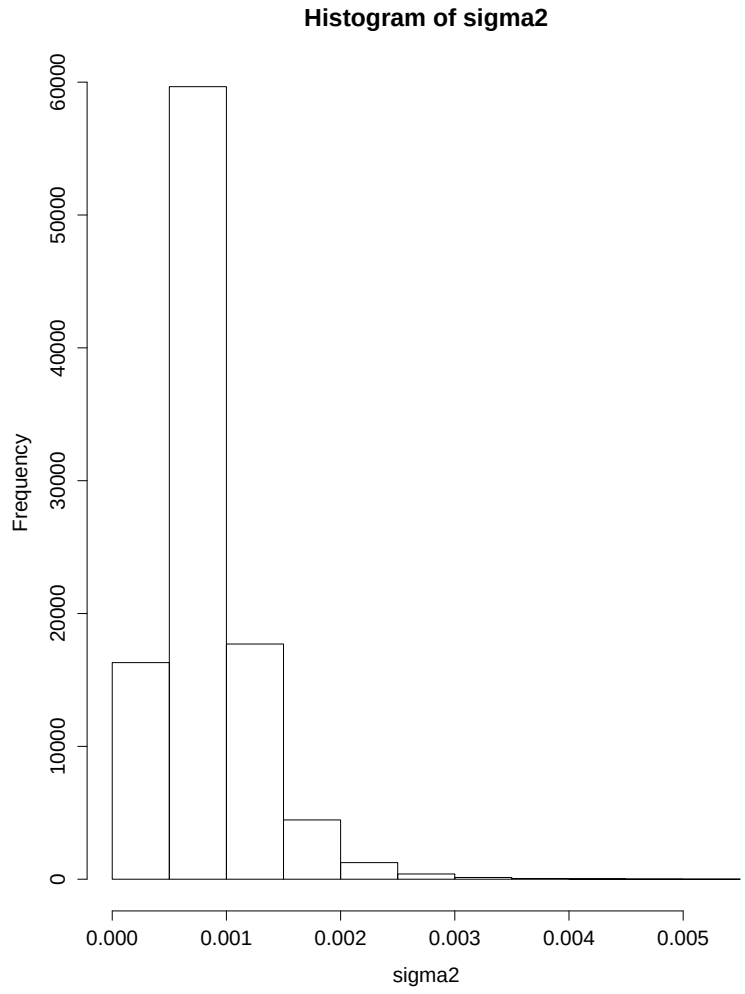




Los Alamos
NATIONAL LABORATORY
EST.1943

Gaussian Example with Unknown Mean and Variance





Los Alamos
NATIONAL LABORATORY
EST.1943

Black Box Functions

- $y|\theta \sim N(\eta(\theta), \sigma^2)$
- $\theta \sim \pi(\theta)$

MCMC only requires that you can evaluate $\eta(\cdot)$.

Black Box Function

```
source('REmu.R')

# Mean is given by the Cosmic Emu
# Read in some useful stuff for making up a problem
M000 <- read.table('smoothM000.txt')
M000 <- M000[42,2]

# Make some data that looks like the simulation of the best fit co
obs.mean <- M000
obs.var <- .0009
y <- rnorm(n=15, mean=obs.mean, sd=sqrt(obs.var))

# Fix the known parameters
wm = .1296
wb = .0224
ns = .97
s8 <- .8
z <- 1

w.upper <- -0.7
w.lower <- -1.3
```

Black Box Function

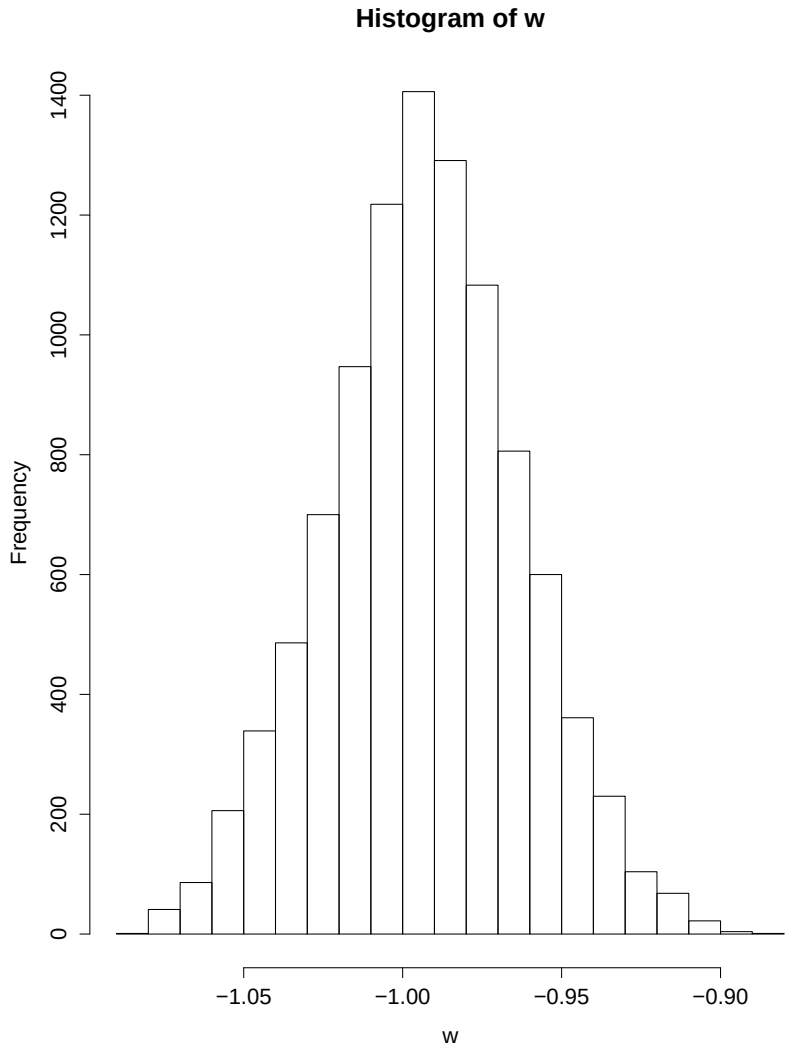
```
# Posterior for w
log.post <- function(wm, wb, ns, s8, w, z, y) {
  if((w <= w.lower) || (w > w.upper)) {
    lp <- -Inf
  } else {
    mu <- cosmic.emu(wm, wb, ns, s8, w, z)
    mu <- mu[42,2]
    lp <- -.5*sum((y-mu)^2)/obs.var
  }
  return(lp)
}
```

Los Alamos
NATIONAL LABORATORY
EST.1943

Black Box Function

```
# MCMC
M <- 10000
ss <- .01
w <- numeric(M)
w[1] <- -1
for(i in 2:M) {
   print(i)
   w.cand <- w[i-1] + rnorm(n=1,sd=ss)
   acc.ratio <- log.post(wm, wb, ns, s8, w.cand, z, y)
   acc.ratio <- acc.ratio - log.post(wm, wb, ns, s8, w[i-1], z, y)
   if(log(runif(n=1)) <= acc.ratio) {
     w[i] <- w.cand
   } else {
     w[i] <- w[i-1]
   }
}
```

Black Box Function



Los Alamos
NATIONAL LABORATORY
EST.1943

What if the function is slow?

- Some cosmology simulations can take weeks.
- Nuclear DFT model takes minutes.
- MCMC could require a million evaluations.

We need to expand our modeling approach to include estimation of the simulation. The basic idea is to run the simulation over a training set and build a statistical model that predicts the results at untried settings.

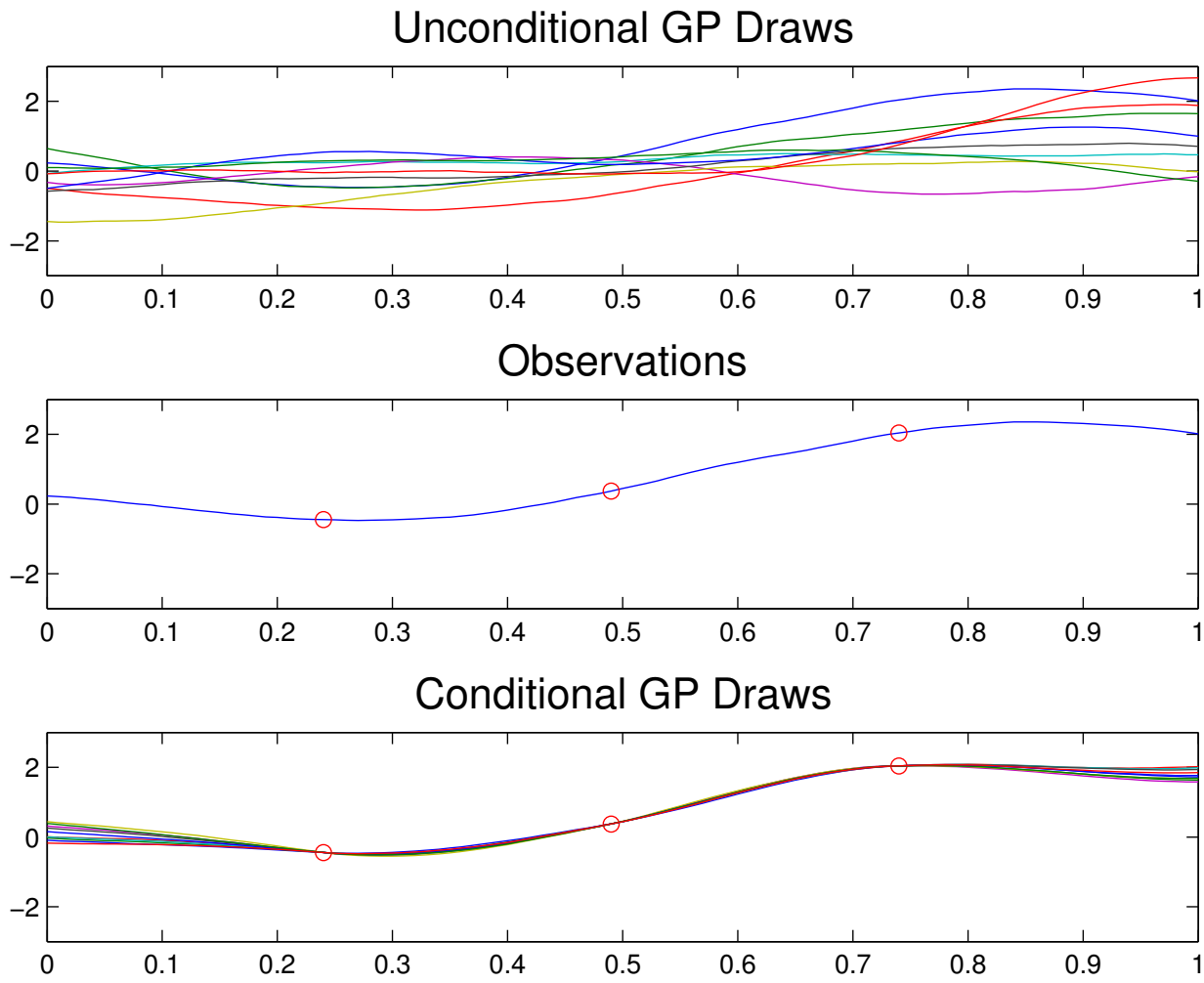
Gaussian Processes: A Distribution for Functions

Assume that univariate y is a function of d-D x. Let $\vec{y}$ be a collection of these points associated with the matrix X (ith row goes with y_i).

$$\begin{aligned} \vec{y} &\sim N\left(\vec{0}, \sigma^2 R(X)\right) \\ R_{i,j} &= \exp\left\{-\sum_{k=1}^{p} \beta_k (X_{i,k} - X_{j,k})^2\right\} \end{aligned}$$

This has the squared exponential covariance which produces continuous and very smooth draws. Given a training set $(\vec{y}, X)$ and priors, the GP parameters σ^2 and $\vec{\beta}$ can be estimated with MCMC.

Gaussian Process Cartoon



Los Alamos
NATIONAL LABORATORY
EST.1943

Conditional GP

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \sim N\left\{\vec{0}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right\}$$

$$y_1|y_2 \sim N\left\{\Sigma_{12}\Sigma_{22}^{-1}y_2, \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right\}$$

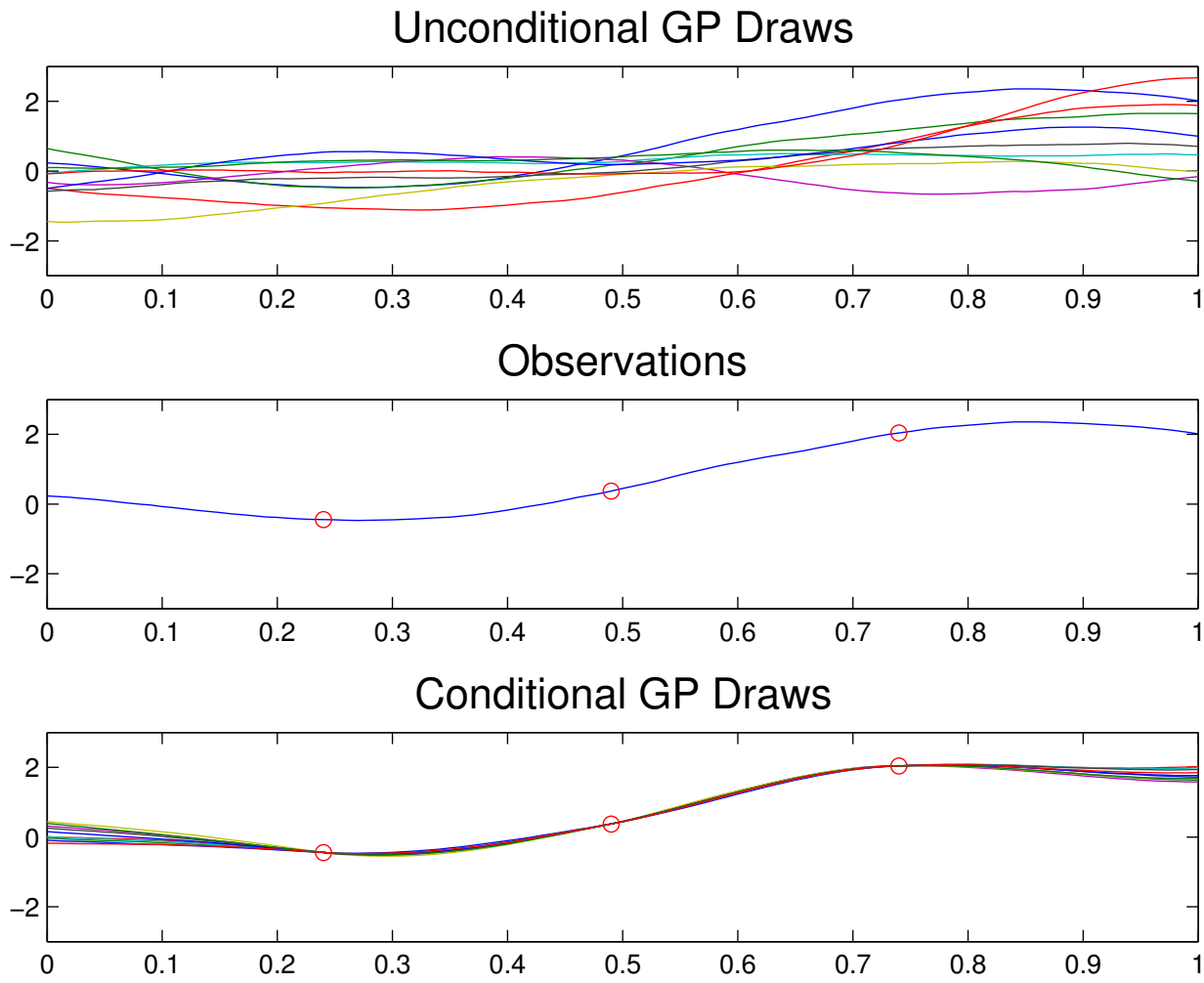
Assume that Σ is the aforementioned function of X, that y_2 are points that we've observed at X_2, and that y_1 are points that we want to predict at X_1. Everything on the right is known and gives us the distribution for the new points.

Conditional GP

$$y_1|y_2 \sim N\left\{\Sigma_{12}\Sigma_{22}^{-1}y_2, \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right\}$$

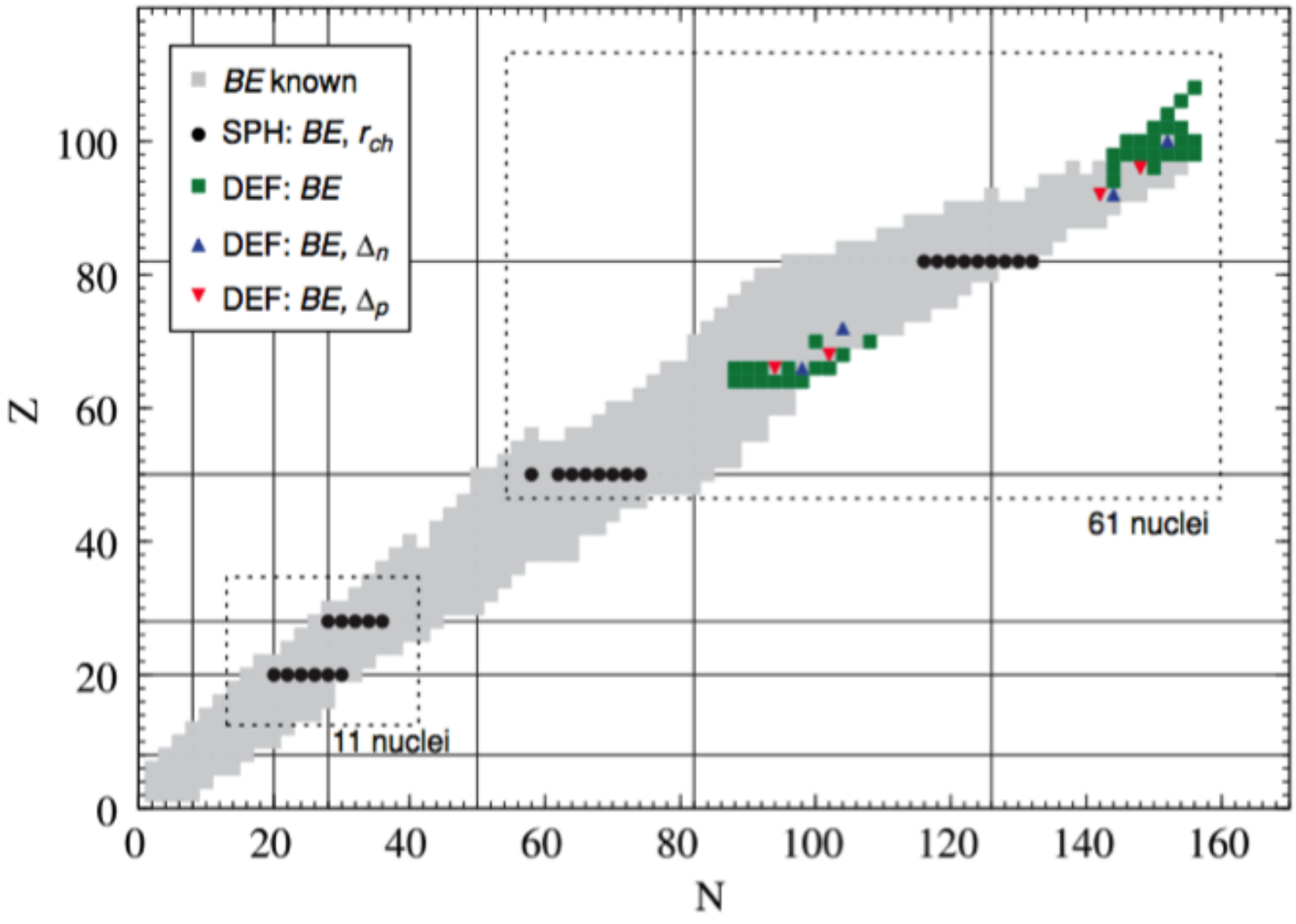
- The mean for new points is a weighted average of the observed points.
- The variance goes to zero as a new point approaches an observed point.

Gaussian Process Cartoon



Los Alamos
NATIONAL LABORATORY
EST.1943

Bayesian Inference for Nuclear Density Functional Theory



Los Alamos
NATIONAL LABORATORY
EST. 1943

Bayesian Inference for Nuclear Density Functional Theory

$$\{BindingEnergies, Radii, Odd\ \ Even\ Scattering, FIEs, \cdots\} =$$
$$\eta(\rho, E/A, K, a_{sym}, L, 1/M_s^*, C_0^{\rho\Delta\rho}, C_1^{\rho\Delta\rho},$$
$$V_0^n, V_0^p, C_0^{\rho\nabla J}, C_1^{\rho\nabla J}, C_0^{JJ}, C_1^{JJ}, 1/M_v^*)$$

Los Alamos
NATIONAL LABORATORY
EST. 1943

Building the Statistical Model

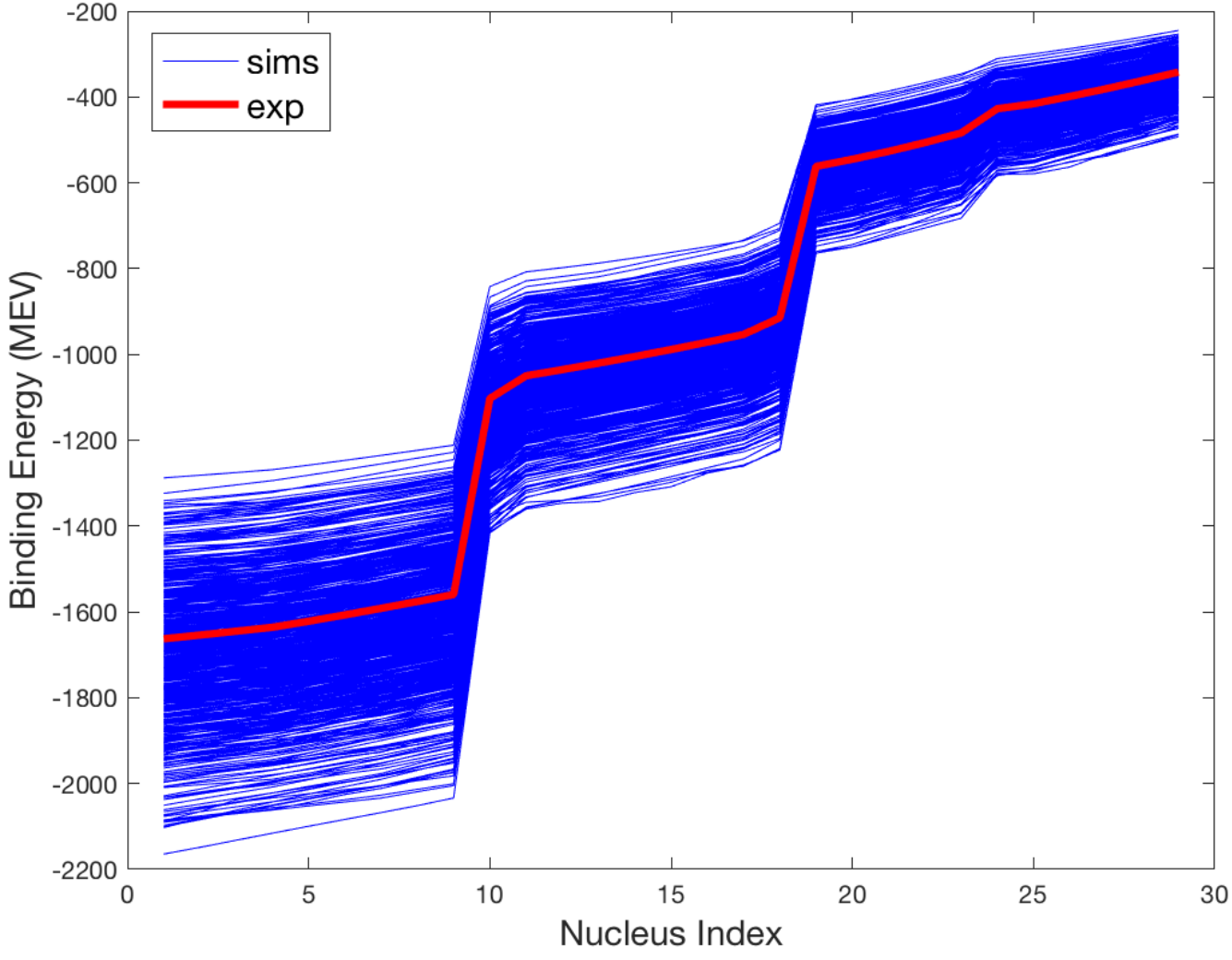
The data is a noisy version of the model at the "correct" input.

$$\begin{aligned} y &= \eta(\theta) + \epsilon \\ \epsilon &\sim \mathsf{N}\left(\vec{0}, \Sigma_y\right) \\ \pi(\theta) &= 1, \theta \in \mathcal{C} \\ L(y|\eta(\theta)) &\propto \exp\left\{\frac{1}{2}(y-\eta(\theta))'\Sigma_y^{-1}(y-\eta(\theta))\right\} \end{aligned}$$

Treat $\eta(\cdot)$ as an unknown function, with observations $\eta^* = (\eta(t_1), \ldots, \eta(t_m))'$.

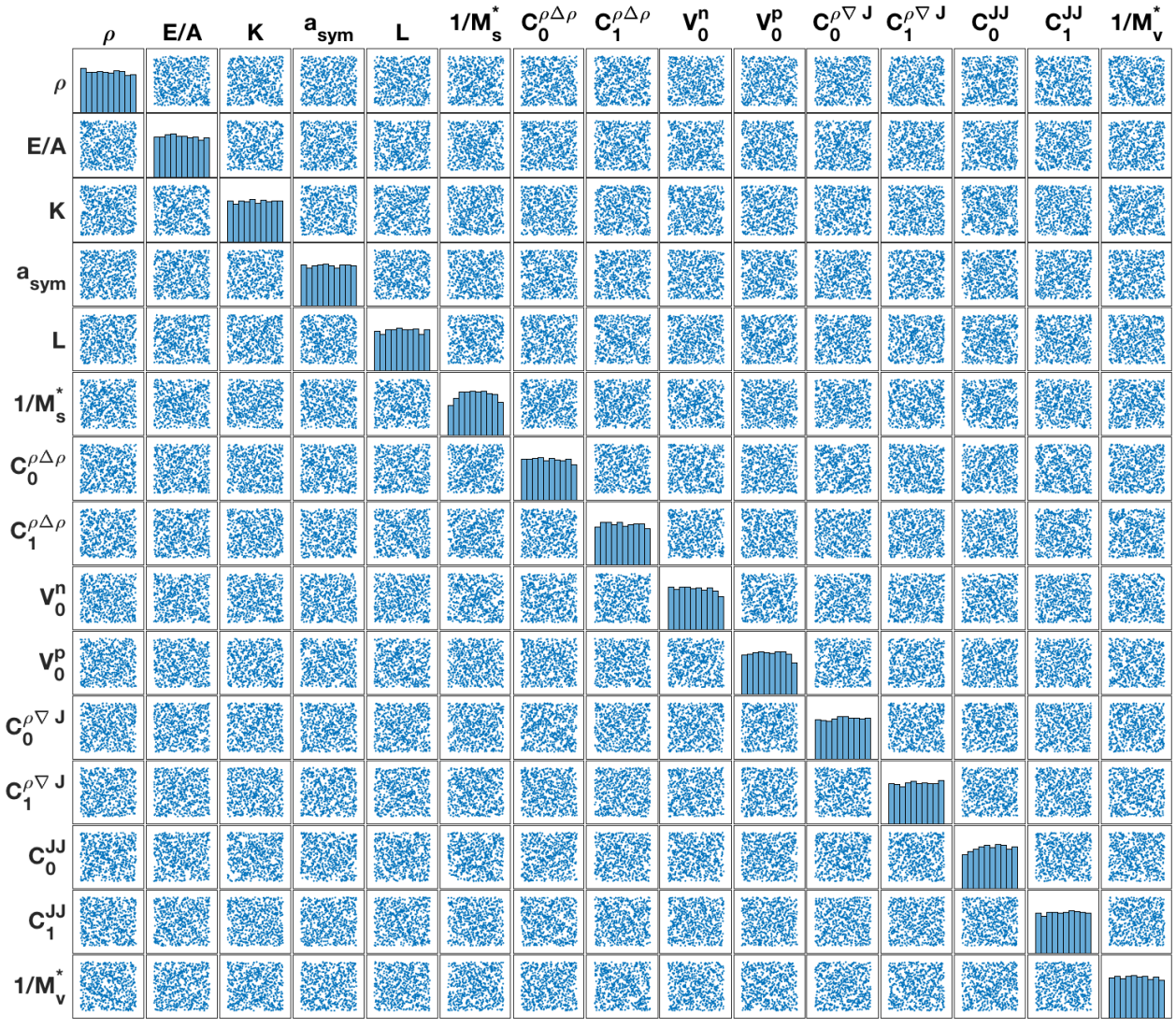
$$\pi(\theta, \eta(\cdot)|y, \eta^*) \propto L(y|\eta(\theta)) \cdot L(\eta^*|\eta(\cdot)) \cdot \pi(\eta(\cdot)) \cdot \pi(\theta)$$

Two Types of Data: Sims and Experimental



Los Alamos
NATIONAL LABORATORY
EST.1943

Choosing the Simulations

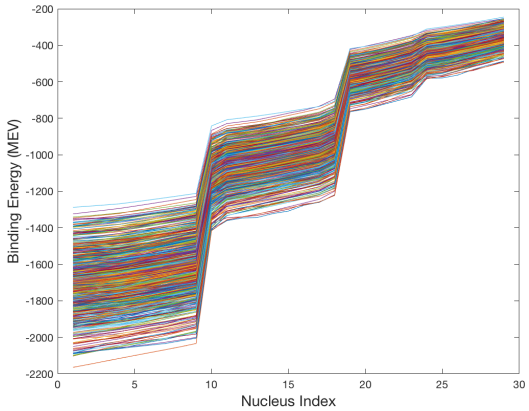


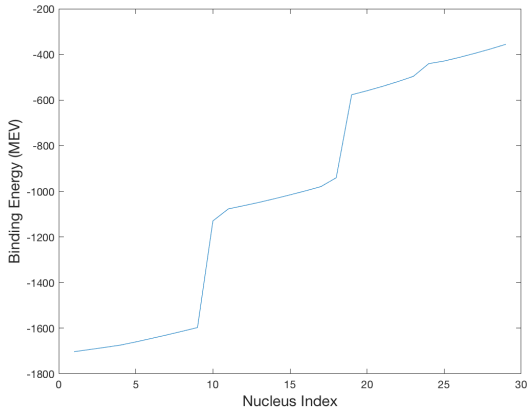
Los Alamos
NATIONAL LABORATORY
EST. 1943

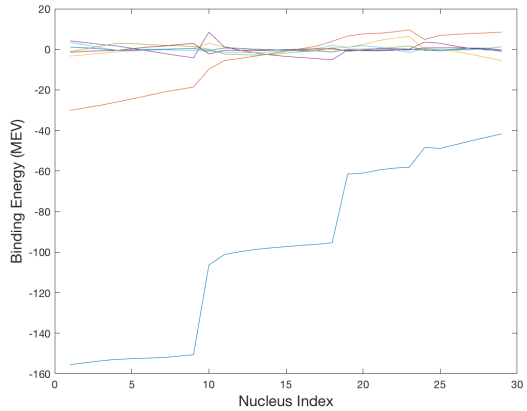
η: Decomposing the Multivariate Output

Compute a principal component basis from the simulations.

$$\eta(t) = \sum_{i=1}^{q} \phi_i w_i(t) + \epsilon$$







η: GPs for Basis Weights

$$
\begin{aligned}
w_i(t) &\sim \mathrm{N}(0, \lambda_{wi}^{-1} R(t; \rho_i)) \\
\mathrm{Corr}(w_i(t), w_i(t')) &= \prod_{k=1}^{p} \rho_{ik}^{4(t_k - t'_k)^2}
\end{aligned}
$$

We call the resulting statistical approximation to the physics model an emulator.

Los Alamos
NATIONAL LABORATORY
EST. 1943

η Likelihood

$$\begin{pmatrix} w_1 \\ \vdots \\ w_q \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} \lambda_{w1}^{-1} R(t;\rho_1) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_{wq}^{-1} R(t;\rho_q) \end{pmatrix} \right)$$

Find η again

$$\eta | w, \lambda_\eta \sim N\left(\Phi w, \frac{1}{\lambda_\eta} I \right)$$

Los Alamos
NATIONAL LABORATORY
EST. 1943

η Priors

$$
\begin{aligned}
\pi(\lambda_{wi}) &\propto \lambda_{wi}^{a_w - 1} e^{-b_w \lambda_{wi}}, \quad i = 1, \ldots, q, \\
\pi(\rho_{ik}) &\propto \rho_{ik}^{a_\rho - 1} (1 - \rho_{ik})^{b_\rho - 1}, \quad i = 1, \ldots, q, \; k = 1, \ldots, p
\end{aligned}
$$

Los Alamos
NATIONAL LABORATORY
EST. 1943

η Posterior

$$\pi(\lambda_\eta, \lambda_w, \rho|\eta) \propto$$
$$\left|(\lambda_\eta \Phi'\Phi)^{-1} + \Sigma_w\right|^{-\frac{1}{2}} \exp\{-\tfrac{1}{2}\hat{w}'([\lambda_\eta \Phi'\Phi]^{-1} + \Sigma_w)^{-1}\hat{w}\} \times$$
$$\lambda_\eta^{a_\eta^*-1} e^{-b_\eta^*\lambda_\eta} \times \prod_{i=1}^{q} \lambda_{wi}^{a_w-1} e^{-b_w\lambda_{wi}} \times \prod_{i=1}^{q}\prod_{j=1}^{p} \rho_{ij}^{a_\rho-1}(1-\rho_{ij})^{b_\rho-1},$$

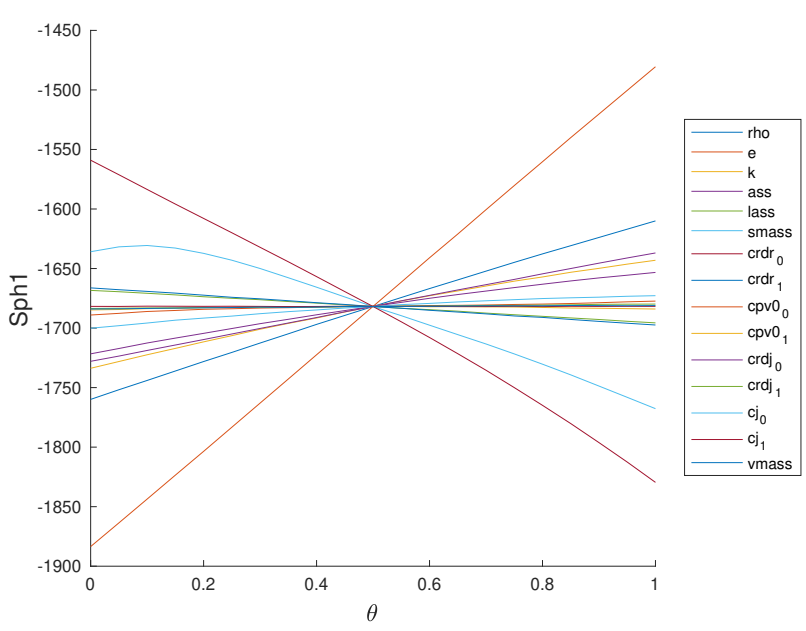
where

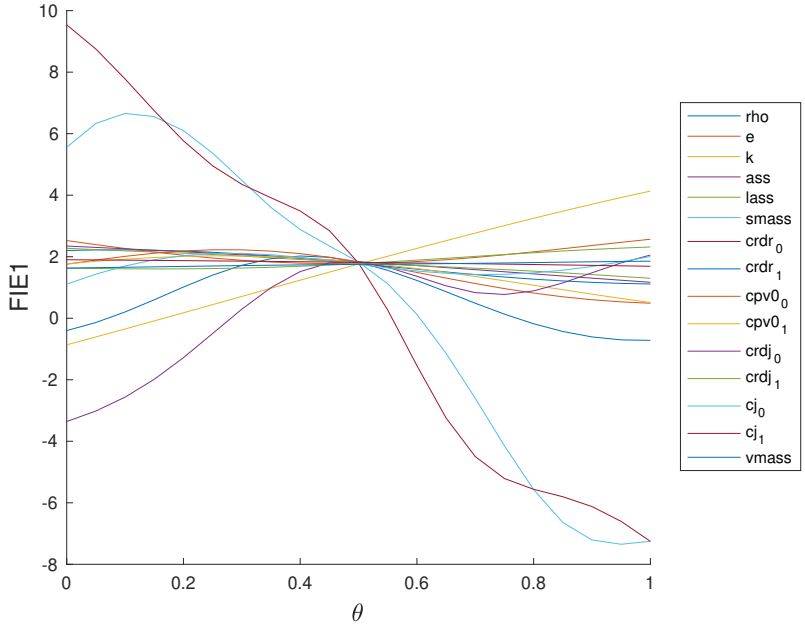
$$\begin{aligned} a_\eta^* &= a_\eta + \frac{m(n_\eta - q)}{2}, \\ b_\eta^* &= b_\eta + \tfrac{1}{2}\eta'(I - \Phi(\Phi'\Phi)^{-1}\Phi')\eta, \text{ and} \\ \hat{w} &= (\Phi'\Phi)^{-1}\Phi'\eta. \end{aligned}$$

Los Alamos NATIONAL LABORATORY EST.1943

The emulator is pretty useful on its own.

Much faster than the original simulation.
Useful for sensitivity or online tasks.





Los Alamos
NATIONAL LABORATORY
EST.1943

Including the Experimental Data

$$
\begin{aligned}
y &= \eta(\theta) + \epsilon, \\
y &= \Phi_y w(\theta) + \epsilon \\
y|w(\theta), \lambda_y &\sim N(\Phi_y w(\theta), (\lambda_y W_y)^{-1}),\ \lambda_y \sim Ga(a_y, b_y)
\end{aligned}
$$

Los Alamos
NATIONAL LABORATORY
EST. 1943

Posterior with Data

$$
\begin{aligned}
\hat{w}_y &= (\Phi_y' W_y \Phi_y)^{-1} \Phi_y' W_y y, \\
a_y^* &= a_y + \tfrac{1}{2}(n-q), \\
b_y^* &= b_y + \tfrac{1}{2}(y - \Phi_y \hat{w}_y)' W_y (y - \Phi_y \hat{w}_y), \\
\Lambda_y &= \lambda_y \Phi_y' W_y \Phi_y, \\
\Lambda_\eta &= \lambda_\eta \Phi' \Phi, \\
I_q &= q \times q \text{ identity matrix}, \\
\Sigma_{w_y w} &= \begin{pmatrix} \lambda_{w1}^{-1} R(\theta, \theta^*; \rho_1) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_{wq}^{-1} R(\theta, \theta^*; \rho_q) \end{pmatrix}, \\
\hat{z} &= \begin{pmatrix} \hat{w}_y \\ \hat{w} \end{pmatrix}, \\
\Sigma_{\hat{z}} &= \begin{pmatrix} \Lambda_y^{-1} & 0 \\ 0 & \Lambda_\eta^{-1} \end{pmatrix} + \begin{pmatrix} I_q & \Sigma_{w_y w} \\ \Sigma_{w_y w}' & \Sigma_w \end{pmatrix}.
\end{aligned}
$$

Los Alamos
NATIONAL LABORATORY
EST. 1943

Posterior with Data

$$\pi(\lambda_\eta, \lambda_w, \rho, \lambda_y, \theta|\hat{z}) \propto$$

$$|\Sigma_{\hat{z}}|^{-\frac{1}{2}} \exp\left\{-\tfrac{1}{2}\hat{z}'\Sigma_{\hat{z}}^{-1}\hat{z}\right\} \times \lambda_\eta^{a_\eta^*-1} e^{-b_\eta^*\lambda_\eta} \times \prod_{i=1}^{q} \lambda_{wi}^{a_w-1} e^{-b_w\lambda_{wi}} \times$$

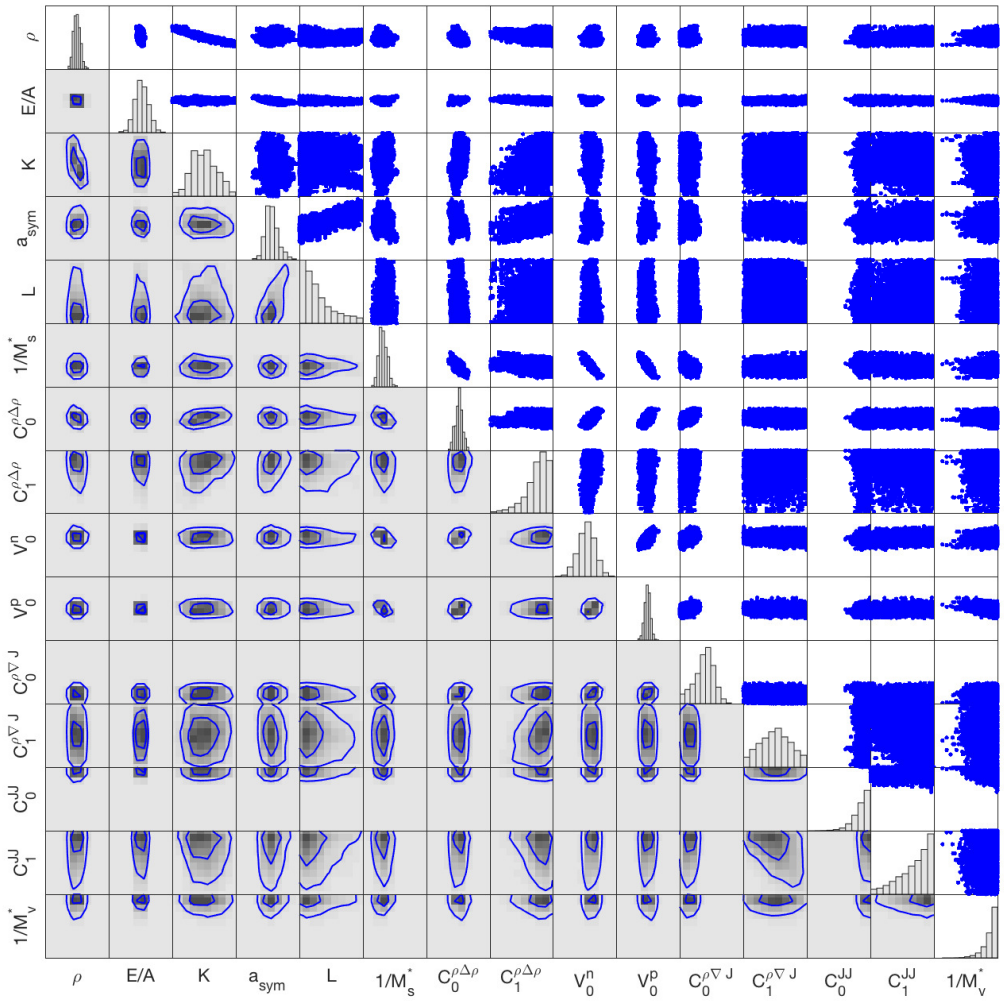
$$\prod_{i=1}^{q}\prod_{k=1}^{p} \rho_{ik}^{a_\rho-1}(1-\rho_{ik})^{b_\rho-1} \times \lambda_y^{a_y^*-1} e^{-b_y^*\lambda_y} \times I[\theta \in C],$$

Los Alamos
NATIONAL LABORATORY
EST. 1943

Estimation is Still "Simple"

All of these parameters are estimated with one-at-a-time Metropolis-Hastings MCMC.

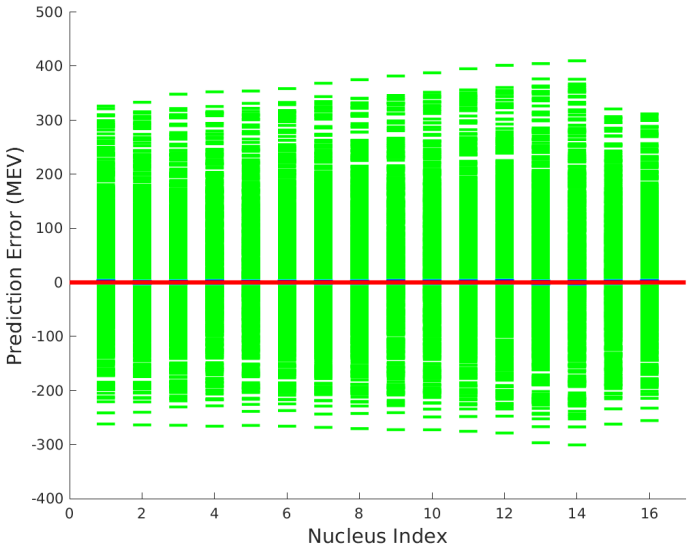
Posterior for Scientific Parameters

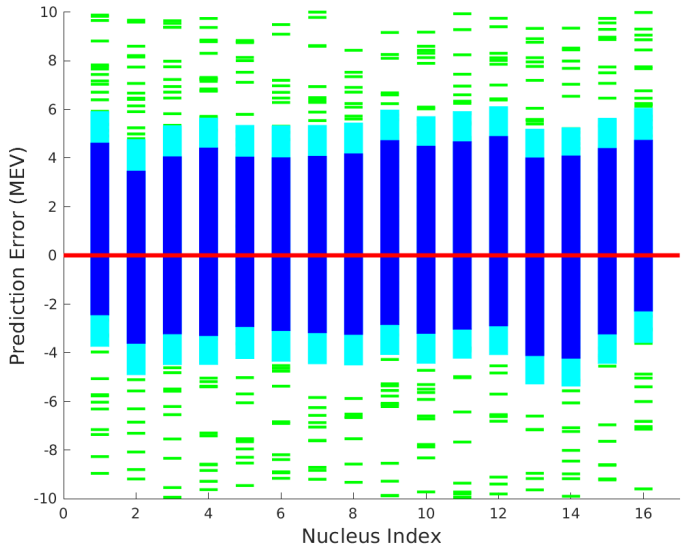


Los Alamos NATIONAL LABORATORY EST. 1943

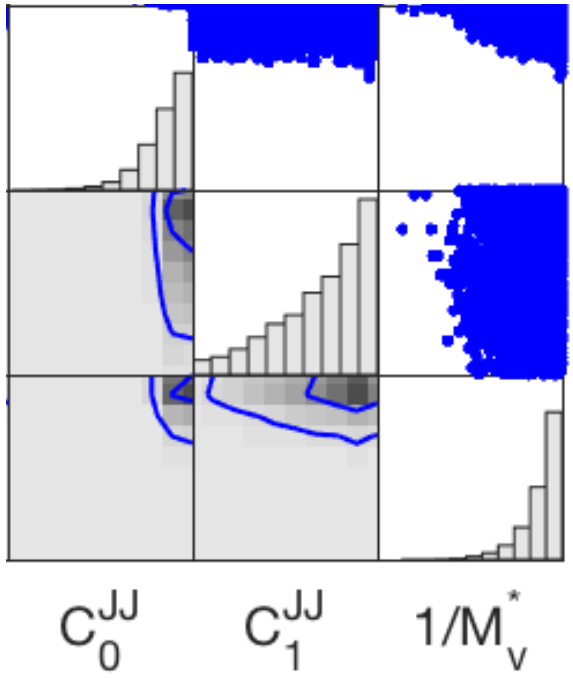
Calibrated Predictions with Quantified Uncertainty

Predictions for new experiments that include parameter uncertainty, measurement error, and emulation uncertainty.



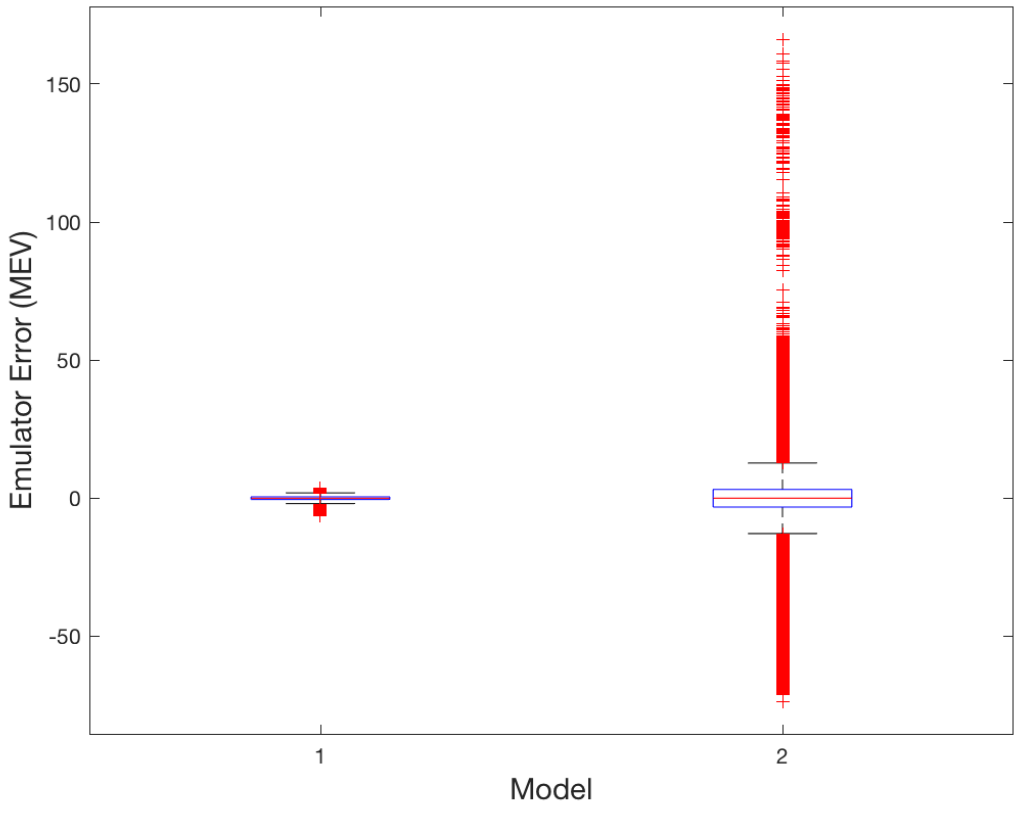


Potential Issues: Calibrated to the Edge

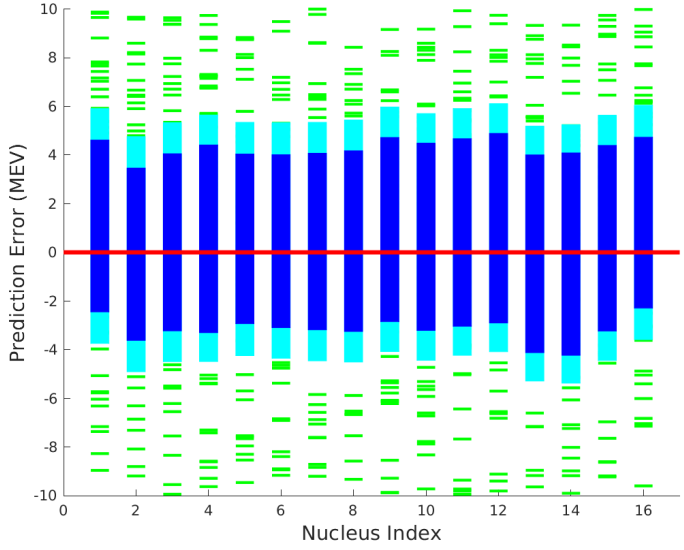


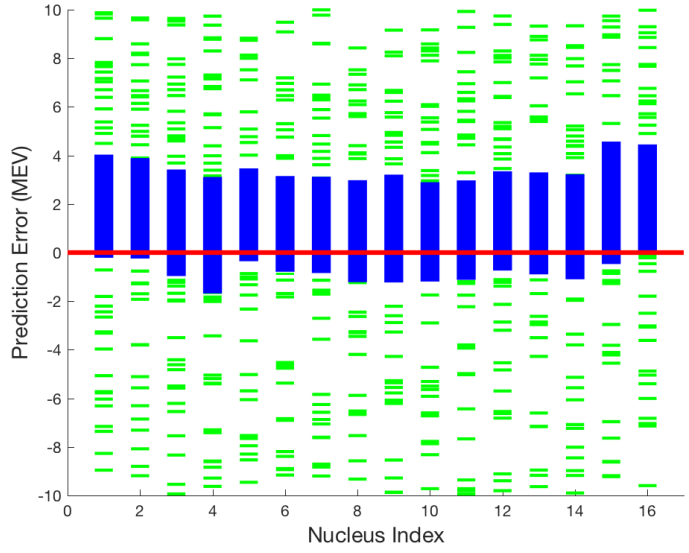
Could indicate trouble matching the data. If there is not a good match, the estimation can try to force one at the edges where the emulator is most uncertain.

Potential Issues: Emulator Error



Potential Issues





Los Alamos
NATIONAL LABORATORY
EST.1943

Potential Issues: Answers that Witek Doesn't Like

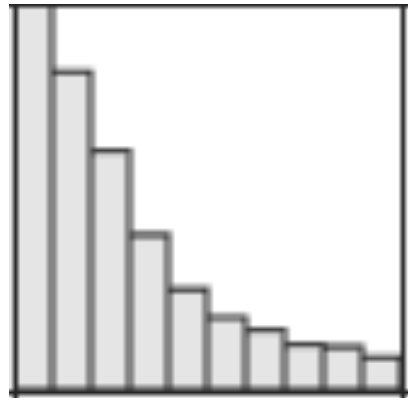
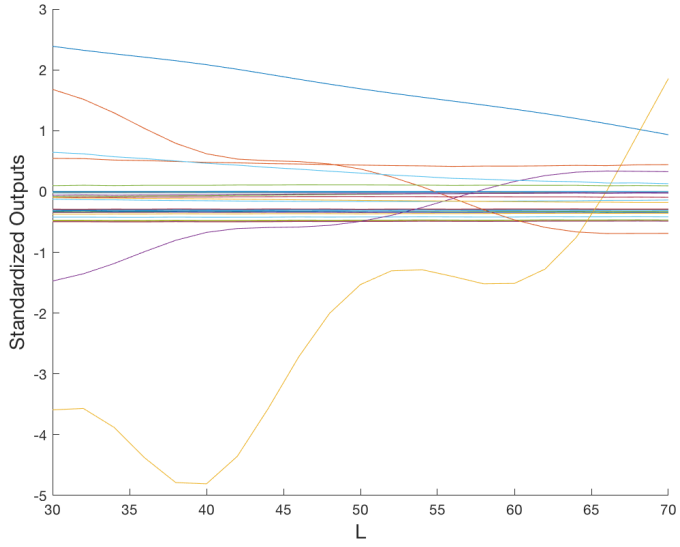


Figure: Posterior Distribution for L

Potential Issues: Answers that Witek Doesn't Like



This parameter doesn't do much. Uninformative prior is probably a bad choice here.

The Most Important Thing That I've Left Out

Model Discrepancy: Sometimes (all the time?), the physics model isn't exactly correct. This approach can be extended to include and estimate this bias.

An important place for prior information, both for specific scientific issues (e.g. amount of magicalness?) or even just general function information (e.g. smoothness).

June 13: Statistics and Beer Day

Celebrate all of the ways in which statistics has made beer, and thus the world, better.

William Sealy Gosset, Statistician, Brewer, Hero.

Los Alamos
NATIONAL LABORATORY
EST. 1943