Concept Test

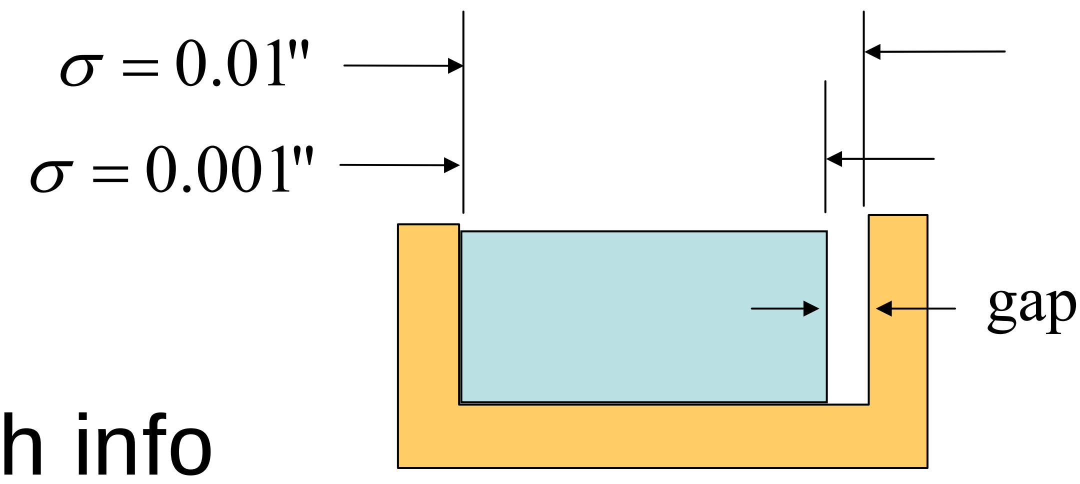
- A bracket holds a component as shown. The dimensions are <u>independent</u> random variables with standard deviations as noted. Approximately what is the standard deviation of the gap?

  A) 0.011”

  B) 0.01”

  C) 0.001”

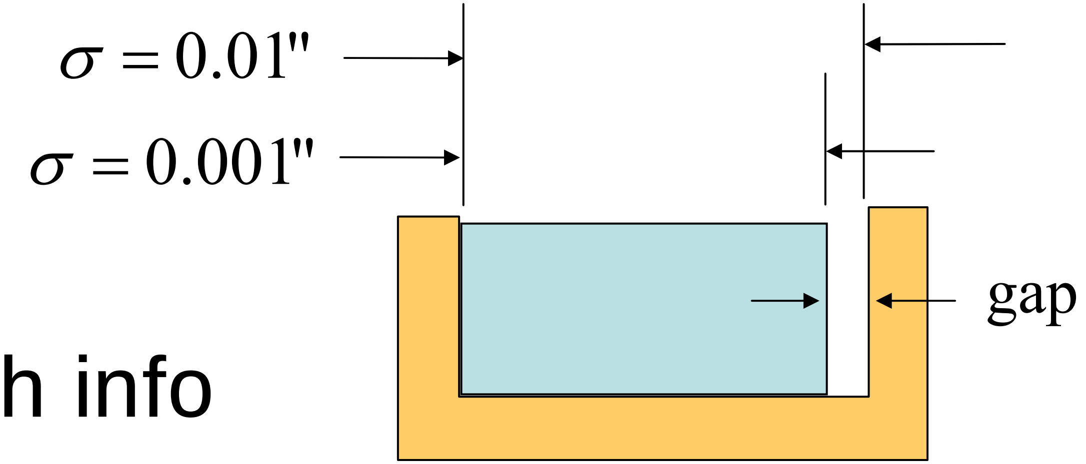
  D) not enough info



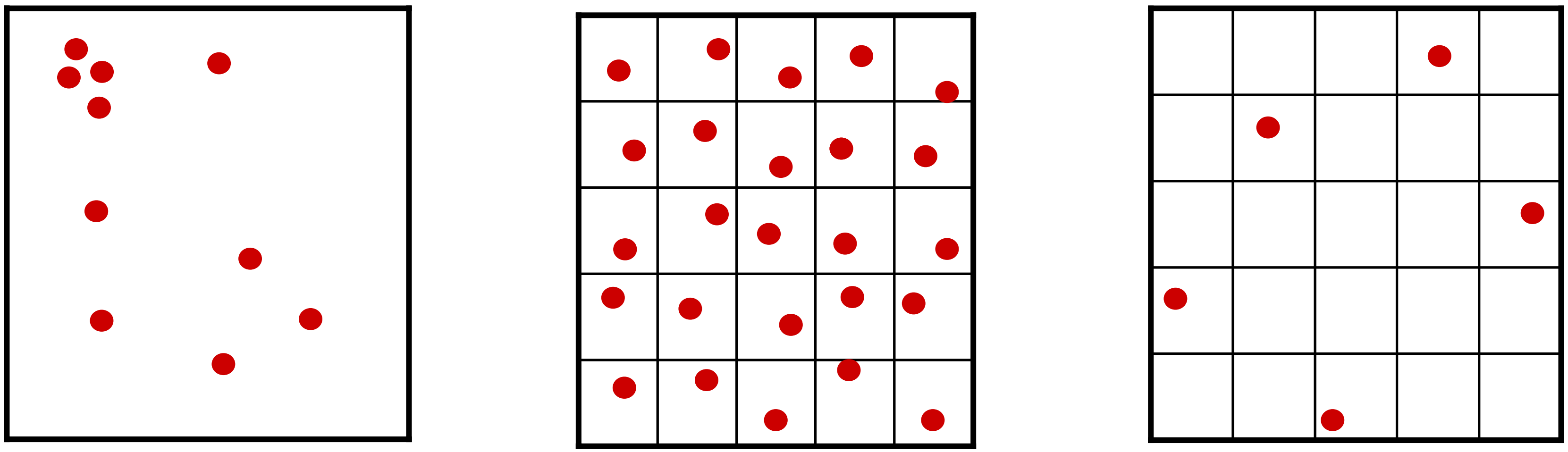
Concept Test

- A bracket holds a component as shown. The dimensions are strongly correlated random variables with standard deviations as noted. Approximately what is the standard deviation of the gap?

A) 0.011”
B) 0.01”
C) 0.009”
D) not enough info



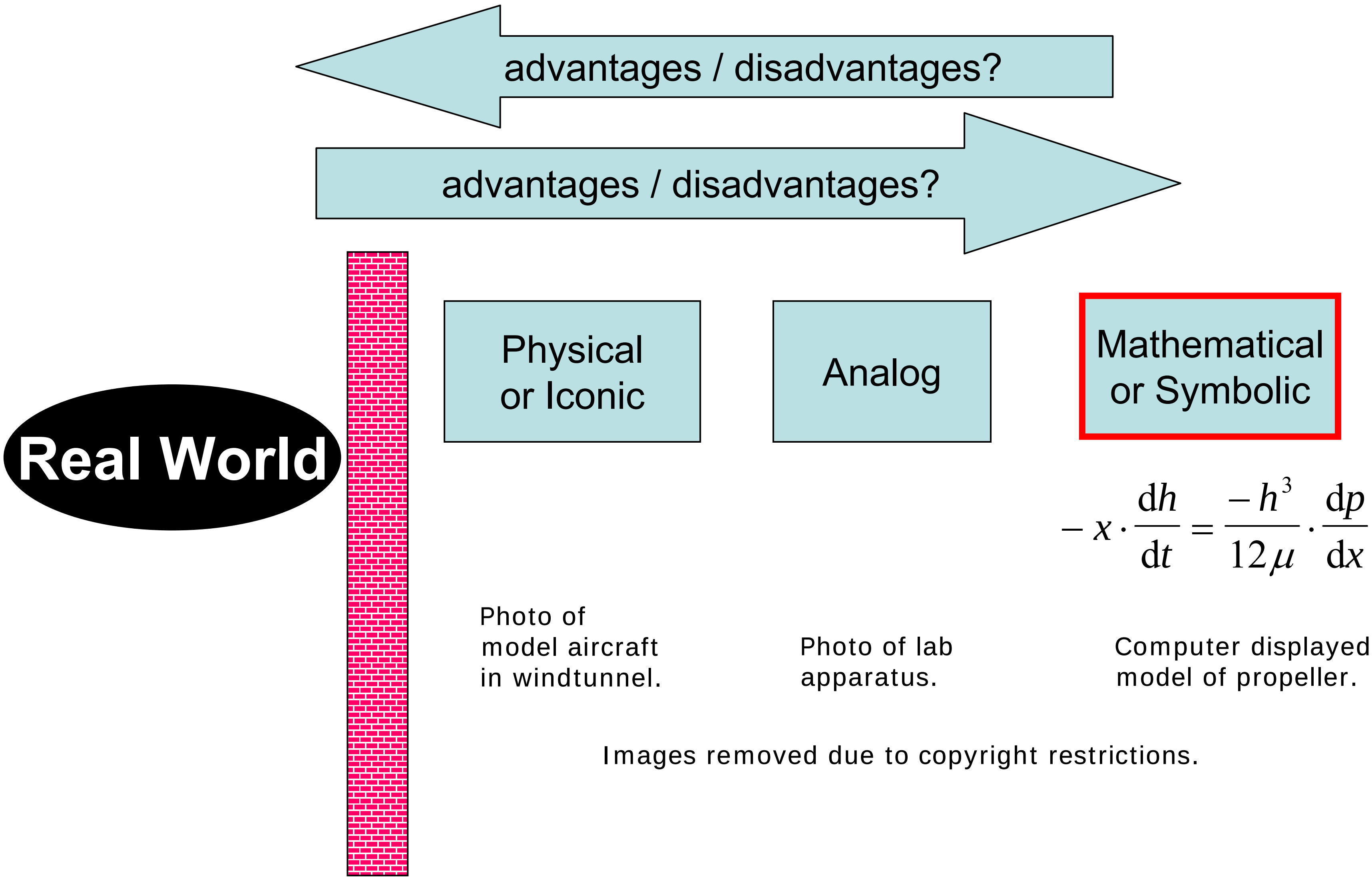
Design of Computer Experiments



Dan Frey

Associate Professor of Mechanical Engineering and Engineering Systems

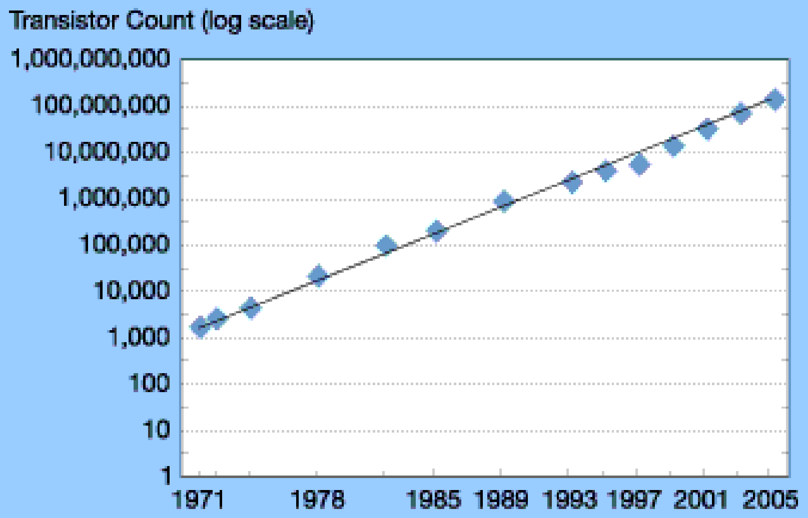
Classification of Models



Mathematical Models Are Rapidly Growing in Power

- Moore's Law – density ↑ 2X / 18 months
- Better algorithms being developed

Figure 8-1.
Moore's Law: 1971–2005



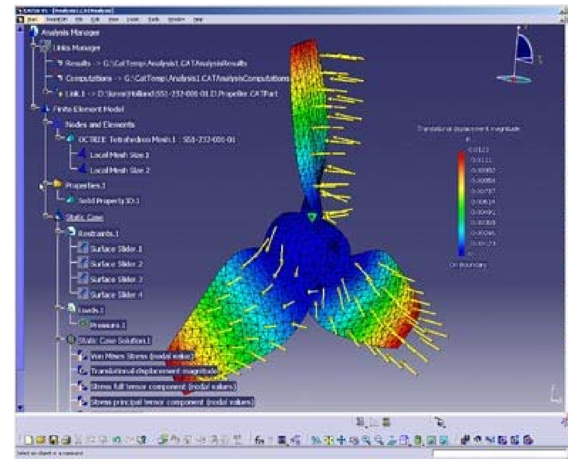
NOTES: The line on the graph represents the trend that defines Moore's Law. The data points reflect actual (1971–2001) and projected (2003–2005) data.

See appendix table 8-1. *Science & Engineering Indicators – 2002*

Source: NSF

Mathematical Models are Becoming Easier to Use

- A wide range of models are available
  - Finite Element Analysis
  - Computational fluid dynamics
  - Electronics simulations
  - Kinematics
  - Discrete event simulation



- Sophisticated visualization & animation make results easier to communicate
- Many tedious tasks are becoming automated (e.g., mesh generation and refinement)

Computational Complexity and Moore's Law

- Consider a problem that requires 3^n flops
- World's fastest computer ~ 36 Teraflops/sec
- In a week, you can solve a problem where

  n=log(60*60*24*7*36*10^{12})/log(3)=40
- If Moore's Law continues for 10 more years

  n=log($2^{10/1.5}$*60*60*24*7*36*10^{12})/log(3)=44
- We will probably not reach n=60 in my lifetime

Outline

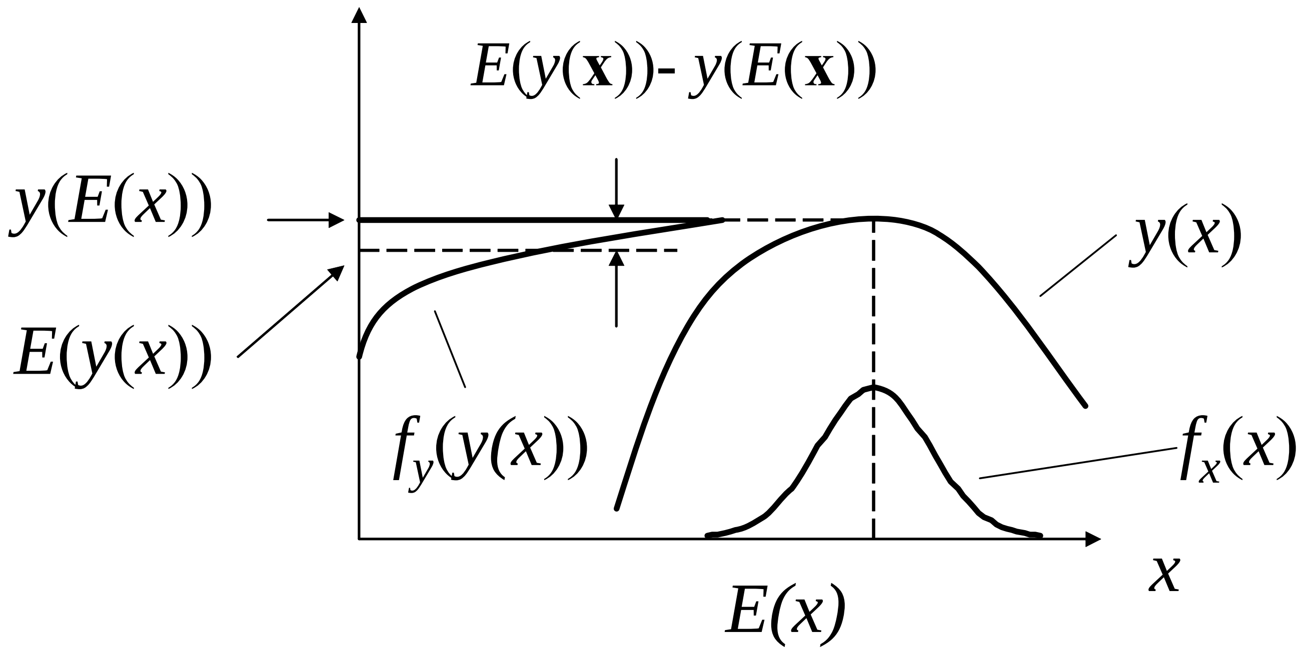
- Motivation & context
- Techniques for “computer experiments”
  - Monte Carlo
  - Importance sampling
  - Latin hypercube sampling
  - Hammersley sequence sampling
  - Quadrature and cubature
- Some cautions

Need for Computer Experiments

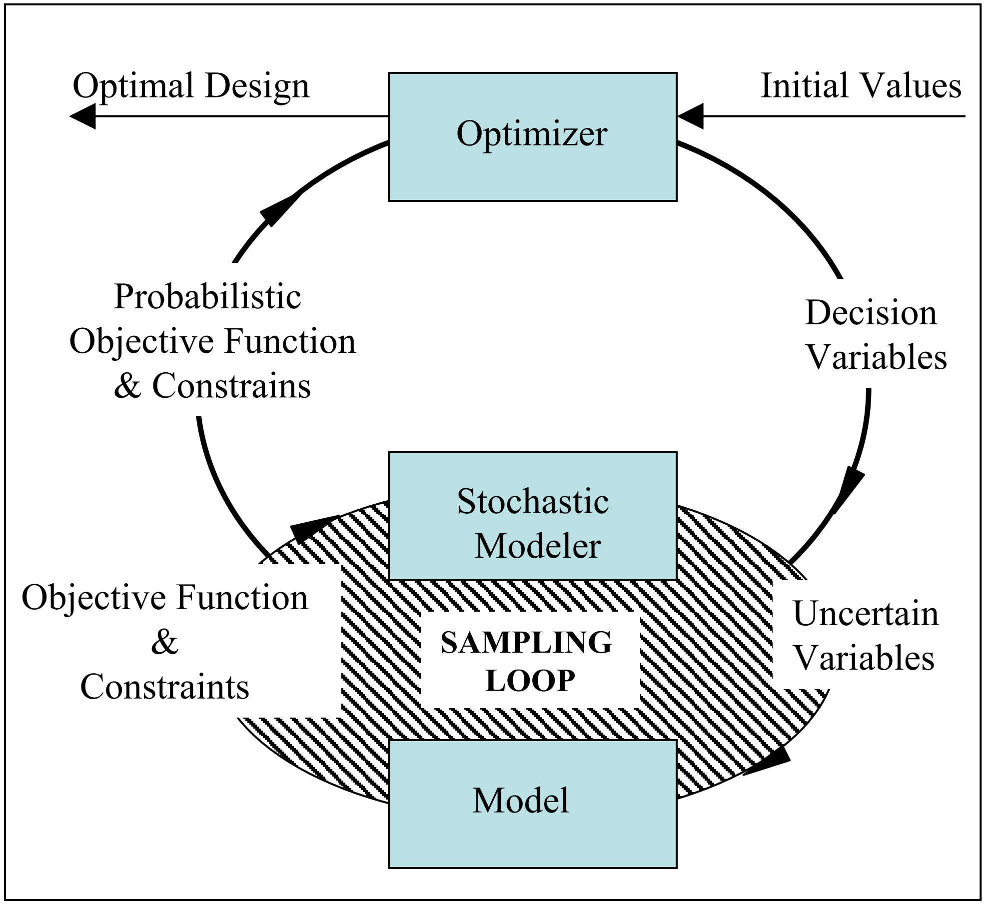
- There are properties of engineering systems we want to affect via our design / policy
- Let's call these properties a function $y(\mathbf{x})$ where $\mathbf{x}$ is a vector random variables
- Often y is a estimated by a computer simulation of a system
- We may want to <u>know</u> some things such as $\mathrm{E}(y(\mathbf{x}))$ or $\sigma(y(\mathbf{x}))$
- We often want to <u>improve upon</u> those same things
- This is deceptively complex

Expectation of a Function

$$E(y(\mathbf{x})) \neq y(E(\mathbf{x}))$$



Resource Demands of System Design



- The resources for system design typically scale as the <u>product</u> of the iterations in the optimization and sampling loops

Adapted from Diwekar U.M., 2003, "A novel sampling approach to combinatorial optimization under uncertainty" *Computational Optimization and Applications* 24 (2-3): 335-371.

Outline

- Motivation & context
- Techniques for “computer experiments”
  - Monte Carlo
  - Importance sampling
  - Latin hypercube sampling
  - Hammersley sequence sampling
  - Quadrature and cubature
- Some cautions

Monte Carlo Method

- Let's say there is a function $y(\mathbf{x})$ where $\mathbf{x}$ is a vector random variables
- Create samples $\mathbf{x}^{(i)}$
- Compute corresponding values $y(\mathbf{x}^{(i)})$
- Study the population to obtain estimates and make inferences
  - Mean of $y(\mathbf{x}^{(i)})$ is an unbiased estimate of $\mathrm{E}(y(\mathbf{x}))$
  - Stdev of $y(\mathbf{x}^{(i)})$ is an unbiased estimate of $\sigma(y(\mathbf{x}))$
  - Histogram of $y(\mathbf{x}^{(i)})$ approaches the pdf of $y(\mathbf{x})$

Fishman, George S., 1996, *Monte Carlo: Concepts, Algorithms, and Applications*, Springer.

Example: A Chemical Process

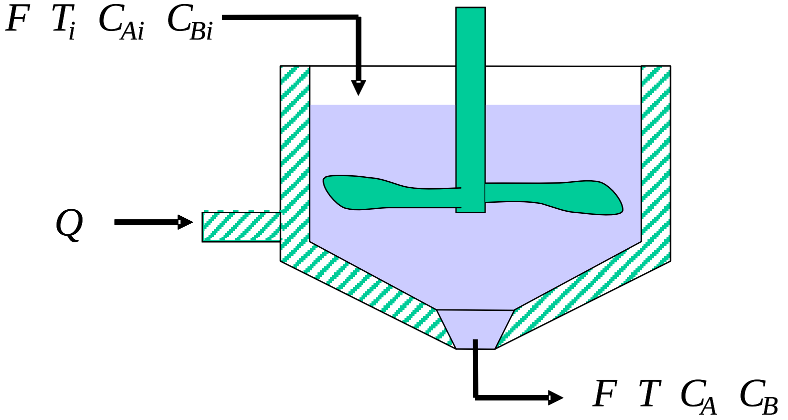
- Objective is to generate chemical species B at a rate of 60 mol/min

$$Q = F\rho C_p (T - T_i) + V(r_A H_{RA} + r_B H_{RB})$$

$$C_A = \frac{C_{Ai}}{1 + k_A^0 e^{-E_A/RT}\tau} \quad C_B = \frac{C_{Bi} + k_A^0 e^{-E_A/RT}\tau C_A}{1 + k_B^0 e^{-EB/RT}\tau}$$

$$-r_A = k_A^0 e^{-E_A/RT} C_A$$

$$-r_B = k_B^0 e^{-E_B/RT} C_B - k_A^0 e^{-E_A/RT} C_A$$



Adapted from Kalagnanam and Diwekar, 1997, "An Efficient Sampling Technique for Off-Line Quality Control", *Technometrics* (39 (3) 308-319.

Monte Carlo Simulations

What are They Good at?

$$\text{Accuracy} \propto \frac{1}{\sqrt{N}} \qquad N \equiv \#\text{Trials}$$

- Above formulae apply regardless of dimension
- So, Monte Carlo is good for:
  - Rough approximations <u>or</u>
  - Simulations that run quickly
  - <u>Even if</u> the system has many random variables

Fishman, George S., 1996, *Monte Carlo: Concepts, Algorithms, and Applications*, Springer.

Monte Carlo vs Importance Sampling

Monte Carlo

$$E\left(y(\mathbf{x})\right) = \int_{\Omega} y(\mathbf{x})\, f_{\mathbf{x}}(\mathbf{x})\, d\mathbf{x}$$

$\mathbf{X}^{(1)},\ldots,\mathbf{X}^{(n)}$ denote independent random vectors sampled from $f_{\mathbf{x}}(\mathbf{x})$

$\frac{1}{n}\sum_{i=1}^{n} y\left(\mathbf{X}^{(i)}\right)$ is an unbiased estimator of $E\left(y(\mathbf{x})\right)$

Importance Sampling

$$E\left(y(\mathbf{x})\right) = \int_{\Omega} \frac{y(\mathbf{x})\, f_{\mathbf{x}}(\mathbf{x})}{f^{+}{}_{\mathbf{x}}(\mathbf{x})}\, f^{+}{}_{\mathbf{x}}(\mathbf{x})\, d\mathbf{x}$$

$\mathbf{X}^{(1)},\ldots,\mathbf{X}^{(n)}$ denote independent random vectors sampled from $f^{+}{}_{\mathbf{x}}(\mathbf{x})$

$\frac{1}{n}\sum_{i=1}^{n} \frac{y\left(\mathbf{X}^{(i)}\right) f_{\mathbf{x}}\left(\mathbf{X}^{(i)}\right)}{f^{+}{}_{\mathbf{x}}\left(\mathbf{X}^{(i)}\right)}$ is an unbiased estimator of $E\left(y(\mathbf{x})\right)$

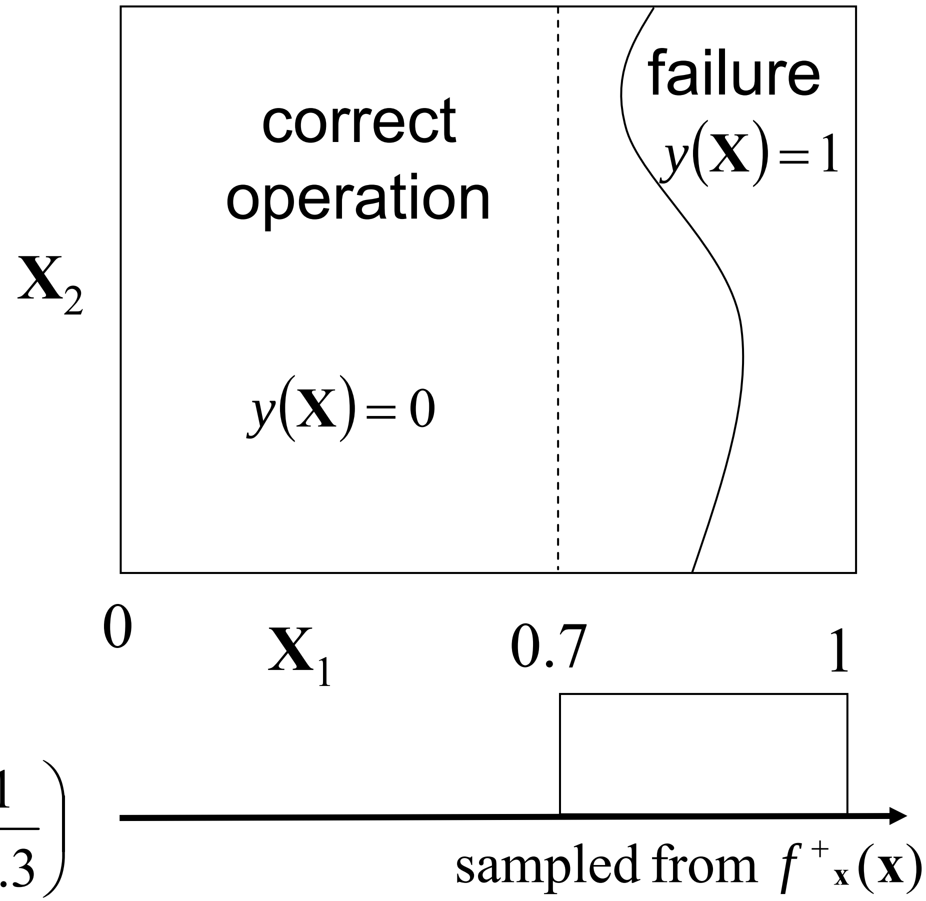
Importance Sampling Example

The variables $\mathbf{X}_1$ and $\mathbf{X}_2$ are uniformly distributed within the indicated rectangle.

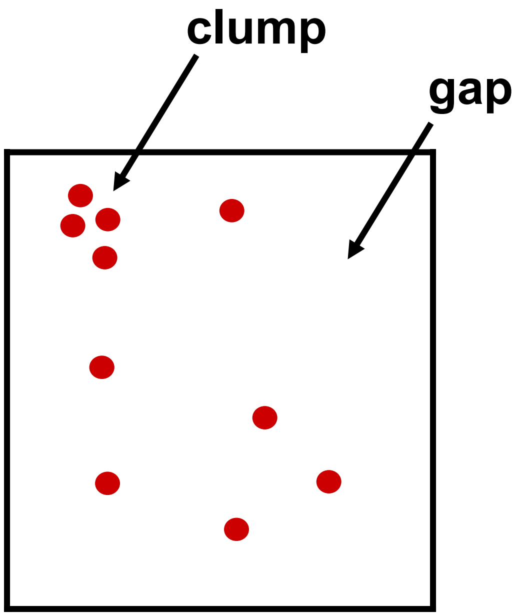
The physics of the problem suggests that the failure mode boundary is more likely somewhere in the right hand region.

Sample only on the right but weight them to correct for this.

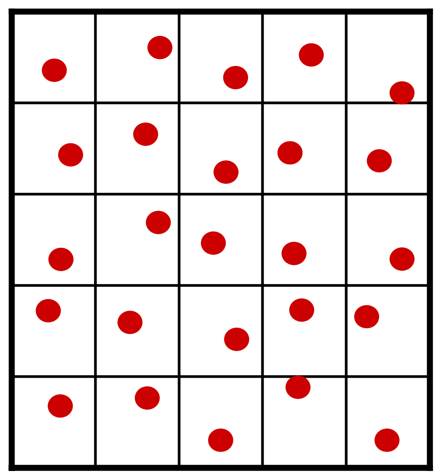
$$\frac{1}{n}\sum_{i=1}^{n} y\left(\mathbf{X}^{(i)}\right)\cdot\left(\frac{1}{0.3}\right)$$



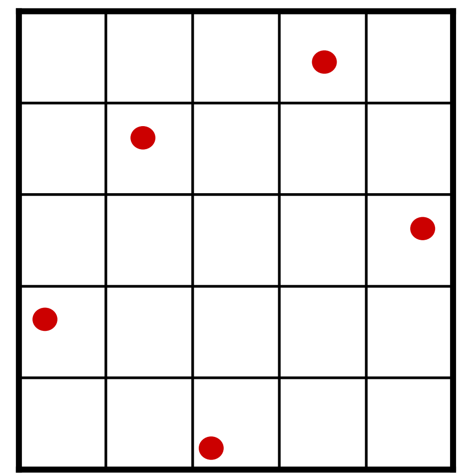
Sampling Techniques for Computer Experiments



Random Sampling

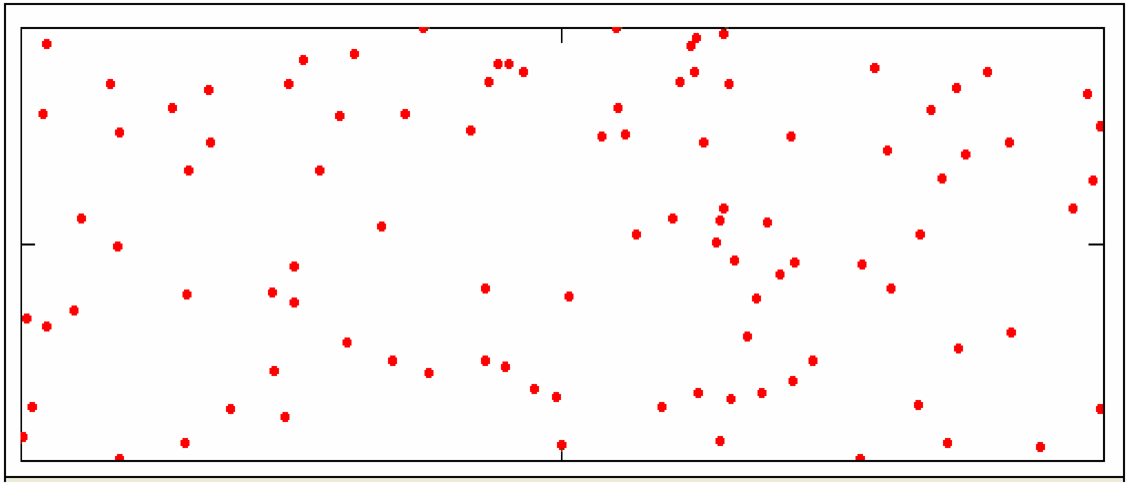


Stratified Sampling

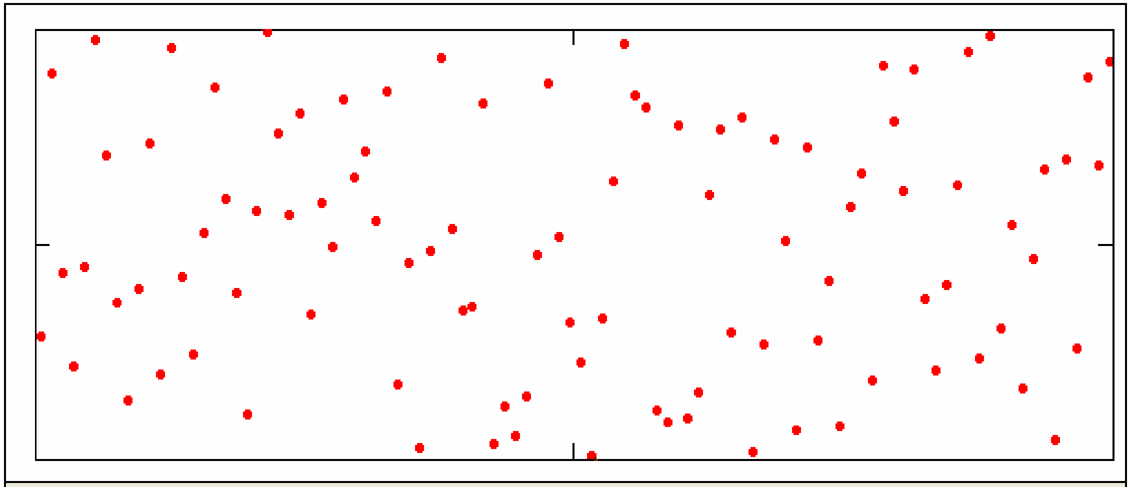


Latin Hypercube Sampling

Latin Hypercube Sampling



100 Monte Carlo Samples

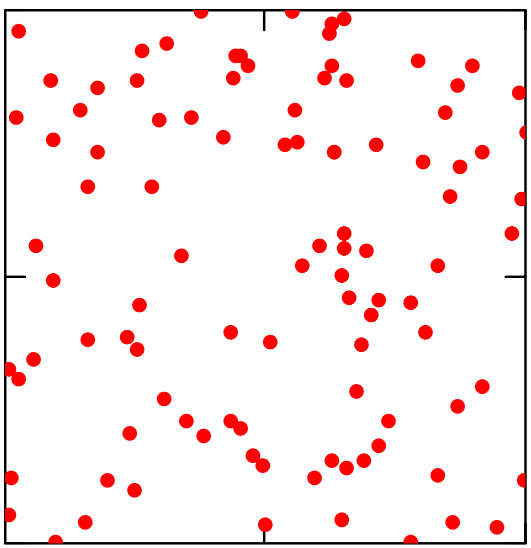


100 Latin Hypercube Samples

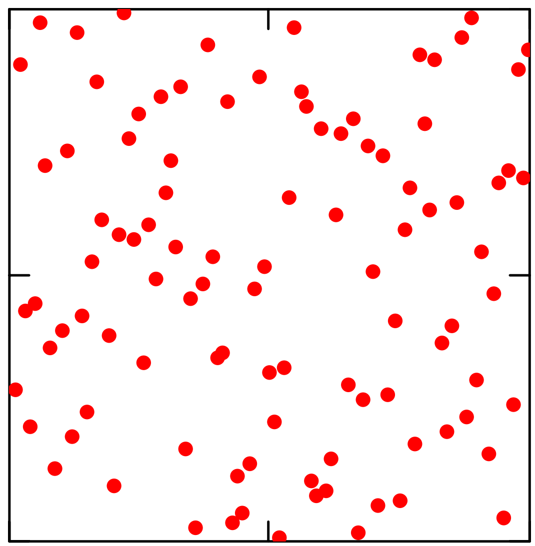
McKay, Beckman, and Conover, [1979, *Technometrics*] proved that LHS converges more quickly than MCS assuming monotonicity of the response.

Hammersley Sequence Sampling

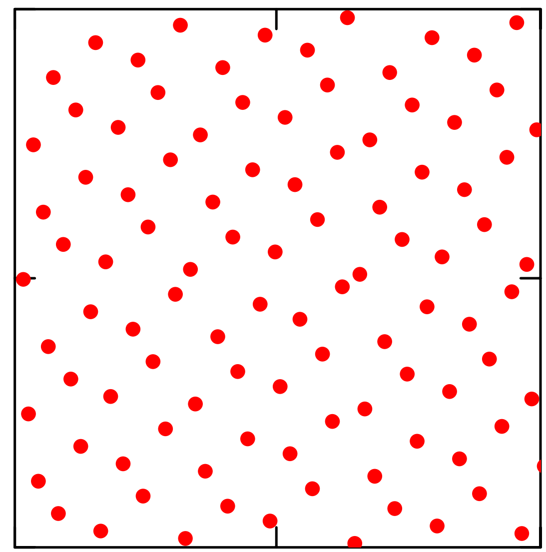
- A sampling scheme design for low "discrepancy"
- Demonstrated to converge to 1% accuracy 3 to 40 times more quickly than LHS [Kalagnanam and Diwekar, 1997]



Monte Carlo

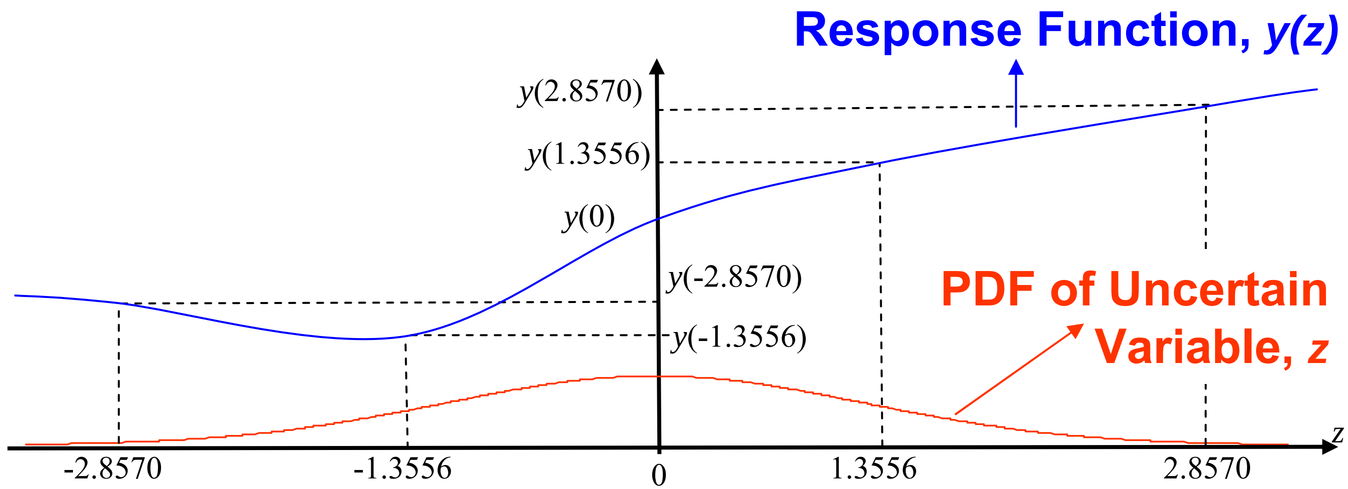


Latin Hypercube



Hammersley

Five-point Gaussian Quadrature Integration



Mean of Response

$$E(y(z)) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2} y(z)dz \approx$$

$$y(0) + \frac{1}{\sqrt{\pi}} \begin{bmatrix} A_1[y(1.3556) - y(0)] + A_1[y(-1.3556) - y(0)] + \\ A_2[y(2.8570) - y(0)] + A_2[y(-2.8570) - y(0)] \end{bmatrix}$$

A_1=0.39362, and A_2=0.019953

Variance of Response

$$E\left((y(z) - E(y(z)))^2\right) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2} (y(z) - E(y(z)))^2 dz \approx$$

$$[y(0) - E(y(z)]^2 +$$

$$\frac{1}{\sqrt{\pi}} \begin{bmatrix} A_1(y(1.3556) - E(y(z)))^2 + A_1(y(-1.3556) - E(y(z)))^2 + \\ A_2(y(2.8570) - E(y(z)))^2 + A_2(y(-2.8570) - E(y(z)))^2 \end{bmatrix}$$

Five point formula gives exact calculation of the mean of the response for the <u>family</u> of all 8th order polynomials

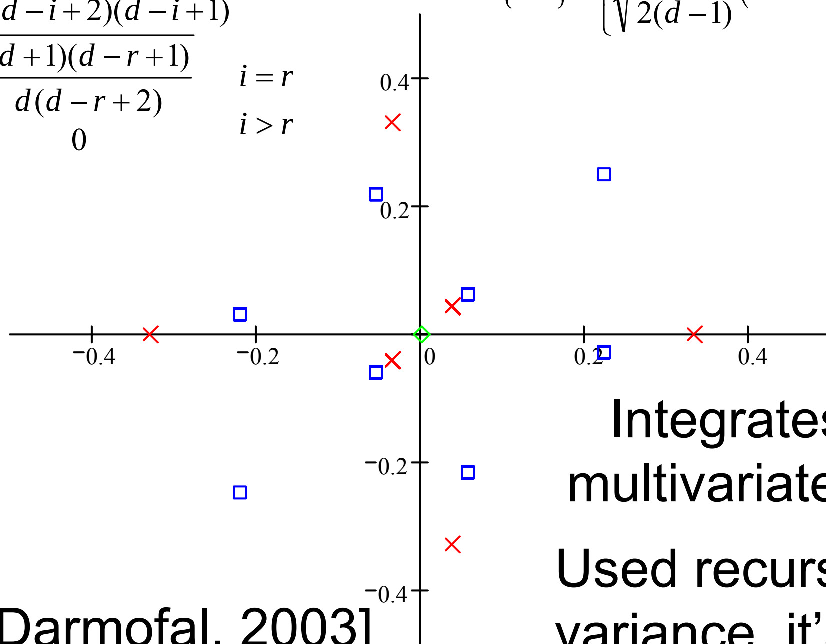
Cubature

$$E(y(\mathbf{z})) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\frac{1}{(2\pi)^{n/2}}e^{-\frac{1}{2}\mathbf{z}^T\mathbf{z}}\,y(\mathbf{z})\,d\mathbf{z}_1 d\mathbf{z}_2\ldots d\mathbf{z}_n \approx$$

$$\frac{2}{d+2}y(\mathbf{0}) + \frac{d^2(7-d)}{d(d+1)^2(d+2)^2}\sum_{j=1}^{d+1}\left[y\left(\mathbf{a}^{(j)}\right)+y\left(-\mathbf{a}^{(j)}\right)\right] + \frac{2(d-1)^2}{(d+1)^2(d+2)^2}\sum_{j=1}^{d(d+1)/2}\left[y\left(\mathbf{b}^{(j)}\right)+y\left(-\mathbf{b}^{(j)}\right)\right]$$

$$\mathbf{a}_i^{(r)} \equiv \begin{cases} -\sqrt{\dfrac{d+1}{d(d-i+2)(d-i+1)}} & i<r \\ \sqrt{\dfrac{(d+1)(d-r+1)}{d(d-r+2)}} & i=r \\ 0 & i>r \end{cases}$$

$$\left\{\mathbf{b}^{(j)}\right\} \equiv \left\{\sqrt{\frac{d}{2(d-1)}}\left(\mathbf{a}^{(k)}+\mathbf{a}^{(l)}\right) : k<l, l=1,2,\ldots d+1\right\}$$



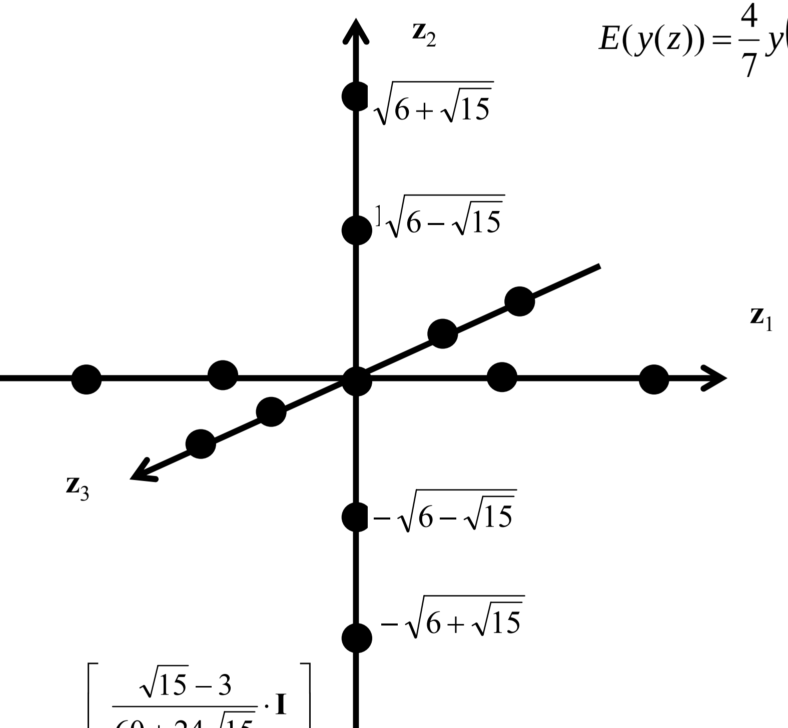
Sampling pattern for d=9 projected into a plane $d^2+3d+3=111$

Integrates exactly all Gaussian weighted multivariate polynomials of degree 5 or less.

Used recursively to estimate transmitted variance, it's exact up to second degree.

[Lu and Darmofal, 2003]

My New Technique: Based on Partial Separability of the Response



$$E(y(z)) = \frac{4}{7} y\left(\mathbf{D}^{\langle 2d+1\rangle}\right) + \frac{\sqrt{15}-3}{60+24\sqrt{15}}\left[y\left(\mathbf{D}^{\langle i\rangle}\right)+y\left(\mathbf{D}^{\langle 3d+1+i\rangle}\right)\right] + \frac{3\sqrt{15}+11}{28\sqrt{15}}\left[y\left(\mathbf{D}^{\langle d+i\rangle}\right)+y\left(\mathbf{D}^{\langle 2d+1+i\rangle}\right)\right]$$

$$E\left[\left(y(\mathbf{z})-E(y(\mathbf{z}))\right)^2\right] = (1+\varepsilon)\sum_{i=1}^{d}\left[\begin{Bmatrix} y\left(\mathbf{D}^{\langle i\rangle}\right) \\ y\left(\mathbf{D}^{\langle d+i\rangle}\right) \\ y\left(\mathbf{D}^{\langle 2d+1\rangle}\right) \\ y\left(\mathbf{D}^{\langle 2d+1+i\rangle}\right) \\ y\left(\mathbf{D}^{\langle 3d+1+i\rangle}\right) \end{Bmatrix}\right]^T \mathbf{W} \begin{bmatrix} y\left(\mathbf{D}^{\langle i\rangle}\right) \\ y\left(\mathbf{D}^{\langle d+i\rangle}\right) \\ y\left(\mathbf{D}^{\langle 2d+1\rangle}\right) \\ y\left(\mathbf{D}^{\langle 2d+1+i\rangle}\right) \\ y\left(\mathbf{D}^{\langle 3d+1+i\rangle}\right) \end{bmatrix}$$

$$\mathbf{W} = \begin{bmatrix} 0.004877409 & 0.005583293 & 0 & 0.00376481 & 0.000013884 \\ 0.005583293 & 0.223202195 & 0 & 0.018541911 & 0.00376481 \\ 0 & 0 & 0.24178391 & 0 & 0 \\ 0.00376481 & 0.018541911 & 0 & 0.223202195 & 0.005583293 \\ 0.000013884 & 0.00376481 & 0 & 0.005583293 & 0.004877409 \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} \frac{\sqrt{15}-3}{60+24\sqrt{15}}\cdot\mathbf{I} \\ \hline \frac{3\sqrt{15}+11}{28\sqrt{15}}\cdot\mathbf{I} \\ \hline 0 \quad 0 \quad \cdots \quad 0 \\ \hline -\frac{3\sqrt{15}+11}{28\sqrt{15}}\cdot\mathbf{I} \\ \hline -\frac{\sqrt{15}-3}{60+24\sqrt{15}}\cdot\mathbf{I} \end{bmatrix}$$

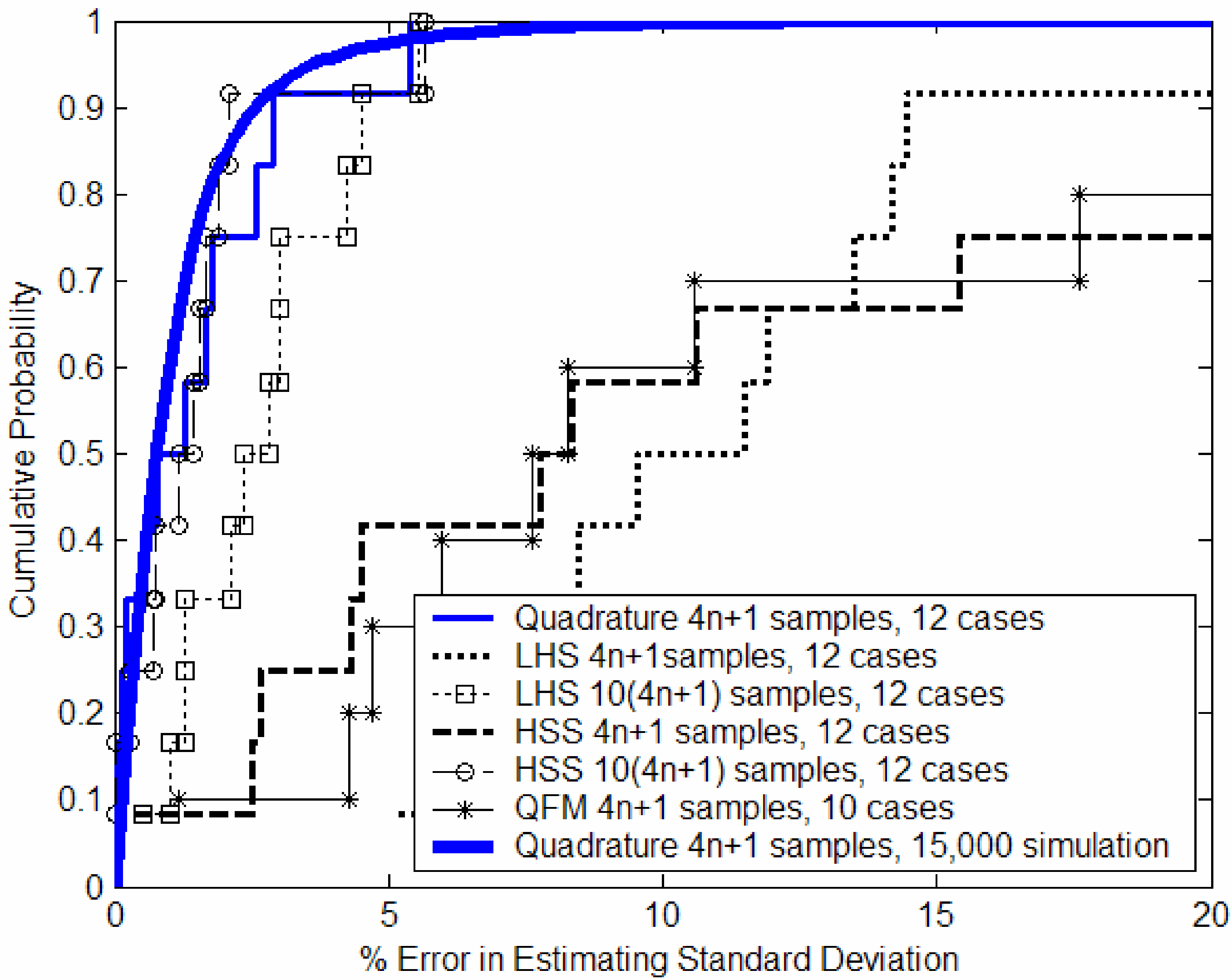
$$\varepsilon = \frac{\sum_{i=1}^{d}\beta_{iii}^{\ 2}}{\sum_{i=1}^{d}\beta_i^{\ 2} + 2\beta_{ii}^{\ 2} + 6\beta_i\beta_{iii} + 15\beta_{iii}^{\ 2} + 24\beta_{ii}\beta_{jjj} + 96\beta_{iii}^{\ 2} + 30\beta_i\beta_{iiiii} + 210\beta_{iii}\beta_{iiiii} + 945\beta_{iiiii}^{\ 2}}$$

$$\sigma^2(\varepsilon) = 2r^6\int_0^{\infty}\left[\frac{\sum_{i=1}^{3}\sum_{j=1}^{3}\left[\frac{\mathbf{P}_{3i}\mathbf{P}_{3j}}{\sqrt{1+2t\lambda_i}\sqrt{1+2t\lambda_j}}\right]^2}{\prod_{j=1}^{5}\sqrt{1+2t\lambda_j}}\right]t\,dt + r^6\int_0^{\infty}\left[\frac{\left[\sum_{i=1}^{3}\sum_{j=1}^{3}\frac{\mathbf{P}_{3i}\mathbf{P}_{3j}}{\sqrt{1+2t\lambda_i}\sqrt{1+2t\lambda_j}}\right]^2}{\prod_{j=1}^{5}\sqrt{1+2t\lambda_j}}\right]t\,dt$$

$$\mathbf{M} = \begin{bmatrix} r & 0 & 3r^2 & 0 & 15r^3 \\ 0 & 2r^2 & 0 & 12r^3 & 0 \\ 3r^2 & 0 & 15r^3 & 0 & 105r^4 \\ 0 & 12r^3 & 0 & 96r^4 & 0 \\ 15r^3 & 0 & 105r^4 & 0 & 945r^5 \end{bmatrix}$$

Used to estimate transmitted variance, it’s very accurate up to fifth degree.

Results of Model-Based and Case-Based Evaluations



Outline

- Motivation & context
- Techniques for “computer experiments”
  - Monte Carlo
  - Importance sampling
  - Latin hypercube sampling
  - Hammersley sequence sampling
  - Quadrature and cubature
- ➡ Some cautions

Why Models Can Go Wrong

- Right model → Inaccurate answer
  - Rounding error
  - Truncation error
  - Ill conditioning
- Right model → Misleading answer
  - Chaotic systems
- Right model → No answer whatsoever
  - Failure to converge
  - Algorithmic complexity
- Not-so right model → Inaccurate answer
  - Unmodeled effects
  - Bugs in coding the model

Errors in Scientific Software

- Experiment T1
  - Statically measured errors in code
  - Cases drawn from many industries
  - ~10 serious faults per 1000 lines of commercially available code
- Experiment T2
  - Several independent implementations of the same code on the same input data
  - One application studied in depth (seismic data processing)
  - Agreement of 1 or 2 significant figures on average

Hatton, Les, 1997, "The T Experiments: Errors in Scientific Software", *IEEE Computational Science and Engineering*.

Definitions

- **Accuracy** – The ability of a model to faithfully represent the real world
- **Resolution** – The ability of a model to distinguish properly between alternative cases
- **Validation** – The process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model. (AIAA, 1998)
- **Verification** – The process of determining that a model implementation accurately represents the developer's conceptual description of the model and the solution to the model. (AIAA, 1998)

Model Validation in Engineering

- A model of an engineering system can be validated using data to some degree within some degree of confidence
- Physical data on <u>that</u> specific system cannot be gathered until the system is designed and built
- Models used for design are ***never fully validated*** at the time design decisions must be made

Next Steps

- Friday 4 May
  - Exam review
- Monday 7 May – Frey at NSF
- Wednesday 9 May – Exam #2
- Wed and Fri, May 14 and 16
  - Final project presentations