VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF CIVIL ENGINEERING | DEPARTMENT OF STRUCTURAL MECHANICS

Monte Carlo method

- Law of large numbers
- Generators of (pseudo)random numbers
- SBRA method, AntHill software

Probabilistic Methods

Simulation methods

Simple simulation Monte Carlo,

Stratified simulation techniques:

Latin Hypercube Sampling – LHS,
Stratified Sampling – SC.

Advanced simulation methods:

Importance Sampling – IS,
Adaptive Sampling – AS,
Axis Orthogonal Importance Sampling,
Directional Sampling – DS,
Line Sampling – LS,
Design Point Sampling,
Subset Simulations,
Descriptive Sampling, Slice Sampling.

Approximation methods

- First (Second) Order Reliability Method - FORM (SORM),
- Response Surface Method – RSM,
- Perturbation techniques – e.g. Stochastic Finite Element Method (SFEM),
- Artificial Neural Network – ANN.

Pure numerical methods

(without simulations and approximations)

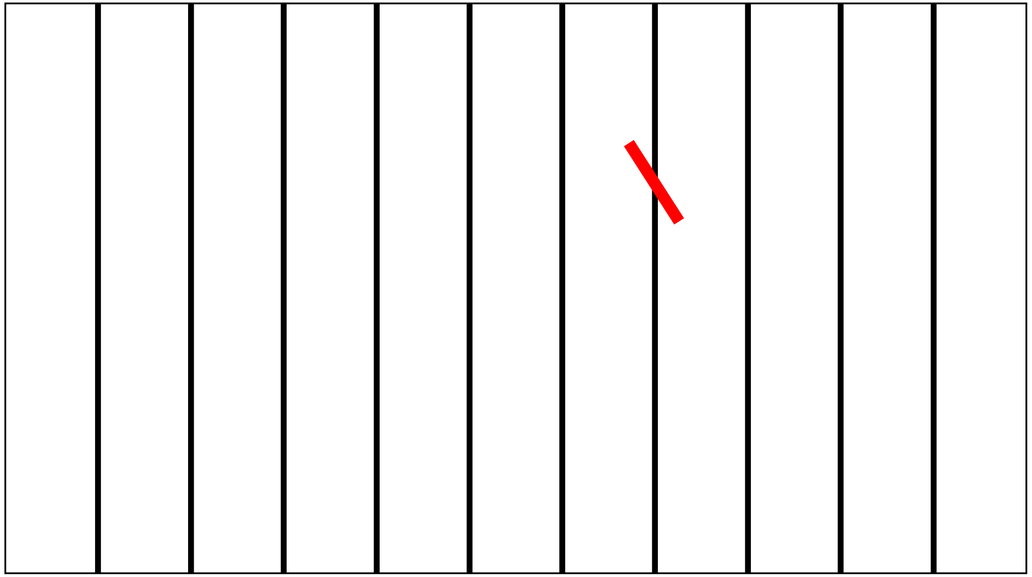
- Point Estimate Method – PEM,
- Direct Optimized Probabilistic Calculation – DOProC .

Overview e.g.:
Krejsa & Králik (2015)

Buffon‘s Needle

One of the oldest methods described use cases is the **Buffon's needle problem**, titled according the French mathematician Georges-Louis Leclerc, Comte de Buffon, who in 1777 attempted to estimate the value of π number using p random throwing needles on lined paper.

Georges-Louis Leclerc, Comte de Buffon
(1707-1788)



Probability of an event when the needle is the same length as the distance between the lines after the impact of the paper will lie on the paper so that it intersects one of the lines equals:

$$p = \frac{2}{\pi}$$

Calculation of the Number π

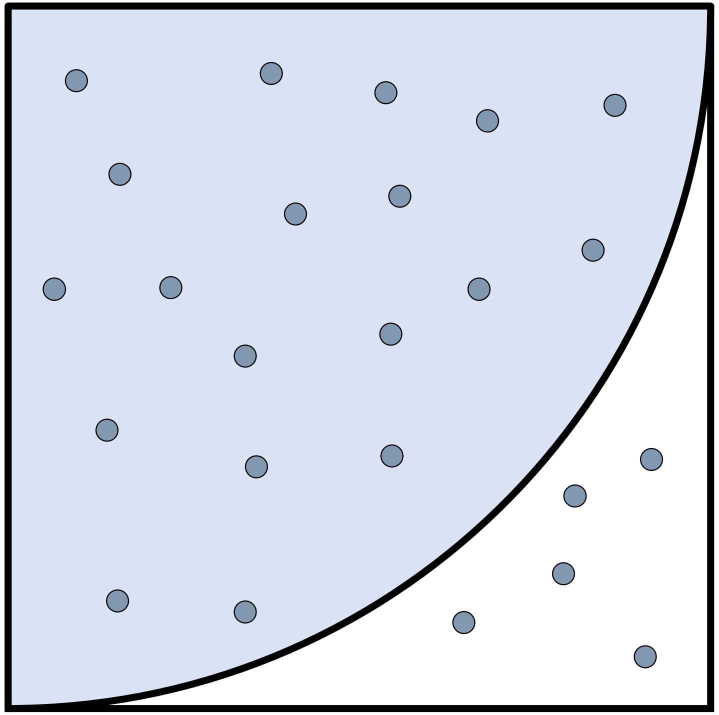
Similarly, the value of π number is possible to determine as follows :

Quarter of circle area: $S_1 = \frac{\pi . r^2}{4}$

Square area: $S_2 = r^2$

Both area ration: $\frac{S_1}{S_2} = \frac{\pi \cdot r^2}{4 \cdot r^2} = \frac{\pi}{4}$

π number equals: $\pi = 4 \cdot \frac{S_1}{S_2}$



The calculation of π number is based on a square of radius r, into which small objects are randomly thrown. The resulting ratio of the total number of throws and throws into quadrant determines the value of π number.

The First Systematic Use of Method

Enrico Fermi
(1901-1954)

Probably the first systematic use of **Monte Carlo simulation** with real results is dated to 1930, when the Nobel Prize winning Italian physicist **Enrico Fermi** used this approach to generate random numbers to calculate the properties at the time the newly discovered particles - neutrons.

The method name comes from **Stanislaw Ulam**, polish mathematic, who named it by the famous casino in Monaco (Ulam's uncle gambled here). The method was previously used as **Statistical sampling**.

The randomness of events and repeat their occurrence are identical to the activities carried out at the Casino. (Roulette is a simple physical random number generator, much like dice.)

Stanislaw Ulam
(1909-1984)

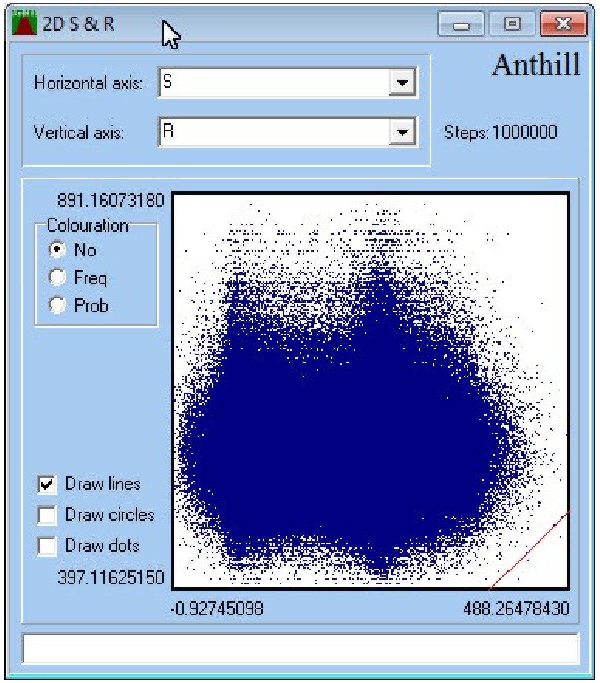
Numerical Integration by Monte Carlo Simulation

The **Monte Carlo simulation method** is mainly used for the calculation of integrals of the probability density functions of continuous random variables, namely multi-dimensional, where conventional methods are not effective.

Monte Carlo simulation method is widely used to:

- simulate random experiments,
- numerical integration of definite integrals,
- numerical solution of differential equations.

The principles of a simple **Monte Carlo simulation method** is based on several probabilistic methods – such as **SBRA** (Simulation-Based Reliability Assessment).



Horizontal axis: S

Vertical axis: R

Steps: 1000000

Colouration

● No

○ Freq

○ Prob

☑ Draw lines

□ Draw circles

□ Draw dots

Advantages and Disadvantages of Monte Carlo

Monte Carlo method is based on the implementation of random experiments with a model of the system and their evaluation. The result of the implementation of a large number of experiments is usually the probability of the random phenomenon under analysis.

The advantage is a **simple implementation**, the disadvantage - relatively low **effectiveness** and **accuracy**:

$$err = \sqrt{\frac{B}{N}}$$

where N is number of random experiments (simulations, simulation steps, samples) and B is constant, depends on specification of problem under solution.

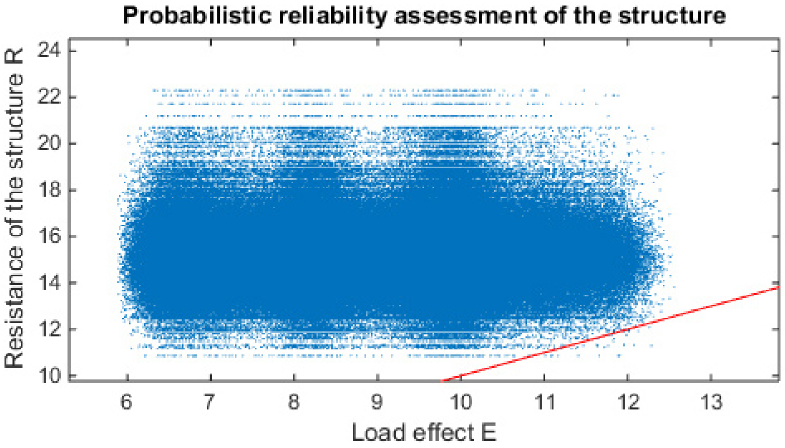
To increase the accuracy of the results of one order is necessary to increase the number of simulations at least two orders of magnitude.

Error Calculation in the Monte Carlo Method

When probabilistic reliability assessment and calculation of the failure probability P_f, the **estimate of accuracy** depends not only on the total number of simulations N, but also on the magnitude of the result in the probability calculation determined by the probability of failure P_f.

The **coefficient of variation** for the small failure probabilities v_{P_f} can be defined using form:

$$v_{P_f} = \frac{1}{\sqrt{N.P_f}}$$



$$P_f = \frac{N_f}{N} \leq P_d$$

Error Calculation in the Monte Carlo Method

E.g.: If the estimated value of the failure probability P_f be in the order of 10^{-4} and calculation was performed with the number of simulation steps $N = 10^4$, coefficient of variation of the failure probability is equal to:

$$v_{P_f} = \frac{1}{\sqrt{10^4 \cdot 10^{-4}}} = 1$$

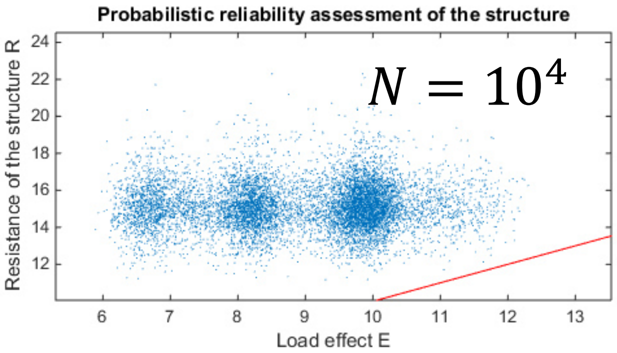
Estimation errors resulting failure probability P_f is therefore 100%.

Increasing the number of simulations to $N = 10^6$ the coefficient of variation of the failure probability achieves more favorable values:

$$v_{p_f} = \frac{1}{\sqrt{10^6 \cdot 10^{-4}}} = 0.1$$

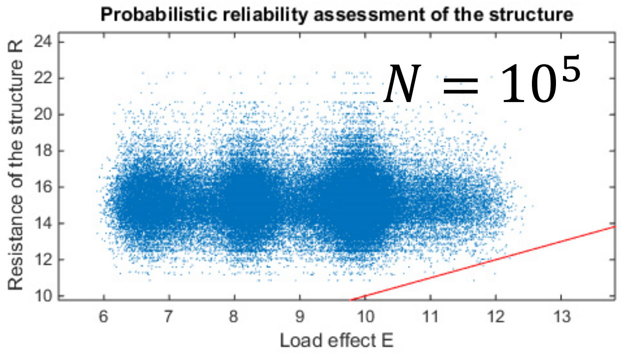
and the results should not be compared to the exact solution differ by 10%.

Error Calculation in the Monte Carlo Method



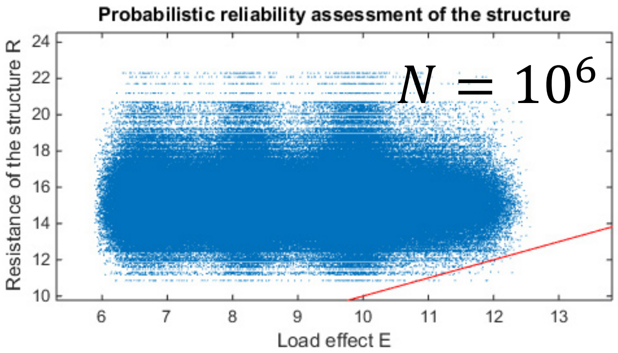
$N = 10^4$

$$P_f = \frac{N_f}{N} = \frac{0}{10^4} = 0$$



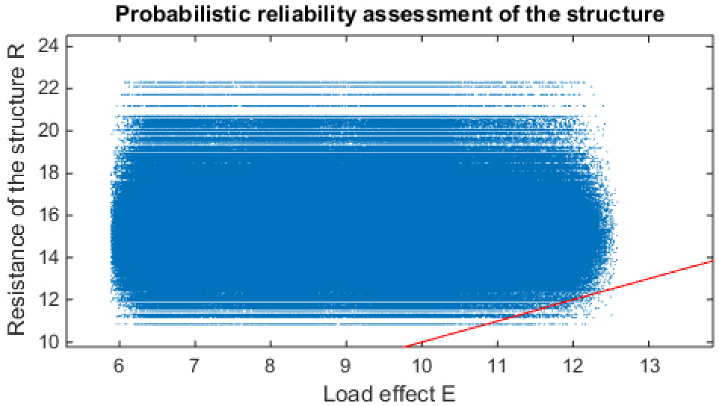
$N = 10^5$

$$P_f = \frac{N_f}{N} = \frac{4}{10^5} = 4 \cdot 10^{-5}$$



$N = 10^6$

$$P_f = \frac{N_f}{N} = \frac{57}{10^6} = 5.7 \cdot 10^{-5}$$



$N = 10^7$

$$P_f = \frac{N_f}{N} = \frac{470}{10^7} = 4.7 \cdot 10^{-5}$$

Law of Large Numbers

With many independent trials can almost certainly expect that the relative frequency will be close to the theoretical value of the probability.

Can be described using the mean random variables:

$$\bar{X}_n = \frac{1}{N}(X_1 + \ldots + X_n)$$

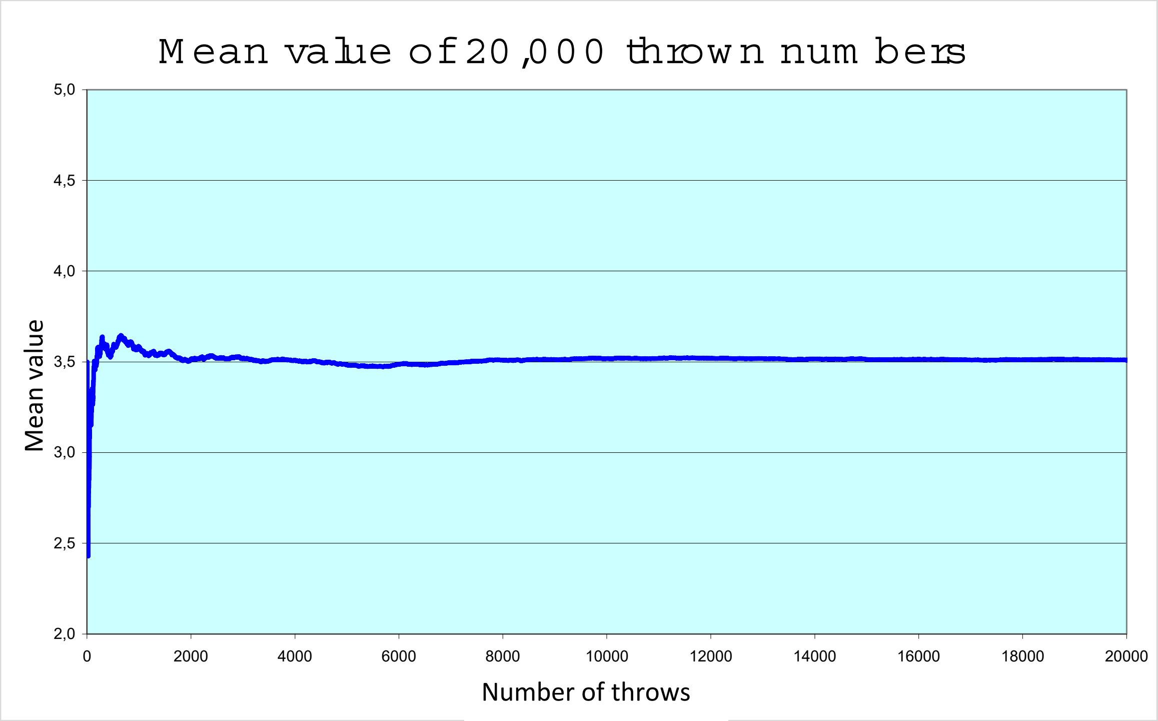
where $X_1, X_2, \cdots, X_n$ an infinite sequence of mutually independent random numbers with finite mean $\mu < \infty$.

With the increasing number of simulations $N \to \infty$ will mean generated sequences converge to the average value $\bar{X}_n \to \mu$, which can be illustrated by a simple example with dice.

Law of Large Numbers

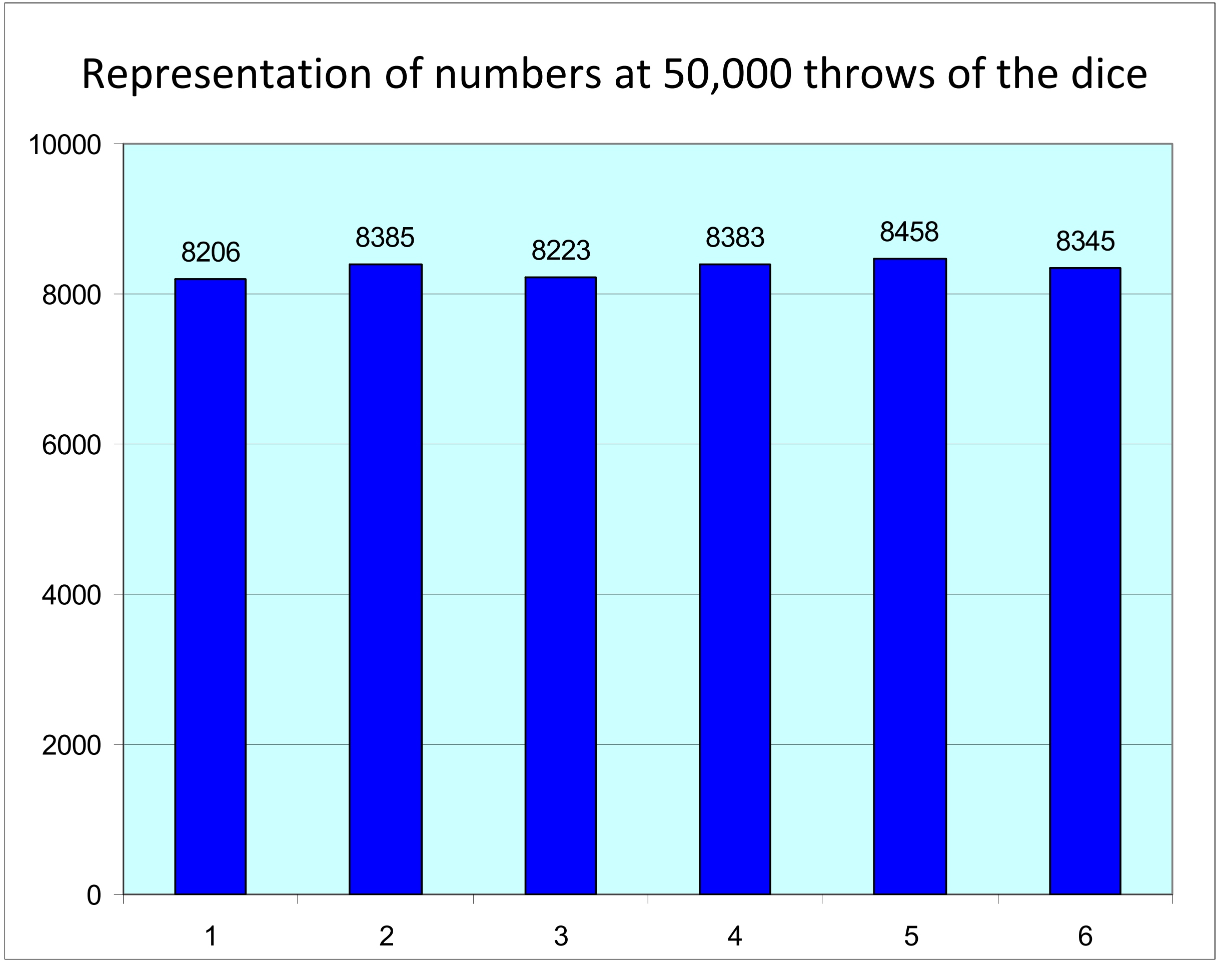
In the case of six-sided dice the mean value of the total of numbers on each side equal to:

$$\mu = \frac{1+2+3+4+5+6}{6} = \frac{21}{6} = 3.5$$



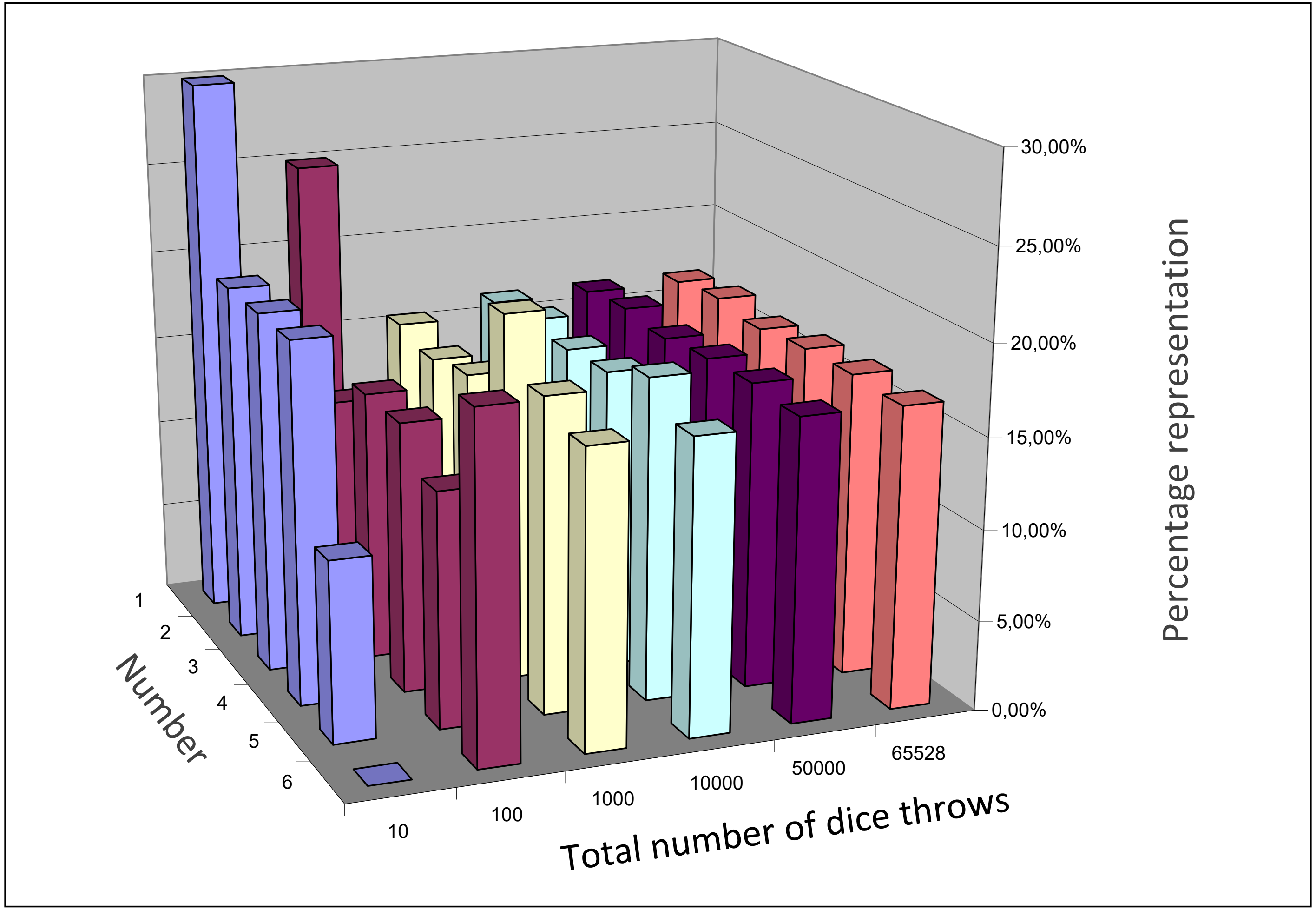
Development of the calculated mean value of 20,000 thrown numbers.

Law of Large Numbers



Representation of numbers at 50,000 throws of the dice

Law of Large Numbers



Percentage of thrown numbers

The total number of throws 65528 is limited by the capacity possibilities of the Excel spreadsheet

Generators of (Pseudo)Random Numbers

Physical random number generators

Congruential <u>pseudo</u>random number generator

$$U_{n+1} = (A \cdot U_n + C) \bmod M$$

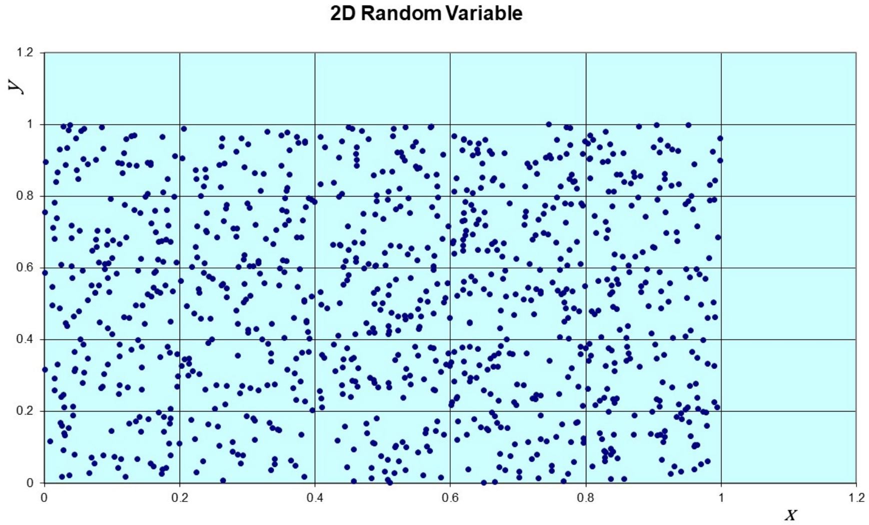
TOUR OF ACCOUNTING
OVER HERE WE HAVE OUR RANDOM NUMBER GENERATOR.
NINE NINE NINE NINE NINE NINE
ARE YOU SURE THAT'S RANDOM?
THAT'S THE PROBLEM WITH RANDOMNESS: YOU CAN NEVER BE SURE.

Congruential Random Number Generator

Derrick Henry Lehmer
(1905-1991)

The most widely used **random number generator**, first introduced by American mathematician Lehmer in 1948.

Used to generate a sequence of random variables with **uniform distribution**.

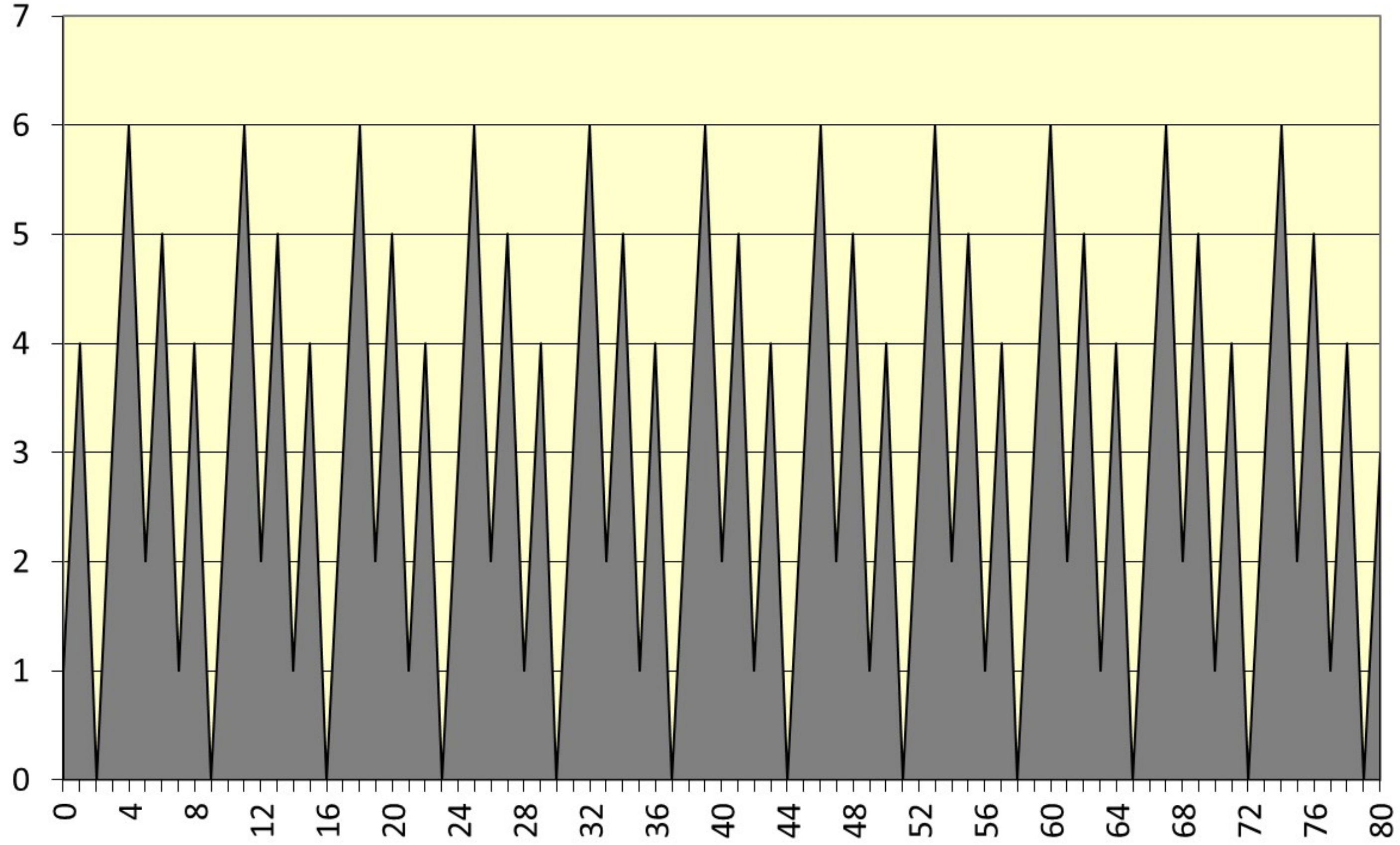


Generating random numbers with recurrent relation:

$$U_{n+1} = (A \cdot U_n + C) \mod M$$

where constants A, C and M determine the statistical quality of the generator.

Input Constants Effect on the Generated Numbers

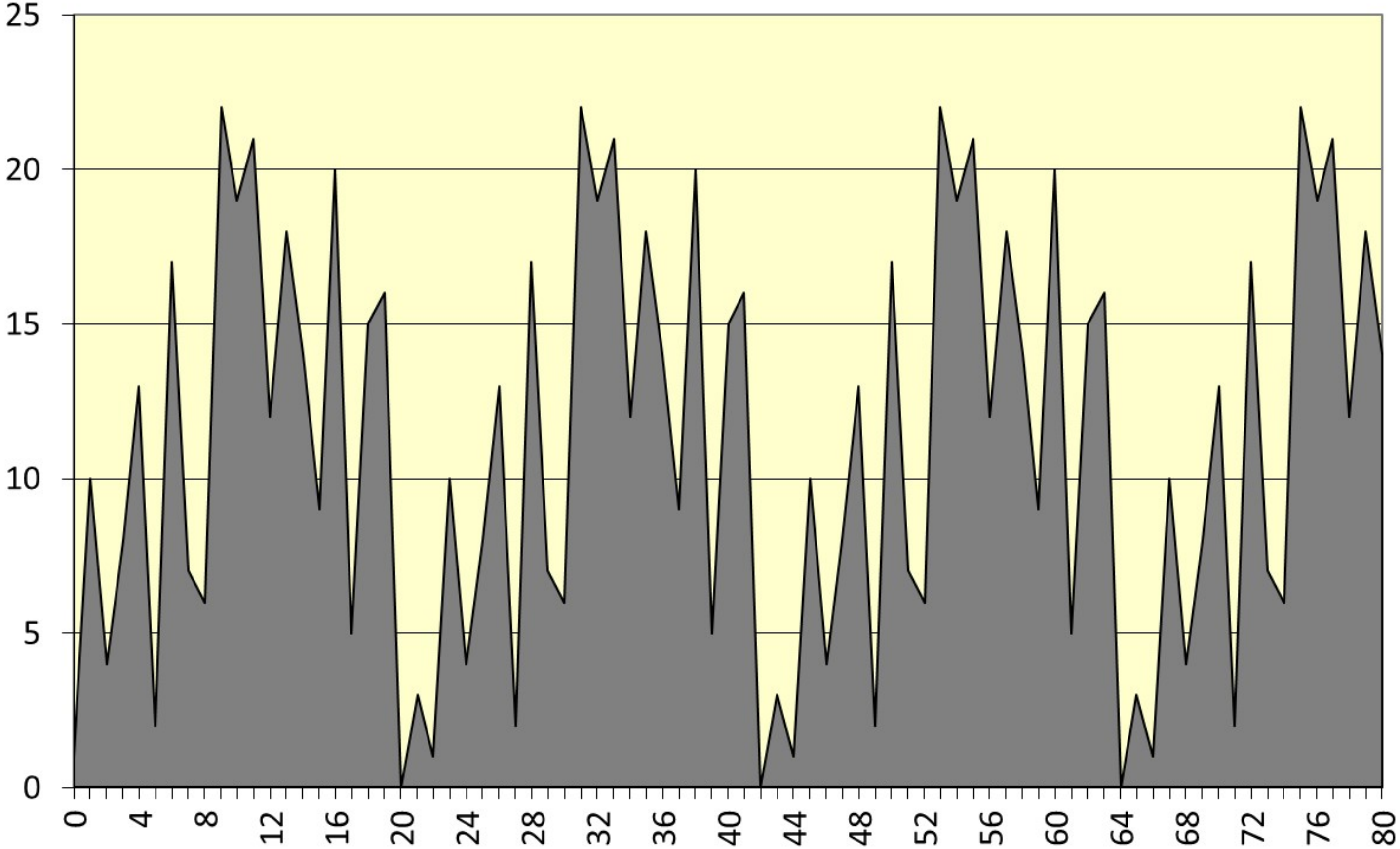


$$U_{n+1} = (A \cdot U_n + C) \bmod M$$

Input values

U_0	**1**
A	**1**
C	**3**
M	**7**

Input Constants Effect on the Generated Numbers

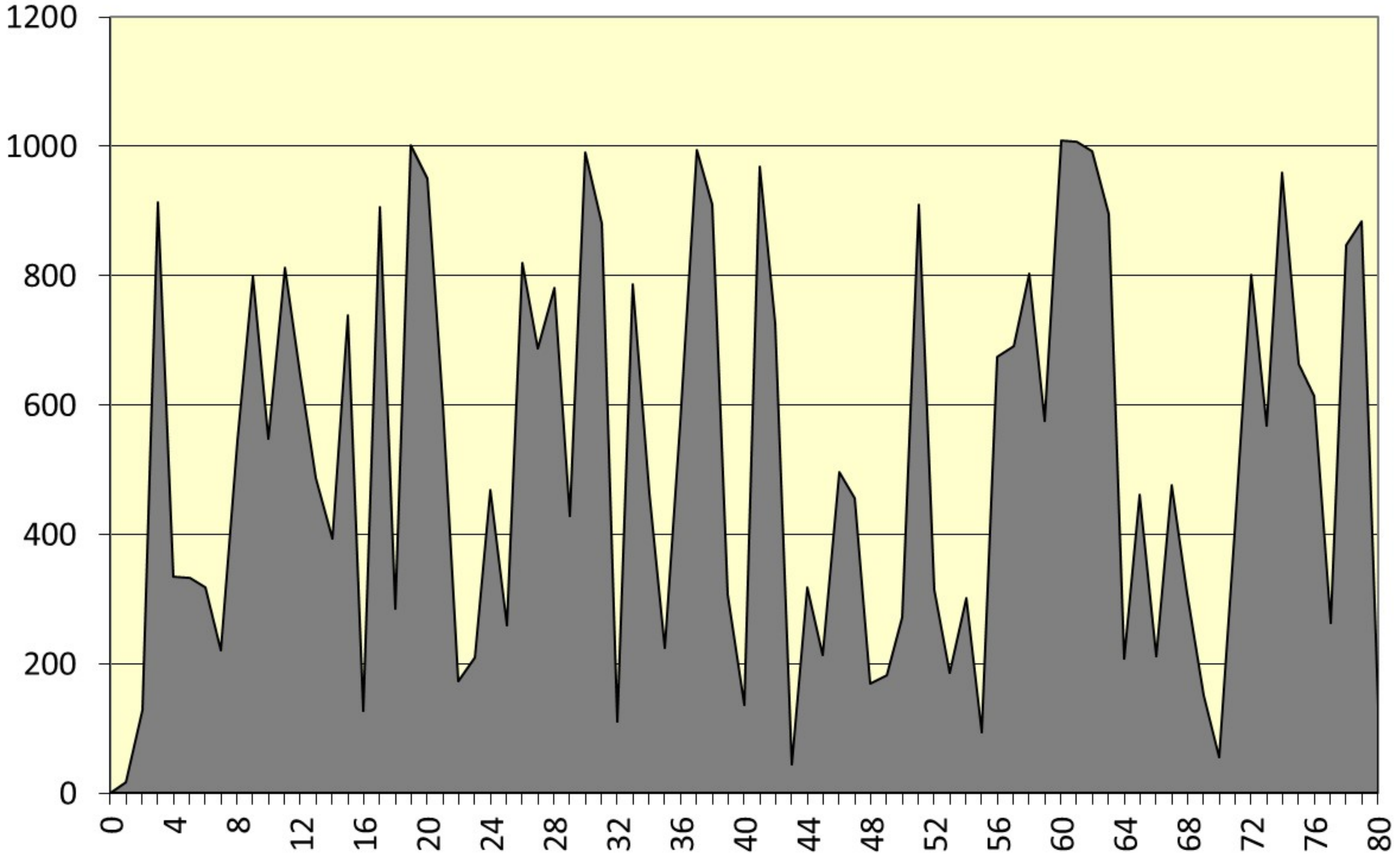


$$U_{n+1} = (A \cdot U_n + C) \bmod M$$

Input values

U_0	**1**
A	**7**
C	**3**
M	**23**

Input Constants Effect on the Generated Numbers

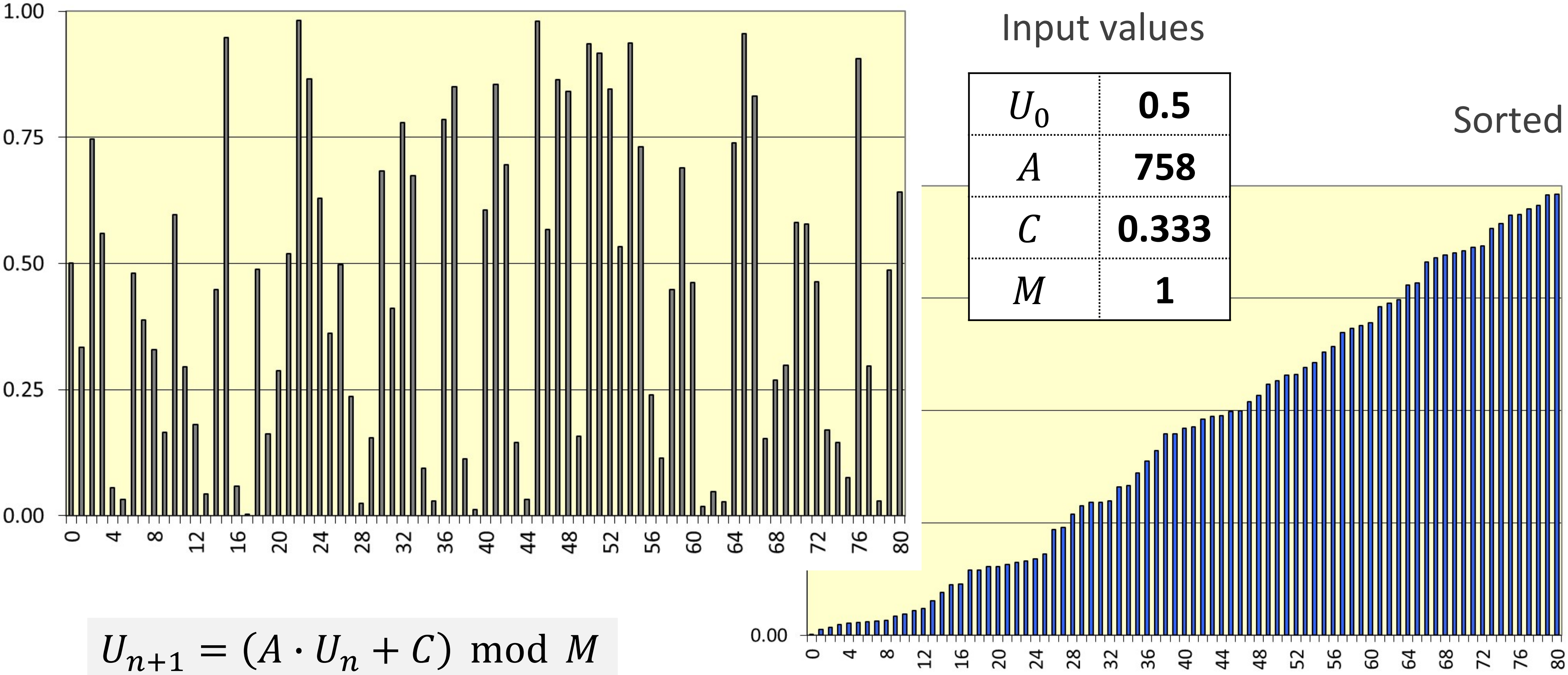


$$U_{n+1} = (A \cdot U_n + C) \bmod M$$

Input values

U_0	**1**
A	**7**
C	**10**
M	**1011**

Input Constants Effect on the Generated Numbers



Input values

U_0	**0.5**
A	**758**
C	**0.333**
M	**1**

$$U_{n+1} = (A \cdot U_n + C) \bmod M$$

Numerical Integration by Monte Carlo Simulation

Monte Carlo method is used most often to solve **multi-dimensional integrals**.

$$I = \int_{x_d}^{x_h} \int_{y_d}^{y_h} f(x, y, \dots)\, \mathrm{d}x\, \mathrm{d}y \dots = \int_{V} f(x, y, \dots)\, \mathrm{d}x\, \mathrm{d}y \dots$$

Numerical integration using Monte Carlo method consists in determining the values of the function f in N random points that lie in the integrated area V. Then, the resulting integral can be defined as:

$$I(f; N) \approx V.\langle f \rangle = \frac{V}{N}.\sum_{i=1}^{N} f_i$$

where $\langle f \rangle$ is mean value of the function f, calculated in N random points.

Numerical Integration by Monte Carlo Simulation

Deviance from the mean value of the function f shows the **standard deviation**:

$$\sigma(f;N) = \sqrt{\frac{\sum_{i=1}^{N}(\langle f\rangle - f_i)^2}{N}}$$

Similarly, deviation can be determined from the mean value of the resulting integral I:

$$\sigma(I;N) = \frac{V}{N}\cdot\sqrt{\sum_{i=1}^{N}(\langle f\rangle - f_i)^2}$$

which can be considered as an **indicator of inaccuracies** in calculation.

Numerical Integration by Monte Carlo Simulation

Algorithm for calculating integrals using **Monte Carlo method** can be more effective. In the integrated area V, it is possible to focus on its known part V^*, where it is easier to generate random points. Established function $\tilde{f}$ then takes the values:

$$\tilde{f} = \begin{cases} 0 & x \notin V \\ f(x) & x \in V \end{cases}$$

After generating N random points lying in the area V^*, then the approximate value of the resulting integral is equal to:

$$I \approx \frac{V^*}{N} \sum_{i=1}^{N} \tilde{f}(x)$$

To demonstrate the process of numerical integration using Monte Carlo simulations serve the following Example 1.

Example 1: Calculation of the Hemisphere Volume

An integrated function is the hemisphere equation: $f(x,y)=\sqrt{r^2-x^2-y^2}$

Area V is the circle with the radius r, area V^* is the square with side length equal to $2\cdot r$, into which the circle is inscribed.
The function $\tilde{f}$ will then take values:

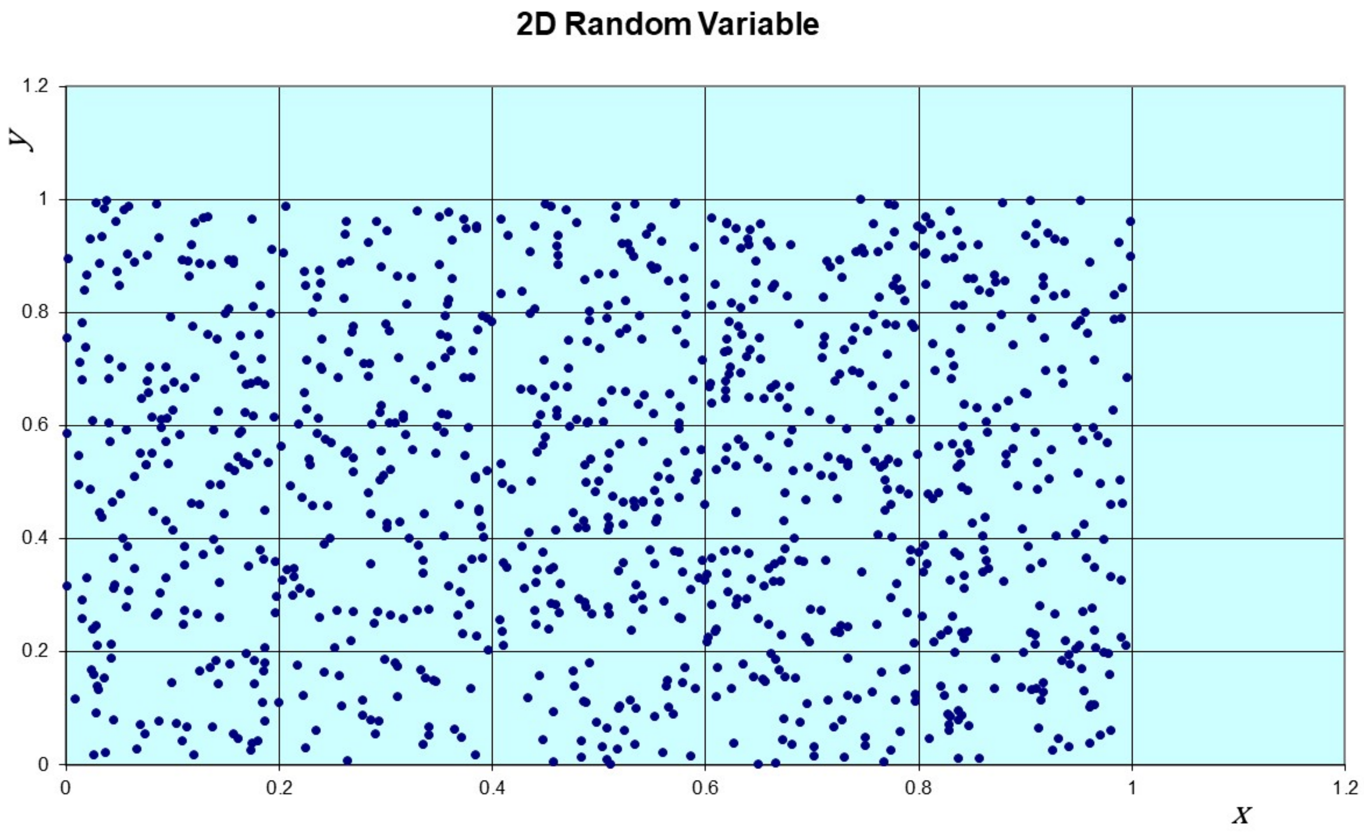
$$\tilde{f}=\begin{cases} 0 & x^2+y^2>r^2 \\ f(x,y) & x^2+y^2\leq r^2 \end{cases}$$

and the standard deviation of the calculated integral estimate will be equal to:

$$\sigma(I,N)=\frac{V^*}{N}\cdot\sqrt{\sum_{i=1}^{n}\left(\langle\tilde{f}\rangle-\tilde{f}_i\right)^2}$$

A sample calculation was made for the 1000 generated points, while the hemisphere radius r was equal to 1 m.

Render Pairs of (Pseudo)Random Numbers



2D chart of **randomly generated points** - a thousand pairs of generated pseudo-random numbers for the calculation of the hemisphere volume.

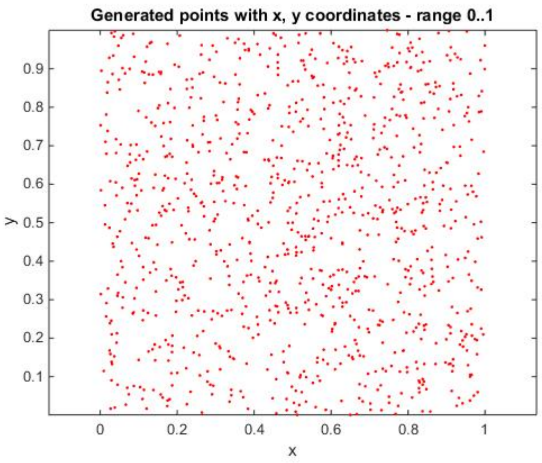
Input values

x_0	**0.5**
A	**758**
C	**0.333**
M	**1**

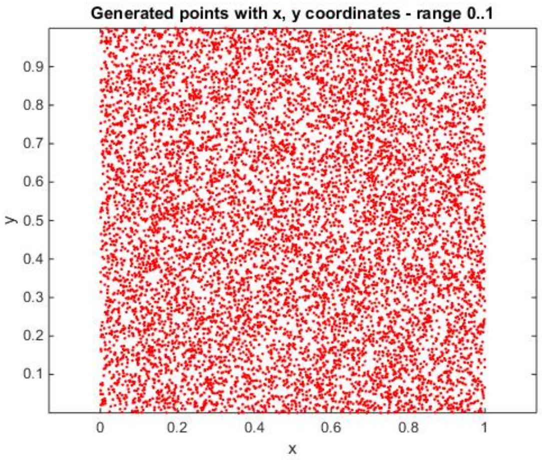
y_0	**0.5**
A	**239**
C	**0.666**
M	**1**

Render Pairs of (Pseudo)Random Numbers

$N = 1{,}000$



$N = 10{,}000$



$N = 100{,}000$

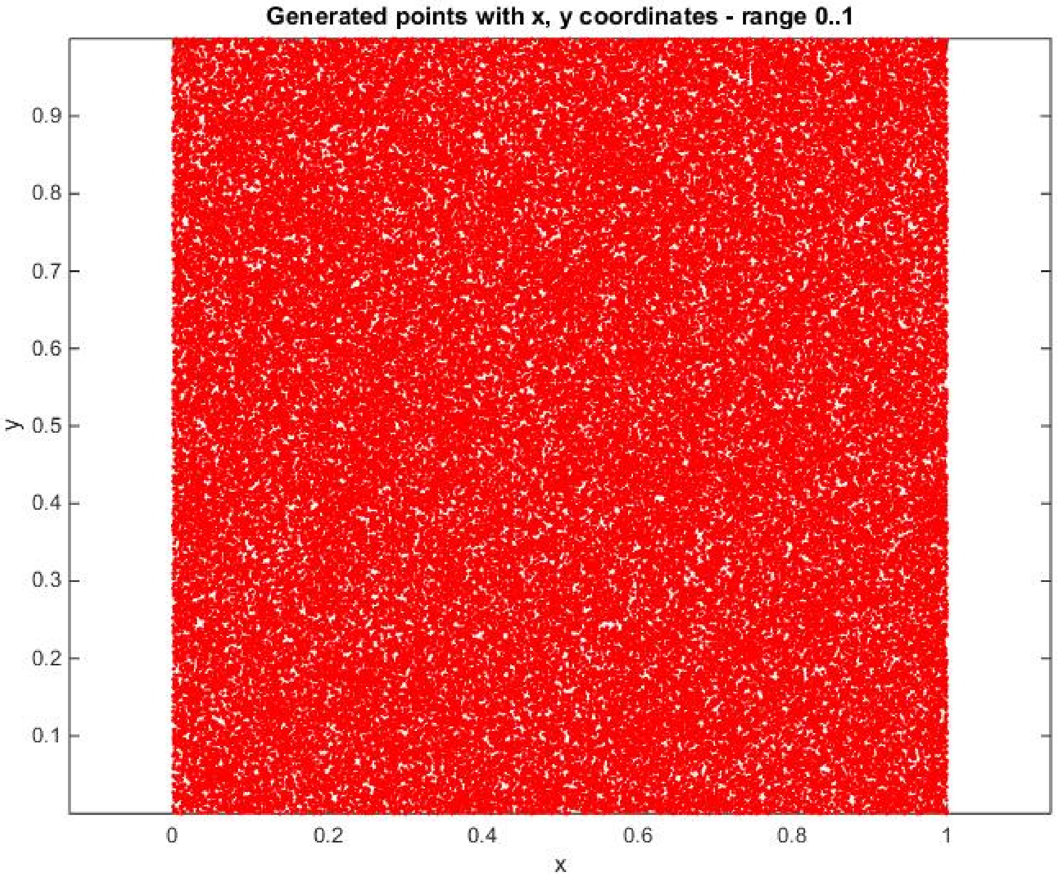


Chart of values f in $N = 1000$ Generated Points

The resulting estimate of the integral value:

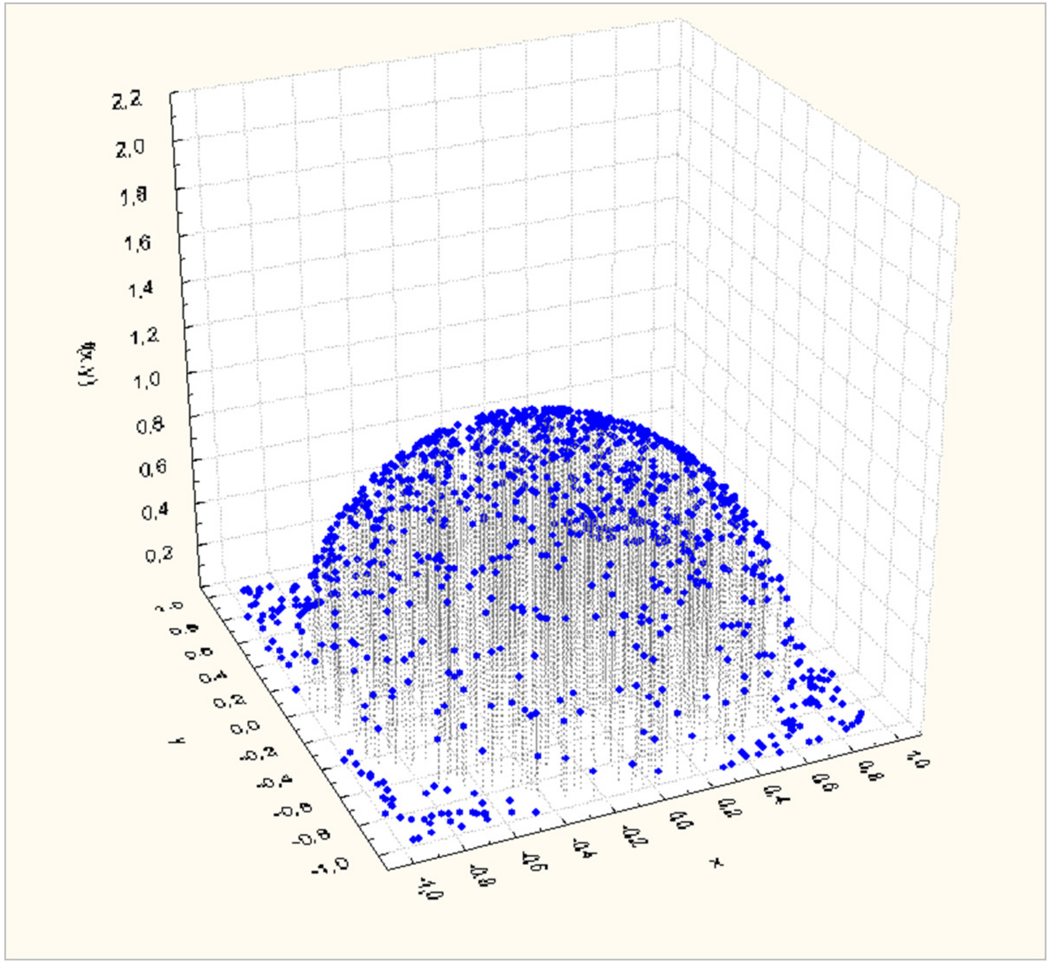
$$I \approx \frac{V^*}{N} \sum_{i=1}^{N} \tilde{f}(x) \approx 2.1771\,\mathrm{m}^3$$

The exact value:

$$\frac{2}{3} \cdot \pi \cdot r^3 = 2.0944\,\mathrm{m}^3$$

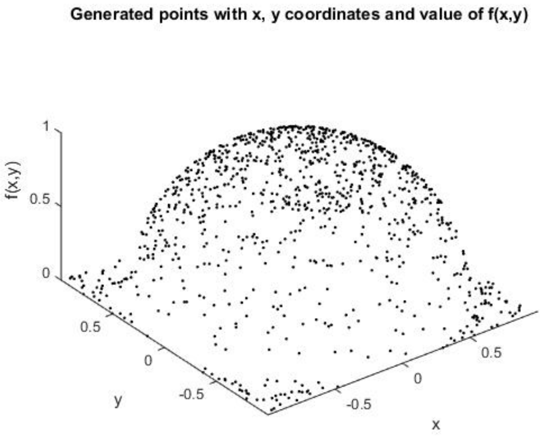
The standard deviation of the integral estimate:

$$\sigma(I, N) \approx 0.0435\,\mathrm{m}^3$$



Render Pairs of (Pseudo)Random Numbers

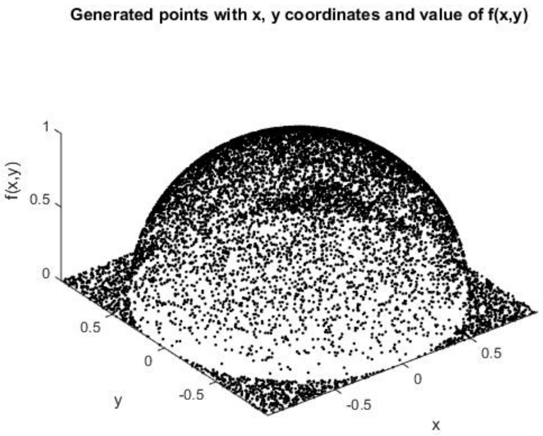
$N = 1{,}000$


Generated points with x, y coordinates and value of f(x,y)

$I = 2.1771\ \mathrm{m}^3$

$\sigma = 0.0435\ \mathrm{m}^3$

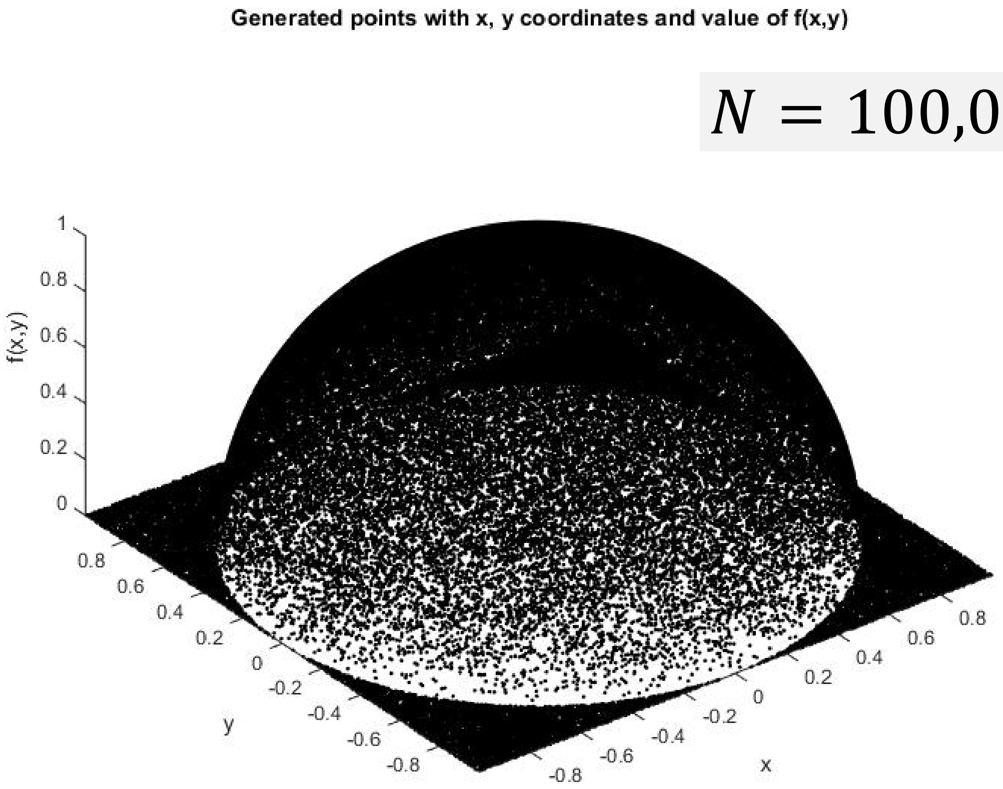
$N = 10{,}000$


Generated points with x, y coordinates and value of f(x,y)

$I = 2.1092\ \mathrm{m}^3$

$\sigma = 0.0137\ \mathrm{m}^3$

$N = 100{,}000$

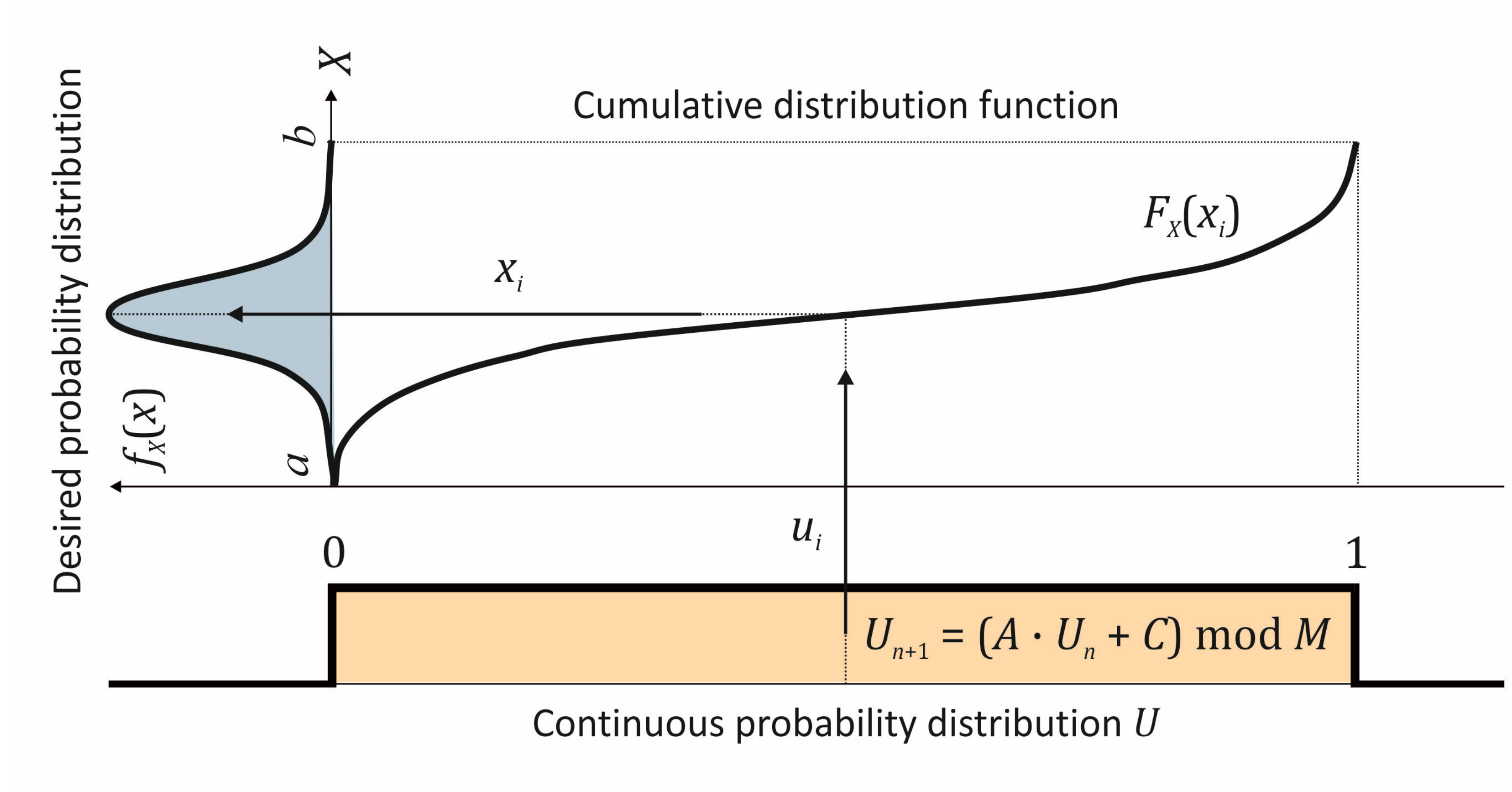

Generated points with x, y coordinates and value of f(x,y)

$I = 2.0999\ \mathrm{m}^3$

$\sigma = 0.0043\ \mathrm{m}^3$

The Principle of SBRA Method

Generating limited distribution and transformation of the desired distribution



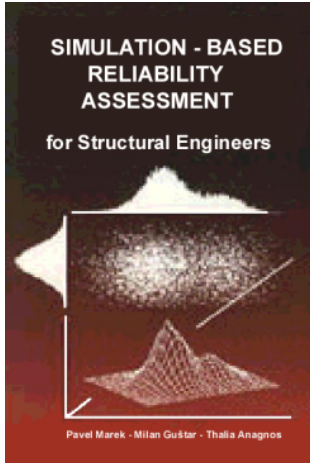
Desired probability distribution
X
Cumulative distribution function
b
$F_X(x_i)$
x_i
$f_X(x)$
a
u_i
0
1
$U_{n+1} = (A \cdot U_n + C) \bmod M$
Continuous probability distribution U

Reliability Assessment Using SBRA

- Input variables characterized using **bounded histograms** with nonparametric probability distribution.
- Analysis of the reliability function using **Monte Carlo method**.
- **Reliability** is expressed as $P_f < P_d$, where P_f is **probability of failure**, and P_d is **designed value** of failure probability:

$$P_f = \Sigma \,/\, \Sigma < P_d$$

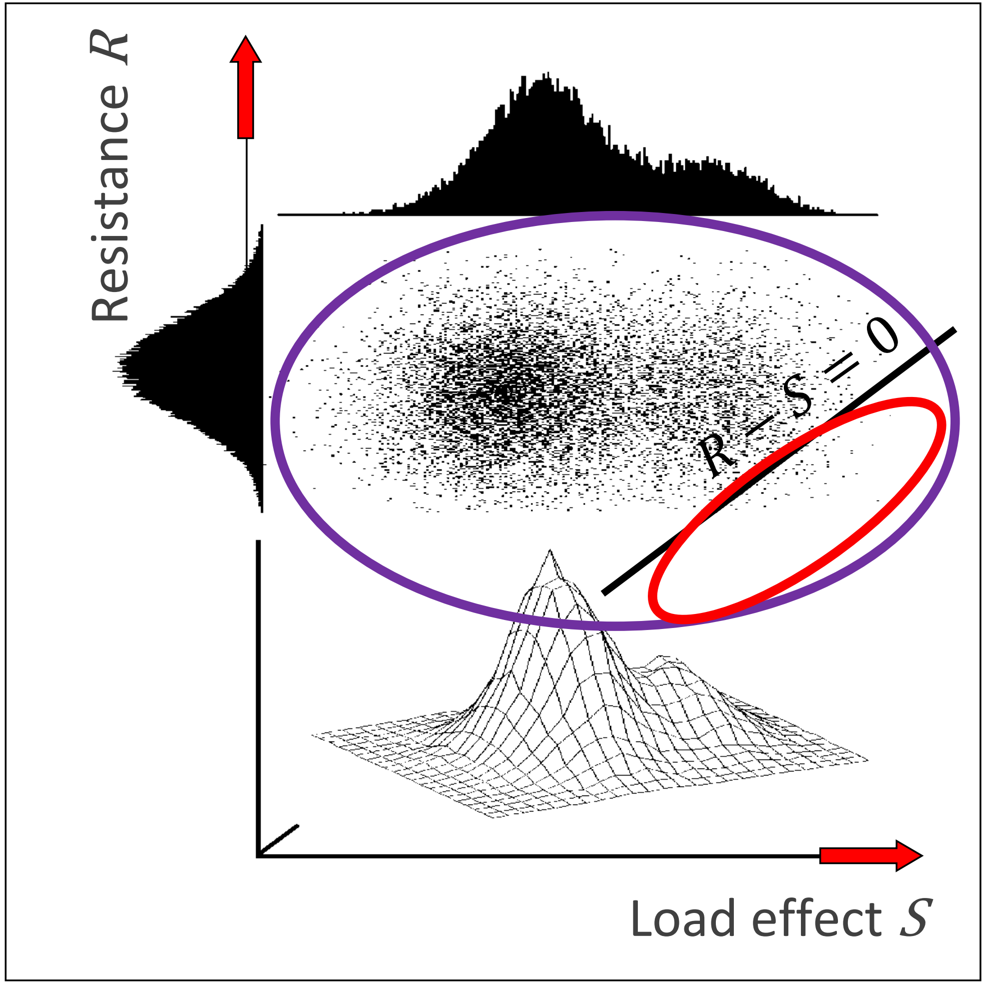
E.g., Marek et al.,
CRC Press, 1995.



SIMULATION - BASED RELIABILITY ASSESSMENT

for Structural Engineers

Pavel Marek · Milan Guštar · Thalia Anagnos



Resistance R
$R - S = 0$
Load effect S