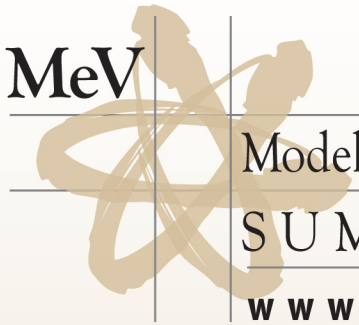
LA-UR-14-26011
Approved for public release; distribution is unlimited.

Title: Design of Experiments, Model Calibration and Data Assimilation

Author(s): Williams, Brian J.

Intended for: use as future presentation materials

Issued: 2014-07-30



Modeling • Experimentation • Validation

SUMMER SCHOOL

www.MeVSchool.org

Design of Experiments, Model Calibration and Data Assimilation

Brian Williams

Statistical Sciences Group

Los Alamos National Laboratory

Session #7

23 July 2014

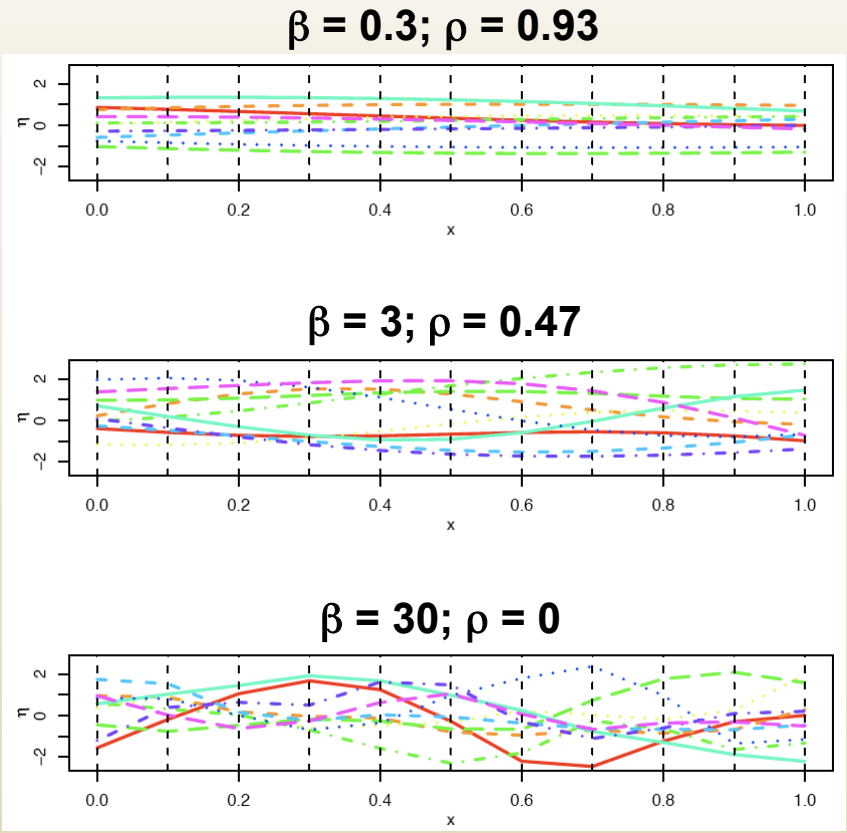
8 – 9:30 AM

Abstract

This presentation provides an overview of emulation, calibration and experiment design for computer experiments. Emulation refers to building a statistical surrogate from a carefully selected and limited set of model runs to predict unsampled outputs. The standard kriging approach to emulation of complex computer models is presented. Calibration refers to the process of probabilistically constraining uncertain physics/engineering model inputs to be consistent with observed experimental data. An initial probability distribution for these parameters is updated using the experimental information. Markov chain Monte Carlo (MCMC) algorithms are often used to sample the calibrated parameter distribution. Several MCMC algorithms commonly employed in practice are presented, along with a popular diagnostic for evaluating chain behavior. Space-filling approaches to experiment design for selecting model runs to build effective emulators are discussed, including Latin Hypercube Design and extensions based on orthogonal array skeleton designs and imposed symmetry requirements. Optimization criteria that further enforce space-filling, possibly in projections of the input space, are mentioned. Designs to screen for important input variations are summarized and used for variable selection in a nuclear fuels performance application. This is followed by illustration of sequential experiment design strategies for optimization, global prediction, and rare event inference.

Emulation of Code Output

Surrogate: Gaussian Process (GP)



Semiparametric regression model for emulating code $\eta(x)$

- Joint distribution of surrogate outputs is multivariate Gaussian
- Mean zero, precision λ
- Correlation function:

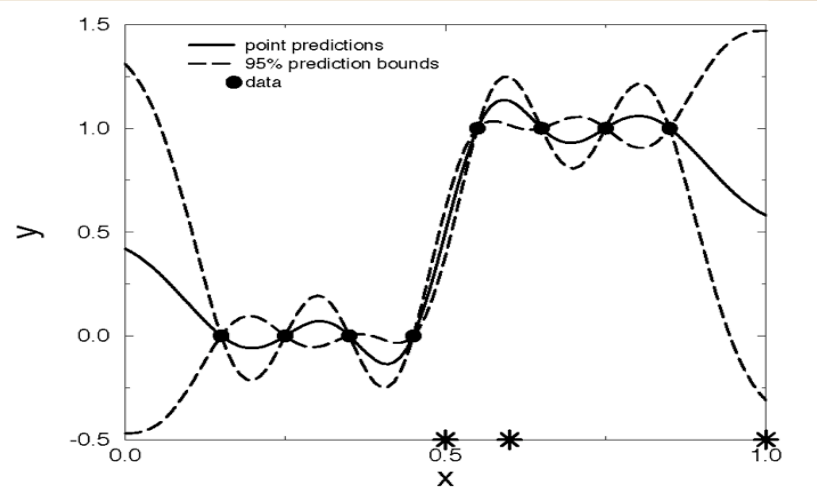
$$R\left(\eta(\mathbf{x}_1),\eta(\mathbf{x}_2)\,|\,\boldsymbol{\beta}\right)=\exp\left(-\sum_{j=1}^{d}\beta_j\left(x_{1,j}-x_{2,j}\right)^2\right)$$

- Define correlation length:
  $\rho_j = \exp(-\beta_j / 4)$
- Notation: GP(0; λ, ρ)

Correlation lengths ρ_j determine *complexity* of process realizations

Emulator

- Use training runs to develop a statistical surrogate model for the complex code (i.e., the *emulator*)
  - Deterministic code is interpolated with zero uncertainty
- Kriging Predictor



outputs evaluated at training runs $\mathbf{x}_1,\ldots,\mathbf{x}_m$

$$\hat{\eta}(\mathbf{x};\boldsymbol{\beta}) = \mathbf{r}^T(\mathbf{x};\boldsymbol{\beta})\mathbf{R}^{-1}(\boldsymbol{\beta})\boldsymbol{\eta}$$

correlations between prediction site $\mathbf{x}$ and training runs $\mathbf{x}_1,\ldots,\mathbf{x}_m$

pairwise correlations between training runs $\mathbf{x}_1,\ldots,\mathbf{x}_m$

- Kriging Variance $Var\left[\eta(\mathbf{x};\boldsymbol{\beta})\right] = \sigma^2\left(1 - \mathbf{r}^T(\mathbf{x};\boldsymbol{\beta})\mathbf{R}^{-1}(\boldsymbol{\beta})\mathbf{r}(\mathbf{x};\boldsymbol{\beta})\right)$
  - Maximum likelihood or full Bayes inference for $\boldsymbol{\beta}$

References

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). "Design and analysis of computer experiments" (with discussion), *Statistical Science*, 4, 409-435.

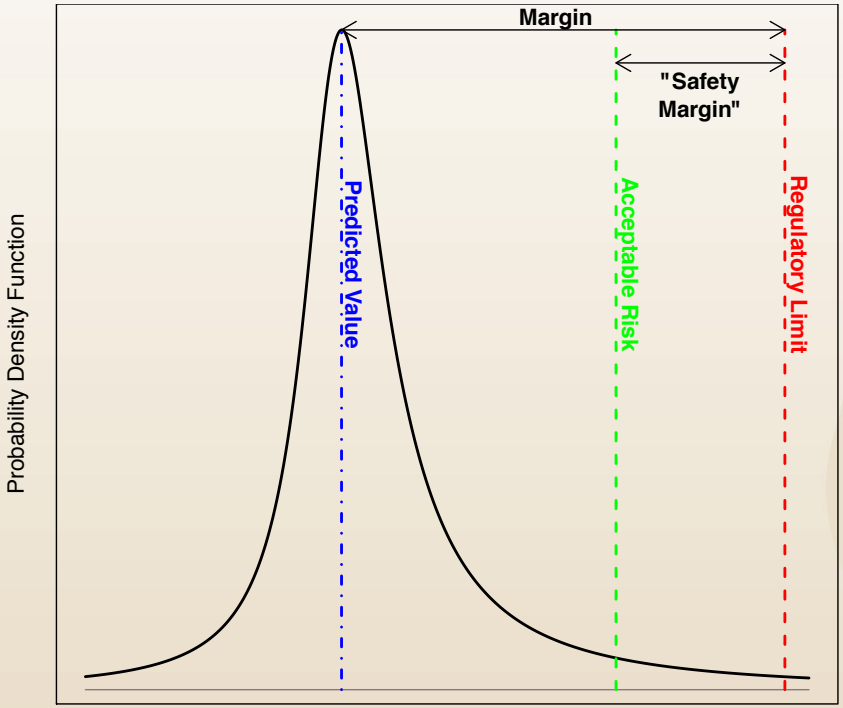
Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991). "Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments," *Journal of the American Statistical Association*, 86, 953-963.

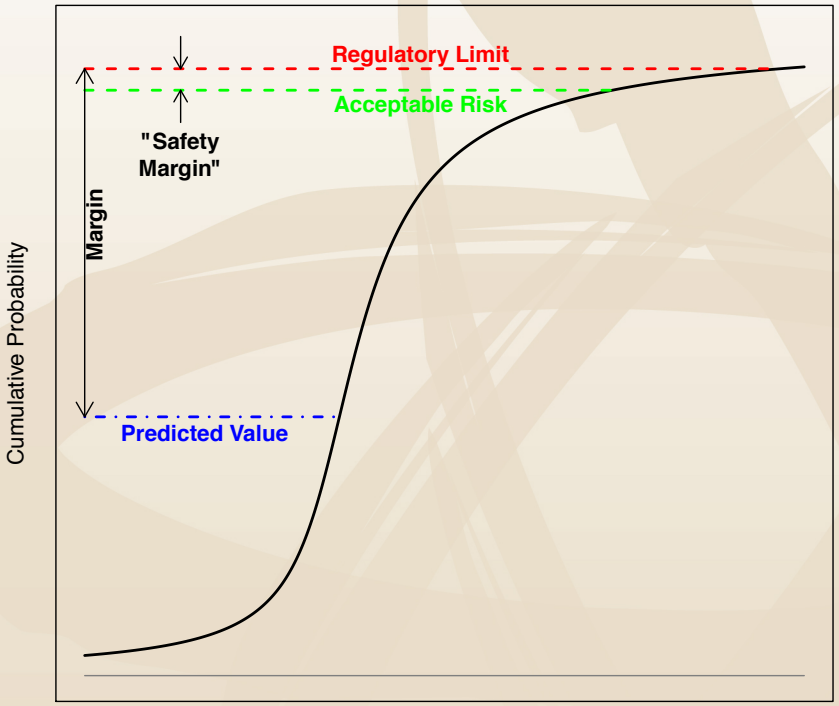
Santner, T. J., Williams, B. J., and Notz, W. I. (2003). *The Design and Analysis of Computer Experiments*. Springer, New York.

Loeppky, J. L., Sacks, J., and Welch, W. J. (2009). "Choosing the sample size of a computer experiment: A practical guide," *Technometrics*, 51, 366-376.

Model Calibration

The Ultimate Objective





- Uncertainty quantification results in distribution of QoI predictions
- Uncertainty reduction in QoI predictions via code calibration results in larger “Safety Margin”

Meaning of *Calibration*

- Computer model $\eta(\boldsymbol{x}, \boldsymbol{t})$
- $\boldsymbol{x}$: design inputs (controlled by experimenter)
- $\boldsymbol{t}$: physics inputs (controlled in model runs)
- $\boldsymbol{t} = \boldsymbol{\theta}$: "best", unknown setting, $[\boldsymbol{\theta}] \sim \pi$ (*prior* PDF)
- Observational data $y(\boldsymbol{x}_1),\ldots,y(\boldsymbol{x}_n)$

– Given θ:

- $y(\boldsymbol{x}_i) = \eta(\boldsymbol{x}_i,\boldsymbol{\theta}) + \varepsilon(\boldsymbol{x}_i)$, $[\varepsilon] \sim f$ (observational error PDF)
- $y(\boldsymbol{x}_i) = \eta(\boldsymbol{x}_i,\boldsymbol{\theta}) + \delta(\boldsymbol{x}_i) + \varepsilon(\boldsymbol{x}_i)$, $[\delta] \sim$ GP
- Calibration results in a *posterior* PDF for $\boldsymbol{\theta}$
- *Bayesian* approach that integrates information from prior constraints and observational data

Example: Linear Model

- Assume linear approximation to $\eta(\theta)$:

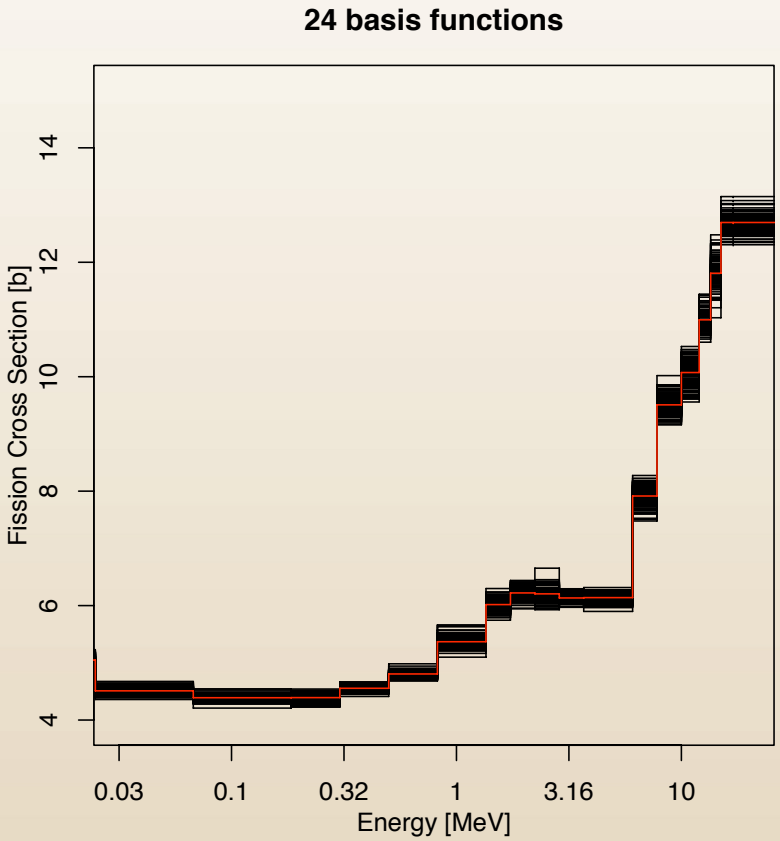
$$\boldsymbol{\eta}(\boldsymbol{\theta}) = \boldsymbol{\eta}(\boldsymbol{\theta}_0) + \boldsymbol{C}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)\,, \text{ where } C_{ij} = \frac{\partial \eta_i}{\partial \theta_j}(\boldsymbol{\theta}_0)$$

- Gaussian prior distribution for θ:
  - $\theta \sim N(\theta_0, \Sigma_0)$
- Experimental data:
  - **Given** θ: $\mathbf{y} = \boldsymbol{\eta}(\boldsymbol{\theta}) + \boldsymbol{\varepsilon}$, $[\boldsymbol{\varepsilon}] \sim N(\mathbf{0}, \Sigma_d)$
- Gaussian posterior distribution for θ:

$$[\boldsymbol{\theta}|\boldsymbol{y}] \sim N\left(\boldsymbol{\theta}_0 + \boldsymbol{P}\boldsymbol{C}^T\boldsymbol{\Sigma}_d^{-1}(\boldsymbol{y} - \boldsymbol{\eta}(\boldsymbol{\theta}_0)), \boldsymbol{P}\right)\,,$$

$$\boldsymbol{P} = (\boldsymbol{\Sigma}_0^{-1} + \boldsymbol{C}^T\boldsymbol{\Sigma}_d^{-1}\boldsymbol{C})^{-1}$$

Cross Section Calibration to Jezebel Critical Assembly Data



Example: Kawano et al., *Nuclear Science and Engineering* 153, 1-7 (2006)

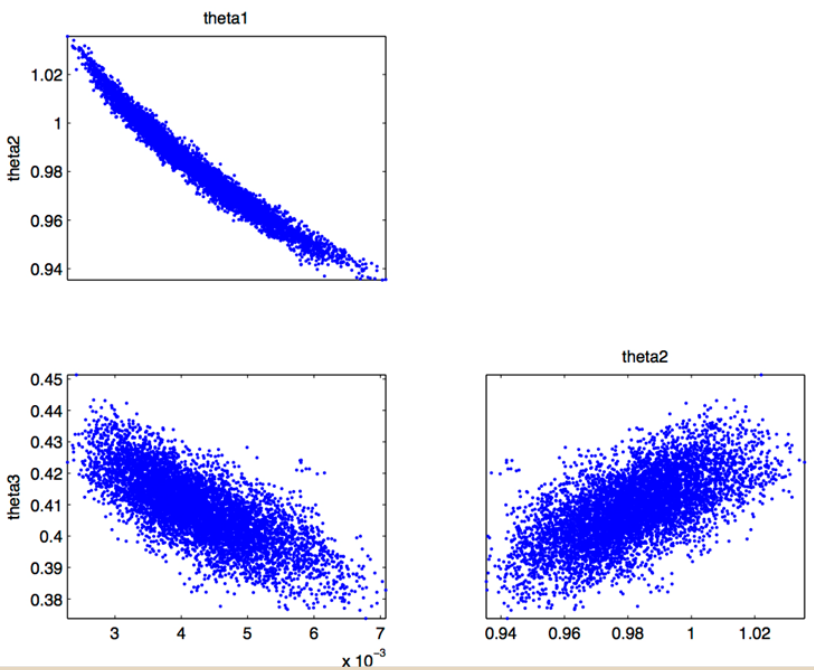
- $\theta = (\theta_1,\ldots,\theta_{30})$ represents 30-group fission cross section (XS) values
- Calibration to differential data provides Gaussian prior distribution of θ for calibration to Jezebel k_{eff}
- $k_{eff}(\theta)$ is linearized around θ_0 based on PARTISN calculations of unperturbed and 1% perturbed group XS values
- Correlation structure (compensating errors) in Gaussian θ posterior introduced by k_{eff} calibration

Example: Nonlinear Model

- Prior distribution for θ: $\theta \sim \pi(\theta)$
  - May be Gaussian
  - Often uniform on ranges provided by SMEs
- Computational model $\eta(\boldsymbol{x}, \theta)$ nonlinear on high prior probability region of parameter space
- Experimental data: $\boldsymbol{y}(\boldsymbol{x}) = \eta(\boldsymbol{x}, \theta) + \varepsilon(\boldsymbol{x})$
- Posterior distribution for θ:
  - Sampled via Markov chain Monte Carlo (MCMC)
  - Metropolis within Gibbs and Adaptive Metropolis

Calibration of the Dittus-Boelter Correlation

Bivariate projections of the joint posterior distribution of $\theta = (\theta_1, \theta_2, \theta_3)$



- Dittus-Boelter is an empirical model for single phase heat transfer involving forced convection and turbulent flow
- Nusselt number (Nu): Ratio of convective to conductive heat transfer across a boundary
- Reynolds number (Re): Ratio of inertial to viscous forces
- Prandtl number (Pr): Ratio of momentum to thermal diffusivity

$$Nu = \theta_1 Re^{\theta_2} Pr^{\theta_3}$$

- Prior for θ: Uniform on SME provided ranges
- Posterior for θ: Sampled with Delayed Rejection Adaptive Metropolis

MCMC: Gibbs Sampler

- Objective: Generate a sample from the target probability distribution described by density function proportional to $\pi(\mathbf{x})$
- Divide parameter $\mathbf{x}$ into b blocks: $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_b)$
  - $\mathbf{x}_{-j} = (\mathbf{x}_1, \ldots, \mathbf{x}_{j-1}, \mathbf{x}_{j+1}, \ldots, \mathbf{x}_b)$
  - Any block may consist of a single parameter
- Assume all conditional distributions $\pi(\mathbf{x}_j \mid \mathbf{x}_{-j})$ available
- Algorithm:
  - Repeat for $i = 1, 2, \ldots, M$
  - Given i, repeat for $j = 1, 2, \ldots, b$
  - Suppose the first $(j - 1)$ blocks of $\mathbf{x}$ have already been updated. The j-th block is updated as follows:
    - Let $\mathbf{x}_i = (\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,j-1}, \mathbf{x}_{i-1,j}, \ldots, \mathbf{x}_{i-1,b})$
    - Sample $\mathbf{y} \sim \pi(\mathbf{x}_j \mid \mathbf{x}_{-j})$
    - Set $\mathbf{x}_i = (\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,j-1}, \mathbf{y}, \mathbf{x}_{i-1,j+1}, \ldots, \mathbf{x}_{i-1,b})$
  - Return values $\mathbf{x}_1, \ldots, \mathbf{x}_M$

MCMC: Metropolis-Hastings

- Objective: Generate a sample from the target probability distribution described by density function proportional to $\pi(\mathbf{x})$
- Algorithm:
  - Repeat for j = 1, 2, …, M
  - Generate **y** from $q(\mathbf{x}_j, \bullet)$ and u from Uniform(0, 1) — proposal density
  - If

$$u \le \alpha(\mathbf{x}_j, \mathbf{y}) \text{ for } \alpha(\mathbf{x}, \mathbf{y}) = \begin{cases} \min\left[\frac{\pi(\mathbf{y})q(\mathbf{y},\mathbf{x})}{\pi(\mathbf{x})q(\mathbf{x},\mathbf{y})}, 1\right], & \text{if } \pi(\mathbf{x})q(\mathbf{x},\mathbf{y}) > 0 \\ 1, & \text{otherwise} \end{cases}$$

    set $\mathbf{x}_{j+1} = \mathbf{y}$
  - Else, set $\mathbf{x}_{j+1} = \mathbf{x}_j$
  - Return values $\mathbf{x}_1, \ldots, \mathbf{x}_M$
- Implementation:
  - Discard initial m_0 samples as ``burn-in”
  - Metropolis: symmetric proposal distribution $q(\mathbf{y},\mathbf{x}) = q(\mathbf{x},\mathbf{y})$

$$\alpha(\mathbf{x}, \mathbf{y}) = \min\left[\frac{\pi(\mathbf{y})}{\pi(\mathbf{x})}, 1\right]$$

  - Challenge is choosing $q(\mathbf{x}, \bullet)$ for effective “mixing”
    - 23.4% multi-parameter, 44% single parameter, 57.4% Langevin diffusion

MCMC: Metropolis Within Gibbs

- Algorithm:
  - Repeat for i = 1, 2, …, M
  - Given i, repeat for j = 1, 2, …, p (number of parameters in $\mathbf{x}$)
  - Suppose the first (j – 1) parameters in $\mathbf{x}$ have already been updated. The j-th parameter is updated as follows:
    - Let $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,j-1}, x_{i-1,j}, x_{i-1,j+1}, \ldots, x_{i-1,p})$
    - Generate y_j from $q_j(x_{i-1,j}, \bullet)$ and u from Uniform(0, 1)
    - Set $\mathbf{y} = (x_{i,1}, \ldots, x_{i,j-1}, y_j, x_{i-1,j+1}, \ldots, x_{i-1,p})$
    - If
      $$u \leq \alpha(\mathbf{x}_i, \mathbf{y}) \text{ for } \alpha(\mathbf{x}, \mathbf{y}) = \min\left[\frac{\pi(\mathbf{y})q_j(\mathbf{y}, \mathbf{x})}{\pi(\mathbf{x})q_j(\mathbf{x}, \mathbf{y})}, 1\right]$$
      set $\mathbf{x}_i = \mathbf{y}$
    - Else, set $y_j = x_{i-1,j}$ and $\mathbf{x}_i = \mathbf{y}$
  - Return values $\mathbf{x}_1, \ldots, \mathbf{x}_M$

MCMC: Adaptive Metropolis

- Algorithm:
  - Proposal distribution: $q_\theta(\mathbf{x}, \mathbf{y})$ is Gaussian($\mathbf{y}$; $\mathbf{x}$, $\lambda\Sigma$) for $\theta = (\lambda, \Sigma)$

mean covariance

  - Repeat for *i* = 1, 2, …, M
    - Generate **y** from $q_{\theta_i}(\mathbf{x}_i, \cdot)$ and *u* from Uniform(0, 1)
    - If

$$u \leq \alpha(\mathbf{x}_i, \mathbf{y}) \text{ for } \alpha(\mathbf{x}, \mathbf{y}) = \min\left[\frac{\pi(\mathbf{y})}{\pi(\mathbf{x})}, 1\right]$$

      set $\mathbf{x}_{i+1} = \mathbf{y}$
    - Else, set $\mathbf{x}_{i+1} = \mathbf{x}_i$
    - Update

e.g. $\gamma_i = i^{-\upsilon}$ for $\upsilon \in (0.5, 1]$

target acceptance rate

$$\log(\lambda_{i+1}) = \log(\lambda_i) + \gamma_{i+1}\left[\alpha(\mathbf{x}_i, \mathbf{y}) - \alpha_t\right],$$

$$\mu_{i+1} = \mu_i + \gamma_{i+1}(\mathbf{x}_{i+1} - \mu_i),$$

$$\Sigma_{i+1} = \Sigma_i + \gamma_{i+1}\left[(\mathbf{x}_{i+1} - \mu_i)(\mathbf{x}_{i+1} - \mu_i)^T - \Sigma_i\right]$$

  - Return values $\mathbf{x}_1, \ldots, \mathbf{x}_M$

MCMC: Delayed Rejection Adaptive Metropolis

- Algorithm is same as Adaptive Metropolis, *except* for delayed rejection of candidates
  - At *i*-th step
    - Generate **y** from $q_1(\mathbf{x}_i, \bullet)$ and u_1 from Uniform(0, 1)
    - If

$$u_1 \leq \alpha_1(\mathbf{x}_i, \mathbf{y}) \text{ for } \alpha_1(\mathbf{x}, \mathbf{y}) = \min\left[\frac{\pi(\mathbf{y})}{\pi(\mathbf{x})}, 1\right]$$

      set $\mathbf{x}_{i+1} = \mathbf{y}$
    - Else, generate **z** from $q_2(\mathbf{x}_i, \bullet)$ and u_2 from Uniform(0, 1)
    - If

$$u_2 \leq \alpha_2(\mathbf{x}_i, \mathbf{y}, \mathbf{z}) \text{ for } \alpha_2(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \min\left[\frac{\pi(\mathbf{z})q_1(\mathbf{z}, \mathbf{y})(1 - \alpha_1(\mathbf{z}, \mathbf{y}))}{\pi(\mathbf{x})q_1(\mathbf{x}, \mathbf{y})(1 - \alpha_1(\mathbf{x}, \mathbf{y}))}, 1\right]$$

      set $\mathbf{x}_{i+1} = \mathbf{z}$
    - Else, set $\mathbf{x}_{i+1} = \mathbf{x}_i$

In AM: Fix $\lambda = 2.38^2/p$

  - $q_1(\mathbf{x}, \mathbf{y})$ is set to $q_\theta(\mathbf{x}, \mathbf{y})$ [Gaussian($\mathbf{y}$; $\mathbf{x}$, $\lambda\Sigma$)] from AM algorithm
  - $q_2(\mathbf{x}, \mathbf{z})$ is set to $q_\theta(\mathbf{x}, \mathbf{z})$ but with scaled covariance matrix γ ($\lambda\Sigma$) for γ generally in (0, 1)

MCMC: Diagnostics I

- Independently simulate m ≥ 2 sequences of length 2M
  - Starting points drawn from overdispersed distribution
  - Discard first M iterations of each sequence
- For each scalar parameter of interest, calculate:

$B/M =$ the variance between the m sequence means, $\bar{x}_{i.}$, each based on M values of x,

$$B/M = \sum_{i=1}^{m} \left(\bar{x}_{i.} - \bar{x}_{..}\right)^2 /(m-1)\,;\ \text{and}$$

$W =$ the average of the m within-sequence variances, s_i^2, each based on $M-1$ degrees of freedom, $W = \sum_{i=1}^{m} s_i^2/m\,.$

MCMC: Diagnostics II

- Estimate the target mean μ
  - Sample mean of the mM simulated values of x: $\hat{\mu} = \bar{x}_{..}$
- Estimate the target variance σ^2

$$\hat{\sigma}^2 = \frac{M-1}{M}W + \frac{1}{M}B$$

  - For finite M, W less than σ^2 (have not yet sampled all of target)
  - Overestimates σ^2 assuming starting distribution is overdispersed
  - Unbiased in limit M $\rightarrow \infty$
- Estimate approximate target distribution for x

$$x \approx \mathcal{T}\left(\mathrm{df} = 2\hat{V}^2/\widehat{\mathrm{var}}\left(\hat{V}\right), \hat{\mu}, \sqrt{\hat{V}} = \sqrt{\hat{\sigma}^2 + B/(mM)}\right)$$

MCMC: Diagnostics III

$$\widehat{\text{var}}\left(\hat{V}\right) = \left(\frac{M-1}{M}\right)^2 \frac{1}{m}\widehat{\text{var}}\left(s_i^2\right) + \left(\frac{m+1}{mM}\right)^2 \frac{2}{m-1}B^2 + 2\frac{(m+1)(M-1)}{mM^2} \times \frac{M}{m}\left[\widehat{\text{cov}}\left(s_i^2, \bar{x}_{i.}^2\right) - 2\bar{x}_{..}\widehat{\text{cov}}\left(s_i^2, \bar{x}_{i.}\right)\right]$$

- Estimated variances and covariances obtained from *m* samples of $\bar{x}_{i.}$ and s_i^2

- Estimate potential scale reduction factor

$$\text{PSRF} = \sqrt{\hat{R}} = \sqrt{\left(\hat{V}/W\right)\text{df}/(\text{df}-2)} \downarrow 1 \text{ as } M \to \infty$$

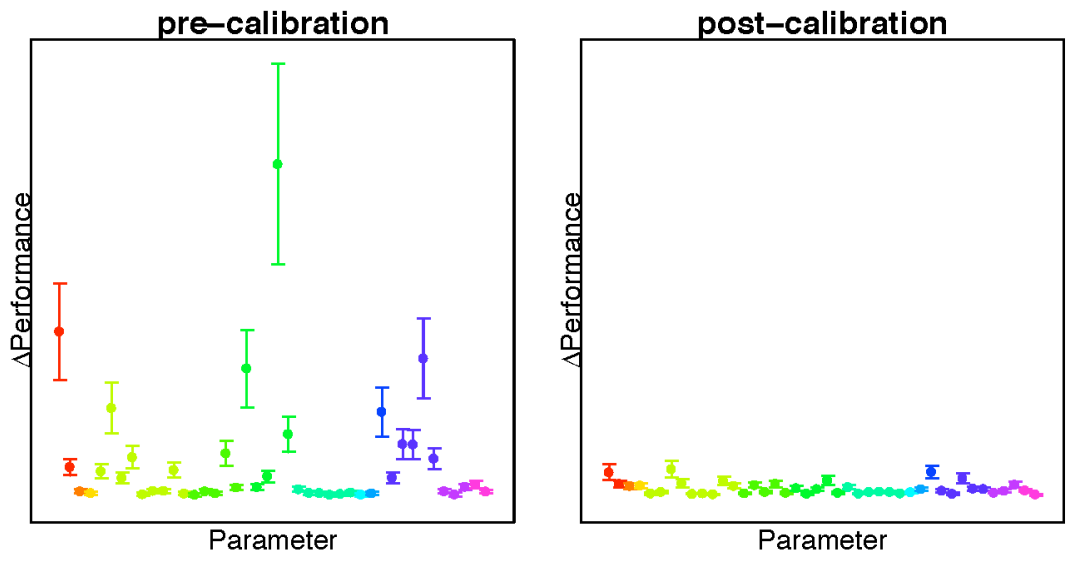
  - Indicates amount of scale reduction expected as M $\rightarrow \infty$
  - Rule of thumb: PSRF < 1.1 desirable; PSRF > 1.2 failure

MCMC: Summary

- Many MCMC algorithms implemented in software
  - MCMCpack, mcmc, adaptMCMC, AMCMC in R
    - http://cran.r-project.org
  - OpenBUGS, WinBUGS
    - http://www.mrc-bsu.cam.ac.uk/software/bugs/
  - DRAM
    - http://helios.fmi.fi/~lainema/dram/
- Failure (large PSRF) can happen for two reasons:
  - One or more chains have not converged to the target
  - One or more chains have not sufficiently explored the target
- Multivariate form of PSRF available
  - Bounds above univariate PSRF for any linear combination of the variables
- R package coda provides a suite of MCMC diagnostic tools

Sensitivity Analysis and Resource Allocation

Output 1		Output 2	
Variation	%Contribution	Variation	%Contribution
age	1.27%	age	7.68%
Stockpile 1	0.18%	Stockpile 1	0.05%
Stockpile 2	0.01%	Stockpile 2	0.01%
Stockpile 3	0.01%	Stockpile 3	0.02%
Stockpile 4	0.01%	Stockpile 4	0.65%
Stockpile 5	6.63%	Stockpile 5	8.76%
Stockpile 6	0.00%	Stockpile 6	0.00%
Physics 1	17.35%	Physics 1	23.60%
Physics 2	4.81%	Physics 2	4.10%
Physics 3	53.19%	Physics 3	19.87%
Physics 4	0.00%	Physics 4	14.94%
Physics 5	15.41%	Physics 5	18.06%



- Sensitivity analysis (SA) summarizes input-output relationships
- SA can be conducted assuming any distribution for input parameters

  For physics parameters, most relevant input distributions are pre-calibration (prior) and post-calibration (posterior)

- Post-calibration SA identifies parameters/ models for further uncertainty reduction

Post-Calibration Sensitivity Analysis

- Begin with MCMC sample of calibrated parameters $\{\theta_1, \ldots, \theta_M\}$
- For the *j*-th parameter, estimate its *main effect*
  - Divide the range of $\theta_{i,j}$, $i = 1, \ldots, M$, into bins
  - For each bin, average the output predictions $\eta(\mathbf{x}, \theta)$ corresponding to samples for which $\theta_{i,j}$ is contained in the bin
  - If desired, apply a smoother to the average output predictions across all bins to estimate the main effect function for input *j*
- For the *j*-th parameter, estimate its main effect variance
  - Compute variance of estimated main effect function with respect to the calibrated marginal distribution of input *j*
  - Calibrated marginal distributions estimated by histograms of parameter values developed for main effect function estimation
- Rank parameters in decreasing order of main effect variance

References

Andrieu, C. and Thoms, J. (2008). "A tutorial on adaptive MCMC," *Statistics and Computing*, 18, 343-373.

Casella, G. and George, E. (1992). "Explaining the Gibbs sampler," *The American Statistician*, 46, 167-174.

Chib, S. and Greenberg, E. (1995). "Understanding the Metropolis-Hastings algorithm," *The American Statistician*, 49, 327-335.

Gelman, A. and Rubin, D.B. (1992). "Inference from iterative simulation using multiple sequences," *Statistical Science*, 7, 457-472.

Haario, H., Laine, M., and Mira, A. (2006). "DRAM: Efficient adaptive MCMC," *Statistics and Computing*, 16, 339-354.

Kawano, T., Hanson, K.M., Frankle, S., Talou, P., Chadwick, M.B., and Little, R.C. (2006). "Evaluation and propagation of the ^{239}Pu fission cross-section uncertainties using a Monte Carlo technique," *Nuclear Science and Engineering*, 153, 1-7.

Unal, C., Stull, C.J., and Williams, B.J. (2013). "Parametric uncertainty in a thermal conductivity model of uranium oxide light water nuclear reactor fuel," *Review of Applied Physics*, 2, 39-48.

Unal, C., Williams, B.J., Yacout, A., and Higdon, D.M. (2013). "Application of advanced validation concepts to oxide fuel performance codes: LIFE-4 fast-reactor and FRAPCON thermal-reactor fuel performance codes," *Nuclear Engineering and Design*, 263, 102-128.

Experiment Design

Why Experiment Design?

- Originally introduced to detect treatment effects in agricultural experiments
  - Overcome limitations of one-factor-at-a-time experiments
  - Blocking, randomization, replication
- How are runs determined?
  - Design criterion
  - Parameter estimation, detecting treatment effect(s), prediction
- How is any of this relevant to deterministic computer models?
  - "Slow" computer models and finite resources
  - Prediction of model output at *unsampled* inputs

Regression and Design

- Consider the traditional linear model framework:

$$y(\mathbf{x}_i) = \mathbf{f}^T(\mathbf{x}_i)\boldsymbol{\beta} + \varepsilon_i, \quad \varepsilon_i \sim N(0,\sigma^2)$$

regression model

regression parameters

- Regression predictor and variance

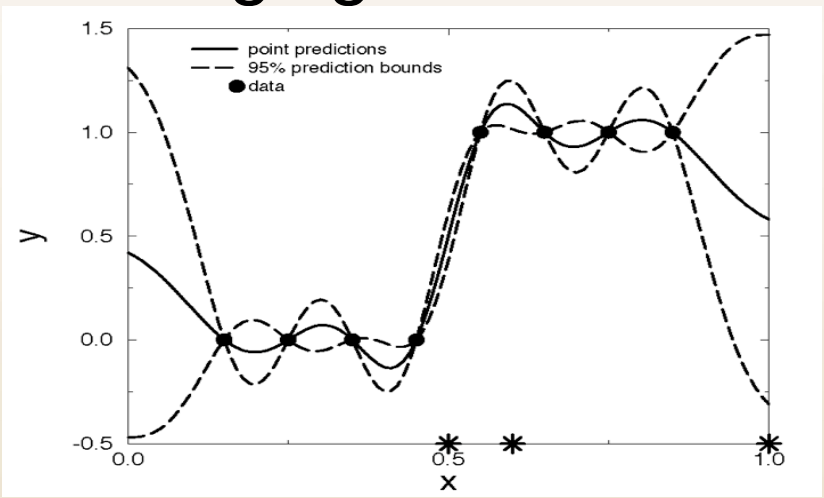
matrix of regression functions for proposed runs $\mathbf{x}_1,\ldots,\mathbf{x}_m$

$$\hat{y}(\mathbf{x}) = \mathbf{f}^T(\mathbf{x})\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\beta}} = \left(\mathbf{F}^T\mathbf{F}\right)^{-1}\mathbf{F}^T\mathbf{y}, Var\left[y(\mathbf{x})\right] = \sigma^2\left(1 + \mathbf{f}^T(\mathbf{x})\left(\mathbf{F}^T\mathbf{F}\right)^{-1}\mathbf{f}(\mathbf{x})\right)$$

- Design Problem: Choose runs $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ to minimize some functional of prediction variance
  - Maximum variance (G-optimal)
  - Average variance (I- or L-optimal)

Kriging and Design

- Kriging Predictor



outputs evaluated at proposed runs $\mathbf{x}_1,\ldots,\mathbf{x}_m$

$$\hat{\eta}(\mathbf{x};\boldsymbol{\beta}) = \mathbf{r}^T(\mathbf{x};\boldsymbol{\beta})\mathbf{R}^{-1}(\boldsymbol{\beta})\boldsymbol{\eta}$$

correlations between prediction site $\mathbf{x}$ and proposed runs $\mathbf{x}_1,\ldots,\mathbf{x}_m$

pairwise correlations between proposed runs $\mathbf{x}_1,\ldots,\mathbf{x}_m$

- Kriging Variance

$$Var\left[\eta(\mathbf{x};\boldsymbol{\beta})\right] = \sigma^2\left(1 - \mathbf{r}^T(\mathbf{x};\boldsymbol{\beta})\mathbf{R}^{-1}(\boldsymbol{\beta})\mathbf{r}(\mathbf{x};\boldsymbol{\beta})\right)$$

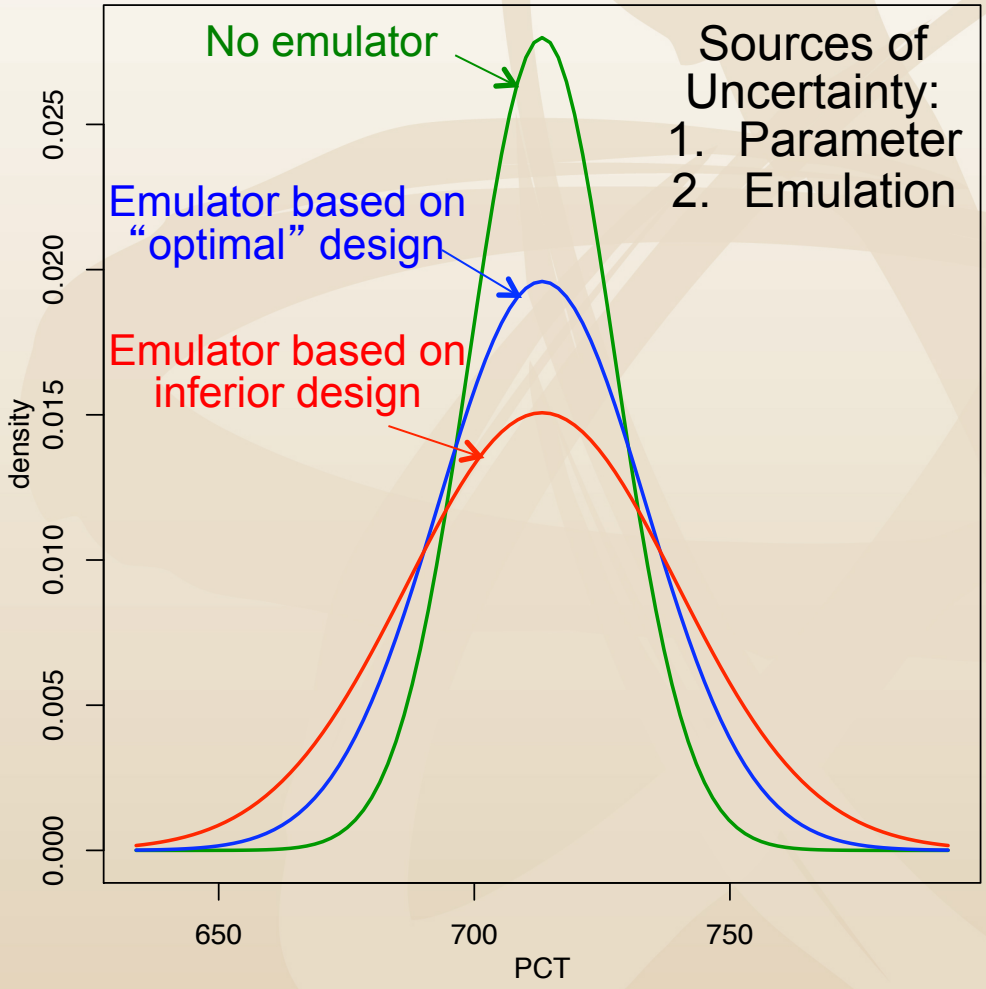
  - Function of unknown parameters β

- Design Problem: Select runs $\mathcal{D} = \{x_1, \ldots, x_m\}$, but how?
  - Heuristics, Bayesian design

Choice of Design Affects Prediction Quality

- Uncertainty in PCT as function of initial ambient conditions and system degradation
  - Fong *et al.* 2009, RESS

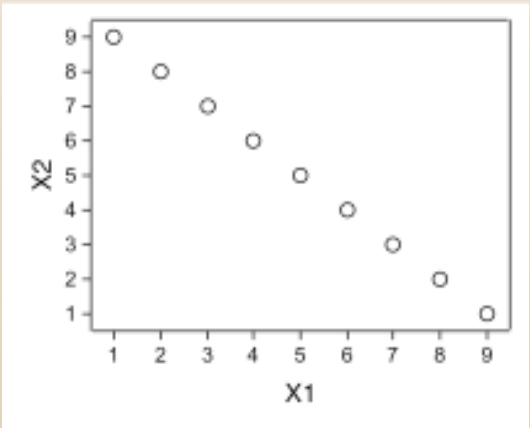
Variable	Distribution	20%	80%	Rate
P_Tubes	Exponential	-	0.15	10.7
P_Temp	Truncated Normal	7	47	-
R_Emis	Truncated Normal	0.65	0.85	-
R_Block	Exponential	-	0.15	10.7
RVACS_T	Truncated Normal	7	47	-



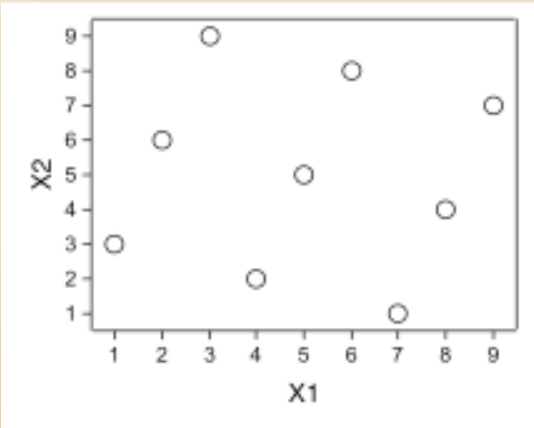
Latin Hypercube Designs (LHDs)

- Each column is a permutation of the numbers from 1 to *m*
- For any number of factors *p* and runs *m*, there are *many* LHDs $[(m!)^{p-1}]$
- Therefore, we specify another criterion to select one:
  - Maximize minimum distance between points
  - Minimize column correlations
- Random LHDs are potentially problematic

Unfortunate random LHD



Maximin Distance LHD



Minimum Average Distance Criteria

- Optimize over projections of design
- Criterion function

$$av_{(\rho,\ell)}(X) = \left(\frac{1}{\sum_{j \in J} C(p,j)} \sum_{j \in J} \sum_{k=1}^{C(p,j)} \left[m_{(\rho,\ell)}(X_{jk}) \right]^{\ell} \right)^{1/\ell}$$

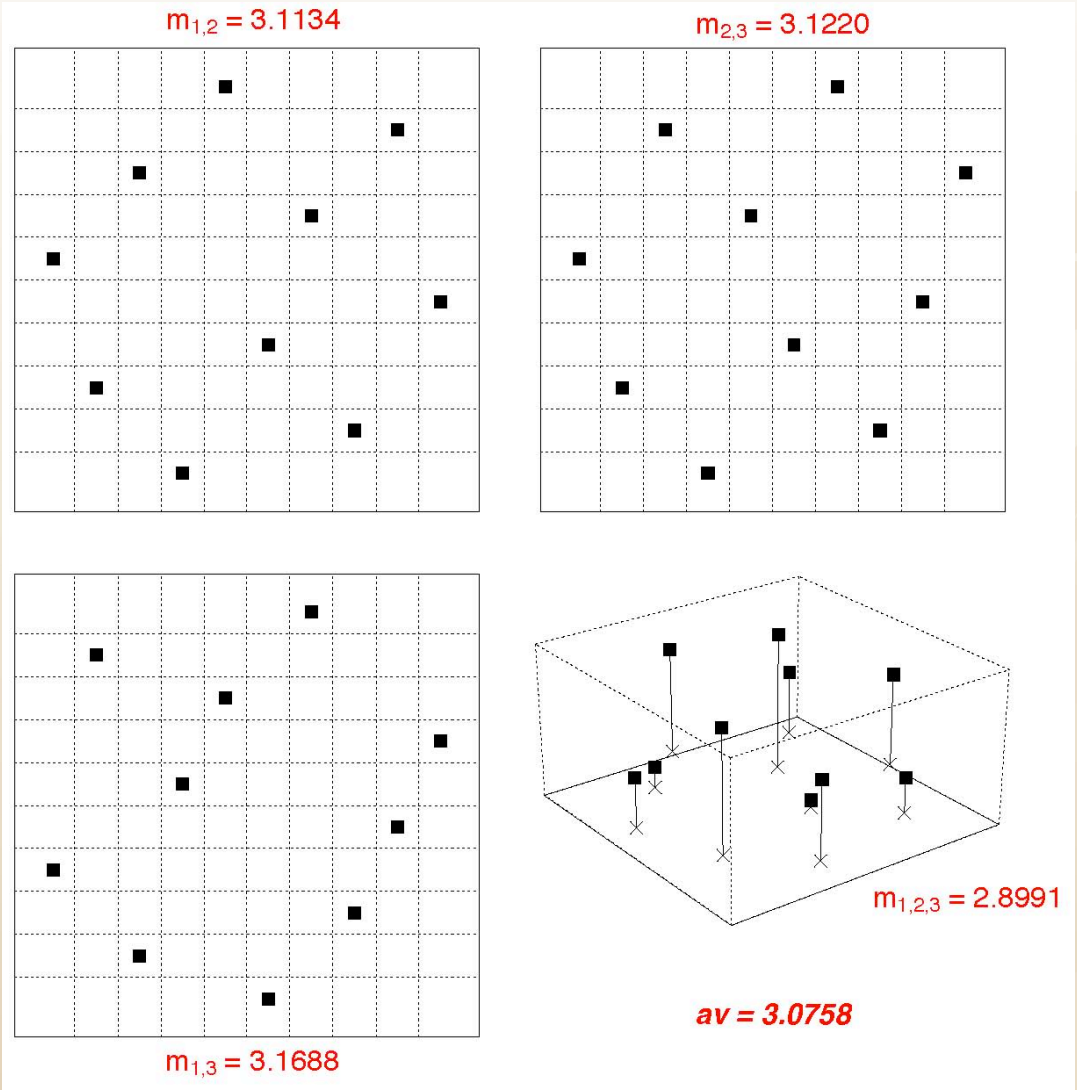
  - $J =$ set of projection dimensions; $C(p,j) = p!/[j!(p-j)!]$
  - X is candidate LHD; $X_{jk} = k$-th projection into j dimensions

- Criterion value for projected design

$$m_{(\rho,\ell)}(X_{jk}) = \left(\frac{1}{C(m,2)} \sum_{1 \le h < i \le m} \left[\frac{j^{1/\rho}}{d_{\rho}(\boldsymbol{x}_h^{jk}, \boldsymbol{x}_i^{jk})} \right]^{\ell} \right)^{1/\ell}$$

  - $d_{\rho}(\cdot,\cdot) = L_{\rho}$ distance
  - $\ell \to \infty$ maximizes minimum distance

Example: Minimum Average Distance LHD



$m = 10$
$p = 3$
$\rho = 1$
$\ell = 1$
$J = \{2,3\}$

OA-based LHDs

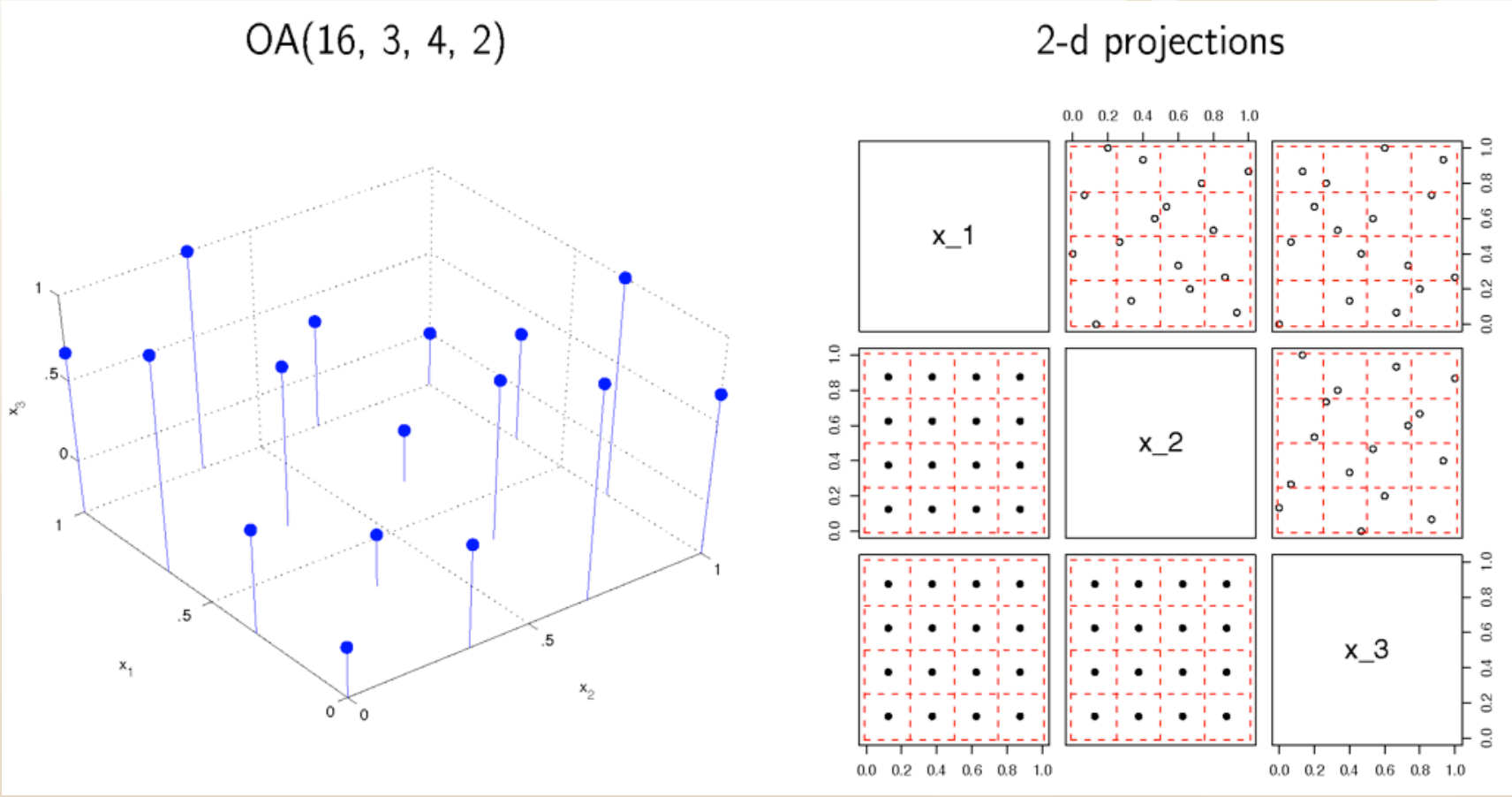
- Begin with a balanced orthogonal array OA(*m*, *p*, *s*, *t*)
  - *m* x *p* design, each column has *s* levels, strength *t*
  - All s^t level combinations for any *t* columns occur equally often
  - If possible, $t \geq 3$, often $s = 2$
- Convert to LHD via Tang (1993)
  - Example: Convert OA(8,3,2,3) to LHD

$$\begin{matrix} 0&0&0 \\ 0&0&1 \\ 0&1&0 \\ 0&1&1 \\ 1&0&0 \\ 1&0&1 \\ 1&1&0 \\ 1&1&1 \end{matrix} \rightarrow \begin{matrix} 0&3&2 \\ 2&1&5 \\ 1&6&0 \\ 3&4&6 \\ 4&2&3 \\ 6&0&7 \\ 7&5&1 \\ 5&7&4 \end{matrix}$$

OA level	Design column	Random permutation
	1	$\{0,2,1,3\}$
0	2	$\{3,1,2,0\}$
	3	$\{2,0,3,1\}$
	1	$\{4,6,7,5\}$
1	2	$\{6,4,5,7\}$
	3	$\{5,6,7,4\}$

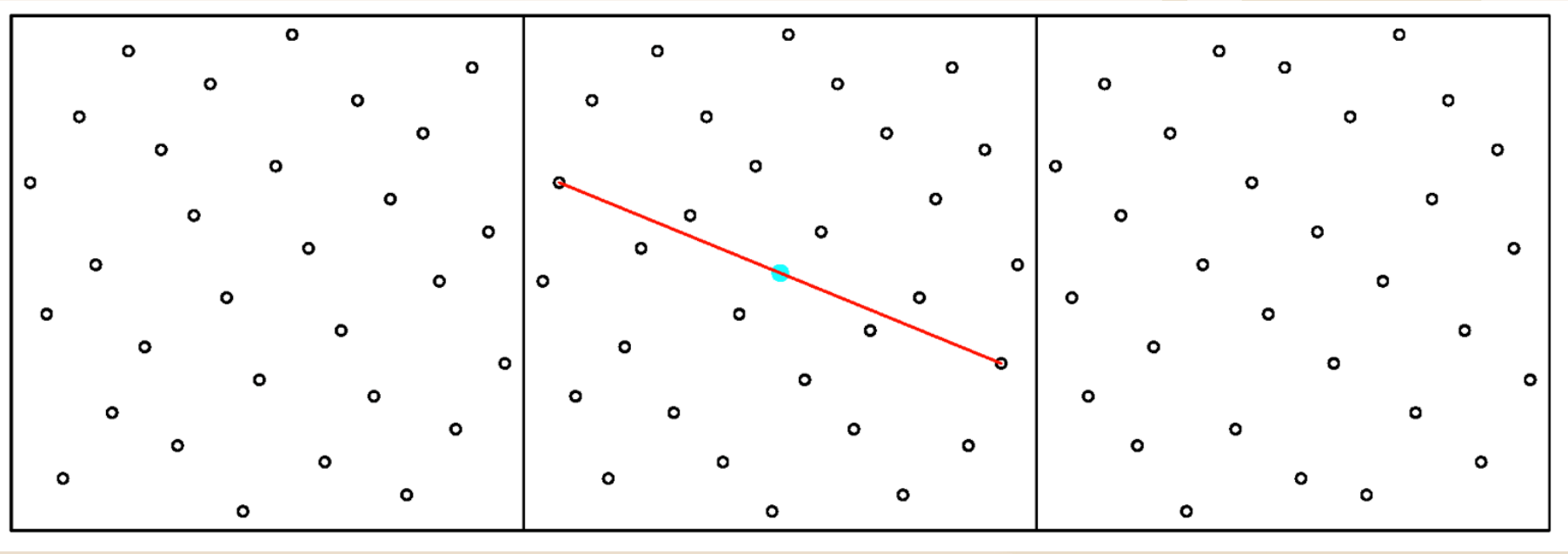
- Combine with distance-based criteria

OA-based LHDs



- OA designs good for estimating low order effects
- OA-based LHD good for GP response surface modeling

Symmetric LHDs



- Symmetry with respect to midpoint of input domain
- Optimize with respect to a distance criterion
  - maximin distance, average projected distance
  - columnwise-pairwise or simulated annealing algorithms

Columnwise-pairwise (CP) Algorithm

- Columnwise algorithms make exchanges on the columns in a design
  - Useful for designs with structure requirements on the columns
- Modification is necessary to accommodate symmetry
  - Two simultaneous pair exchanges
- Algorithm:
  1. Start with a random *m* x *p* symmetric LH design
  2. Each iteration has *p* steps. At the *i*-th step, the best two simultaneous exchanges within column *i* are found. The design matrix is updated accordingly.
  3. If the resulting design is better w.r.t. the criterion, repeat Step 2. Otherwise the search is terminated at the current design.

Simulated Annealing (SA) Algorithm

- Let $\phi(D)$ denote the criterion value for design D
- Algorithm:
  1. *Initialization*. Define t_0, I_{max}, FAC_t, and t_{min}. Randomly select an initial symmetric LH design D. Set $D_{best} = D$, $t = t_0$.
  2. *Temperature loop*: Set $FLAG = 0$, $I = 1$.
  3. *Perturbation loop*: Set D_{try} to D. Randomly select a column of D_{try}, then randomly select two elements within this column. Simultaneously exchange these two elements and their symmetric pairs.
  4. If $\phi(D_{try}) < \phi(D)$, or with probability $\exp[-(\phi(D_{try}) - \phi(D))/t]$, set D to D_{try} and $FLAG$ to 1.
  5. If $\phi(D_{try}) < \phi(D_{best})$, set D_{best} to D_{try} and I to 1. Otherwise increment I by 1.
  6. If $I < I_{max}$, branch to Step 3.
  7. If $FLAG = 1$ and $t > t_{min}$, multiply t by FAC_t and branch to Step 2.
  8. Stop and report D_{best}.

Quasi-Monte Carlo (MC) Sequences

- Better error properties than MC for approximating integrals
  - "low-discrepancy" sequences, e.g. Sobol' or Niederreiter

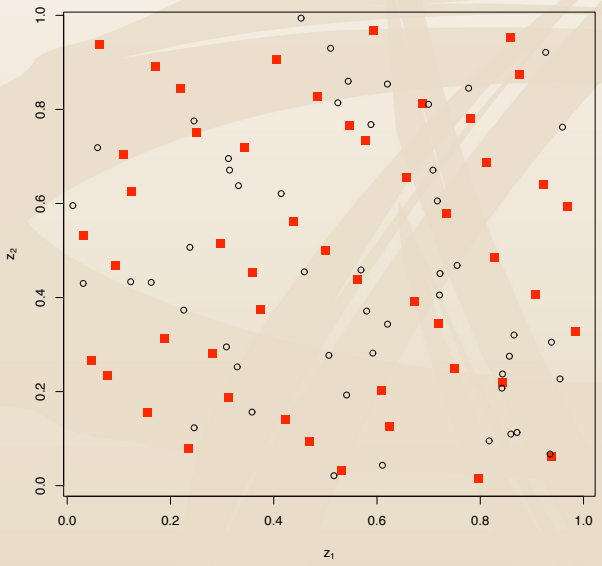
```
#include <stdio.h>
#include <gsl/gsl_qrng.h>

int main( void )
{
   int i, j;
   int m=50, p=2;
   gsl_qrng *q;

   q = gsl_qrng_alloc( gsl_qrng_sobol, p );

   for( i=0; i<m; i++ ) {
      double v[p];
      gsl_qrng_get( q, v );
      for( j=0; j<p-1; j++ ) {
         printf( "%.5f ", v[j] );
      }
      printf( "%.5f\n", v[p-1] );
   }

   gsl_qrng_free( q );
   return 0;
}
```



Sobol' sequence

Random sample

- 50 runs, 2 parameters

Performance of Space-Filling Designs

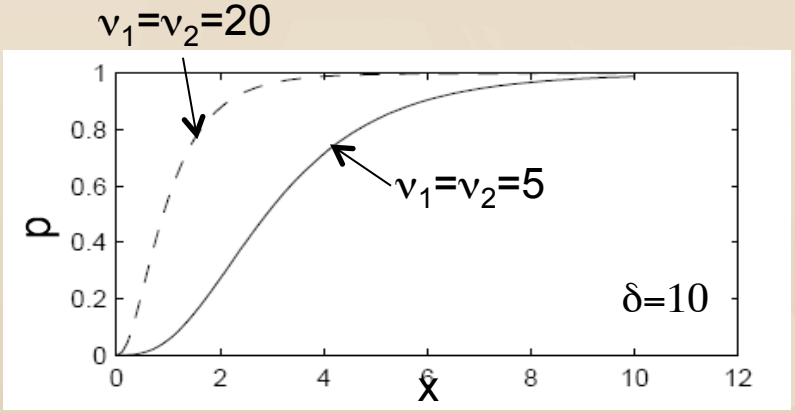
- Substantial empirical evidence supporting use of space-filling designs for GP modeling
- Example: F-Quantile Function

The noncentral F cdf is:

$$F(x|\nu_1, \nu_2, \delta) = \sum_{j=0}^{\infty} \left(\frac{\left(\frac{1}{2}\delta\right)^j}{j!} e^{-\frac{\delta}{2}} \right) I\left(\frac{\nu_1 \cdot x}{\nu_2 + \nu_1 \cdot x} \middle| \frac{\nu_1}{2} + j, \frac{\nu_2}{2} \right)$$

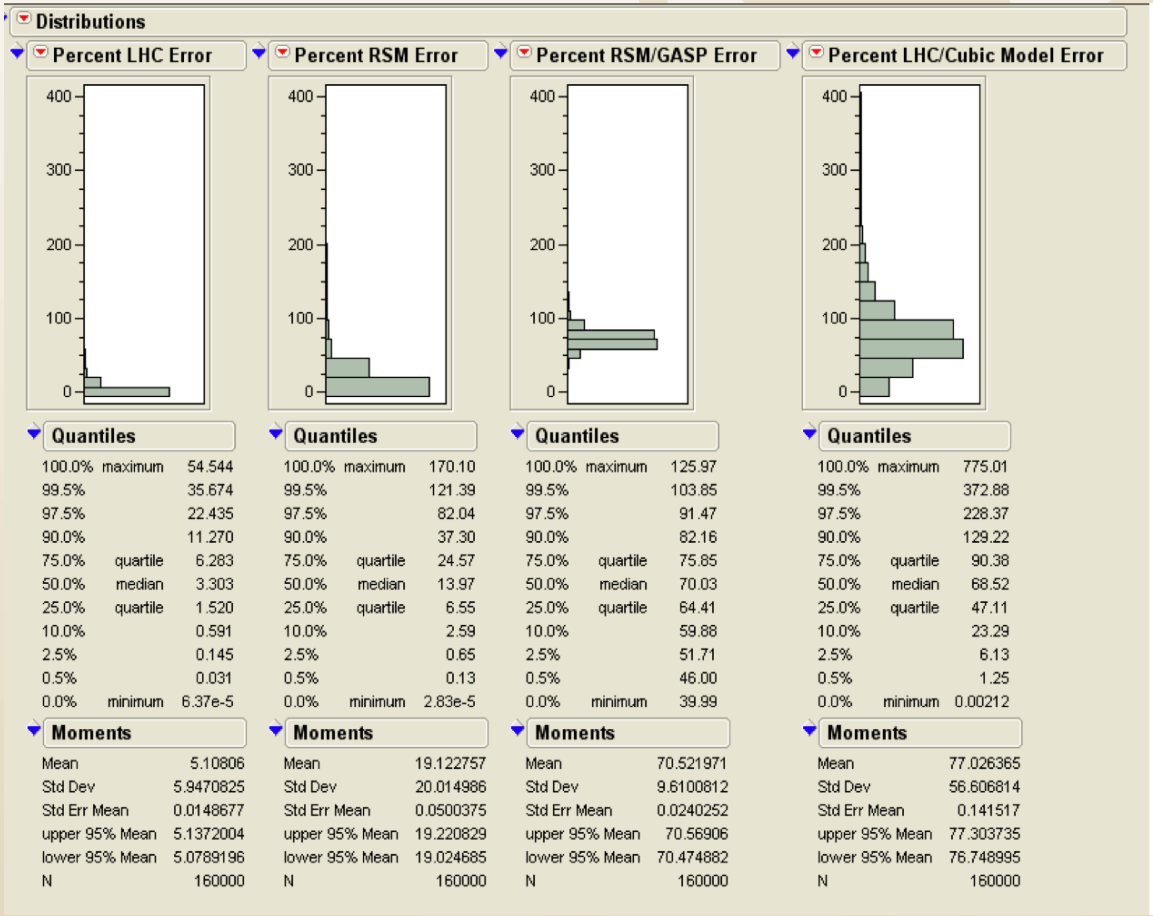
where $I(x/a,b)$ is the incomplete beta function with parameters a and b.

  - Given (ν_1, ν_2, δ, p), compute x



Results

- Two 35-run designs
  - Maximin Distance LHD
  - D-optimal
- Two surrogates
  - GP
  - full cubic polynomial
- Prediction quality
  - Measured by % error
  - 160,000 test samples



Example: Rachel Johnson and Brad Jones

Conclusions

- Experiment design strategy must be tailored to inference objectives
  - (fractional) factorial, alphabet optimal designs work well with regression, ANOVA
  - space-filling designs work well with GP models typically used in computer model emulation
- Space-filling experiment designs can be optimized
  - Obtain desirable projection properties
  - Restricted permutations
  - Optimization algorithms tailored to design column structure

References

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, 21, 239-245.

Johnson, M., Moore, L., and Ylvisaker, D. (1990). "Minimax and maximin distance designs," *Journal of Statistical Planning and Inference*, 26, 131-148.

Tang, B. (1993). "Orthogonal array-based Latin hypercubes," *Journal of the American Statistical Association*, 88, 1392-1397.

Ye, K. Q., Li, W. and Sudjianto, A. (2000). "Algorithmic construction of optimal symmetric Latin hypercube designs," *Journal of Statistical Planning and Inference*, 90, 145-159.

Morris, M. D. and Mitchell, T. J. (1995). "Exploratory designs for computational experiments," *Journal of Statistical Planning and Inference*, 43, 381-402.

Sobol, I. M. (1967). "On the distribution of points in a cube and the approximate evaluation of integrals," *USSR Computational Mathematics and Mathematical Physics*, 7, 86-112.

Screening Experiments

Why Screening?

- Often effect sparsity pertains, in which most of the output variation (~80-90%) is explained by relatively few inputs (~10-20%)
- Emulation of complex computer models often becomes substantially more difficult as input dimension increases
  - Minimum inter-point distance increases, potentially having a negative impact on covariance estimation
- Resources often used more efficiently with the two-stage procedure of screening followed by production analysis
  - Removal of inactive inputs results in (perhaps substantially) fewer computer model runs required for production analysis

R7: Screening and the "20-80 Rule"

Variable	Main Effect %	Total Effect %
PrzPreEnd	0.02%	0.21%
PrzDnTime	0.00%	0.01%
PumpTripPre	0.00%	0.01%
PumpStopTime	0.65%	2.76%
PumpPow	54.39%	59.15%
SCRAMtemp	0.12%	1.67%
CRinject	0.00%	0.01%
CRtime	0.00%	0.02%
HRadCore	0.11%	0.17%
HRadHX	0.12%	0.15%
TwallHX	26.64%	27.19%
Qcore	1.04%	1.08%
PheadNom	0.33%	0.37%
UncFric	0.00%	0.19%
UncNu	11.59%	11.81%
sph	0.00%	0.00%
vis	0.02%	0.12%
dPREdDEN	0.01%	0.53%
dPREdTEM	0.00%	0.01%
alphaLLF	0.02%	0.12%

Output: Peak Coolant Temperature

Two Stage Screening Algorithm

1. 100 runs in stage 1
2. 30 runs in stage 2

Three parameters selected: PumpPow, TwallHX, and UncNu

Results are consistent with global sensitivity analysis based on 300 runs (sensitivity indices presented in table)

Method of Morris

- Factorial Sampling Plans for Preliminary Computational Experiments
- *f* inputs scaled to unit interval, restricted to a *p*-level grid:

$$x_i \in \{0, 1/(p-1), 2/(p-1), \ldots, 1\}$$

- Inference based on *elementary effects*:

$$d_i(\mathbf{x}) = \left[y(x_1, x_2, \ldots, x_{i-1}, x_i + \Delta, x_{i+1}, \ldots, x_f) - y(\mathbf{x}) \right] / \Delta$$

  - Output $y(\mathbf{x})$, $x_i \le 1 - \Delta$
  - Δ is a pre-specified multiple of $1/(p-1)$
- Goal: Important effects identified with a design having number of runs proportional to *f*
  - Estimate finite distribution F_i of $p^{f-1}(p - \Delta(p-1))$ elementary effects

Method of Morris: Interpretation

- Large measure of (absolute) central tendency for F_i indicates input x_i has an important overall influence on y
  - Linear effects
- Large measure of spread indicates x_i is highly dependent on the values of the inputs **x**
  - Nonlinear or interaction effects
- Use estimates of the mean and standard deviation (SD) of each F_i to screen for important effects
  - K-means clustering can be used to separate the absolute means and SDs of each F_i into two clusters: active and inactive inputs
- Sampling plans provide random samples from each F_i on which these estimates are based

Method of Morris: Examples

- Elementary effect distributions F_i for 3 canonical functions

  μ_i = mean of F_i and σ_i = standard deviation of F_i; $\Delta = p/[2(p-1)]$

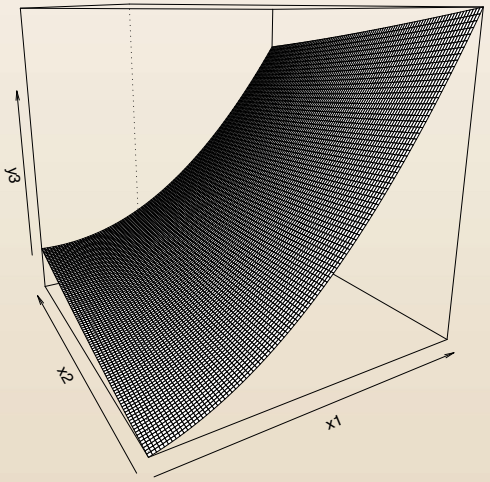
$$y_1(x_1, \ldots, x_f) = a_0 + \sum_{j=1}^{f} a_j x_j \quad \Rightarrow \quad \begin{aligned} &\text{Prob}[d_i = a_i] = 1 \\ &\mu_i = a_i \text{ and } \sigma_i = 0 \end{aligned}$$

$$y_2(x_1, \ldots, x_f) = a_0 + \sum_{j=1}^{f} a_j x_j^2 \quad \Rightarrow \quad \begin{aligned} &\text{Prob}[d_i = a_i(4j+p)\Delta/p] = 2/p \\ &\text{for } j \in \{0, 1, 2, \ldots, (p-2)/2\} \\ &\mu_i = a_i \text{ and } \sigma_i = |a_i| \sqrt{\frac{(p-2)(p+2)}{12(p-1)^2}} \end{aligned}$$

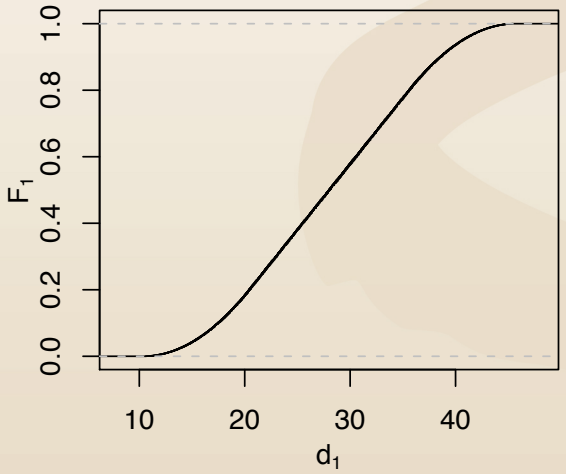
$$y_3(x_1, \ldots, x_f) = a_0 + \sum_{j=1}^{f} a_j x_j + \left(\sum_{j=1}^{f} b_j x_j \right)^2 \quad \Rightarrow \quad \begin{aligned} &d_i(\boldsymbol{x}) = a_i + b_i^2 \Delta + 2\, b_i \sum_{j=1}^{f} b_j x_j \\ &\mu_i = a_i + b_i \sum_{j=1}^{f} b_j \text{ and} \\ &\sigma_i = |b_i| \sqrt{\frac{4(p^2-1)\sum_{j=1}^{f} b_j^2 - 3p^2 b_i^2}{12(p-1)^2}} \end{aligned}$$

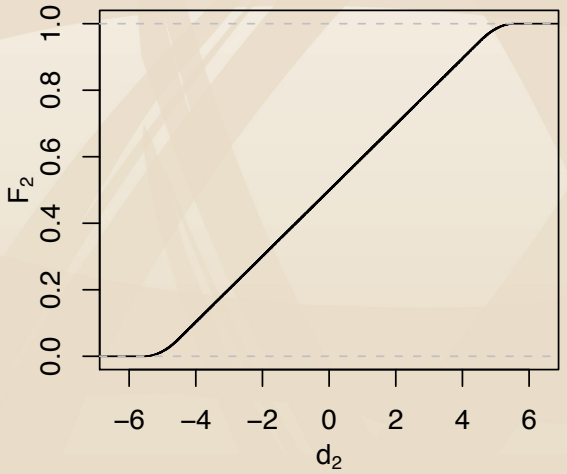
Example: A Quadratic Function

- Function: $y_3(x_1,x_2)$, $x_i \in [0,1]$
  - **a** = (-4, 8, 4)
  - **b** = (5, -1)



Elementary Effect Distributions





- Summary statistics
  - Grid size p = 100
  - $\Delta = p/[2(p-1)]$

	Mean	Standard Deviation
F_1	28	7.85
F_2	0	2.93

GSinCE

- **G**roup **S**creening **in C**omputer **E**xperiments
- Two-stage procedure
  - Stage 1: Identify groups of active parameters
  - Stage 2: Identify active inputs within active groups
- In Stage 1, group total effects are compared with total effects of "low-impact" inputs for selection of active groups
- In Stage 2, total effects of the individual inputs in selected groups are compared with total effects of "low-impact" inputs for selection of active inputs
- Selected active inputs are used in follow-on studies

Example: Halden Rod 1

- IFA-432 Test
  - 6-rod assembly irradiated in the Halden heavy boiling water reactor in Norway from 1975 to 1984
  - Test long-term steady-state performance of BWR-6 type fuel rods, operated at upper bound power levels for full-length commercial fuel rods
  - Centerline thermocouple in top and bottom end of fuel column
- FRAPCON Calculations
  - 61 parameters (17 design, 44 physics)
  - 4 outputs
    - Average of total fuel radius change (microns) for axial regions 1-4 at time steps 38 (y_1) and 44 (y_2)
    - Centerline temperature (Kelvin) for axial region 4 at time step 38 (y_3)
    - Fission gas release at time step 44 (y_4)

FRAPCON: Method of Morris Results

Number	Variable	Model	Outputs
1	rp_qc_coeff	Rod surface heat flux at elevation z on the rod axis	1,2,3,4
14	rp_ec_coeff	Clad emissivity	2
17	rp_rphonon_coeff	Expression in fthcon.f used in computing conductivity	1,2,3,4
19	rp_fm_coeff	Multiplier effect of porosity	1,2,3,4
20	rp_con_coeff	Fuel thermal conductivity	1,2,3,4
26	rp_celmod_coeff	Elasticity modulus	2
27	rp_cshear_coeff	Shear modulus	2
31	rp_delta_coeff	As fabricated fuel-cladding gap size	1,2,3
36	rp_fit_coeff	Multiplier in grain boundary accumulation model	4
37	rp_rns_coeff	Saturation area density gas	1,3
38	rp_dmultiplier_coeff	Burnup-enhancement factor of 14 applied to diffusion constant	1,2,3,4
39	rp_dconstant_coeff	Diffusion constant 2 1381<T<1650	1,2,4
50	dspg	Spring diameter	2

Selection of 13 active effects based on K-means clustering (K = 2) applied to (a) absolute means, and (b) standard deviations of samples from elementary effect distributions

FRAPCON: GSinCE Results

			Morris	GSinCE
Number	Variable	Model	Outputs	Outputs
1	rp_qc_coeff	Rod surface heat flux at elevation z on the rod axis	1,2,3,4	1,2,3,4
14	rp_ec_coeff	Clad emissivity	2	
17	rp_rphonon_coeff	Expression in fthcon.f used in computing conductivity	1,2,3,4	1,3,4
19	rp_fm_coeff	Multiplier effect of porosity	1,2,3,4	1,3,4
20	rp_con_coeff	Fuel thermal conductivity	1,2,3,4	1,3,4
26	rp_celmod_coeff	Elasticity modulus	2	
27	rp_cshear_coeff	Shear modulus	2	
31	rp_delta_coeff	As fabricated fuel-cladding gap size	1,2,3	2
36	rp_fit_coeff	Multiplier in grain boundary accumulation model	4	4
37	rp_rns_coeff	Saturation area density gas	1,3	
38	rp_dmultiplier_coeff	Burnup-enhancement factor of 14 applied to diffusion constant	1,2,3,4	1,4
39	rp_dconstant_coeff	Diffusion constant 2 1381<T<1650	1,2,4	1,4
50	dspg	Spring diameter	2	

Selection of 8 active effects based on groups defined by
(a) exploratory data analysis (EDA),
(b) subject matter expert (SME), and
(c) EDA within SME groups

EDA uses Fisher transformed Pearson correlation coefficients between columns of design and computer model output

FRAPCON: GSinCE — EDA Groups

Stage 1

Output	r_j^* range			Individual Inputs						Groups	Selection
y_1	-0.467	to	-0.398	19	17	20				g1	√
	-0.149	to	-0.109	36	10	9				g2	
	-0.093	to	-0.069	6	37	45	25			g3	
	-0.056	to	-0.042	41	60	59	56			g4	
	-0.038	to	-0.031	31	51	4	28			g5	
	-0.025	to	-0.021	21	13	34	54	11		g6	
	-0.018	to	-0.012	47	53	42	49	40		g7	
	-0.009	to	0.000	18	30	43	12	29	5	g8	
	0.005	to	0.013	24	2	55	61			g9	
	0.022	to	0.029	35	33	50				g10	
	0.030	to	0.050	22	14	26	48	7		g11	
	0.050	to	0.058	27	23	3	57			g12	
	0.062	to	0.081	58	46	8	15			g13	
	0.091	to	0.108	44	52	32	16			g14	
	0.164	to	0.304	38	39	1				g15	√
y_2			-1.942	31						g1	√
	-0.060	to	-0.054	43	54	6				g2	
	-0.045	to	-0.033	36	30	27				g3	
	-0.029	to	-0.023	3	4	9				g4	
	-0.019	to	-0.011	25	14	41	50	23		g5	
	-0.010	to	-0.006	12	56	5	58			g6	
	-0.003	to	-0.001	26	42	37	40	49		g7	
	0.000	to	0.003	29	19	34	61	55	22	g8	
	0.004	to	0.006	24	7	13				g9	
	0.010	to	0.017	15	2	18	17	16		g10	
	0.020	to	0.029	59	11	52	35	21	39	g11	
	0.032	to	0.034	46	48	60				g12	
	0.035	to	0.039	10	8	45	38			g13	
	0.040	to	0.048	57	28	51				g14	
	0.056	to	0.073	32	53	47	33	20		g15	
			0.241	1						g16	√

Output	r_j^* range			Individual Inputs						Groups	Selection
y_3	-0.524	to	-0.481	17	19	20				g1	√
	-0.108	to	-0.088	10	9					g2	
	-0.067	to	-0.060	6	51	47	25	45		g3	
	-0.057	to	-0.051	56	37	36				g4	
	-0.048	to	-0.042	41	60	21				g5	
	-0.037	to	-0.033	13	59	11				g6	
	-0.028	to	-0.023	4	42	34	53			g7	
	-0.017	to	-0.014	28	33	49				g8	
	-0.006	to	-0.003	29	24	5				g9	
			0.000	55	35	40				g10	
	0.004	to	0.007	18	30	2				g11	
	0.010	to	0.024	54	43	61	12			g12	
	0.027	to	0.035	50	22	46	27	48		g13	
	0.047	to	0.060	14	52	38	8			g14	
	0.062	to	0.069	26	7	58	3			g15	
	0.071	to	0.088	44	15	23	57			g16	
	0.097	to	0.111	32	16	39				g17	
	0.221	to	0.252	1	31					g18	√
y_4	-0.416	to	-0.371	19	17	20				g1	√
	-0.253	to	-0.137	36	37					g2	√
	-0.098	to	-0.073	10	25	33				g3	
	-0.062	to	-0.052	45	56	6	60			g4	
	-0.049	to	-0.042	59	41	47	28			g5	
	-0.030	to	-0.020	21	9	53	12	51		g6	
	-0.020	to	-0.014	11	13	2				g7	
	-0.009	to	-0.004	4	30	54	42	35	55	g8	
	0.000	to	0.010	49	40	46	5			g9	
	0.012	to	0.019	61	24	34	18	29		g10	
	0.024	to	0.027	14	48	8				g11	
	0.033	to	0.040	44	27	58	3			g12	
	0.041	to	0.044	22	7	43	50			g13	
	0.053	to	0.058	57	26	52				g14	
	0.072	to	0.095	23	15	16				g15	
	0.105	to	0.143	32	31					g16	
	0.224	to	0.417	1	38	39				g17	√

Output	Stage 1 Selection									Stage 2 Selection								
y_1	1	17	19	20				38	39	1	17	19	20					
y_2	1				31					1				31				
y_3	1	17	19	20	31					1	17	19	20					
y_4	1	17	19	20		36	37	38	39	1	17	19	20		36		38	39
Union	1	17	19	20	31	36	37	38	39	1	17	19	20	31	36		38	39

Analysis: H. Moon

Stage 2

FRAPCON: GSinCE — SME Groups

Individual inputs	Groups	Stage 1 Selection				Stage 2 Selection			
		y_1	y_2	y_3	y_4	y_1	y_2	y_3	y_4
1 2 3	g1	√	√	√	√	1	1	1	1
4 5 6 7 8 9	g2								
10 11 12 13 14 15	g3								
16 17 18 19 20 21	g4	√	√	√	√	17 19 20		17 19 20	17 19 20
22 23 24 25	g5								
26 27 28 29 30	g6								
31	g7		√				31		
32 33 34 35 36 37 38 39 40 41 42 43	g8				√	38 39			36 38 39
44	g9								
45 46 48 50 51 53 56 57 59 60 61	g10								
47 49 52 54 55 58	g11								

Analysis: H. Moon

SME groups have potential disadvantages:
(1) Reduction of Stage 1 selection efficiency due to the existence of large groups
(2) Mitigation of group effect through cancellation involving inputs having opposing functional relationships with output

FRAPCON: GSinCE — EDA Within SME Groups

input	expert group	Trans Corr y_1	Trans Corr y_3	sub group	new group
1	1	0.3042	0.2210	1-1	g1
2	1	0.0073	0.0069	1-2	g2
3	1	0.0562	0.0692	1-2	g2
4	2	-0.0344	-0.0280	2-1	g3
5	2	-0.0004	-0.0032	2-1	g3
6	2	-0.0932	-0.0665	2-1	g3
7	2	0.0495	0.0636	2-2	g4
8	2	0.0710	0.0599	2-2	g4
9	2	-0.1085	-0.0878	2-1	g3
10	3	-0.1099	-0.1077	3-1	g5
11	3	-0.0205	-0.0325	3-1	g5
12	3	-0.0039	0.0241	3-2	g6
13	3	-0.0219	-0.0365	3-1	g5
14	3	0.0335	0.0473	3-2	g6
15	3	0.0811	0.0763	3-2	g6
16	4	0.1075	0.0974	4-1	g7
17	4	-0.4604	-0.5240	4-2	g8
18	4	-0.0087	0.0036	4-1	g7
19	4	-0.4668	-0.4853	4-2	g8
20	4	-0.3977	-0.4813	4-2	g8
21	4	-0.0245	-0.0422	4-1	g7
22	5	0.0301	0.0284	5-1	g9
23	5	0.0523	0.0779	5-1	g9
24	5	0.0052	-0.0052	5-2	g10
25	5	-0.0687	-0.0615	5-2	g10
26	6	0.0346	0.0616	6-1	g11
27	6	0.0503	0.0342	6-1	g11
28	6	-0.0311	-0.0170	6-2	g12
29	6	-0.0015	-0.0062	6-2	g12
30	6	-0.0085	0.0052	6-2	g12
31	7	-0.0382	0.2523	7	g13
32	8	0.1073	0.0965	8-1	g14
33	8	0.0256	-0.0164	8-2	g15
34	8	-0.0215	-0.0257	8-2	g15
35	8	0.0219	-0.0001	8-2	g15
36	8	-0.1486	-0.0507	8-3	g16
37	8	-0.0919	-0.0564	8-3	g16
38	8	0.1635	0.0553	8-1	g14
39	8	0.2157	0.1112	8-1	g14
40	8	-0.0122	0.0000	8-2	g15
41	8	-0.0556	-0.0480	8-3	g16
42	8	-0.0155	-0.0276	8-2	g15
43	8	-0.0044	0.0169	8-2	g15
44	9	0.0906	0.0705	9	g17
45	10	-0.0725	-0.0602	10-1	g18
46	10	0.0642	0.0294	10-2	g19
48	10	0.0437	0.0352	10-2	g19
50	10	0.0291	0.0266	10-2	g19
51	10	-0.0345	-0.0646	10-1	g18
53	10	-0.0178	-0.0233	10-1	g18
56	10	-0.0422	-0.0565	10-1	g18
57	10	0.0584	0.0882	10-2	g19
59	10	-0.0487	-0.0350	10-1	g18
60	10	-0.0538	-0.0472	10-1	g18
61	10	0.0126	0.0223	10-2	g19
47	11	-0.0180	-0.0627	11-1	g20
49	11	-0.0149	-0.0143	11-1	g20
52	11	0.0990	0.0536	11-2	g21
54	11	-0.0210	0.0102	11-1	g20
55	11	0.0121	-0.0003	11-2	g21
58	11	0.0622	0.0688	11-2	g21

Individual inputs						Groups	Stage 1 Selection				Stage 2 Selection			
							y_1	y_2	y_3	y_4	y_1	y_2	y_3	y_4
1						g1	√	√	√	√	1	1	1	1
2	3					g2								
4	5	6	9			g3								
7	8					g4								
10	11	13				g5								
12	14	15				g6								
16	17	18				g7	√	√	√	√	17		17	17
19	20	21				g8	√	√	√	√	19 20		19 20	19 20
22	23					g9								
24	25					g10								
26	27					g11								
28	29	30				g12								
31						g13		√				31		
32	38	39				g14	√			√	38			38 39
33	34	35	40	42	43	g15								
36	37	41				g16	√			√				36
44						g17								
45	51	53	56	59	60	g18								
46	48	50	57	61		g19								
47	49	54				g20								
52	55	58				g21								

Analysis: H. Moon

In this example, all 3 grouping methods result in the same Stage 2 selection of active inputs: 1, 17, 19, 20, 31, 36, 38, and 39

References

Morris, M. D. (1991). "Factorial sampling plans for preliminary computational experiments," *Technometrics*, 33, 161-174.

Moon, H., Dean, A.M., and Santner, T.J. (2012). "Two-stage sensitivity-based group screening in computer experiments," *Technometrics*, 54, 376-387.

Schonlau, M. and Welch, W. J. (2006). "Screening the input variables to a computer model via analysis of variance and visualization." In *Screening: Methods for Experimentation in Industry, Drug Discovery, and Genetics*, A. M. Dean and S. Lewis (eds.), 308-327, Springer, New York.

Linkletter, C., Bingham, D., Hengartner, N., Higdon, D., and Ye, K. Q. (2006). "Variable selection for Gaussian process models in computer experiments," *Technometrics*, 48, 478-490.

Sequential Experiment Design

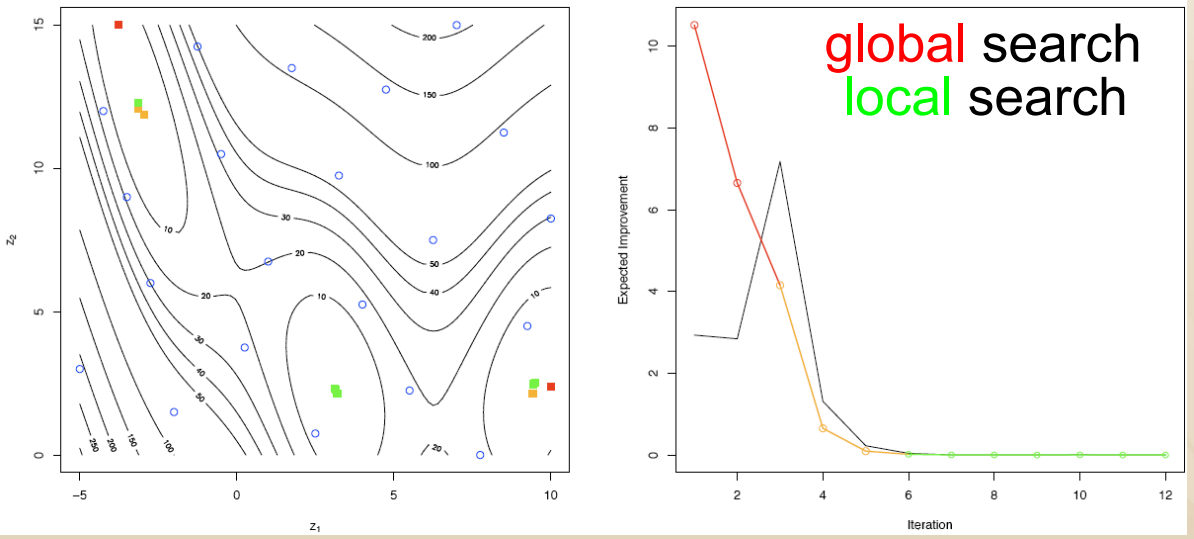
Introduction

- Sequential experiment design is utilized to more efficiently pursue an objective
  - Optimization
  - Global prediction
  - Calibration/discrepancy inference
  - Contour estimation
  - Quantile estimation
- How does it work?
  - Analyze runs from initial design
  - Propose additional runs, perhaps in batches, using results from the initial design and a design criterion tailored to the objective
  - Continue until the budget of runs is expended
    - Generally at least 30% of budget is spent on the initial design

Optimization

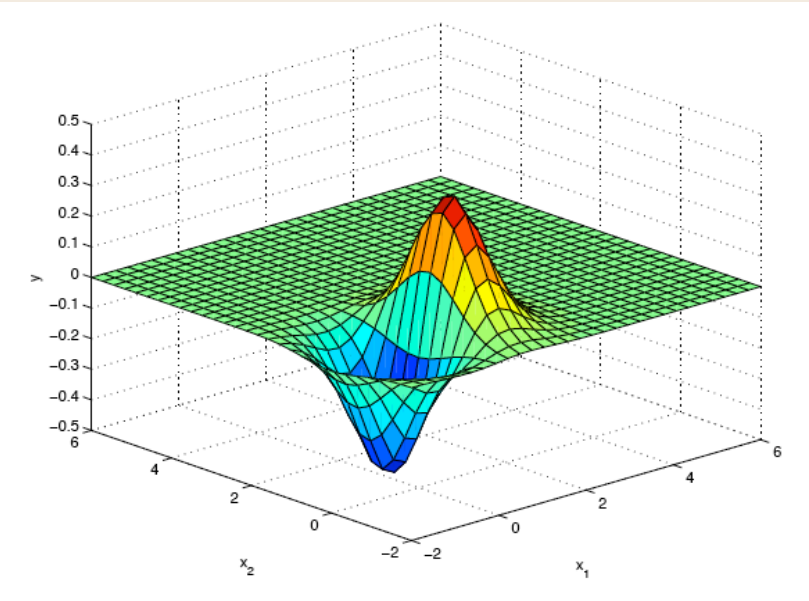
- Sequentially collect additional runs to minimize computer model output
  - Based on an *expected improvement* criterion
  - Balances prediction uncertainty (global) and local optimization
- Facilitates baselining of complex computer models
  - Can be embedded in other algorithms that guarantee convergence (e.g. pattern search)
- Can be applied when some inputs are environmental (have pdfs)

Expected Improvement For Global Optimization (Baselining)

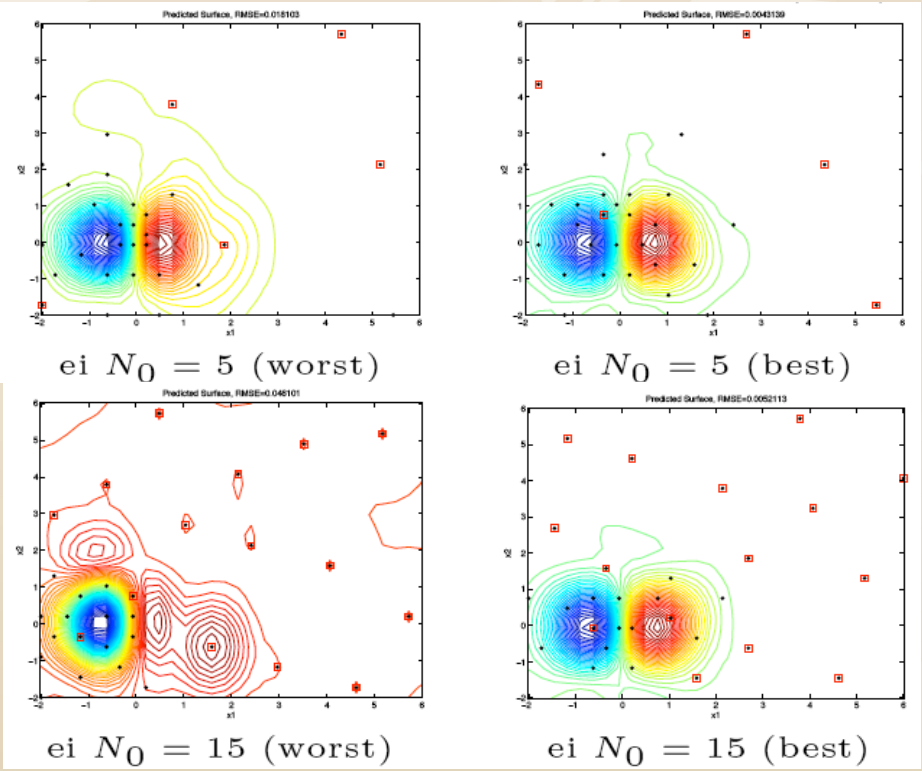


Global Prediction

- Sequentially collect additional runs to efficiently improve global prediction capability
  - Based on an *expected improvement* criterion
  - Balances prediction *uncertainty* and *bias*
- Facilitates control of absolute or relative prediction error



Example: Eric Lam and Bill Notz



Rare Event Estimation

- Interested in rare event estimation
  - Outputs obtained from computational model
  - Uncertainties in operating conditions and physics variables
  - Physics variables calibrated wrt reference experimental data
- In particular, quantile or percentile estimation

$$Pr[\,\eta(\boldsymbol{x},\boldsymbol{\theta}) > q_\alpha\,] = \alpha$$

  - One of q_α or α is specified and the other is to be inferred
  - q_α may be random when inferring α
- Sequential importance sampling for improved inference
  - Oversample region of parameter space producing rare events of interest
  - Sequentially refine importance distributions for improved inference

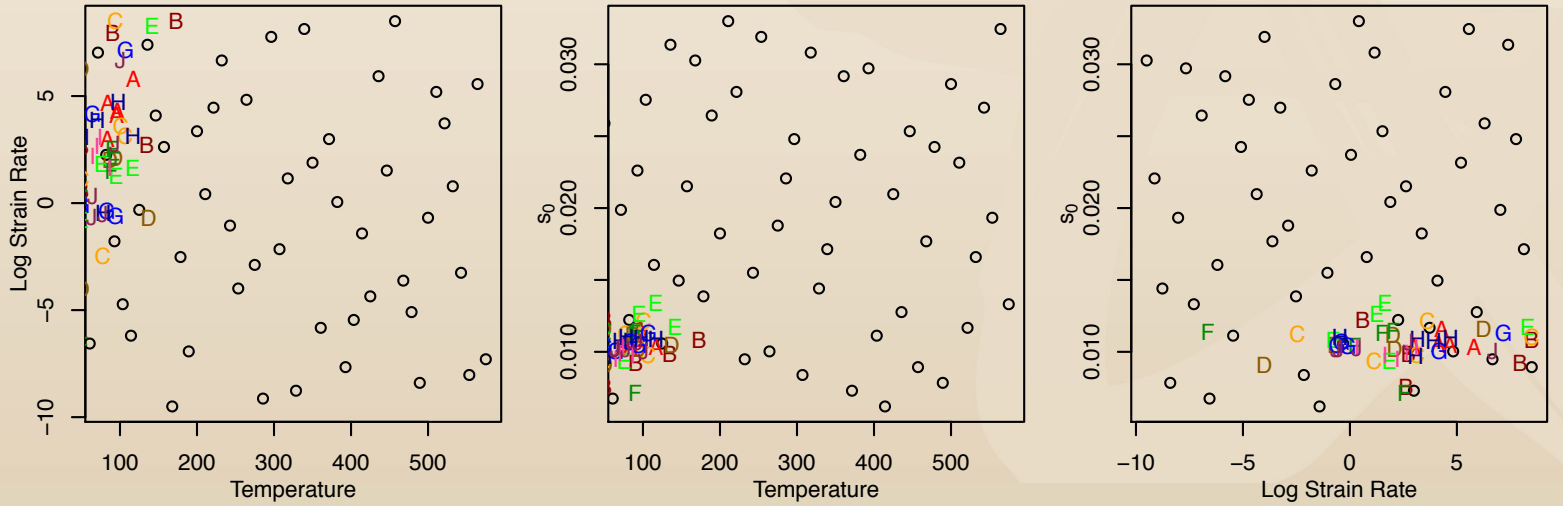
Targeted Sequential Design

- Choose design augmentation that minimizes integrated mean square error with respect to the currently estimated importance distributions for sensitive parameters
  - A version of "targeted" IMSE (tIMSE)

$$IMSE(\mathcal{D}_b) = 1 - \text{trace}\left(\left[\int \boldsymbol{r}_{\{\mathcal{D}_0,\mathcal{D}_b\}}(\boldsymbol{z};\boldsymbol{\beta})\, \boldsymbol{r}^T_{\{\mathcal{D}_0,\mathcal{D}_b\}}(\boldsymbol{z};\boldsymbol{\beta}) \prod_{i=1}^{n_z} w_i(z_i)\, d\boldsymbol{z} \right] \boldsymbol{R}^{-1}_{\{\mathcal{D}_0,\mathcal{D}_b\}}(\boldsymbol{\beta}) \right)$$

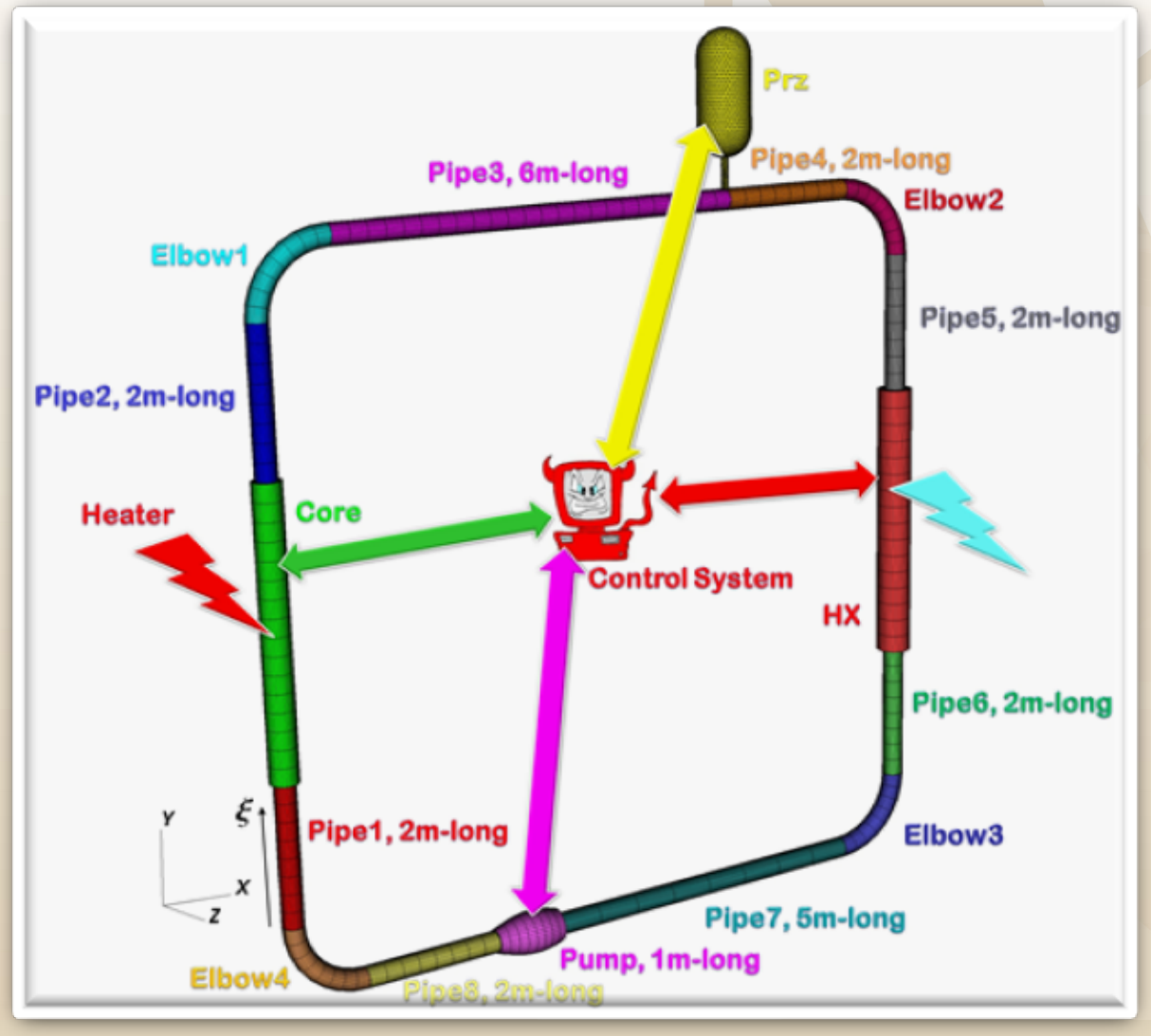
Estimate correlation parameters based on current design $\mathcal{D}_0$

For sensitive input *i*, select $w_i(z_i)$ to be its importance distribution



Example: VR2plus Model

- Scenario: Pressurizer failure, followed by pump trip and initiation of SCRAM (insertion of control rods)
- Goal: Understand behavior of peak coolant temperature (PCT) in the reactor
- Interested in probability that PCT exceeds 700^{o} K



VR2plus Details

- Single thermal-hydraulics loop with 21 components
- Working coolant is water at 16MPa and 600^{o} K, single-phase flow
- Nominal power output of this reactor is 15MW
- Calculations performed with reactor safety analysis code R7 (INL)

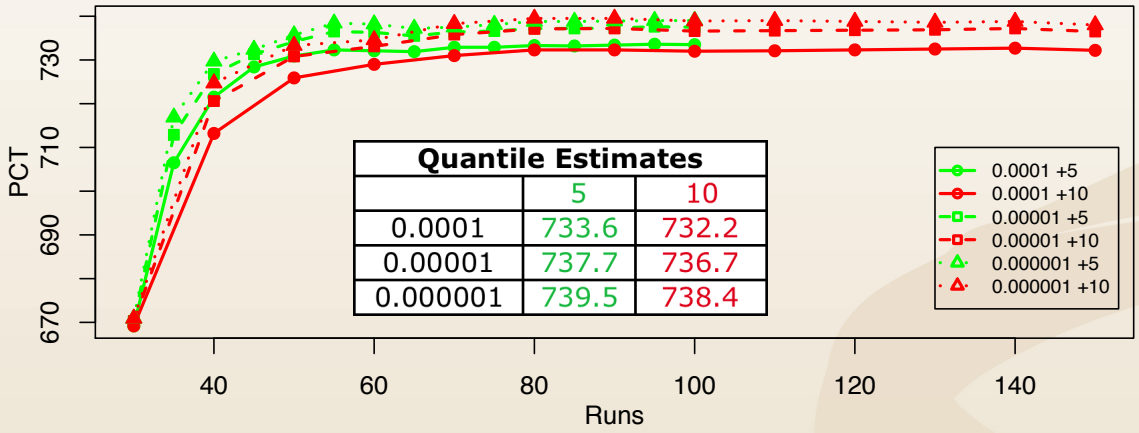
Input Parameter	Min	Max	Description
PumpTripPre	15.6 MPa	15.7 MPa	Min. pump pressure causing trip
PumpStopTime	10 s	100 s	Relaxation time of pump phase-out
PumpPow	0.0	0.4	Pump end power
SCRAMtemp	625^{o} K	635^{o} K	Max. temp. causing SCRAM
CRinject	0.025	0.24	Position of CR at end of SCRAM
CRtime	10 s	50 s	Relaxation time of CR system

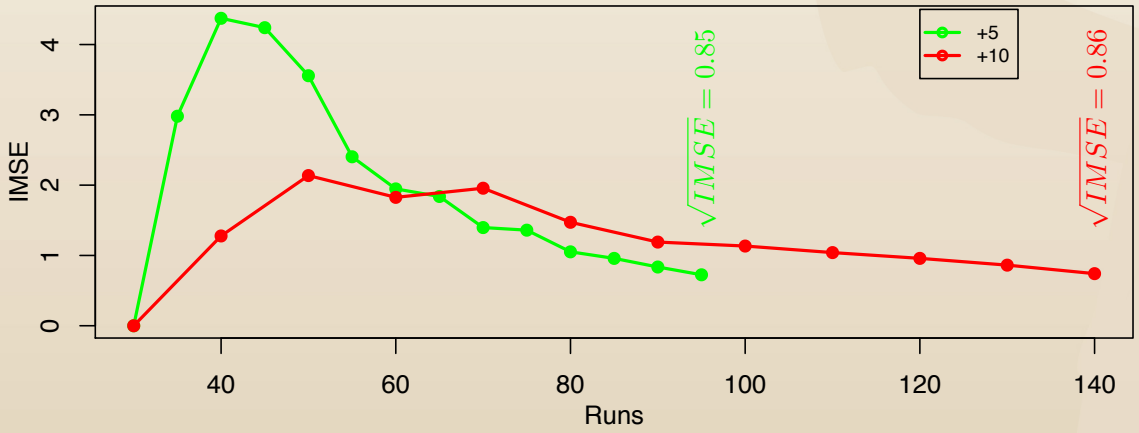
Input parameters assigned independent Uniform distributions on their ranges

VR2plus Analysis

- Quantile inference
  - Target 0.0001 quantile of PCT distribution
- Percentile inference
  - PCT > 700^{o} K
- Importance distribution
  - Independent Beta distributions for sensitive parameters
  - Independent Uniform distributions for insensitive parameters
- PumpPow and CRinject are sensitive

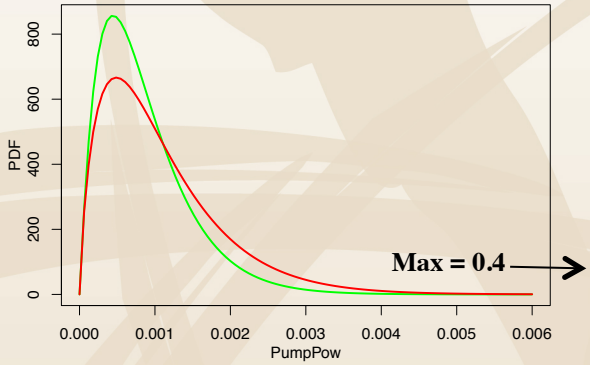
Pressurizer Failure: Quantile Inference

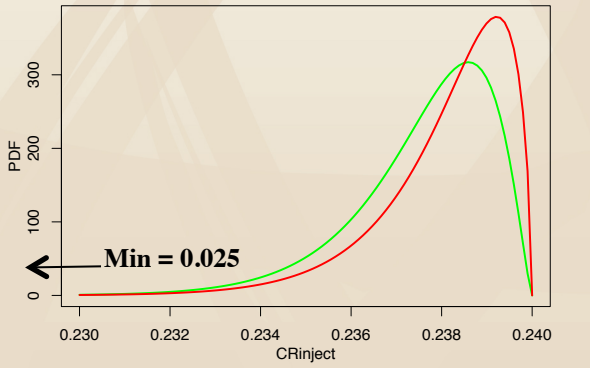




Again, convergence benefits seen with higher frequency updates

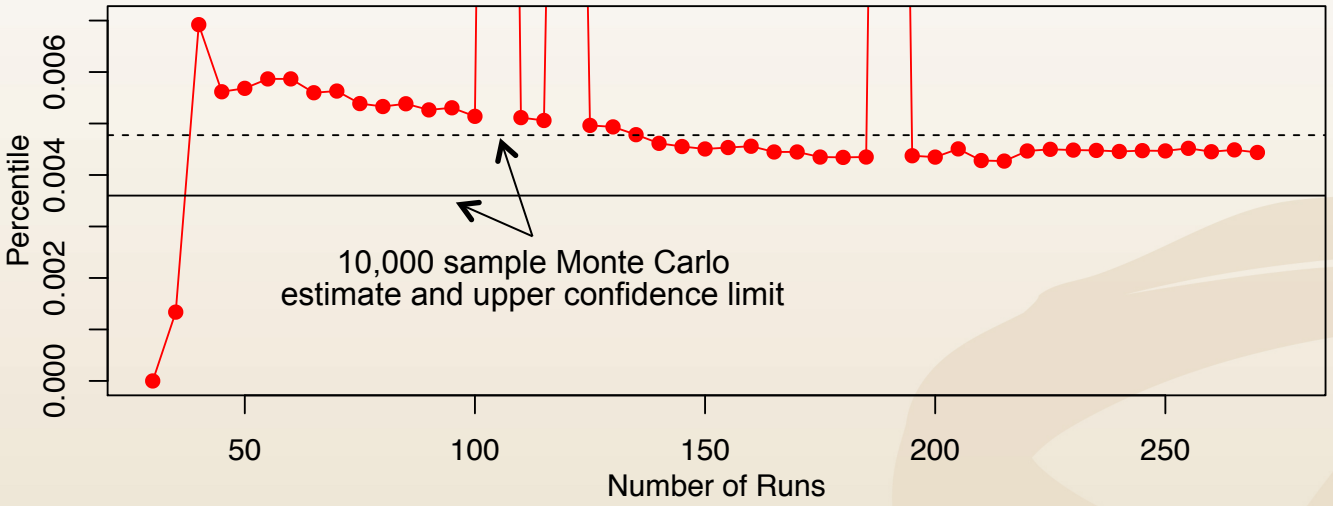
Importance Distributions

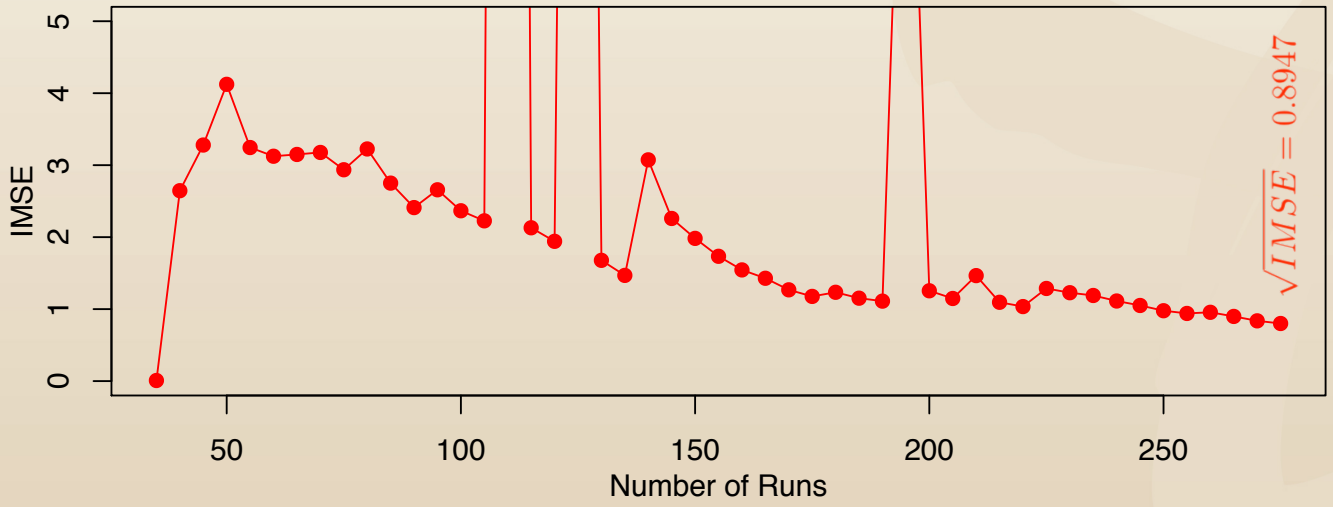


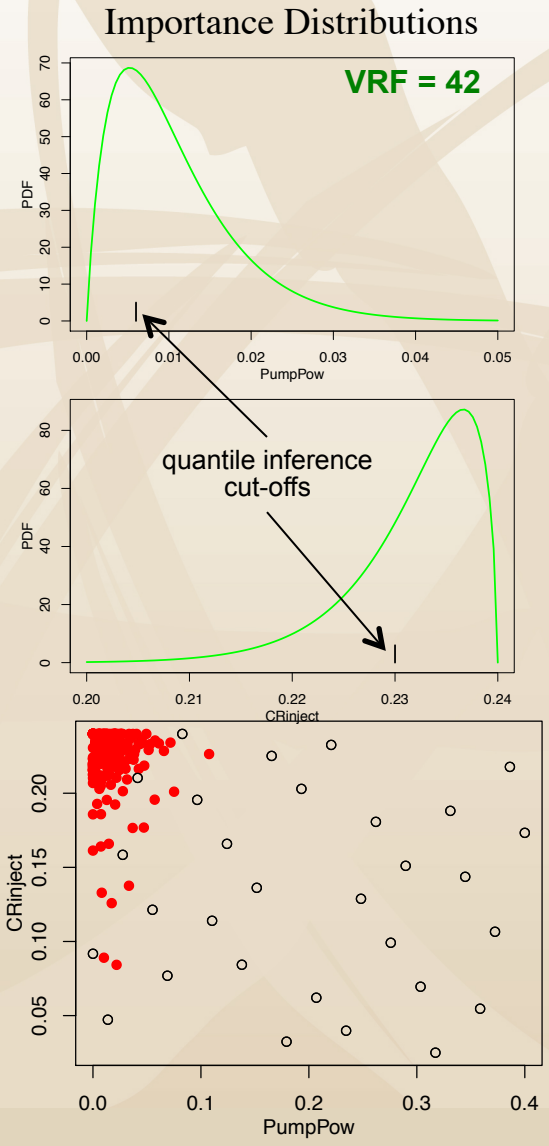


Variance Factors		
	5	10
0.0001	1737	4994
0.00001	2953	7536
0.000001	1749	3159

Pressurizer Failure: Percentile Inference







Additional runs required to reduce IMSE in greater volume of input space

Conclusions

- Space-filling designs work well with the GP models typically used in computer model emulation
- If given a budget of code runs/experiments, single-stage space-filling design generally results in prediction performance no worse than sequential design
  - exceptions can occur with non-stationary output behavior
- Sequential design is particularly efficient for optimization and rare event inference problems

References

Jones, D. R., Schonlau, M., and Welch, W. J. (1998). "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, 13, 455-492.

Lam, C. Q. and Notz, W. I. (2008). "Sequential adaptive designs in computer experiments for response surface model fit," *Statistics and Applications*, 6, 207-233.

Loeppky, J. L., Moore, L. M. and Williams, B. J. (2010). "Batch sequential designs for computer experiments," *Journal of Statistical Planning and Inference*, 140, 1452-1464.

Picard, R. and Williams, B. (2013). "Rare event estimation for computer models," *The American Statistician*, 67, 22-32.

Ranjan, P., Bingham, D., and Michailidis, G. (2008). "Sequential experiment design for contour estimation from complex computer codes," *Technometrics*, 50, 527-541.