Oregon State University

Design Under Uncertainty: Methods

ME 615 Spring 2020

Dr. Chris Hoyle

MIME

Monte Carlo Simulation for calculating a Probability

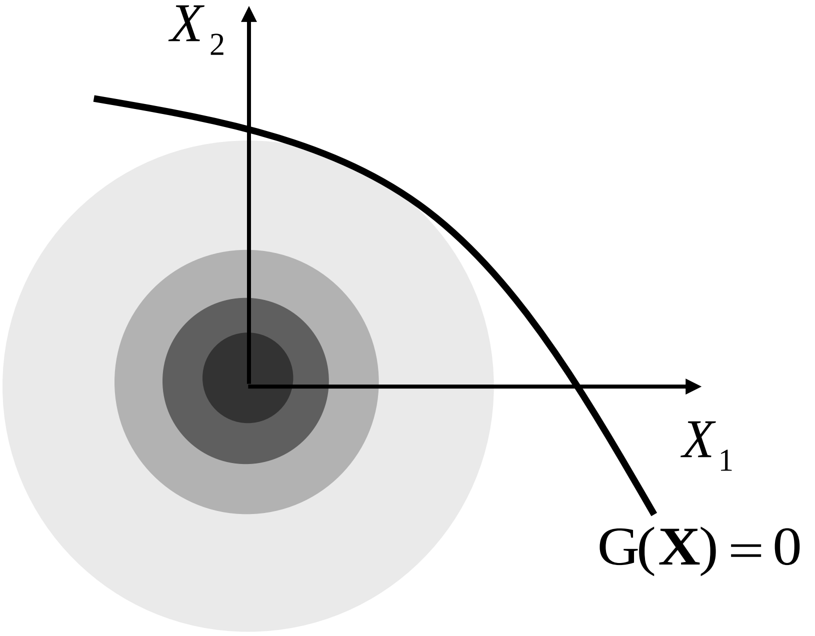
- Generate random samples X_i for X (normrnd, betarnd, weibrnd,… in Matlab)
- Compute $G(X_i)$ and $I[G(X_i)<0]$
- Calculate probability of failure as follows:

$$P_f = \Pr[G(\mathbf{X}) \le 0] = \int_{G(\mathbf{X})\le 0} f(\mathbf{X})d\mathbf{X}$$

$$P_f = \int I[G(\mathbf{X}) \le 0] f(\mathbf{X})d\mathbf{X}$$

$$P_f \approx \frac{1}{N}\sum_{i=1}^{N} I[G(\mathbf{X}_i) \le 0]$$

$I[\bullet]$: Indicator function



Monte Carlo Algorithm for computing a probability

1. Define the random model inputs.
2. Generate a set of inputs randomly from a probability distribution over the domain.
   - In Matlab, you can use normrnd, lognrnd , betarnd, unifrnd
3. Perform a deterministic computation using your system model on this set of input values.
   - This means you will need a method to send the input values generated by Matlab to your system model
4. Check if simulated value meets the requirement.
   - Set I= 1 if it meets requirement
   - Set I = 0 If it does not meet requirement.
5. Repeat 2-4 *N* times, where *N* is the number of samples desired (usually on the order of 10^3-10^6)
6. Sum I and compute probability of meeting requirement as
   - **sum I/N**

Monte Carlo Simulation

- Characteristics of the MCS Method:
  - Can handle any parametric or non-parametric representation of input uncertainty.
  - The output uncertainty is not limited to a parametric distribution.
  - Straightforward implementation.
  - Expense not a function of number of input variables.
- Limitations of the MCS Method:
  - Requires much sampling, even with advanced sampling methods.
  - Difficult to estimate the number of MCS samples needed *a priori*.

Monte Carlo Simulation

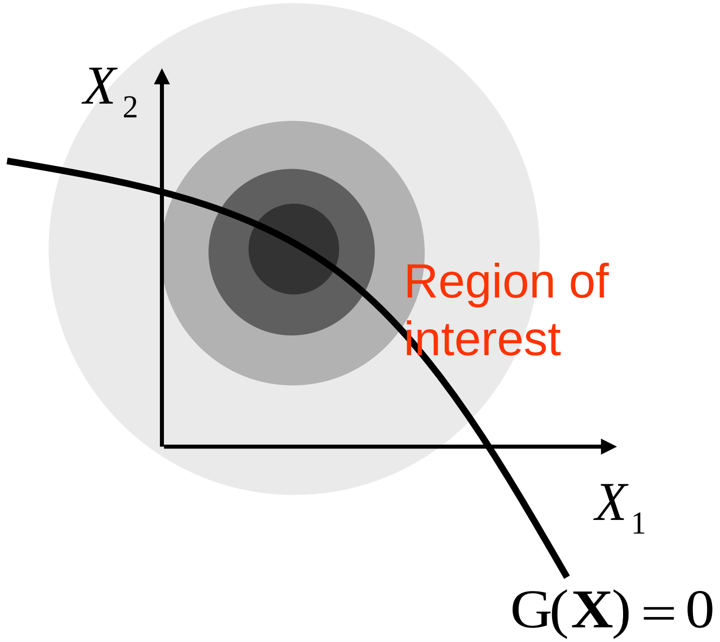
Oregon State University
College of Engineering

- Large sample required ($\sim 10^{-3}\ P_f$)
- Variance in result
- Importance sampling, stratified sampling, antithetic variants,…

$$P_f = \int I\left[G\left(\mathbf{X}\right) \le 0\right] f(\mathbf{X}) d\mathbf{X}$$

$$P_f = \int I\left[G\left(\mathbf{X}\right) \le 0\right] \frac{f(\mathbf{X})}{h(\mathbf{X})} h(\mathbf{X}) d\mathbf{X}$$

$$P_f \approx \frac{1}{N} \sum_i I\left[G\left(\mathbf{X}_i\right) \le 0\right] \frac{f(\mathbf{X}_i)}{h(\mathbf{X}_i)}$$



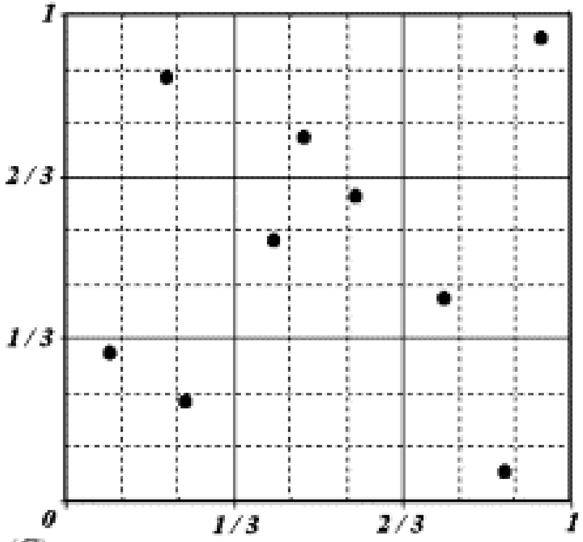
h(x): importance sampling density function

Choosing appropriate h(x) is often tricky.

LHS and Low Discrepancy Sampling

Oregon State University
College of Engineering

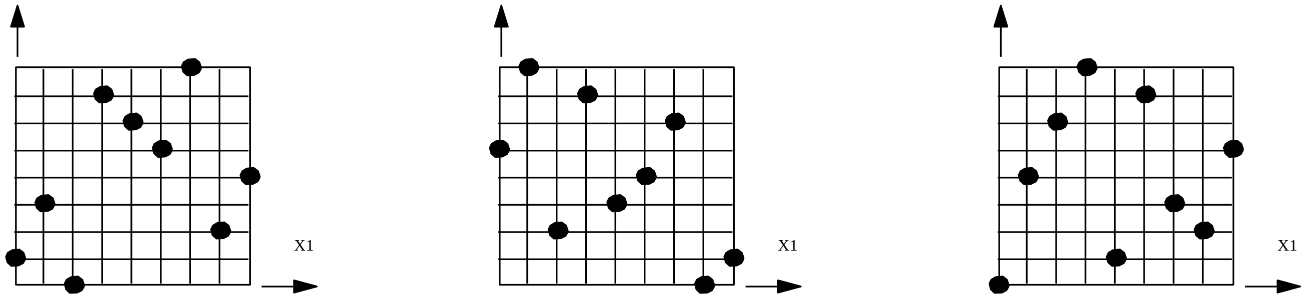
- In the context of statistical sampling, a square grid containing sample positions is a Latin square if (and only if) there is only one sample in each row and each column.
- A **Latin hypercube** is the generalization of this concept to an arbitrary number of dimensions, whereby each sample is the only one in each axis-aligned hyperplane containing it.



(C)

Latin Hypercubes Sampling (LHS)

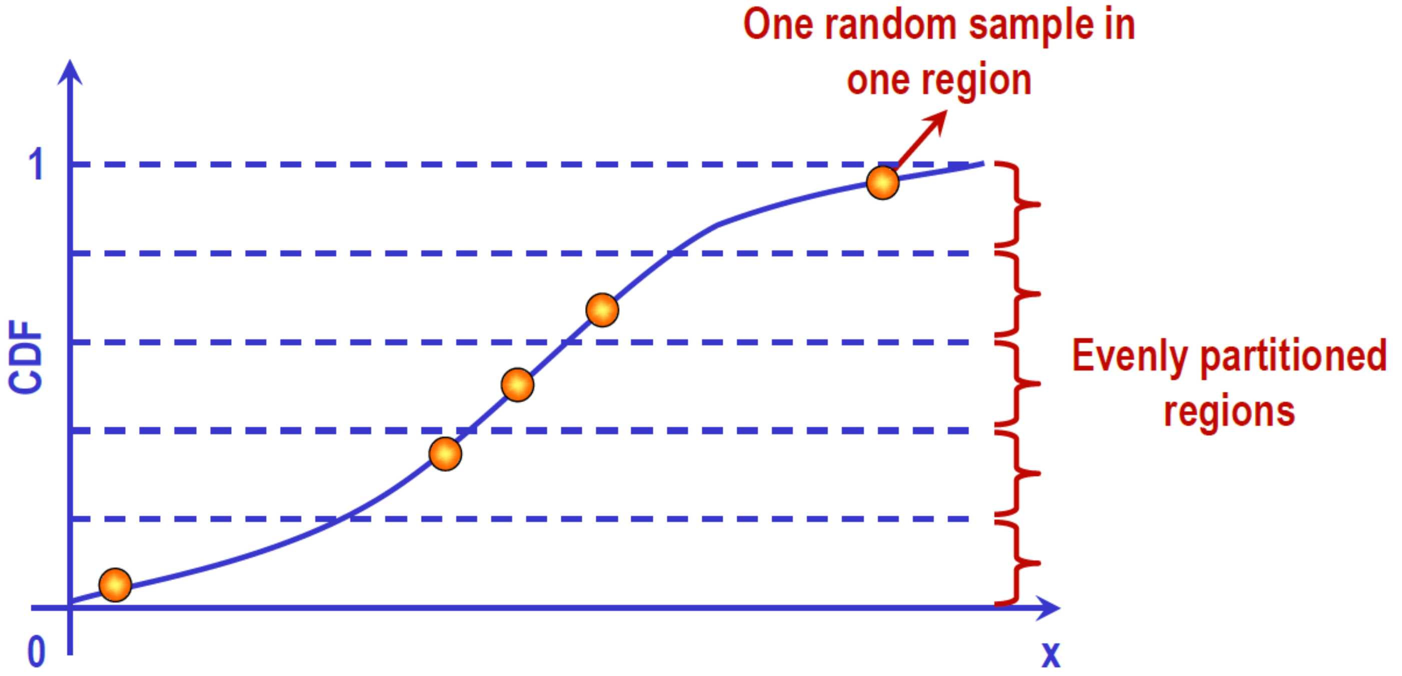
Oregon State University College of Engineering



- For ***n*** samples in ***d*** factors (or variables), ***n*** divisions on each factor ***d*** dimension
  - For example for 9 samples, we create 9 divisions in each of the 2 design variables (or factor) divisions.
  - This creates $9^2 = 81$ possible locations.
- Stratified sampling - each of the factors is sampled at ***n*** levels and evenly distributed when projected to a single dimension.
- Randomly distributed therefore the design is not unique.
  - Good balance of uniform sampling with randomness

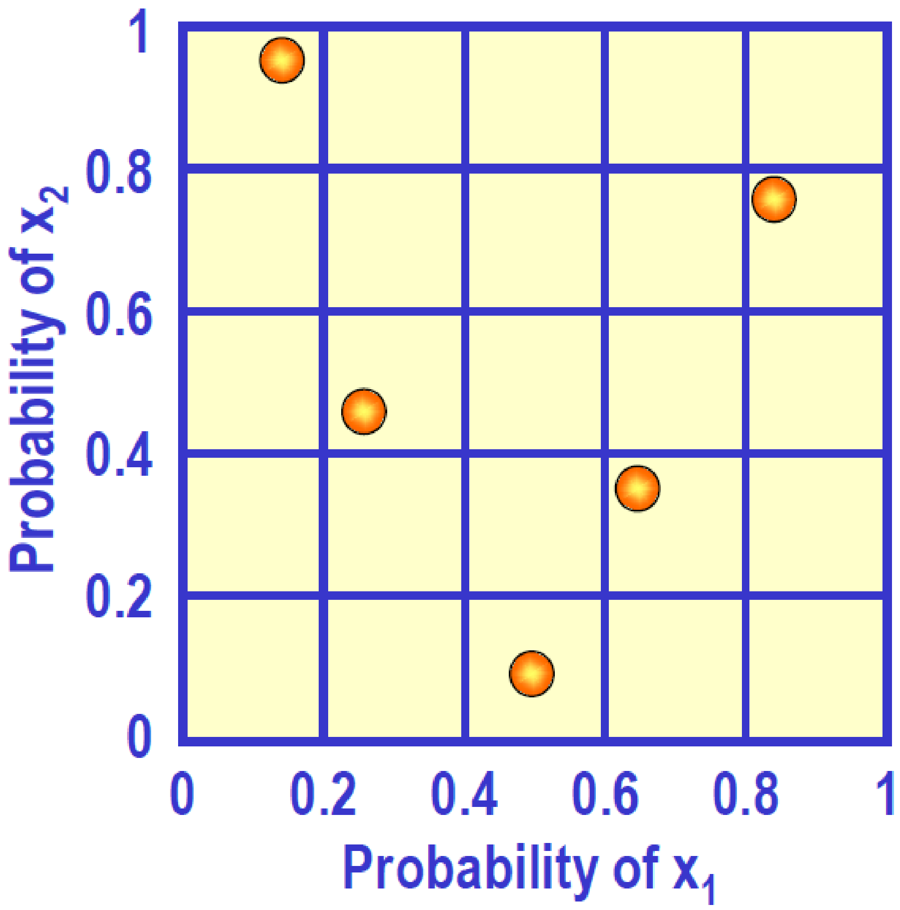
1 Dimensional LHS

- **One dimensional Latin hypercube sampling:**
- Evenly partition CDF into *N* regions
- **Randomly** add one sampling point in each region



2 Dimensional LHS

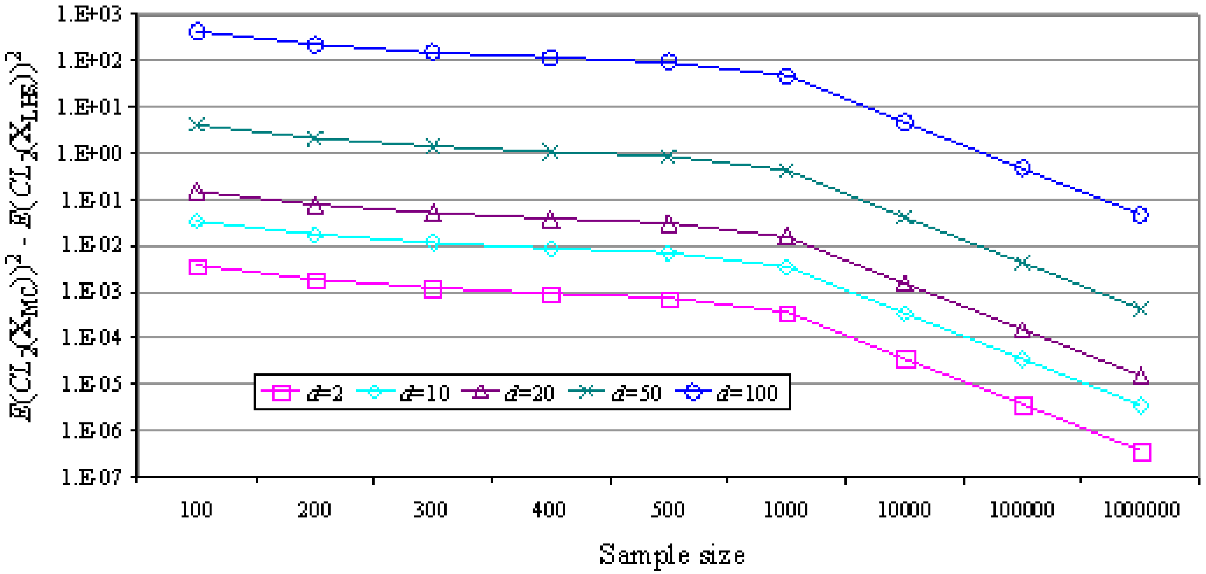
- **Two dimensional Latin hypercube sampling**
- x_1 and x_2 assumed to be independent
- Generate one-dimensional LHS samples for x1
- Generate one-dimensional LHS samples for x2
- Randomly combine the LHS samples to two-dimensional pairs



LHS and Low Discrepancy Sampling

Oregon State University
College of Engineering

- **Latin-Hypercube Sampling** (LHS) is the most beneficial to replace MCS when the sample size n is small and the dimension d is large



- LHS has better (lower) expected value of $CL_2(\mathbf{X})^2$ (measure of discrepancy) compared to that of Monte Carlo random samples (Fang et al. 2002)

$$E\left(CL_2\left(\mathbf{X}_{MC}\right)\right)^2 - E\left(CL_2\left(\mathbf{X}_{LHS}\right)\right)^2 = \left(\frac{13}{12}\right)^{d-1} \frac{d}{6n}\left(1 - \frac{2d+11}{26n}\right) + O\left(n^{-3}\right)$$

Low Discrepancy Sampling

Oregon State University
College of Engineering

- LHS can use the Matlab functions:
  - **lhsdesign**: Uniform sampling
  - **lhsnorm**: Latin hypercube sample from normal distribution
- Another low discrepancy sampling method is the Sobol' sequence.
  - These sequences use a base of two to form successively finer uniform partitions of the unit interval, and then reorder the coordinates in each dimension.
  - **sobolset**: uniform sampling
- LHS and Sobol' Sequence are called Pseudo-random sampling
  - Preserve randomness to avoid spurious correlations
  - Impose rules to ensure good coverage of the sample space.

Generation of Random Variables from a Distribution

Oregon State University
College of Engineering

- If you have random variables from a uniform random distribution:

$$v = Uniform[0,1]$$

- These can be turned in “draws” from a specific distribution using the inverse transform method:

$$F_X(x_i) = v_i \text{ or } x_i = F_X^{-1}(v_i)$$

- Matlab has inverse CDF functions
  – norminv
  – logninv
  – betainv
  – unifinv
  – etc

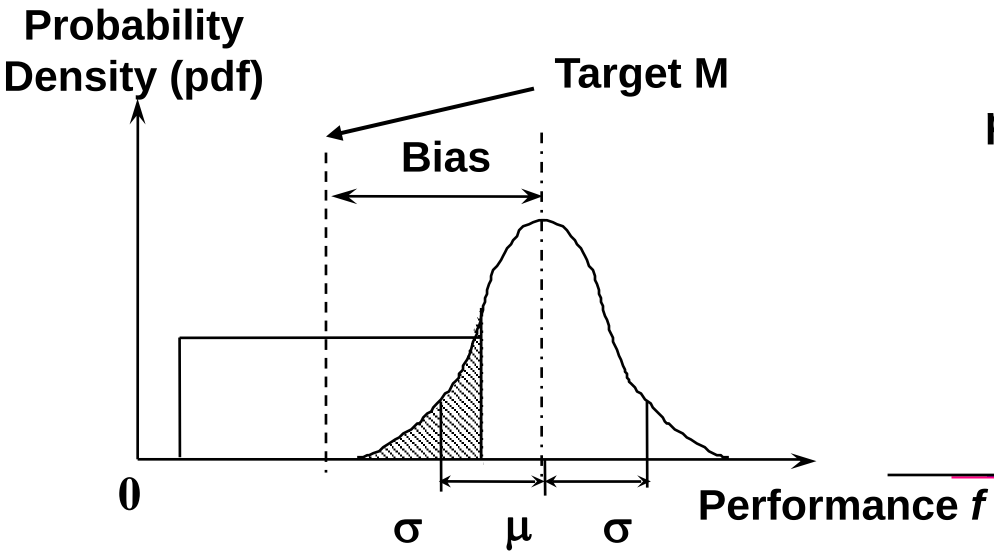
Tips for using MCS in Optimization

- In optimization, it is necessary to have repeatability in moment calculations.
- Use the rng function:
  - rng(1)
- This will ensure the calculation of moments is repeatable.

Objectives and Requirements

Oregon State University College of Engineering

Objectives

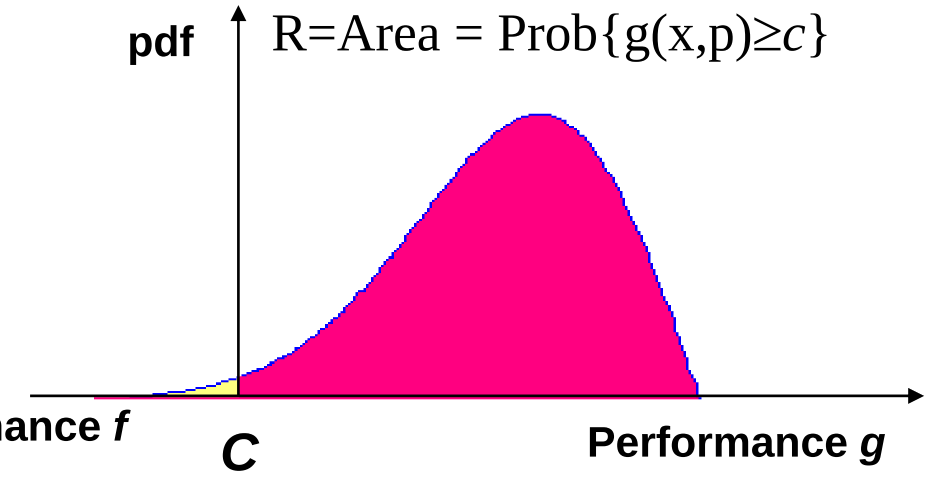


Look at entire distribution

s.t. $x \in X$

Considering the effect of variations without eliminating the causes

Requirements



Satisfy

$R = P\{g(\mathbf{x},\mathbf{p}) \geq c\} \geq R_0$

Limit State

To assure proper levels of “safety” for the system designed