The building block of simulation is being able to sample an iid sequence of Unif(0,1)

$$U_1, \dots, U_n$$

How do we sample Unif(0,1) random numbers on a computer?

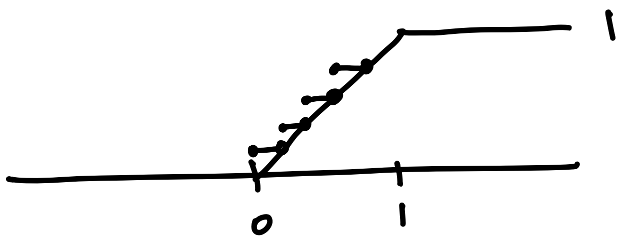
→ approximate a unif(0,1) by first generating a random integer

$x_n = 0, 1, \dots, m-1$ where m is large integer

$\rightarrow \underbrace{P(x_n = i) = \frac{1}{m}}_{\text{for } i=0,\dots,m-1}$

$$U_n = \frac{x_n}{m}$$

cdf of U_n converges to the cdf of Unif(0,1) for m large



→ true random numbers are hard to generate

→ pseudo random numbers
sequence of numbers a pseudo-random number generator

→ way of generating a deterministic sequence that looks random

simple one: (multiplicative congruential generator)

x_0 = seed value

$\vdots$

$x_n = \underbrace{a}\cdot x_{n-1} \bmod \underbrace{m}$

period of MCG is at most m

m is going to be based on the bit size of computer

$m = 2^{31} - 1$

$a = 7^5$

period 16,807

linear congruential generator

$x_n = (a\, x_{n-1} + b) \bmod m$

Mersenne Twister

period of $2^{19,937} - 1$