Uncertainty Analysis: Sampling Methods

Thierry A. Mara

University of Reunion Island

11^{th} SAMO Summer School, June 6-10 2022, (Online event)

Outline

Introduction

Sampling $\mathbf{u} \sim \mathcal{U}(0,1)^d$

- The uniform distribution $\mathcal{U}(0,1)$
- A measure of discrepancy
- Random sampling
- Latin hypercube sampling
- Quasi-Monte Carlo sampling

From $\mathbf{u} \sim \mathcal{U}(0,1)^d$ to $\mathbf{x} \sim p_{x_1} \times \cdots \times p_{x_d}$

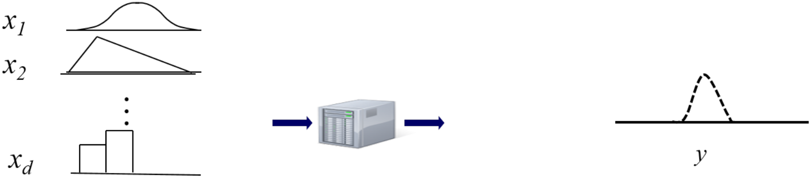
Conclusion

Homework

Uncertainty Analysis by Monte Carlo simulations

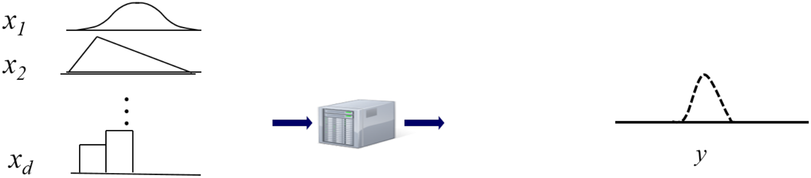
UA

Let $y = \mathcal{M}(\boldsymbol{x})$ be the scalar model response of interest, with $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)$.



UA

Let $y = \mathcal{M}(\boldsymbol{x})$ be the scalar model response of interest, with $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)$.



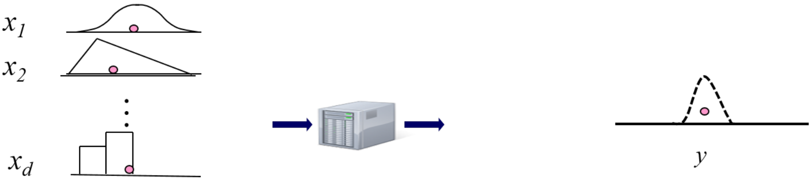
The goal of UA is to infer p_y, the probability density function (pdf) of the model response of interest, knowing p_x the joint pdf of the model input.

For this purpose, Monte Carlo simulations are usually used.

Monte Carlo simulations

Monte Carlo simulations is a numerical way to propagate the input uncertainty into the model.

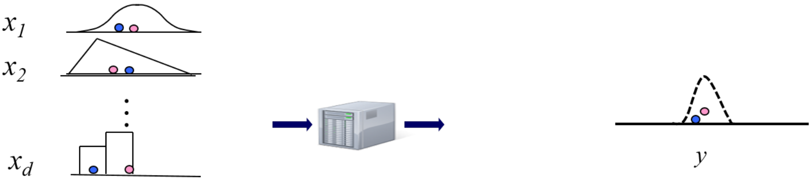
The idea is to generate several draws of $\boldsymbol{x} \sim p_x$ and for each draw, run the model and collect the model response



Monte Carlo simulations

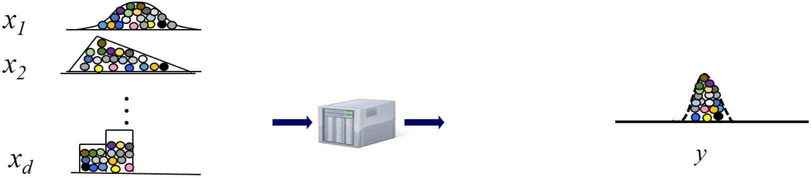
Monte Carlo simulations is a numerical way to propagate the input uncertainty into the model.

The idea is to generate several draws of $\boldsymbol{x} \sim p_x$ and for each draw, run the model and collect the model response



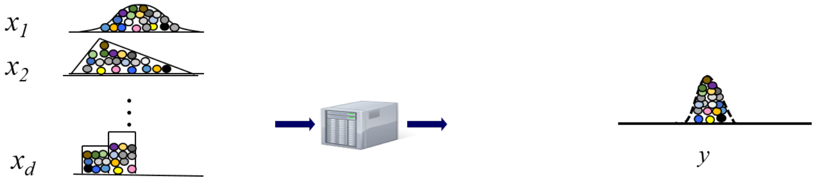
Monte Carlo simulations

Monte Carlo simulations is a numerical way to propagate the input uncertainty into the model.
The idea is to generate several draws of $\boldsymbol{x} \sim p_x$ and for each draw, run the model and collect the model response



Monte Carlo simulations

Monte Carlo simulations is a numerical way to propagate the input uncertainty into the model.
The idea is to generate several draws of $\boldsymbol{x} \sim p_x$ and for each draw, run the model and collect the model response



Then, p_y and other statistics of y can be inferred from the Monte Carlo sample.

How to sample $\boldsymbol{x} \sim p_x$?
First let's see how to sample from $\mathcal{U}(0,1)^d$?

Sampling $\boldsymbol{u} \sim \mathcal{U}(0,1)^d$

The uniform distribution $\mathcal{U}(0,1)$

The uniform distribution $\mathcal{U}(0,1)$

<u>Definition</u>: A random variable (RV) u uniformly distributed over $(0,1)$ has a probability density function (pdf) defined as follows,

$$p_u(u) = \begin{cases} 1 & \text{if } u \in (0,1) \\ 0 & \text{otherwise} \end{cases}$$

The uniform distribution $\mathcal{U}(0,1)$

Definition: A random variable (RV) u uniformly distributed over $(0,1)$ has a probability density function (pdf) defined as follows,

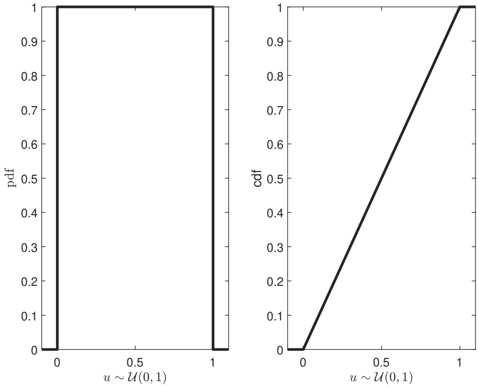
$$p_u(u) = \begin{cases} 1 & \text{if } u \in (0,1) \\ 0 & \text{otherwise} \end{cases}$$

Its cumulative density function (cdf) is,

$$F_u(u) = \int_{-\infty}^{u} \mathrm{d}x = \begin{cases} 0 & \text{if } u < 0 \\ u & \text{if } u \in (0,1) \\ 1 & \text{if } u > 1 \end{cases}$$

u is completely defined by p_u or F_u.

The uniform distribution $\mathcal{U}(0,1)$

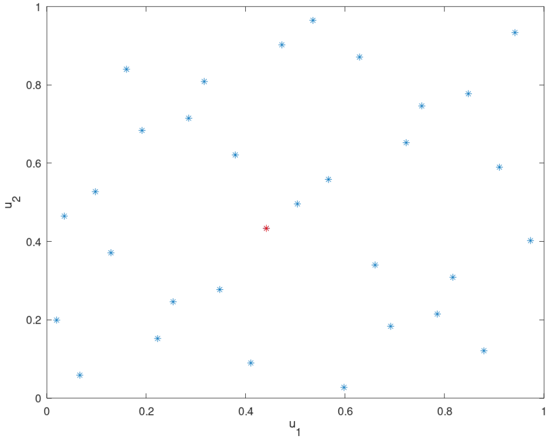


Most of the programming languages contain by default a pseudo-random generator for $\mathcal{U}(0,1)$. For example,
Matlab/Octave $->$ rand
R $->$ runif

A measure of discrepancy

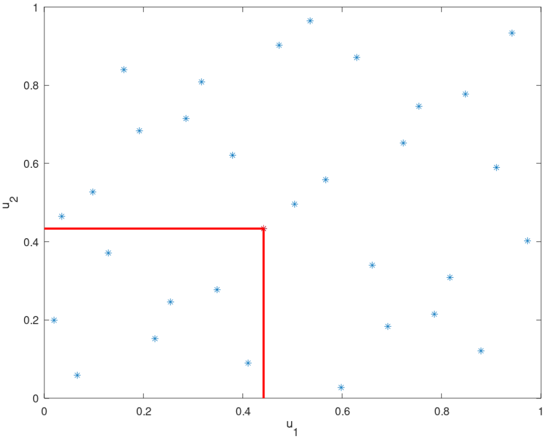
Discrepancy measure

Let $\mathbf{U}$ be a sample of size $N \times d$ generated with a pseudo-random sampler of $\mathcal{U}(0,1)^d$.



Discrepancy measure

Any draw $\mathbf{u}_k$ in this sample (i.e. any row of the sample matrix $\mathbf{U}$) defines a (sub-)hypercube of volume $Vol_k = \prod_{i=1}^{d} u_{ki}$ with u_{ki} the element of $\mathbf{U}$ at row k and column i.



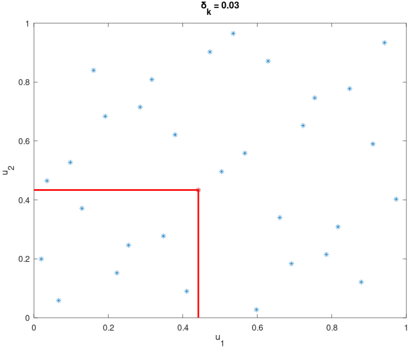
Discrepancy measure

Let us denote by N_k the number of points in this (sub-)hypercube.

It is expected that: $\frac{N_k}{N} \approx Vol_k$

Definition: the **local discrepancy** of $\mathbf{u}_k$ is defined as,

$$\delta_k = \left| \frac{N_k}{N} - Vol_k \right|$$



Discrepancy measure

<u>Definition</u>: the **star discrepancy** of the sample is defined as,

$$D_N = \sup_{k \in \{1,\dots,N\}} \delta_k$$

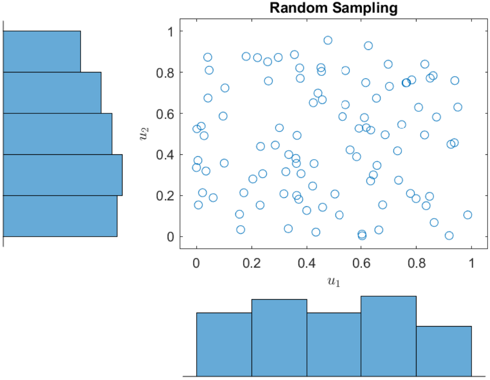
- Several generators exist to sample from $\mathcal{U}(0,1)^d$
- Discrepancies measure how well a given sample uniformly covers the unit hypercube $(0,1)^d$
- The smaller the discrepancy the better is the generator
- **Star discrepancy** is one measure of discrepancy

(Pseudo-)Random Sampling

Random Sampling

Most of the programming languages contain by default a pseudo-random generator also called random sampler of $\boldsymbol{u} \sim \mathcal{U}(0,1)^d$.
For instance,
Matlab/Octave $->$ rand(N,d)
R $->$ runif(N,d)

Random Sampling



In d-dimension, RS does not cover well the unit hypercube (i.e. poor discrepancy measure)

Latin Hypercube Sampling

Latin hypercube sampling

An Intuitive Approach: How to draw N values of u uniformly within $(0,1)$?

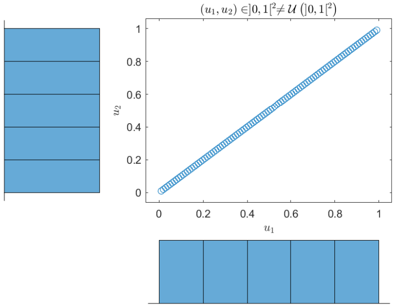
Set $\mathbf{u} = \left(\frac{1-\frac{1}{2}}{N}, \frac{2-\frac{1}{2}}{N}, \ldots, \frac{N-\frac{1}{2}}{N} \right)$.

Latin hypercube sampling

An Intuitive Approach: How to draw N values of u uniformly within $(0, 1)$?

Set $\mathbf{u} = \left(\frac{1-\frac{1}{2}}{N}, \frac{2-\frac{1}{2}}{N}, \ldots, \frac{N-\frac{1}{2}}{N}\right)$.

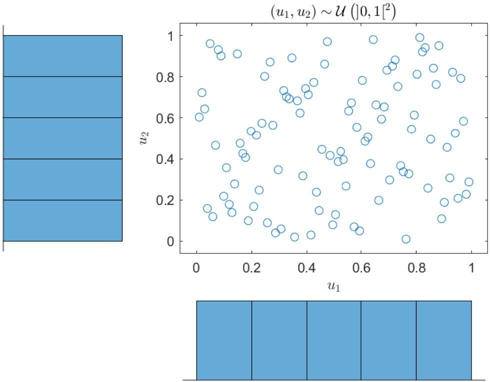
If now we assign these draws to u_1 and u_2, we obtain



which is not $\mathcal{U}(0, 1)^2$.

Latin hypercube sampling

To circumvent this issue, randomize (i.e. randomly permute) the values of u_1 and u_2



This is known as the Latin Hypercube Sampling (LHS).

About Latin hypercube sampling

- High discrepancy of basic LHS is observed at low sample size
- It is possible to optimize the pairing of the scrambled values in order to reduce the discrepancy (e.g., Optimized LHS, ...)

Remark: LHS was developed in the 70's by statisticians from the SANDIA Laboratory (USA).

Low-discrepancy sequences

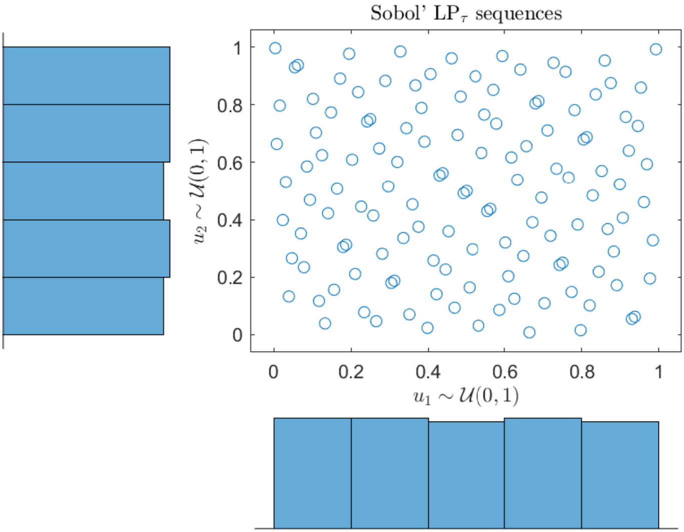
Low-discrepancy sequences

Several statisticians have proposed algorithms to generate draws with low discrepancy (ex, Halton 1960, Faure 1982, Sobol 1967). They are usually called Quasi Monte Carlo (QMC) sample.

- QMC should be generated with N multiple of 2 to ensure a uniform coverage of $(0,1)^d$
- QMC is deterministic, the same input draws are provided (Randomized QMC can circumvent this issue)
- At low sample sizes, QMC draws are correlated

The LP_τ-sequences of Sobol' is one such QMC sampler. They are provided in,
Matlab/Octave $->$ LPTAU51
R $->$ sobol

Low-discrepancy sequences



$\boldsymbol{u} \sim \mathcal{U}(0,1)^d \rightarrow \boldsymbol{x} \sim p_{x_1} \times \cdots \times p_{x_d}$

From $\boldsymbol{u} \in \mathcal{U}(0,1)^d$

to

$\boldsymbol{x} \sim p_{x_1} \times \cdots \times p_{x_d}$

The Integral Transform

The integral transform: Let $\boldsymbol{x} \sim p_{x_1} \times \cdots \times p_{x_d}$ be a random vector of **independent** RVs arbitrary distributed, and $F_{x_1}, \ldots, F_{x_d}$ be their associated cdfs.

The Integral Transform

The integral transform: Let $\boldsymbol{x} \sim p_{x_1} \times \cdots \times p_{x_d}$ be a random vector of **independent** RVs arbitrary distributed, and $F_{x_1}, \ldots, F_{x_d}$ be their associated cdfs.

From a random vector $\boldsymbol{u} \sim \mathcal{U}(0,1)^d$, it is straightforward to derive $\boldsymbol{x}$ with the following integral transformation,

$$x_i = F_{x_i}^{-1}(u_i) \tag{1}$$

with $i = 1, \ldots, d$.

Given a sample $\mathbf{U}$ of $\boldsymbol{u}$ Eq.(1) allows to generate a sample $\mathbf{X}$ of $\boldsymbol{x}$.

The Integral Transform

Some analytical integral transforms,

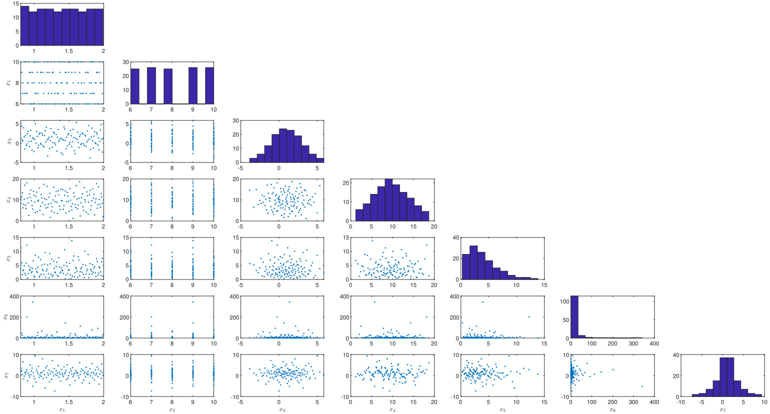
- if $x \sim \mathcal{U}(x|a, b)$ then $x = u(b-a) + a$
- if $x \sim \mathcal{DU}(x|l_1, l_2)$, $l_j \in \mathbb{Z}$, then $x = \mathrm{E}[(l_2 - l_1 + 1)u] + l_1$, where E is the integer part operator
- if $x \sim \mathcal{N}(x|\mu, \sigma^2) = (2\pi\sigma^2)^{-1/2}\mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}$, then $x = \sigma\sqrt{2}\mathrm{erf}^{-1}(2u-1) + \mu$, erf is the error function.
- if $x \sim \mathcal{T}(x|a, b, c)$, then
$$x = \begin{cases} a + \sqrt{u(b-a)(c-a)} & \text{if } 0 < u < \frac{c-a}{b-a} \\ b - \sqrt{(1-u)(b-a)(b-c)} & \text{otherwise} \end{cases}.$$

Useful functions:
Matlab $->$ erfinv ($= \mathrm{erf}^{-1}$), gaminv (inverse cdf of Γ law),
R $->$ qnorm ($= \sqrt{2}\mathrm{erf}^{-1}(2u-1)$), qgamma (inverse cdf of Γ law),

Exercise

Exercise 1: Set $N = 128$, $d = 7$, and generate $\mathbf{U}$ a sample over $\mathcal{U}(0,1)^7$. From the sample $\mathbf{U}$, deduce the sample $\mathbf{X}$ such that

1. x_1 uniformly distributed over $\mathcal{U}(x_1|0.8, 2)$
2. $x_2 \sim \mathcal{DU}(x_2|6, 10)$
3. $x_3 \sim \mathcal{N}\left(x_3|1, 2^2\right)$
4. $x_4 \sim \mathcal{T}(x_4|0, 20, 9)$
5. $x_5 \sim \Gamma(x_5|2, 2)$, $\Gamma(x|k, \theta) \propto x^{k-1} e^{-\frac{x}{\theta}}$, $x \geq 0$
6. $x_6 \sim \mathcal{N}\left(\ln(x_6)|1, 2^2\right)$, i.e. Log-Normal distribution



x_1
x_2
x_3
x_4
x_5
x_6
x_7

Conclusion

$$y = \mathcal{M}(\boldsymbol{x}) = f(\boldsymbol{u})$$

$y = \mathcal{M}(\boldsymbol{x}) = f(\boldsymbol{u})$

Let $y = \mathcal{M}(\boldsymbol{x})$ be the scalar model response of interest with $\boldsymbol{x} \sim p_{x_1} \times \cdots \times p_{x_d}$.

$y = \mathcal{M}(\boldsymbol{x}) = f(\boldsymbol{u})$

Let $y = \mathcal{M}(\boldsymbol{x})$ be the scalar model response of interest with $\boldsymbol{x} \sim p_{x_1} \times \cdots \times p_{x_d}$.
If $\boldsymbol{x}$ is derived from $\boldsymbol{u} \sim \mathcal{U}\left(0, 1\right)^d$ by the integral tranformation,

$$x_i = F_{x_i}^{-1}(u_i)$$

with $i = 1, \ldots, d$.

$y = \mathcal{M}(\boldsymbol{x}) = f(\boldsymbol{u})$

Let $y = \mathcal{M}(\boldsymbol{x})$ be the scalar model response of interest with $\boldsymbol{x} \sim p_{x_1} \times \cdots \times p_{x_d}$.
If $\boldsymbol{x}$ is derived from $\boldsymbol{u} \sim \mathcal{U}(0,1)^d$ by the integral tranformation,

$$x_i = F_{x_i}^{-1}(u_i)$$

with $i = 1, \ldots, d$.

Performing the uncertainty and sensitivity analysis (UASA) of y w.r.t. $\boldsymbol{x}$ boils down to perform UASA w.r.t. $\boldsymbol{u}$, since

$$y = \mathcal{M}(x_1, \ldots, x_d) = \mathcal{M}(F_{x_1}^{-1}(u_1), \ldots, F_{x_d}^{-1}(u_d)) = f(\boldsymbol{u})$$

Homework

Uncertainty analysis of the borehole model

taken from https://www.sfu.ca/~ssurjano/

Homework

The function models water flow through a borehole. The response of interest φ is the flow rate (in m^3/yr) defined as follows,

$$\varphi(\boldsymbol{x}) = \frac{2\pi T_u \left(H_u - H_l\right)}{\ln\left(r/r_w\right)\left(1 + \frac{2LT_u}{\ln(r/r_w) r_w^2 K_w} + \frac{T_u}{T_l}\right)}$$

The uncertain input variables, assumed **independent** of each other, are $\boldsymbol{x} = (r_w, r, T_u,\ H_u, T_l, H_l, L, K_w)$. Their marginal pdfs are given below,

Radius of Borehole (m)	$\mathcal{N}(r_w \vert 0.11, 0.017^2)$
Radius of Influence (m)	$\mathcal{N}(\ln(r) \vert 7.71, 1)$
Transmittivity of Upper Aquifer (m^2/yr)	$\mathcal{T}\left(T_u \vert 63\,070, 115\,600, 100\,000\right)$
Pressure Head of Upper Aquifer (m)	$\mathcal{N}\left(\ln(H_u) \vert 6.95, 0.0167^2\right)$
Transmittivity of Lower Aquifer (m^2/yr)	$\mathcal{U}\left(T_l \vert 63, 116\right)$
Pressure Head of Lower Aquifer (m)	$\mathcal{N}\left(\ln(H_l) \vert 6.6, 0.033^2\right)$
Length of Borehole (m)	$\mathcal{U}(L \vert 1\,120, 1\,680)$
Hydraulic Conductivity (m/yr)	$\mathcal{U}(K_w \vert 9\,855, 12\,045)$

Propagate the input uncertainty into the model response through 1 000 Monte Carlo simulations. Plot the histogram of φ and get an estimate of its mathematical expectation and of its variance.