Review of High-Quality Pseudo Random Number Generators

F. James, L. Moneta (CERN)

based on arXiv:1903.01247

4-8 November 2019, Adelaide

Introduction

- Before 1994 random number generators were based on integer arithmetic and number theory.
  - Linear Congruential Generator studied intensively but still not completed understood
  - Generators with long periods were made and period could be known exactly
  - Marsaglia hyper-planes discovered and their distances could be calculated (spectral test)
  - Algorithms like Marsaglia SWB (RCarry) and Mersenne-Twister become popular
- 1992 paper of Ferrenberg et al:
  Monte Carlo simulations: Hidden errors from "good" random number generators
  - Monte Carlo calculations wrongs when using SWB but were correct with older generator known to have defects (e.g. LCG)

Classical dynamical systems

- Trajectories of a dynamical system described by:

$$x(t+1) = \mathbf{A}\ x(t) \mod 1$$

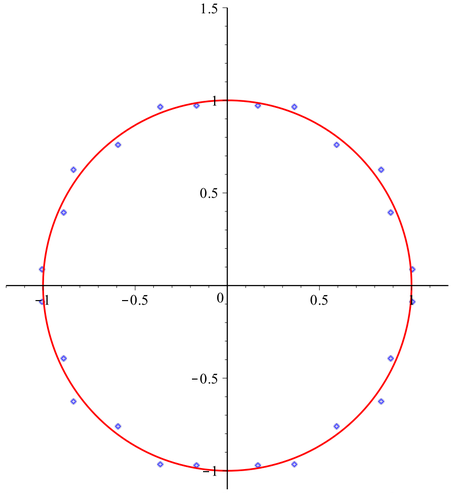
  - x(t) is a vector of *N* real numbers in [0,1), i.e. position of the trajectory in phase space at time t
  - **A** is a *N* x *N* matrix of integers.
- Theory defines conditions for Kolmogorov-Anosov mixing, i.e. when the points on the trajectory appears as random
  - System has K mixing when:
    - Det(**A**) = 1
    - All eigenvalues of **A** have modulo different than 1

Mixing in Classical Systems

- **Zero-mixing** is the same as ergodic motion. It means that the system will asymptotically come arbitrarily close to any point in state space.
- **One-mixing** means that the coverage will be uniform, in the sense that asymptotically, the probability of finding the system in any given region of state space is proportional to the volume of the region.
- **Two-mixing** means that the probability of the system being in one region of state space at one time, and another region at another time, is proportional to the product of the two volumes.
- **n-mixing** means that the probability of finding the system in n different regions at n different times is proportional to the product of the n volumes.
- **K-mixing** is n-mixing for arbitrarily large n.
  Points taken sufficiently far apart on the trajectory of a k-system are therefore independent, identically distributed (i.i.d.), i.e random

RANLUX

- M. Lüscher found the relation between the existing Marsaglia generator (SWB) and a classical system with K mixing (M. Lüscher, 1994)
- Generator sequence is equivalent to :
  - $x(t+1) = \mathbf{A}\ x(t) \mod 1$
  - A is a 24x24 matrix with many zero,
    1 and -1 at right position to get the SWB sequence
  - Determinant is 1 and eigenvalues are all different than 1
- But mixing is an asymptotic property.
- Past is forget only after a certain time
  - for this reason SWB generator fails empirical randomness tests
- Solution: apply decimation by throwing away some generated numbers and use only a fraction of them

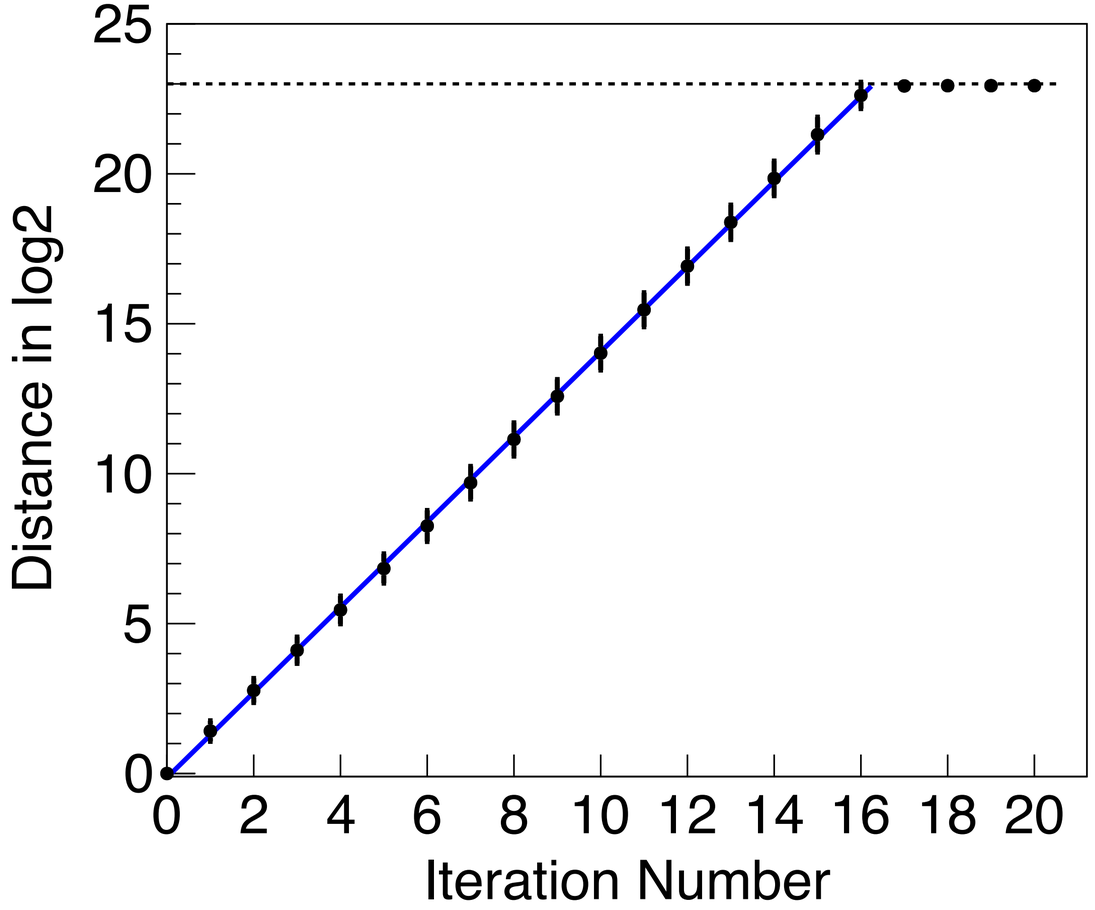


Spectrum of eigenvalues

Exponential Divergency

- Consider trajectories initiated from two closed points:
  - Their distance will grow exponentially
- Speed of divergency is given by the Lyapunov exponent = $\ln |\lambda|_{max}$
  - Property of the K systems !

Skip some iterations to forget past and achieve independent numbers (randomness)



MIXMAX

- G. Savvidy et al. proposed already in 1991 that K systems could be used for MC generators (G. Savvidy et al, 1991)
- The generators proposed by K. Savvidy (K. Savvidy, 2015) is described by the matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & 2 & 1 & 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & 3+s & 2 & 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & 4 & 3 & 2 & 1 & 1 & \dots & 1 & 1 \\ 1 & 5 & 4 & 3 & 2 & 1 & \dots & 1 & 1 \\ & \vdots & & & & & & & \\ & \vdots & & & & & & & \\ 1 & N & N-1 & N-2 & & \dots & & 3 & 2 \end{pmatrix}$$

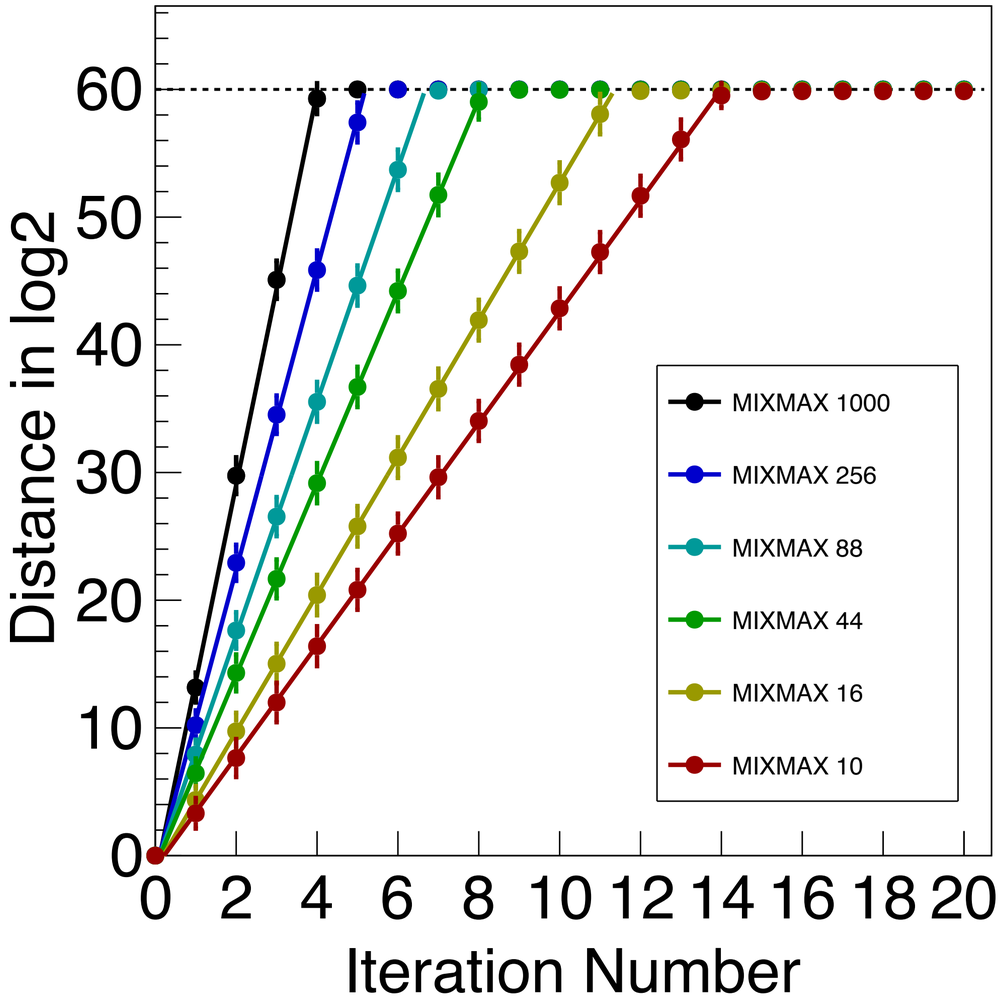
$$a(t+1) = \mathbf{A}\ a(t) \mod p \qquad p = 2^{61} - 1$$

  - different generator for different (N,s) values
    - recommended using N = 256 and s = -1
  - but other generators for different N and s values have been proposed (see table 1 in K. Savvidy, 2015)
    - CPU efficiency of generator independent of N, but not seeding and initialization

Nearby Trajectories for MIXMAX

- We have studied divergency of nearby trajectory for some of the proposed MIXMAX generators
- Exponential divergency with observed rate as expected ($=\ln|\lambda|_{max}$)
- Number of iterations to reach maximum varying with N
  - e.g. for N=256 need 5 iterations

Without decimation, generators with N<88 fail empirical tests (TestU01)



Extended MIXMAX

- The Savvidy's have proposed a new version with an extra parameter (m) (K & G. Savvidy, 2016 and 2018)

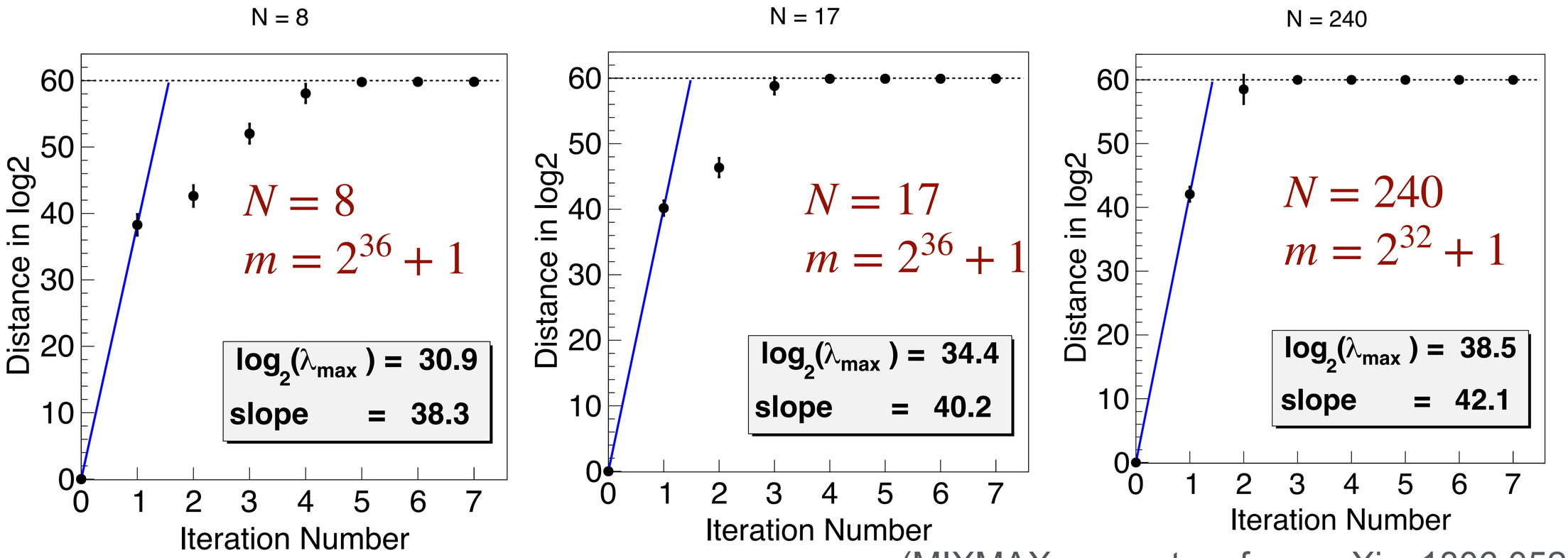
$$A(N,s,m)=\begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 & 1 \\ 1 & 2 & 1 & 1 & \cdots & 1 & 1 \\ 1 & m+2+s & 2 & 1 & \cdots & 1 & 1 \\ 1 & 2m+2 & m+2 & 2 & \cdots & 1 & 1 \\ 1 & 3m+2 & 2m+2 & m+2 & \cdots & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 1 & N-1 & N-2 & N-3 & \cdots & 2 & 1 \\ 1 & (N-2)m+2 & (N-3)m+2 & (N-4)m+2 & \cdots & m+2 & 2 \end{pmatrix}$$

m is a very large number

➡ larger entropy, larger $\ln|\lambda|_{max}$ and faster divergency

Extended MIXMAX passes empirical tests (TestU01) also for smallest *N* (=8)

Exponential Divergency for MiXMAX 2



(MIXMAX generators from arXiv: 1806.05243v2)

- Not clear why divergency is not anymore exponential and why rate of first point is larger than expected
  - caused by using special large *m* values ? $m = 2^{36} + 1$ for *N*=17
  - can these generator still be considered K systems ?

RANLUX++

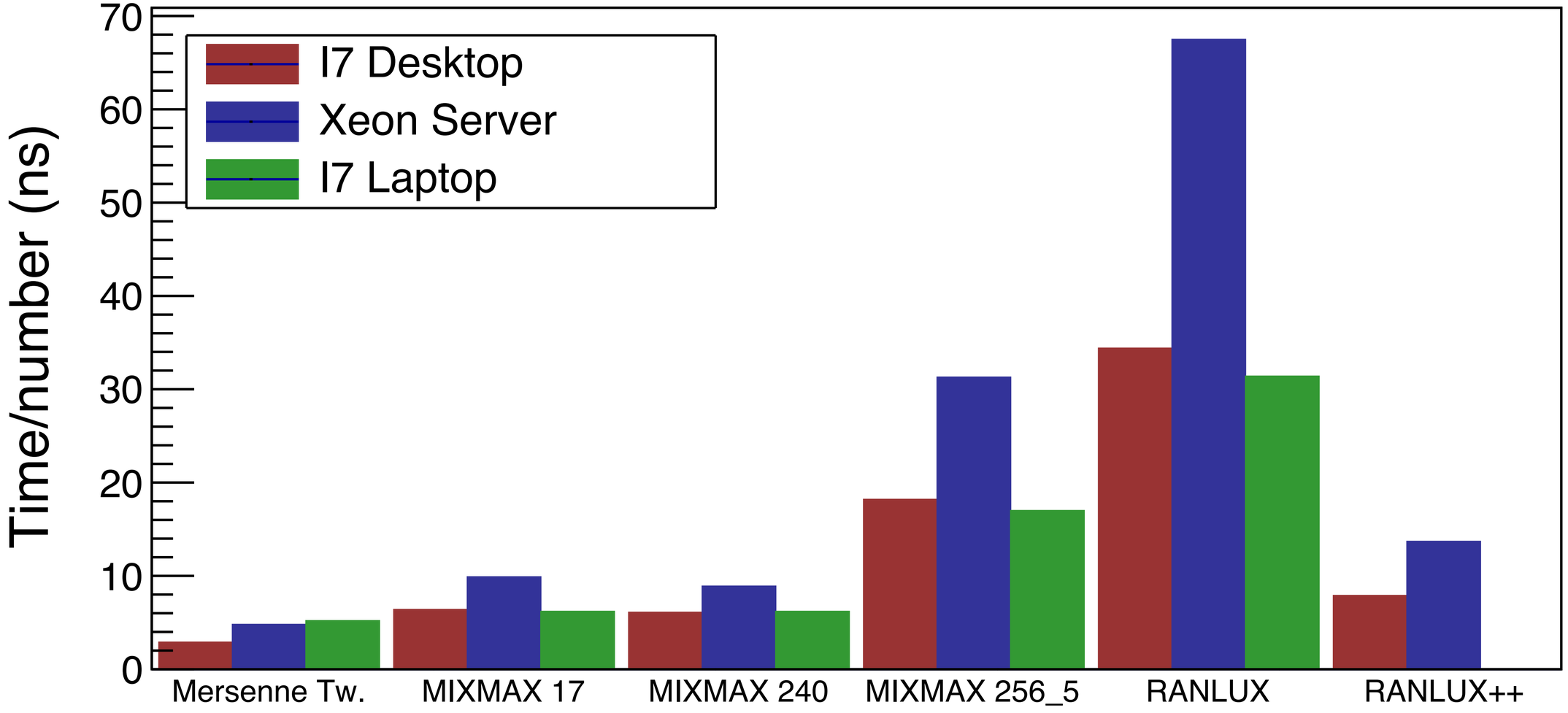
- New version of RANLUX proposed by A. Sibidanov (arXiv:1705.03123v1)
  - Seen as a LCG with an enormous multiplier
  - implemented using 576 (24x24) bit integer operations using assembler vectorised instructions

$$\underbrace{a \cdot (a \cdot (\ldots) \bmod m) \bmod m}_{p \text{ times}} = (a^p \bmod m) \cdot x \bmod m = A \cdot x \bmod m,$$

- Main feature:
  - can produce decimation (jumping head in sequence) without increasing generation time
  - same sequence as RANLUX but much faster
- Very interesting generator that could be integrated in our HEP software tools
  - need a portable implementation working on several architectures

Generators CPU Performance

Time to generate a random number in 3 different hardware architectures



Empirical Tests

- BigCrush of TestU01 (P. Lecuyer)
  - a battery of several tests (~200), which are run in different configurations
  - contain major tests designed in recent years (e.g from Knuth, Marsaglia, Lecuyer, etc..)
  - most stringent test existing
- Results:
  - good for MIXMAX generators
    - N >= 88 for MIXMAX1
    - all MIXMAX 2 (N=8,17,240)
  - RANLUX pass (when skipping is applied)
  - Mersenne-Twister fails dramatically in 2 tests
  - Older generators (LCG) fail also in several tests of BigCrush

Empirical tests cannot guarantee that a generator has not defects !

Spectral Test

- Test of possible dependence in the generated number
- Theoretical test looking at the recursion producing the random numbers
  - study lattice structure of successive values in many dimensions
  - a study has been performed on MIXMAX (P. Lecuyer, 2018)
    - for example using $0 \leq j \leq N-6$ we have that these points

      $\{ x_{4+j}, x_{5+j}, x_{N+3+j}, x_{N+4+j}, x_{N+5+j} \}$ are related (define an hyperplane)
  - but these relationship very difficult to detect empirically when N is large
    - only apply to a subset of the generated numbers
    - new MIXMAX17 and MIXMAX240 have very small distances between these hyperplanes
    - skipping iterations reduces these spectral dependences (as expected by K-system theory)

Generator Seedings

- MIXMAX generators and RANLUX++ can jump ahead in the sequence when a different seed is provided
  - this gives completely independent numbers for every different seed provided
  - period is very large, no problem to divide total sequence in ~2^64 ones
  - but seeding time proportional to N^2 for MIXMAX
- This functionality is not available when using other generators (e.g RANLUX or Mersenne-Twister)

Conclusions

- High Quality generators based on K systems
  - based on a relation (automorphism) described by matrix **A** with Det(**A**) = 1 and eigenvalues all different than 1
  - good approximation to continuous K systems trajectories
  - very long period ($P > 10^{150}$)
  - sufficient decimation to satisfy conditions of nearby trajectories

- Generators available in the ROOT software for HEP usage