

## 19. Pseudo-random number generators. Examples using `randgui`.

Numerical Analysis E2021

Institute of Mathematics  
Aalborg University



AALBORG UNIVERSITY  
DENMARK

# Introduction to pseudo-random number generation

Numerical Analysis  
E2021

Introduction To PRNG

1

Examples Of MCA  
Systems

Further Improvements

- ▶ Computers are deterministic, which is of course shite.
- ▶ The basis of many RNGs in use today is Lehmer's **multiplicative congruential algorithm**:

$$x_{k+1} = ax_k + c \mod m.$$

- ▶ “Problem”: Given fixed parameters, and a fixed seed, the sequence will always be the same, and will always have a period of  $m - 1$ .

4

## Multiplicative congruential algorithm with parameters

$$a = 65539 = 2^{16} + 3, \quad c = 0, \quad m = 2^{31}.$$

### Undesirable property:

$$\begin{aligned} x_{n+2} &= (2^{16} + 3)x_{n+1} = (2^{16} + 3)^2 x_n = (2^{32} + 6 \cdot 2^{16} + 9)x_n \\ &\equiv (6(2^{16} + 3) - 9)x_n \pmod{2^{31}}, \end{aligned}$$

thus we conclude

$$x_{n+2} \equiv 6x_{n+1} - 9x_n \pmod{2^{31}},$$

which leads to a very high correlation throughout the sequence.

MATLAB demo: `randgui (@randssp)`.

# Examples

Old Matlab scheme

Numerical Analysis  
E2021

Introduction To PRNG

Examples Of MCA  
Systems

Further Improvements

3

Matlab used an MCA with parameters  $a = 7^5$ ,  $c = 0$  and  $m = 2^{31} - 1$  for many years.

Troublesome as the period of the algorithm is too short compared to the computational power we have.

MATLAB demo: `randgui (@randmcg)`.

4

# Further Improvements

## Marsaglia's Algorithm

Numerical Analysis  
E2021

Introduction To PRNG

Examples Of MCA  
Systems

Further Improvements

4

- ▶ Not an MCA - do not apply any multiplication or division.
- ▶ Designed to produce floating-point numbers.
- ▶ Not based on single seed. Instead based on 35 numbers which form a *state*.
  - ▶ The cache, consists of 32 floating-point numbers  $z_0, \dots, z_{31}$  all in the interval  $[0, 1]$ .
  - ▶ An integer  $i$  such that  $0 \leq i \leq 31$ .
  - ▶ A random integer  $j$ .
  - ▶ A borrow flag  $b$  from the previous step of the algorithm. Either 0 or a small number.
- ▶ Begins by generating these values. Then determines  $z_i$  as follows:

$$z_{i \bmod 32} = z_{i+20 \bmod 32} - z_{i+5 \bmod 32} - b.$$

MATLAB demo of MCA example.