PRISM

NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems

ME 597/AAE 590 : Introduction to Uncertainty Quantification

Lecture 3: Uncertainty Propagation

Lin Sun and Martin Hunt

Purdue University

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

PURDUE UNIVERSITY

THE UNIVERSITY of NEW MEXICO

In previous lecture,

Local sensitivity

$$J = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$

Scaled sensitivity

$$S = \left\{ x \bullet \frac{dy}{dx} \right\}$$

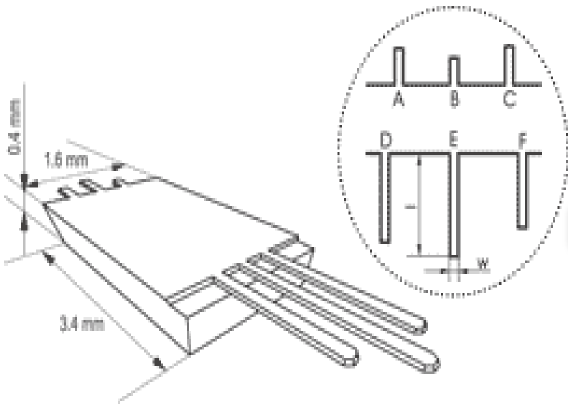
Techniques for computing sensitivity: finite difference, code differentiation, sensitivity equation, complex number method, c++ compiler …

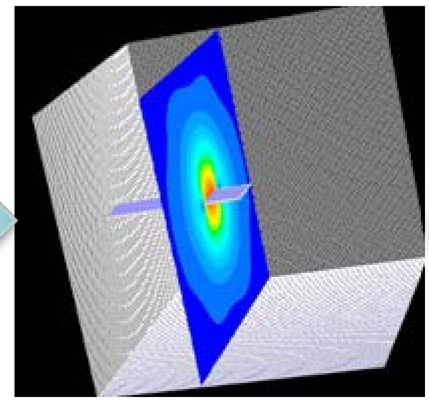
In this lecture,

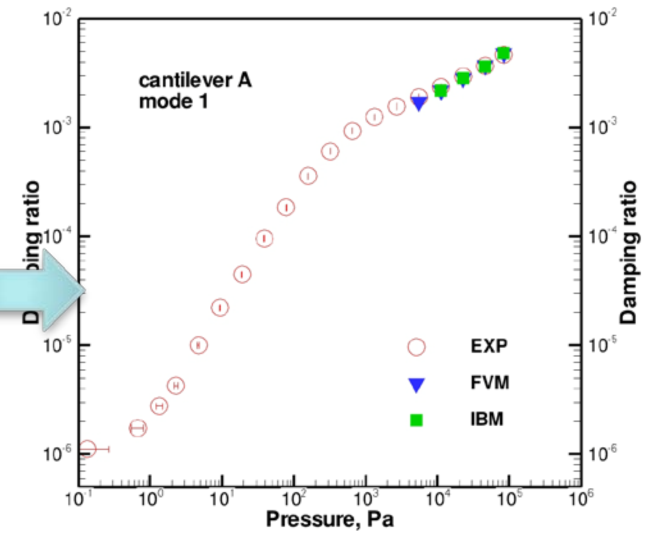
- Variance propagation
- Sampling methods
    - --- Monte Carlo sampling (MCS)
    - --- Latin Hypercube sampling (LHS)
- Sample generation

A typical UQ problem is to answer two closely related questions:

- What is the uncertainty in y(x) given the uncertainty in x?
- How important are the individual elements of x with respect to the uncertainty in y(x)







b=35μm; $\sigma_b = 5\%$;

$t = 1\mu m, \sigma_t = 15\%$;

$L = 100\mu m, \sigma_L = 5\%$;

$\rho_g = 0.9707 kg/m^3, \sigma_g = 10\%$;

$\omega = 1.33\text{x}10^5 Hz, \sigma_\omega = 5\%$;

1. What is the uncertainty of output?
2. Would the nominal value be the mean output?
3. Which parameter is important?
4. …

PRISM
NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems

Variance Propagation

Given input mean and variance, how to calculate output mean and variance?

Expected value:

$$E[T(x,t)] = \int_{-\infty}^{\infty} T(x,t;k) f(k)\,dk$$

Variance:

$$\sigma_T^2(x,t) = \int_{-\infty}^{\infty} \{T(x,t;k) - E[T(x,t)]\}^2 f(k)\,dk$$

Expand output in a Taylor series about nominal input value:

$$T(x,t;k) = T(x,t;\bar{k}) + \left.\frac{\partial T}{\partial k}\right|_{\bar{k}}(k-\bar{k}) + \frac{1}{2}\left.\frac{\partial^2 T}{\partial k^2}\right|_{\bar{k}}(k-\bar{k})^2 + \ldots$$

Expected value:

$$E[T(x,t)] = T(x,t;\bar{k})\int_{-\infty}^{\infty} f(k)\,dk + \left.\frac{\partial T}{\partial k}\right|_{\bar{k}}\int_{-\infty}^{\infty}(k-\bar{k})f(k)\,dk + \frac{1}{2}\left.\frac{\partial^2 T}{\partial k^2}\right|_{\bar{k}}\int_{-\infty}^{\infty}(k-\bar{k})^2 f(k)\,dk + \ldots$$

PRISM
NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems

Variance Propagation Equation (cont.)

$$E[T(x,t)] = T(x,t;\bar{k})\int_{-\infty}^{\infty} f(k)dk + \left.\frac{\partial T}{\partial k}\right|_{\bar{k}} \int_{-\infty}^{\infty}(k-\bar{k})f(k)dk + \left.\frac{1}{2}\frac{\partial^2 T}{\partial k^2}\right|_{\bar{k}} \int_{-\infty}^{\infty}(k-\bar{k})^2 f(k)dk + \ldots$$

- First integral is unity for a properly scaled distribution function
- Second integral will be zero because of definition of $\bar{k}$
- Third integral is definition of variance of k
- Expected value becomes

$$E[T(x,t)] = T(x,t;\bar{k}) + \left.\frac{1}{2}\frac{\partial^2 T}{\partial k^2}\right|_{\bar{k}} \sigma_k^2 + \ldots = T(x,t;\bar{k}) + \left.\frac{\bar{k}^2}{2}\frac{\partial^2 T}{\partial k^2}\right|_{\bar{k}} \left(\frac{\sigma_k}{\bar{k}}\right)^2 + \ldots$$

$$E[T(x,t)] \neq T(x,t;\bar{k})$$

Variance Propagation Equation (cont.)

$$\sigma_T^2 = \left(\frac{\partial T}{\partial k}\bigg|_{\bar{k}}\right)^2 \int_{-\infty}^{\infty} (k-\bar{k})^2 f(k)dk + \frac{\partial T}{\partial k}\bigg|_{\bar{k}} \frac{\partial^2 T}{\partial k^2}\bigg|_{\bar{k}} \int_{-\infty}^{\infty} (k-\bar{k})[(k-\bar{k})^2 - \sigma_k^2] f(k)dk$$

$$+ \frac{1}{4}\left(\frac{\partial^2 T}{\partial k^2}\bigg|_{\bar{k}}\right)^2 \int_{-\infty}^{\infty} [(k-\bar{k})^2 - \sigma_k^2]^2 f(k)dk + \ldots$$

$$\int_{-\infty}^{\infty} (k-\bar{k})^3 f(k)dk = \alpha_3 \sigma_k^3 \qquad \int_{-\infty}^{\infty} (k-\bar{k})^4 f(k)dk = \alpha_4 \sigma_k^4$$

$$\sigma_T^2 = \left(k\frac{\partial T}{\partial k}\bigg|_{\bar{k}}\right)^2 \left(\frac{\sigma_k}{\bar{k}}\right)^2 + \alpha_3 k\frac{\partial T}{\partial k}\bigg|_{\bar{k}} k^2\frac{\partial^2 T}{\partial k^2}\bigg|_{\bar{k}} \left(\frac{\sigma_k}{\bar{k}}\right)^3 + \left(\frac{\alpha_4 - 1}{4}\right)\left(k^2\frac{\partial^2 T}{\partial k^2}\bigg|_{\bar{k}}\right)^2 \left(\frac{\sigma_k}{\bar{k}}\right)^4 + \ldots$$

$$\sigma_T^2 = \left(\frac{\partial T}{\partial \beta_1}\sigma_{\beta_1}\right)^2 + \left(\frac{\partial T}{\partial \beta_2}\sigma_{\beta_2}\right)^2 + \ldots$$

PRISM
NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems

Example

$$E[T(x,t)] = T(x,t;\bar{k}) + \frac{1}{2}\frac{\partial^2 T}{\partial k^2}\bigg|_{\bar{k}} \sigma_k^2 + \ldots = T(x,t;\bar{k}) + \frac{\bar{k}^2}{2}\frac{\partial^2 T}{\partial k^2}\bigg|_{\bar{k}} \left(\frac{\sigma_k}{\bar{k}}\right)^2 + \ldots$$

$$\sigma_T^2 = \left(\frac{\partial T}{\partial \beta_1}\sigma_{\beta_1}\right)^2 + \left(\frac{\partial T}{\partial \beta_2}\sigma_{\beta_2}\right)^2 + \ldots$$

A simple example:

$$y = x^2 \qquad \bar{x} = 10 \qquad \sigma_x = 1.155$$

$$E[y(x)] = y(\bar{x}) + \frac{1}{2}\frac{\partial^2 y}{\partial x^2}\bigg|_{\bar{x}} \sigma^2{}_x = 10\times 10 + 0.5\times 2\times 1.155^2 = 101.33$$

$$\sigma^2{}_y = \left(\frac{dy}{dx}\sigma_x\right)^2 = (2\times 10\times 1.155)^2 = 533.6 \qquad \sigma_y = 23.1$$

PRISM
NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems

Sampling-based Method

Uncertainty in x and its distribution

↓

Random sampling of x

↓

Evaluation of model

↓

Uncertainty in y and its distribution

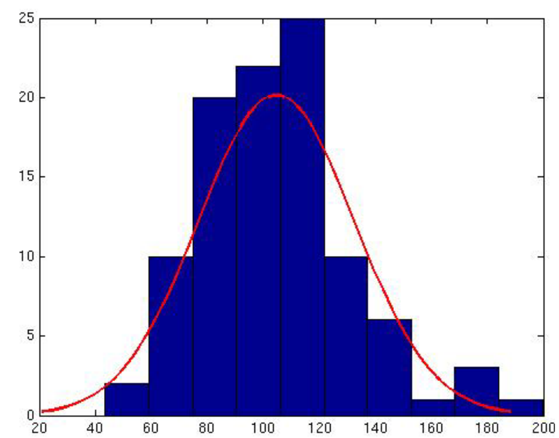
↓

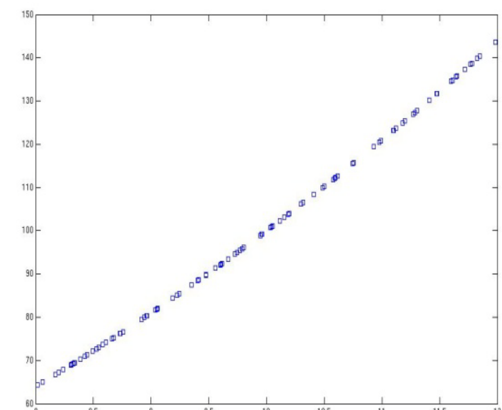
Sensitivity analysis

$$\bar{x} = 10, \quad \sigma = 1.15$$

$$x = \{8.21,\ 9.49,\ 10.13\cdots,\ 11.37\}$$

$$y = x^2 \qquad y = \{67.4,\ 90.1,\ 102.6,\cdots,129.3\}$$





A Closer Look at Sampling Methods

$$x_k = \left[x_{k1}, x_{k2}, \cdots x_{kn_X} \right], \quad \text{k=1, 2, ..., n}_s$$

where n_s is the sampling size

$$y(x_k) = \left[y_1(x_k), y_2(x_k), \cdots y_{n_Y}(x_k) \right]$$

The pairs [x_k, $y(x_k)$] form a mapping form the uncertain inputs to the corresponding uncertain outputs. From this mapping we can perform uncertainty and sensitivity analysis.

- Input uncertainty and distribution;
- Generation of the sample from the distributions;
- Evaluation of the models for each individual elements of the sample;
- Display the uncertainty of the outputs;
- Sensitivity analysis by exploring the mapping

Step1. Input Uncertainty

Aleatoric uncertainty: *Uncertainty due the variability of inputs or models*

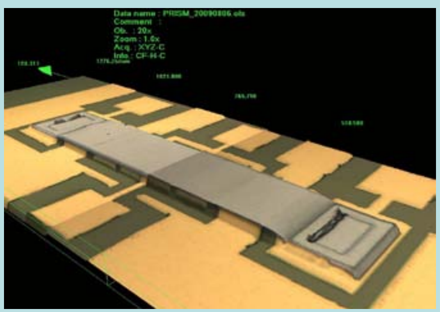
Typically represented as probability density functions

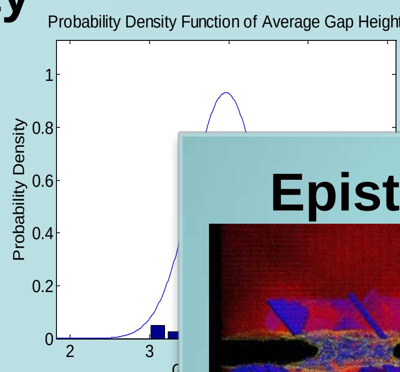
Epistemic uncertainty: *Uncertainty due to unknown processes or mechanisms*

Unknown parameters in a model

Model form error

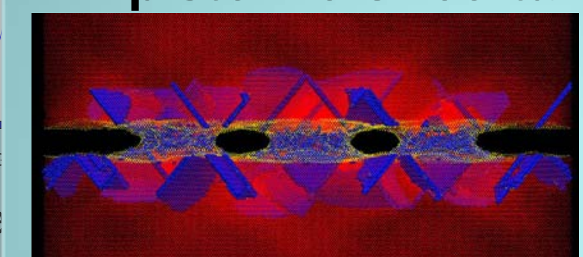
Aleatoric Uncertainty





PRISM device fabricated by Prof. Pe

Epistemic Uncertainty



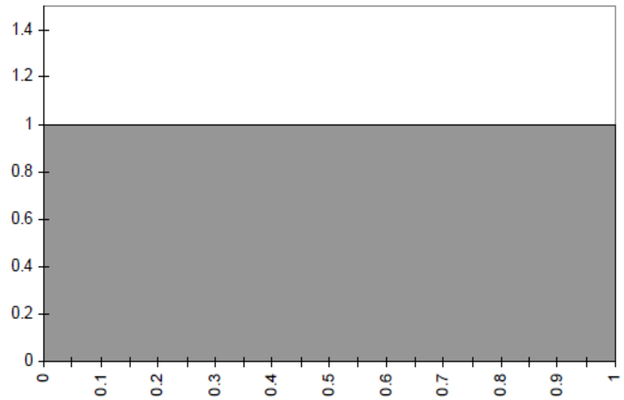
Molecular dynamics simulations of contact

- Form of the interatomic potential?
- Constants in the interatomic potential?

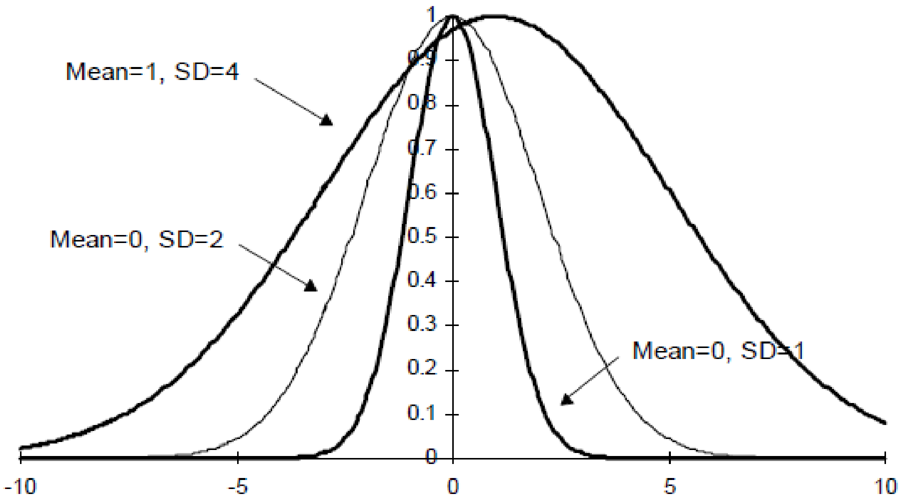
Courtesy Strachan group

PRISM
NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems

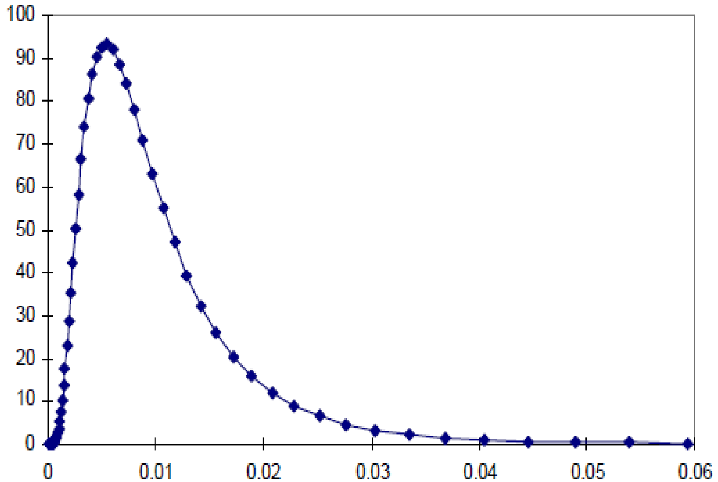
Uncertainty Distribution



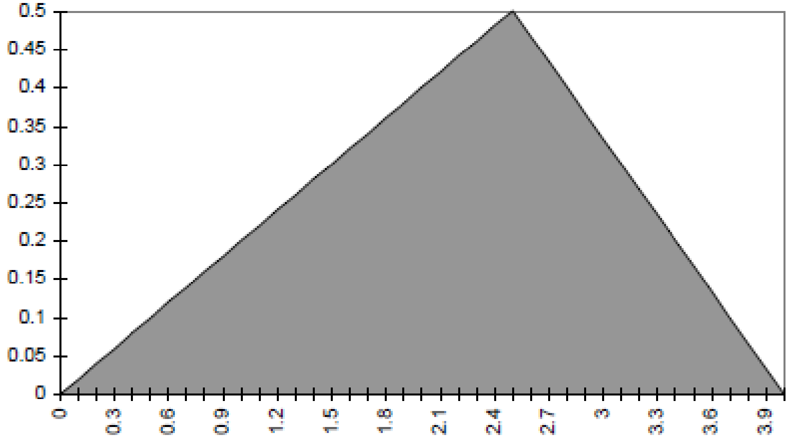
Uniform distribution



Normal(Gaussian) distribution



Lognormal distribution

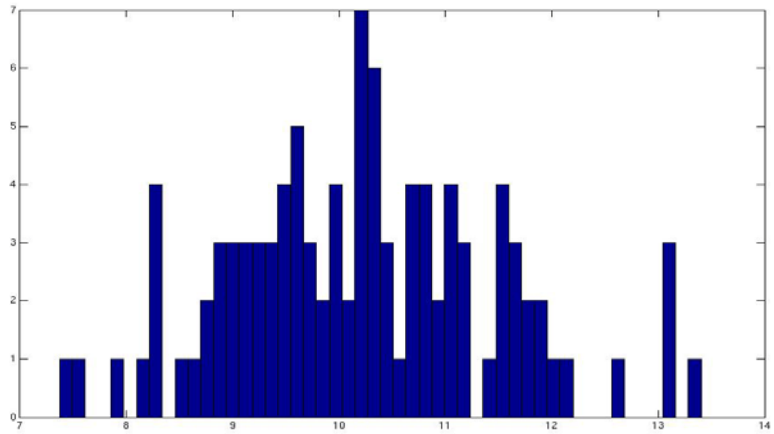


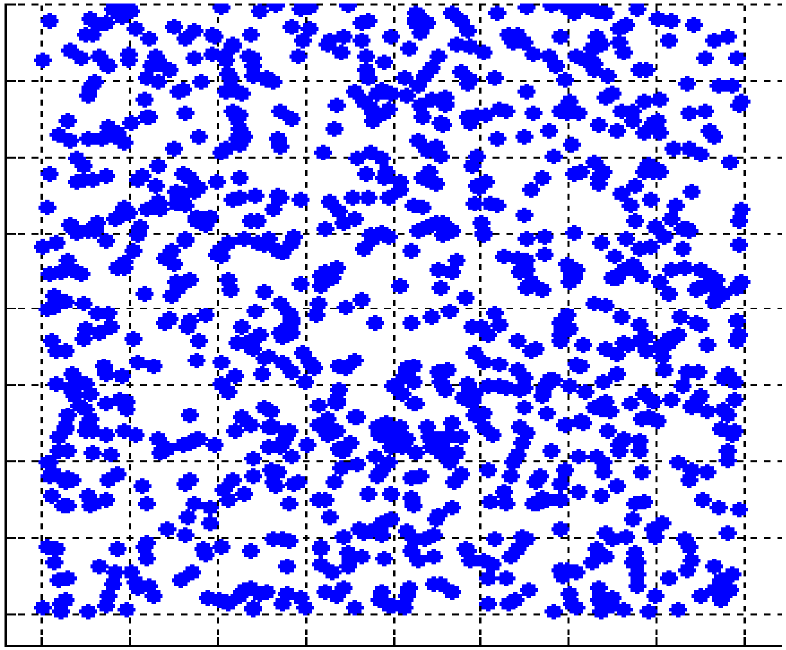
Triangular distribution

Step 2. Sampling

Monte Carlo sampling:

Random sample generator
constrained by distribution function





Pros: easy to implement;

Cons:

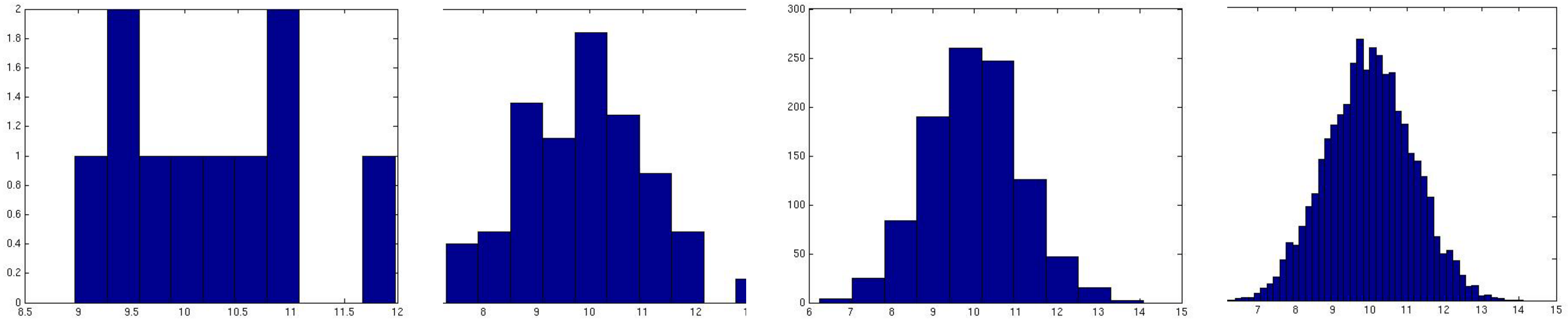
(a) no guarantee to cover all ranges

(b) can’t avoid replicates or clusters

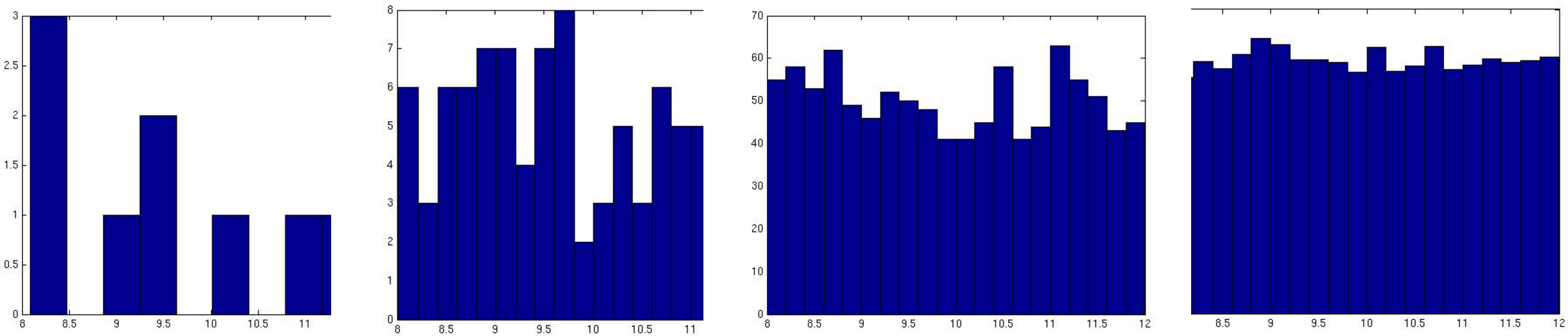
(c) require large number of samples

(d) convergence rate is low

PRISM
NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems



N=10 N=100 N=1000 N=10000



Step2. Sampling---Latin Hypercube

Wikipedia: a **Latin square** is a square grid containing sample positions 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.

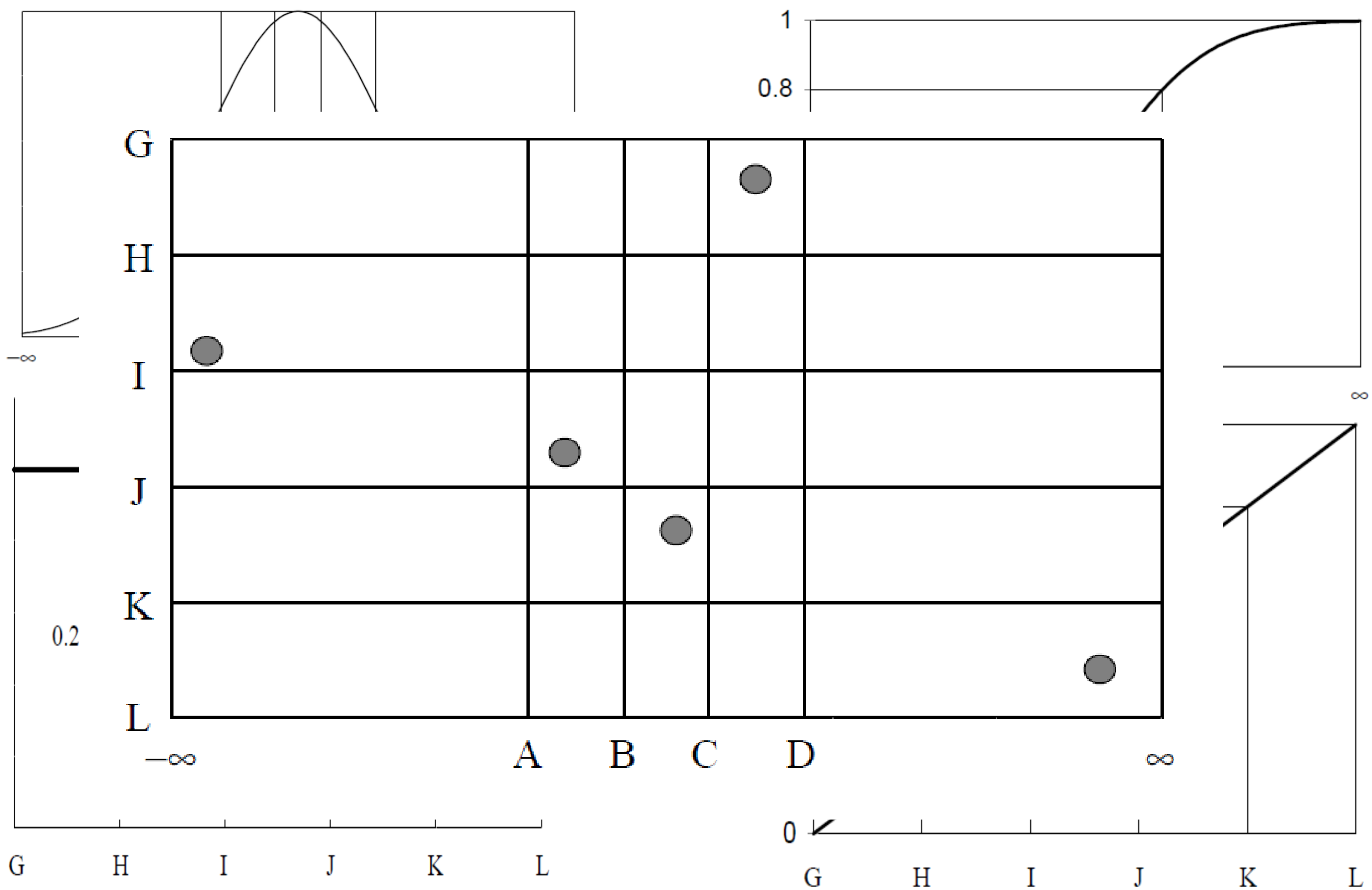
Procedures:

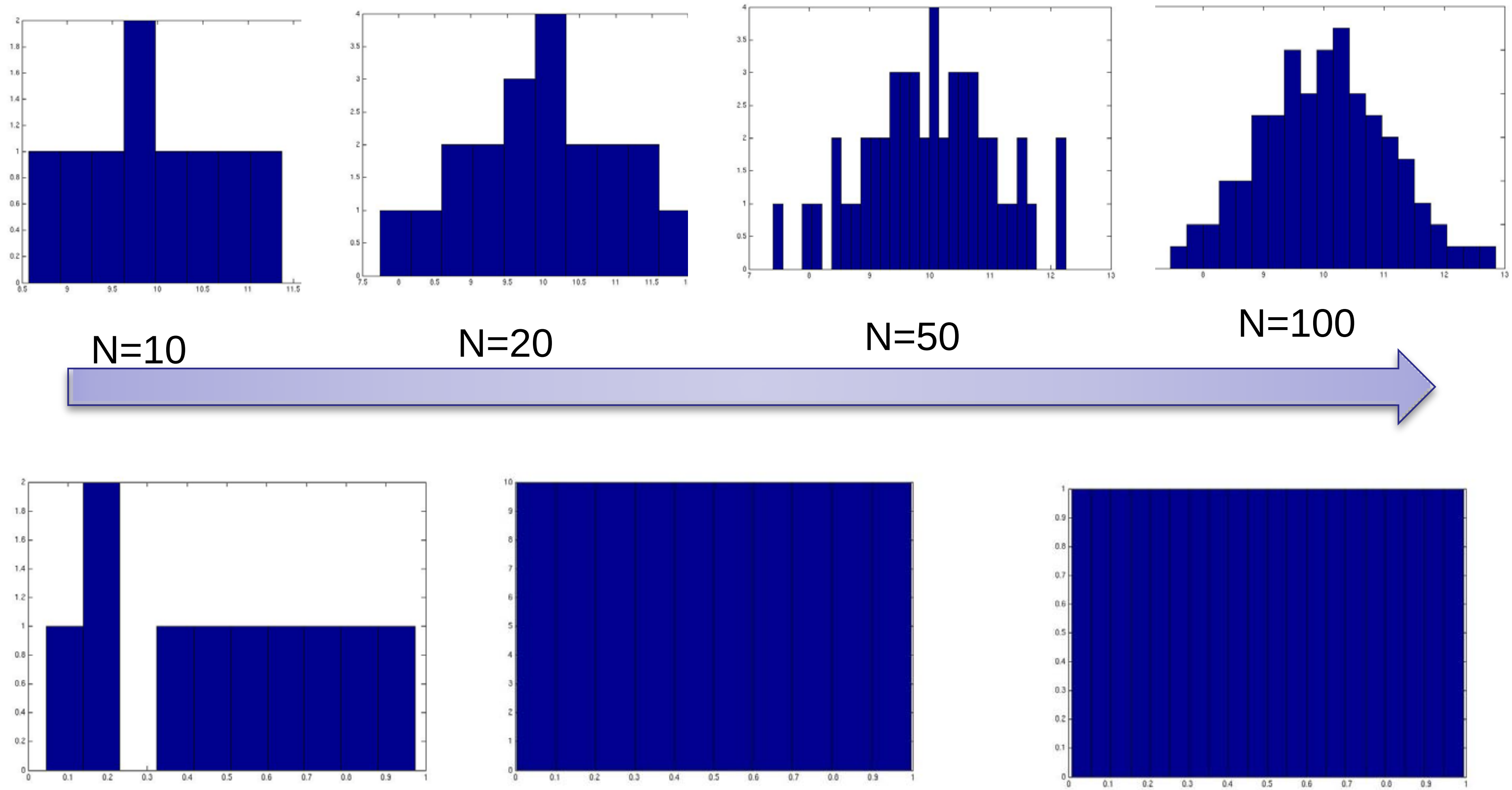
- PDF or CDF divided into bands of equal probability
- Within each band, a random sample is drawn
- This process is repeated for each input parameters
- The N_{lhs} xN_p matrix is randomized

probability band/parameters	→ x_1	x_2	...	x_{np}
sample 1				
sampl				
...				
sample				

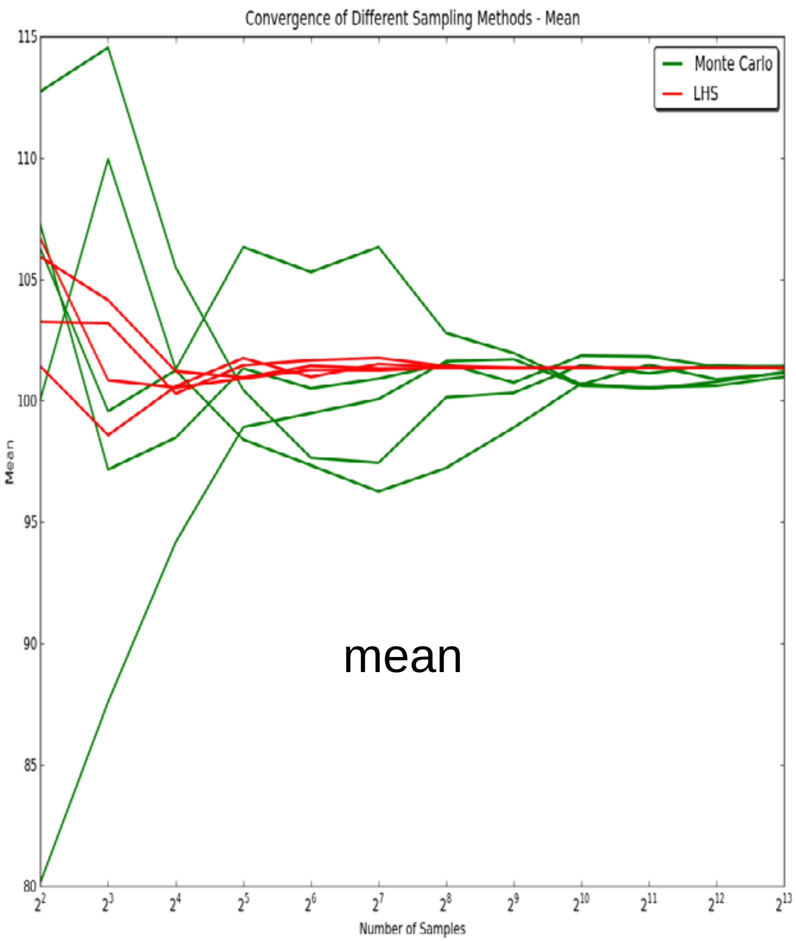
Pros: (a) covers whole range; no clusters
(b) require fewer number of samples than MC
(c) convergence rate is faster
Cons: sample size increases with number of input parameters; choose sample size a priori.

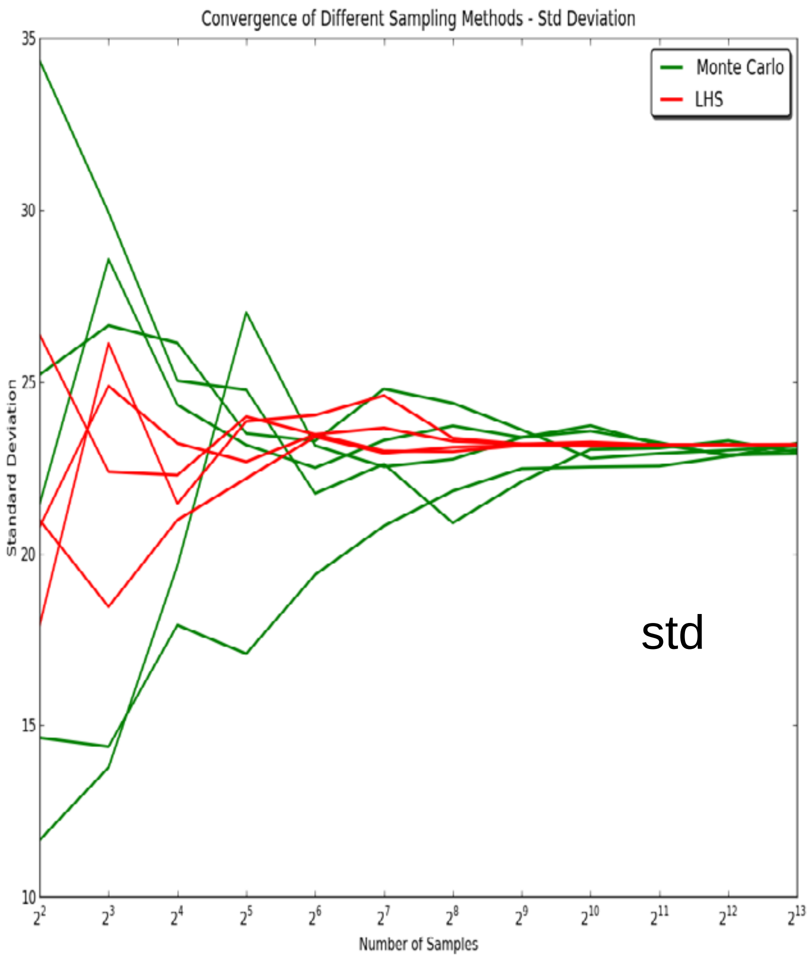
LHS Example (2D)




N=10
N=20
N=50
N=100

Convergence Rate





PRISM
NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems

Step 3. Evaluation of Models

Evaluate the model for each element of the input samples and eventually get the mapping between input and output.

Since the sampling number is generally large, one needs to automate job submission and automate data collection. PUQ tool has all the job management features.

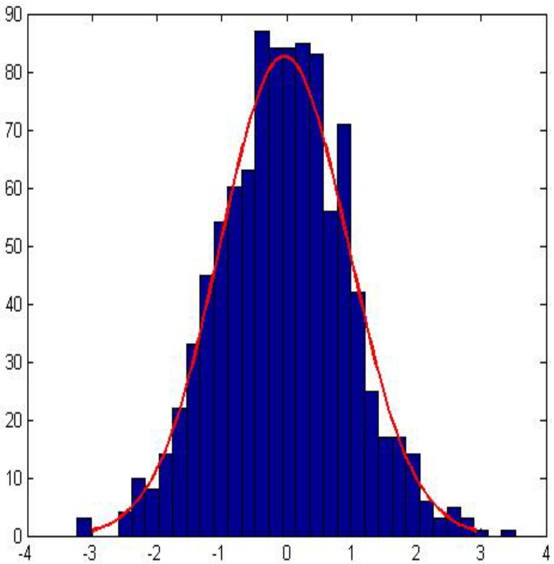
The mapping or response surface can be achieve in some means without doing so many model evaluations. This method will be covered in the next two lectures

Lecture 4&5: Polynomial Chaos --- Galerkin and collocation

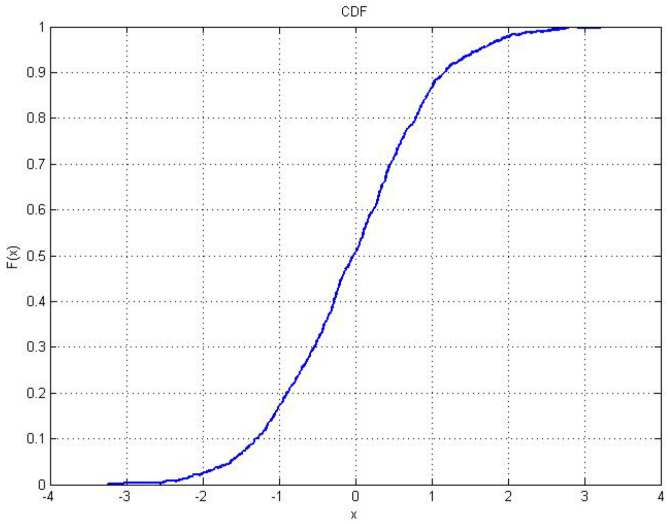
Step 4. Display Uncertainty

probability density function (PDF) of a continuous random variable is a function that describes the relative likelihood for this random variable to occur at a given point in the observation space.

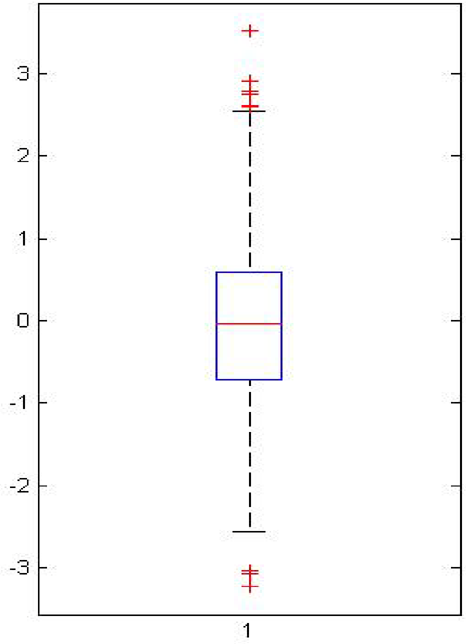
cumulative distribution function (**CDF**), or just **distribution function**, describes the probability that a real-valued random variable *X* with a given probability distribution will be found at a value less than *x*.



Histogram and PDF

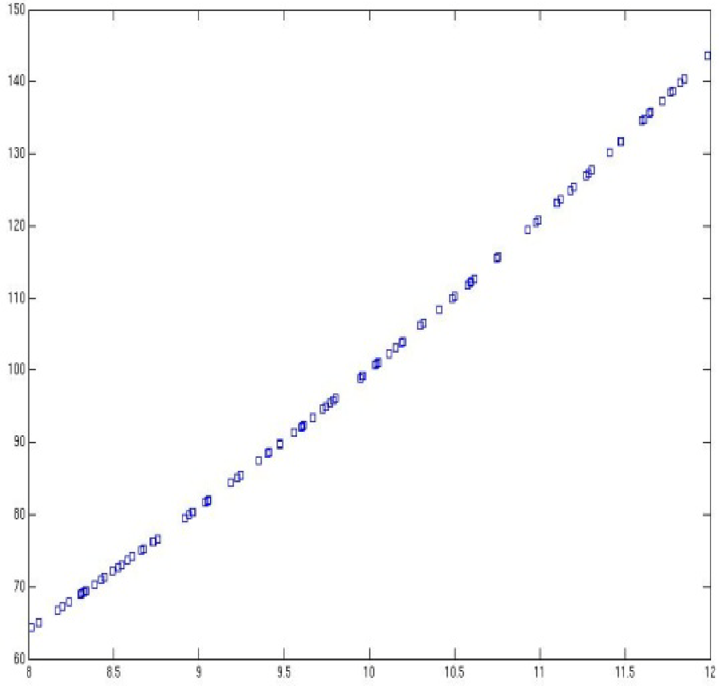


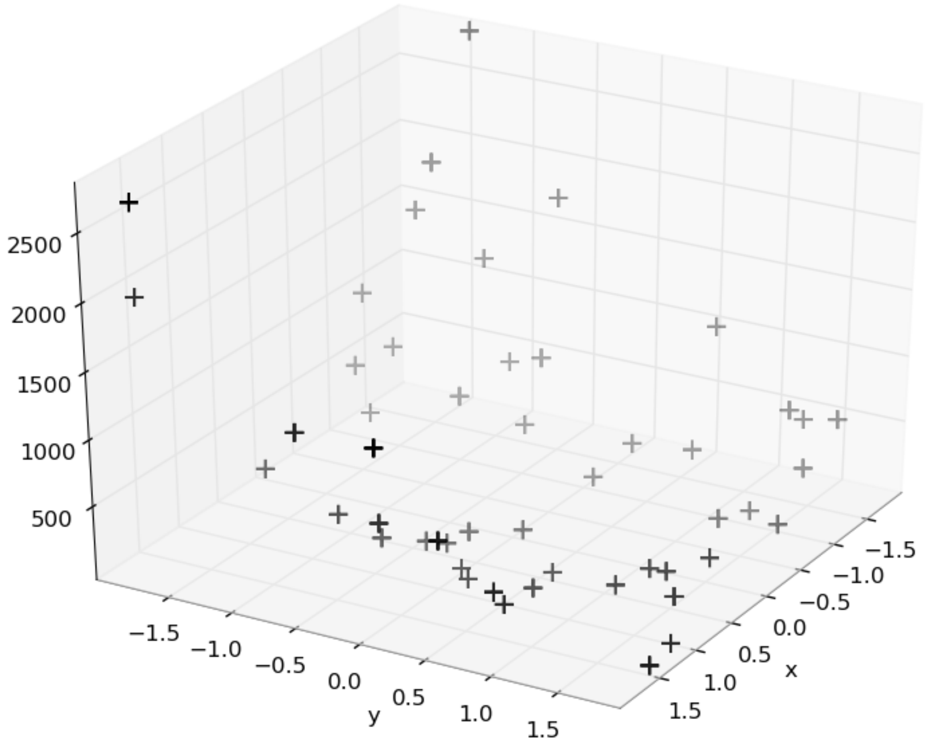
CDF



Boxplot

Step 5. Sensitivity Analysis





Scatter plot

PRISM
NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems

Matlab Exercise

Random number generator:

- rand(m,n) uniform random number between 0 and 1
- randn(m,n) normally distributed random number between 0 and 1
- normrnd(mu,sigma,m,n) mu is mean and sigma is std
- lhsnorm(mu,sigma2,n) mu is mean and sigma2 is var

Statistics:

- mean(A) mean value of an array
- var(A) variance of an array
- std(A) standard deviation of an array

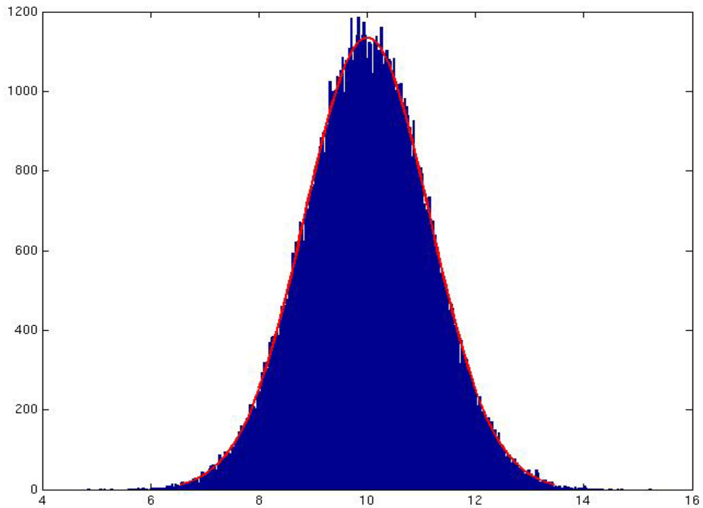
Plot:

- hist(A, nbins) histogram plot of A
- cdfplot(A) or ecdf(A) CDF plot of A

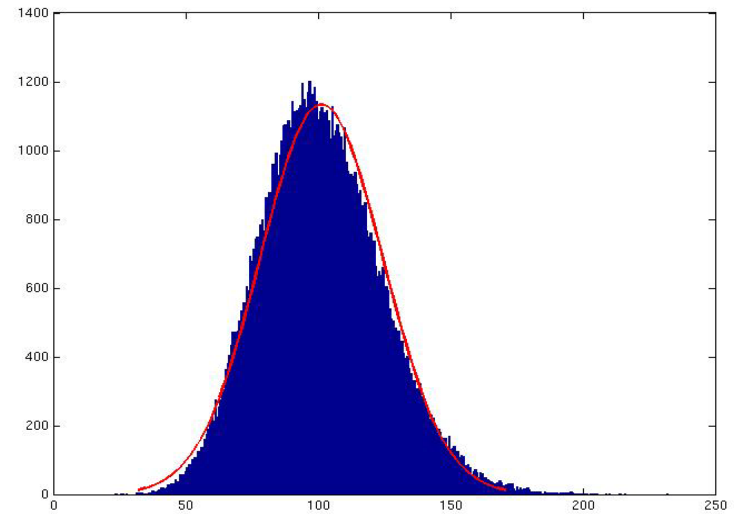
Sampling Methods vs. Variance Propagation

method	X uncertainty	distribution	N sample	Y uncertainty
Monte Carlo	Mean=10; STD=1.155	Normal	10000	Mean=101.331 STD = 23.21
Monte Carlo	Mean=10 Range=[8,12]	Uniform	10000	Mean=101.189 STD=23.11
LHS	Mean=10; STD=1.155	Normal	100	Mean=101.317 STD=23.14
LHS	Mean=10 Range=[8,12]	Uniform	100	Mean=101.333 STD=23.25
Variance	Mean=10; STD=1.155	N/A	N/A	Mean=101.33 STD=23.10

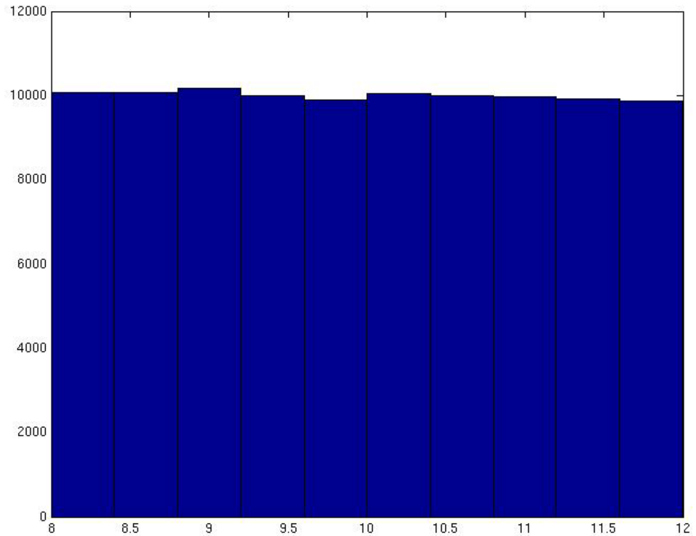
Sampling method gives more than mean and variance



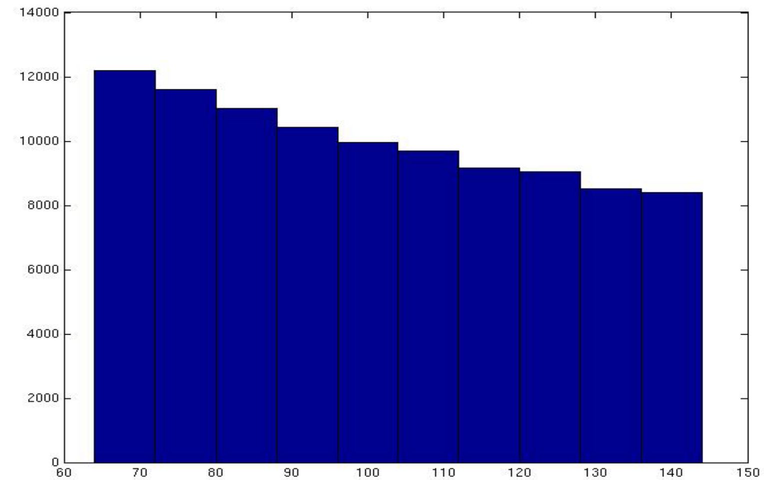
Normal distribution of input



Normal distribution fit of output



Uniform distribution of input



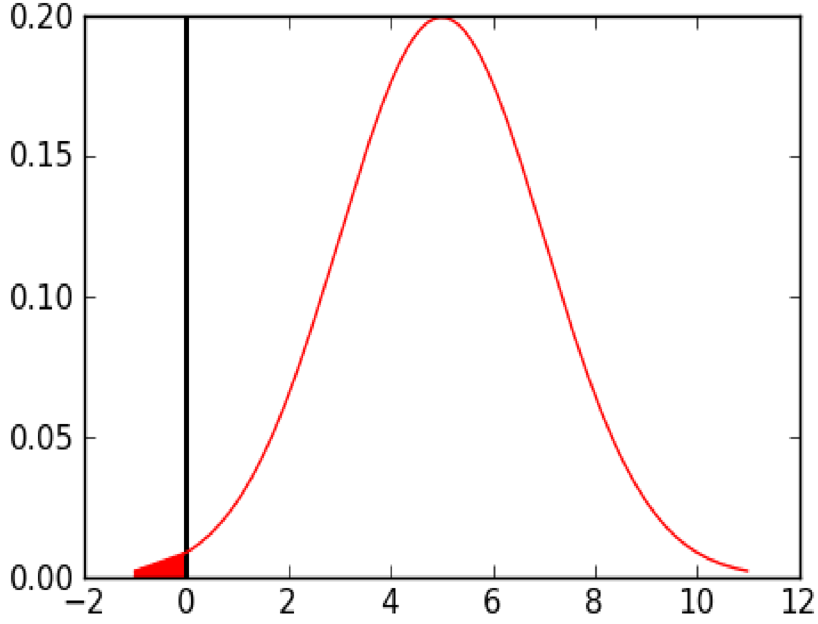
pdf of output

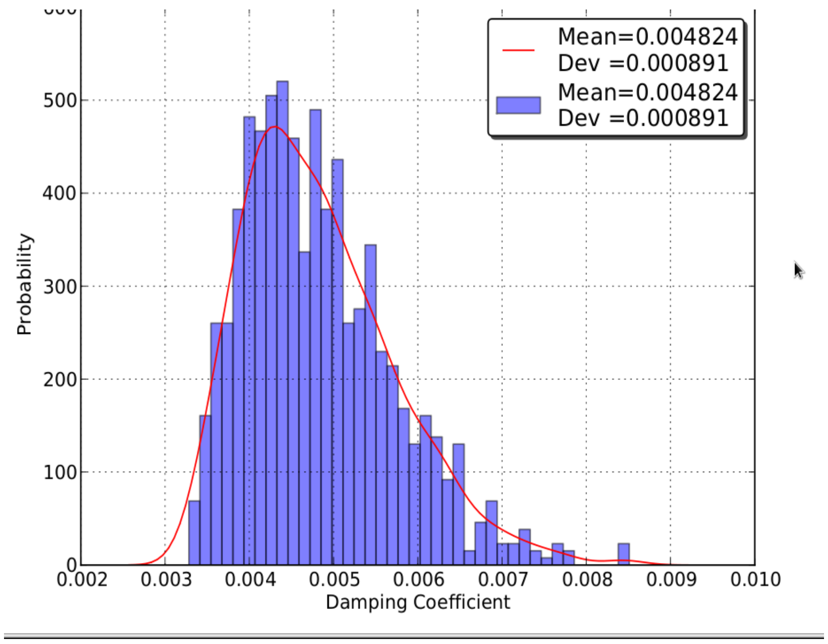
Open question:

how to deal with the uncertainty of input uncertainty distribution?

Non-Standard PDFs

- Often PDFs do not exactly fit any standard analytical PDF.
- They may need to be truncated.
- Or they may just be complicated.

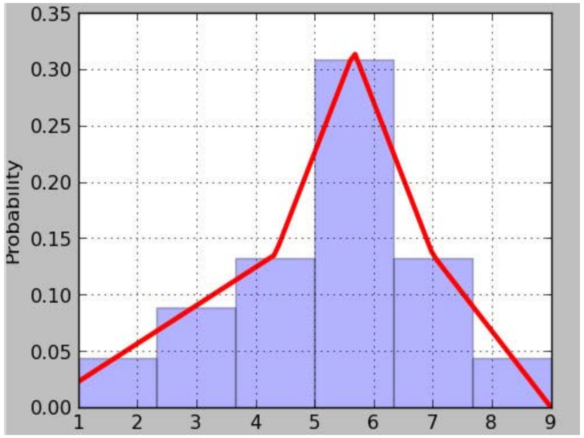


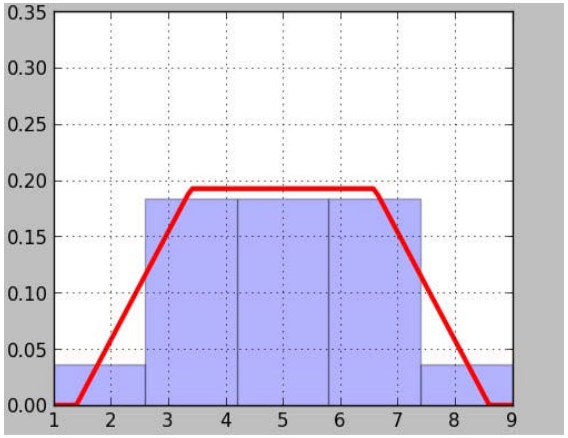


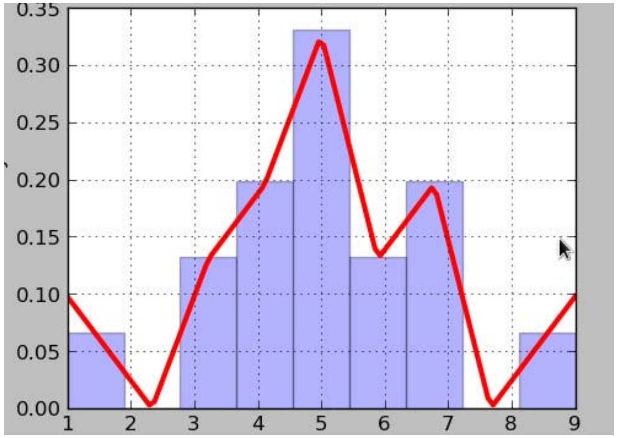
PRISM
NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems

Fitting Experimental Data to a PDF

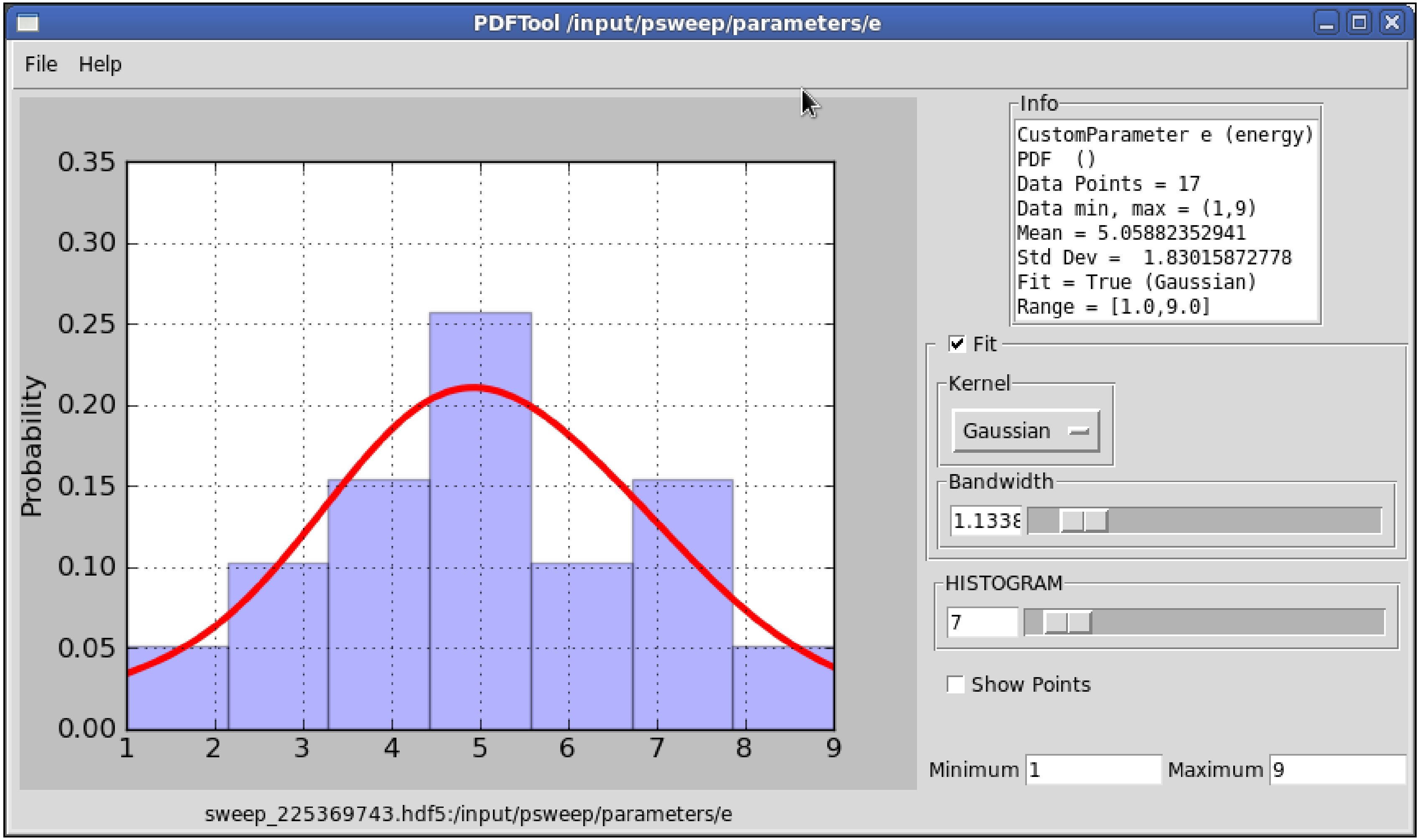
- Histograms and Kernel Density Estimation are useful, but be careful.
- Unless you have plenty of data, no easy solution.







Fitting Experimental Data to a PDF



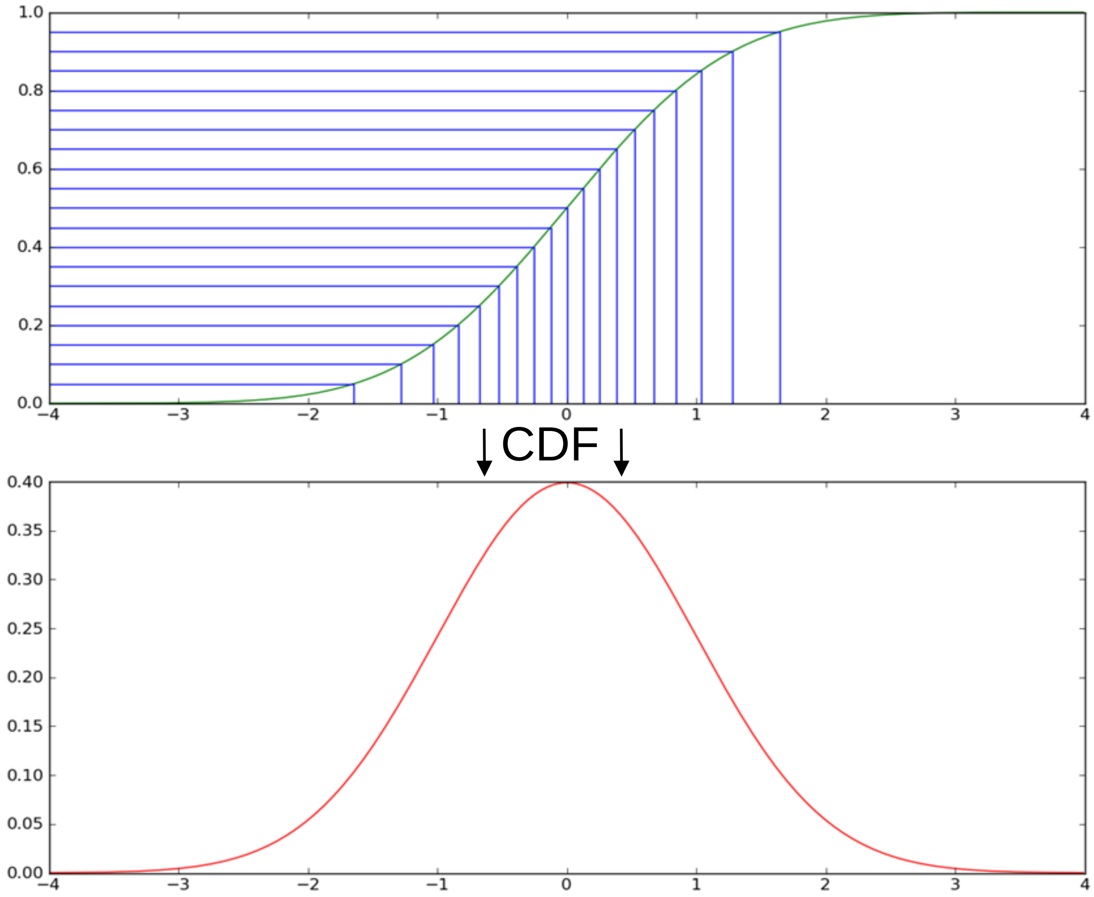
PRISM
NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems

Generating Random Numbers

- We can easily generate random numbers that fit any PDF.
- First we compute a CDF by integrating the PDF.
- Then, using the inverse CDF, or Percent Point Function, we can use uniform random numbers to generate random numbers that fit our PDF.
- This is sometimes called Inverse Transform Mapping

Generating Random Numbers

Uniform Samples



Generating Random Samples

```
def random(self, num):
    return self.ppf(random.uniform(0, 1, num))

def lhs(self, num):
    return random.permutation(self.ppf((arange(0, num) +
random.uniform(0,1,num)) / num))
```

PRISM
NNSA Center for Prediction of Reliability, Integrity and Survivability of Microsystems