

Lecture 7: Basic Sampling

Professor Ilias Bilonis

Pseudo-random number generators

Pseudo-random number generators

- Computers are deterministic machines and therefore they cannot generate completely random numbers?
- Idea: Are there deterministic sequences of numbers that look random?
- Pseudo-random number generators do exactly that.
- We use statistical tests to see how good they are.

Pseudo-random number generators

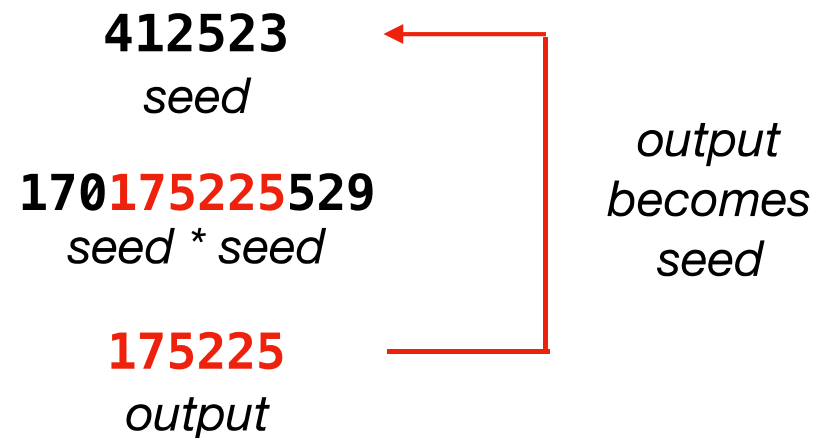
How do you generate a uniform random number?



John von Neumann.
(Los Alamos)

Unless otherwise indicated, this information has been authored by an employee or employees of the Los Alamos National Security, LLC (LANS), operator of the Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396 with the U.S. Department of Energy. The U.S. Government has rights to use, reproduce, and distribute this information. The public may copy and use this information without charge, provided that this Notice and any statement of authorship are reproduced on all copies. Neither the Government nor LANS makes any warranty, express or implied, or assumes any liability or responsibility for the use of this information.

The middle-square method



The first, but it doesn't pass all statistical tests.

Linear congruential generators

Seed x_0

$$x_{i+1} = (ax_i + b) \text{ mod } m$$

Handwritten notes in red:

- Under a : big number, better if prime as well
- Under b : take remainder of division by m

- For every choice of a , b , and m we get a pseudo-random number generator
- This method passes many statistical tests, but still not the best

Mersenne Twister PRNG

- This is what is inside numpy.random.
- Details beyond the scope of this class.

