Random number generators and application

Marcin Chrząszcz
mchrzasz@cern.ch

University of Zurich[UZH]

Experimental Methods in Particle Physics,
19 November, 2015

Random and pseudorandom numbers

John von Neumann:

"Any one who considers arithmetical methods of producing random digits is, of course, in a state of sin. For, as has been pointed out several times, there is no such thing as a random number — there are only methods to produce random numbers, and a strict arithmetic procedure of course is not such a method."

$\Rrightarrow$ Random number: a given value that is taken by a random variable

$\twoheadrightarrow$ by definition cannot be predicted.

$\Rrightarrow$ Sources of truly random numbers:

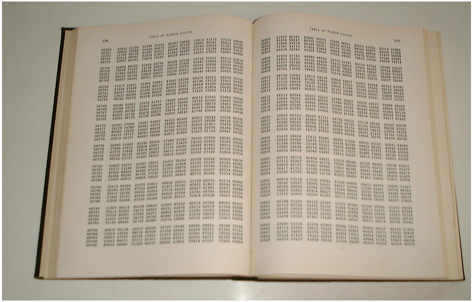
- Mechanical
- Physical

$\Rrightarrow$ Disadvantages of physical generators:

- To slow for typical applications, especially the mechanical ones!
- Not stable; small changes in boundary conditions might lead to completely different results!

Random numbers - history remark

⇛ In the past there were books with random numbers:



⇛ It's obvious that they didn't become very popular ;)

⇛ This methods are comming back!

↠ Storage device are getting more cheap and bigger (CD, DVD).

↠ 1995: G. Marsaglia, $650\mathrm{MB}$ of random numbers, "White and Black Noise".

Pseudorandom numbers

⇛ Pseudorandom numbers are numbers that are generated accordingly to strict mathematical formula.
↬ Strictly speaking they are non random numbers, how ever they have all the statistical properties of random numbers.
↬ Discussing those properties is a wide topic so let's just say that without knowing the formula they are generated by one cannot say if those numbers are random or not.
⇛ Mathematical methods of producing pseudorandom numbers:

- Good statistical properties of generated numbers.
- Easy to use and fast!
- Reproducible!

⇛ Since mathematical pseudorandom genrators are dominantly: pseudorandom $\rightarrowtail$ random.

Middle square generator; von Neumann

⇛ The first mathematical generator (middle square) was proposed by von Neumann (1964).

↬ Formula: $$X_n = \lfloor X_{n-1}^2 \cdot 10^{-m} \rfloor - \lfloor X_{n-1}^2 \cdot 10^{-3m} \rfloor$$

↬ where X_0 is a constant (seed), $\lfloor \cdot \rfloor$ is the cut-off of a number to integer.

⇛ Example:

Let's put $m = 2$ and $X_0 = 2045$:

↬ $X_0^2 = \underbrace{04}_{\text{rej}}1820\underbrace{25}_{\text{rej}}$ $\Rightarrow X_1 = 1820$

↬ $X_1^2 = \underbrace{03}_{\text{rej}}3124\underbrace{00}_{\text{rej}}$ $\Rightarrow X_1 = 3124$

↬ Simple generator but unfortunately quite bad generator. Firstly the sequences are very short and strongly dependent on the X_0 number.

Linear generators

⇛ General equation:

$$X_n = (a_1 X_{n-1} + a_2 X_{n-2} + ... + a_k X_{n-k} + c) \bmod m,$$

↬ where a_i, c, m are parameters of a generator(integer numbers).

↬ Generator initialization ⇄ setting those parameters.

⇛ Very old generators. (often used in Pascal, or first C versions):

$$k = 1: \; X_n = (aX_{n-1} + c) \, \mathrm{mod} m,$$

$$c = \begin{cases} = 0, \mathrm{multiplicative geneator} \\ \neq 0, \mathrm{mix geneator} \end{cases}$$

⇛ The period can be achieved by tuning the seed parameters:

$$P_{\max} = \begin{cases} 2^{L-2}; \text{ for } m = 2^L \\ m - 1; \text{ for } m = \text{prime number} \end{cases}$$

Linear generators; examples

⇛ General equation:

$$X_n = (a_1 X_{n-1} + a_2 X_{n-2} + ... + a_k X_{n-k} + c) \text{ mod } m,$$

↬ where a_i, c, m are parameters of a generator(integer numbers).

↬ Generator initialization ⇄ setting those parameters.

⇛ Very old generators. (often used in Pascal, or first C versions):

$$k = 1: \; X_n = (aX_{n-1} + c) \text{ mod} m,$$

$$c = \begin{cases} = 0, \text{multiplicative geneator} \\ \neq 0, \text{mix geneator} \end{cases}$$

⇛ The period can be achieved by tuning the seed parameters:

$$P_{\max} = \begin{cases} 2^{L-2}; \text{ for } m = 2^L \\ m - 1; \text{ for } m = \text{prime number} \end{cases}$$

Shift register generator

⇛ General equation:

$$b_n = (a_1 X_{n-1} + a_2 X_{n-2} + ... + a_k X_{n-k} + c) \bmod 2,$$

where $a_i \subset (\{0, 1\})$

⇛ Super fast and easy to implement due to: $(a + b) \bmod 2 = a \text{ xor } b$

a	b	a xor b
0	0	0
1	0	1
0	1	1
1	1	0

⇛ Maximal period is $2^k - 1$.

⇛ Example (Tausworths generator):

$a_p = a_q = 1$, other $a_i = 0$ and $p > q$. Then: $b_n = b_{n-p} \text{ xor } b_{n-q}$

⇛ How to get numbers from bits (for example):

$U_i = \sum_{j=1}^{L} 2^{-j} b_{is+j}, \ \ s < L$.

Fibonacci generator

⇛ In 1202 Fibonacci with Leonardo in Piza:

$$f_n = f_{n-2} + f_{n-1},\ n \geqslant 2$$

⇛ Based on this first generator was created (Taussky and Todd, 1956):

$$X_n = (X_{n-2} + X_{n-1}) \bmod m,\ n \geqslant 2$$

This generator isn't so good in terms of statistics tests.
⇛ Generalization:

$$X_n = (X_{n-r} \odot X_{n-s}) \bmod m,\ n \geqslant r,\ s \geqslant 1$$

$\odot$	P_{max}	Stat. properties
$+, -$	$(2^r - 1)2^{L-1}$	good
x	$(2^r - 1)2^{L-13}$	very good
xor	$(2^r - 1)$	poor

Backup