Random Numbers

a quick and dirty guide

Bartosz Kostrzewa

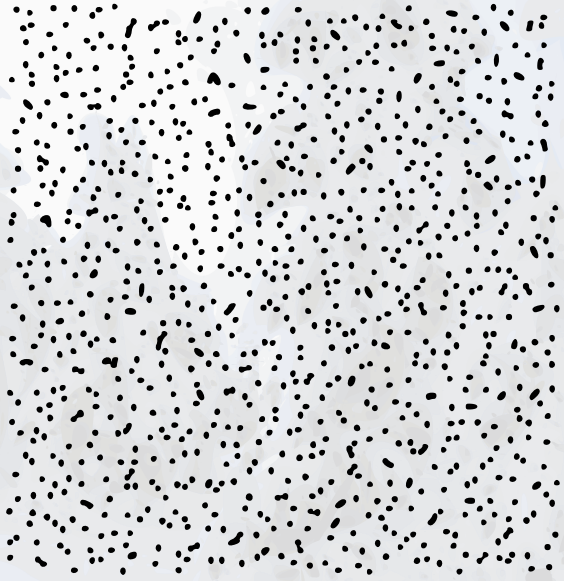
Institute of Physics
Humboldt-Universität zu Berlin
Supported in full by FNR AFR grant 2773315

PhD Seminar
22nd January 2013

Fonds National de la
Recherche Luxembourg

Quiz

Which image is more random?



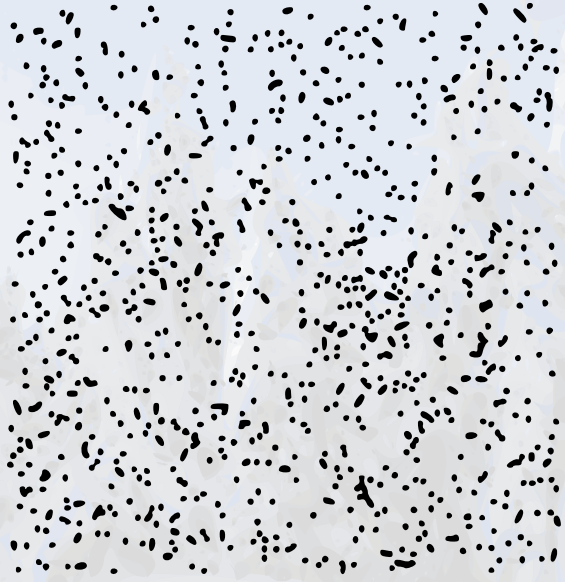


Figure: http://arachnoid.com/randomness/index.html

Overview

Introduction

True random numbers

- + truly random
- - not reproducible
- - rare / slow
- - possible physical bias (temperature etc.)

Pseudo-random numbers

- + reproducible
- + quasi-unlimited / fast
- - possibly correlated

Introduction

Pseudo-random numbers

PRNG = Pseudo-random number generator

- deterministic algorithm to produce numbers that "look random"

Figures of merit

- period – number of updates before the sequence repeats
  - depends on details of generator, can be calculated analytically for some
  - a typical $N_f = 2$ QCD HMC simulation with mass preconditioning uses $\sim 10^{13}$ random numbers per ensemble
- equidistribution
- statistical properties
- tests for specific applications

A simple PRNG

Linear Congruential Generators

Recurrence relation

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

Properties

- need to choose a,c and m judiciously
- for c and m relatively prime, period = m
- typical periods in practice: $2^{31(48)} - 1 \sim 2 \times 10^{9(14)}$

Generalization

Use multiple recurrence relation:

$$x_{i+1} = (a_1 x_{i-j_1} + a_2 x_{i-j_2} + \ldots + a_n x_{i-j_n} + c) \mod m$$

Need multiple seed values $\rightarrow$ correlation in seeds can persist to RNG! [2]

Measuring random number quality

Correlations of random numbers

Pseudo-random numbers can be correlated:

1. between successive draws
2. between n-tuples of successive draws
3. between different random number chains
4. at different timescales

Measuring random number quality

Equidistribution tests

What happens when you arrange sequences in vectors with n components?

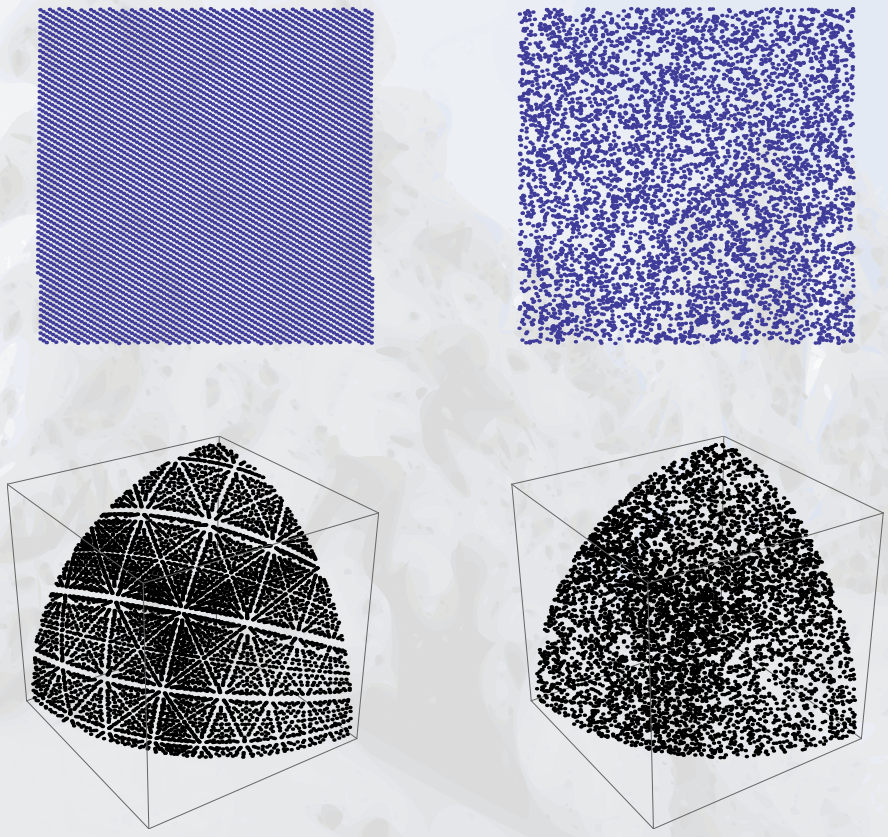


Figure: left Lin.Con.Gen., right MersenneTwister, Helmut G. Katzgraber, arXiv:1005.4117v1

Measuring random number quality

Other tests

Statistical

Deviations from expectation values can be computed analytically and results compared using χ^2 tests.

- tests involving the deviation from the expected mean and moments thereof
- binary-level tests
- n-tuple tests which count the occurence of certain subsequences (pairs, triples, quadruples etc..)

These tests are not exhaustive and not necessarily indicative of performance in actual applications...

Application-based

Calculations for well-known physical models (e.g. Ising) and comparison to theoretical expectations.
Calculation using multiple PRNGs and comparison.

Measuring random number quality

So what makes a good generator?

Depends on requirements but in general:

1. (astronomically) long period
2. equidistribution in as many dimensions as possible (or necessary)
3. no or known dependence of RNG quality on initial seed
4. no disconnected regions in "sequence-space":
   - related to 3: for any set of seeds $\{x_0 \ldots x_n\}$, all possible sequences of length n will be visited
   - this is not true for many types of PRNGs
5. suitability for given application

Random number generators as dynamical systems

Marsaglia-Zaman RCARRY generator

recurrence relation

generate integers in $0 \leq x < b$

$$\delta_n = x_{n-s} - x_{n-r} - c_{n-1}$$

$$x_n = \begin{cases} \delta_n, & c_n = 0 \quad \text{if } \delta_n > 0 \\ \delta_n + b, & c_n = 1 \quad \text{if } \delta_n < 0 \end{cases}$$

properties

- r and s are called "lags", c_n is the "carry bit", indicates addition of "base" b
- Require r initial values to "seed" the RNG
- Recommended values $r = 24$, $s = 10 \rightarrow$ period $\sim 10^{171}$
- for any set of seeds, whole space of r-tuples is produced in whole period, but RNG fails certain short-length tests

Random number generators as dynamical systems

Vector in r-dimensional unit hypercube

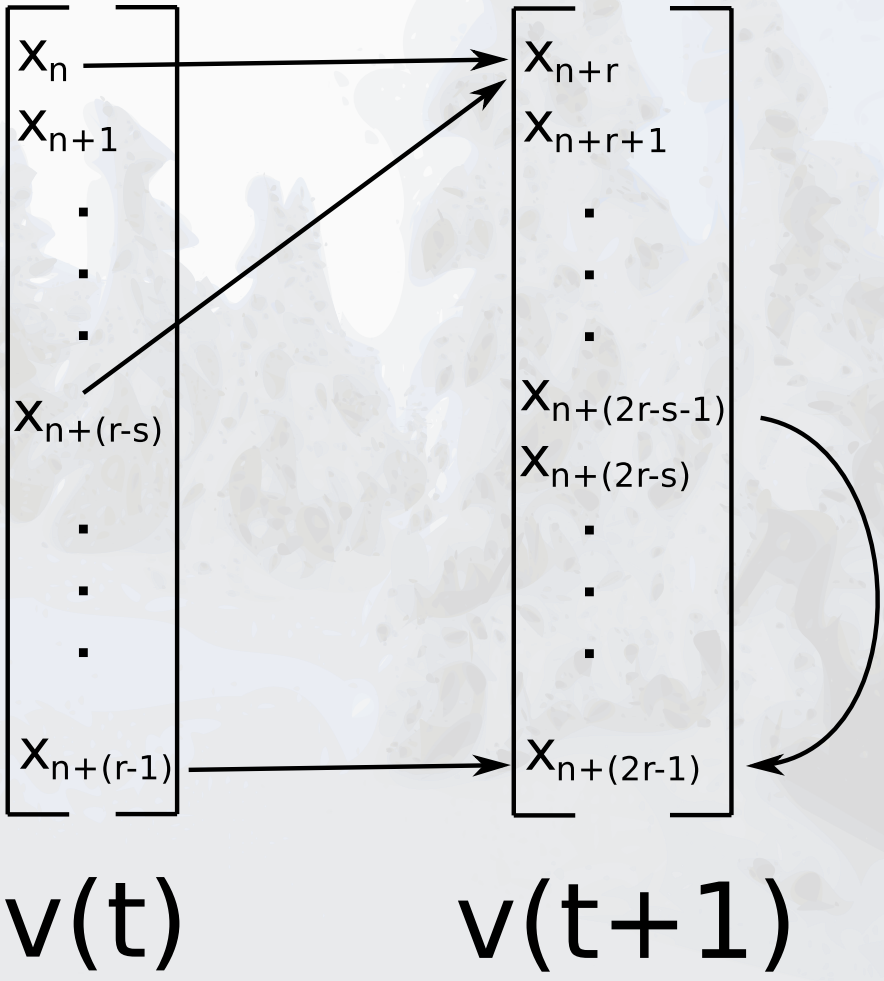
Normalised by b, we can interpret a subsequence of the RNG as a vector in the r-dimensional unit hypercube with lattice spacing $1/b$.

$$\vec{v_n} = \frac{1}{b}(x_n, x_{n+1}, ..., x_{n+r-1})$$

Identifiying the points 0 and 1 in each direction $\rightarrow$ discrete hypertorus with lattice spacing $1/b$.

Random number generators as dynamical systems

RCARRY as a linear transformation



Ignoring the carry bit, we have a linear transformation of the torus:

$$x_n = (x_{n-s} - x_{n-r}) \mod b$$

Or in vectorial form:

$$\vec{v}(t+1) = L^r \vec{v}(t)$$

Using ergodic theory, can make precise notion of "decorrelation" as a measure of RNG quality.

Random number generators as dynamical systems

Random walkers and decorrelation

average distance of random walkers

- Introduce distance function on hypertorus. $d(v, w)$

$$d(v, w) = \max_k \left(\min\left\{|v_k - w_k|, 1 - |v_k - w_k|\right\}\right)$$

- Sample pairs of trajectories $v(t)$ and $w(t)$. Initial separation $= 1/b$.
- Compute average distance

$$\delta(t) = \langle d(v(t), w(t)) \rangle.$$

Random number generators as dynamical systems

Random walkers and decorrelation

Observe exponential divergence $\sim e^t$. At around $t = 16$, reach a plateau of 12/25, the average distance between randomly chosen points on the hypertorus. Exponential divergence indicative of chaos on large timescales from a linear system $\rightarrow$ RCARRY is chaotic.

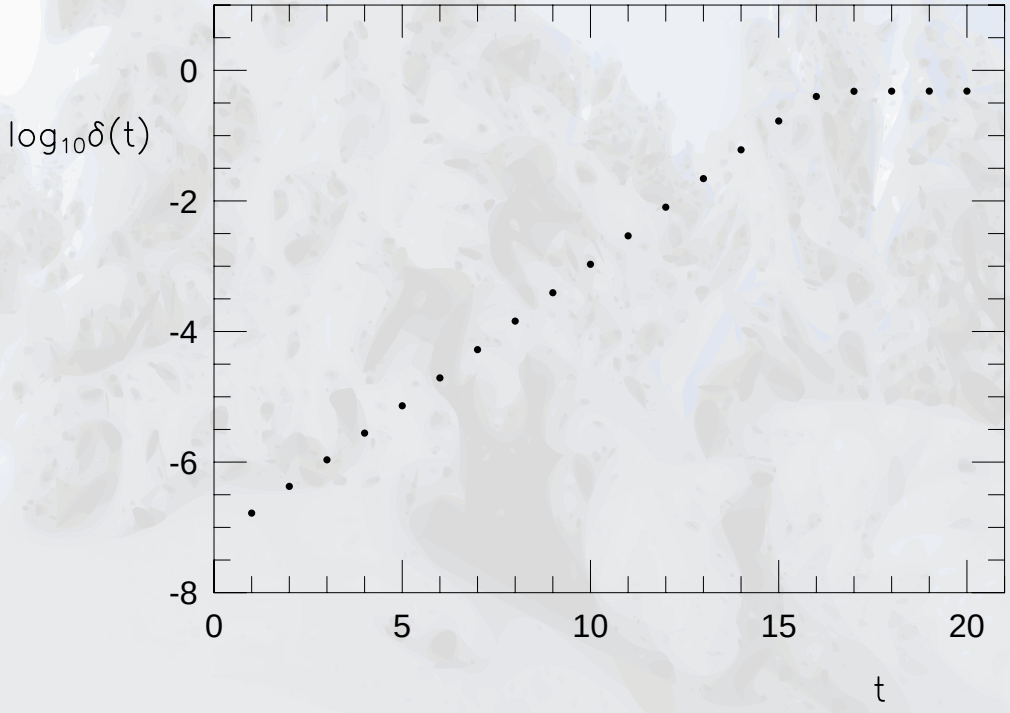


Figure: Martin Luescher, Computer Physics Communications,Volume 79, Issue 1, February 1994, Pages 100–110

Random number generators as dynamical systems

Continuum limit and Lyapunov exponent

In the continuum limit $1/b \to 0$ it can be shown that the evolution is invertible and volume preserving. Defining a distance vector:

$$\vec{u}(t) = (\vec{w}(t) - \vec{v}(t)) \mod \vec{1}$$

It is clear that:

$$\vec{u}(t+1) = L^r \vec{u}(t)$$

Further, the exponential growth of $\|\vec{u}(t)\|$ is determined by the largest eigenvalue of L. The exponent of the fastest-growing separation is termed the Lyapunov exponent:

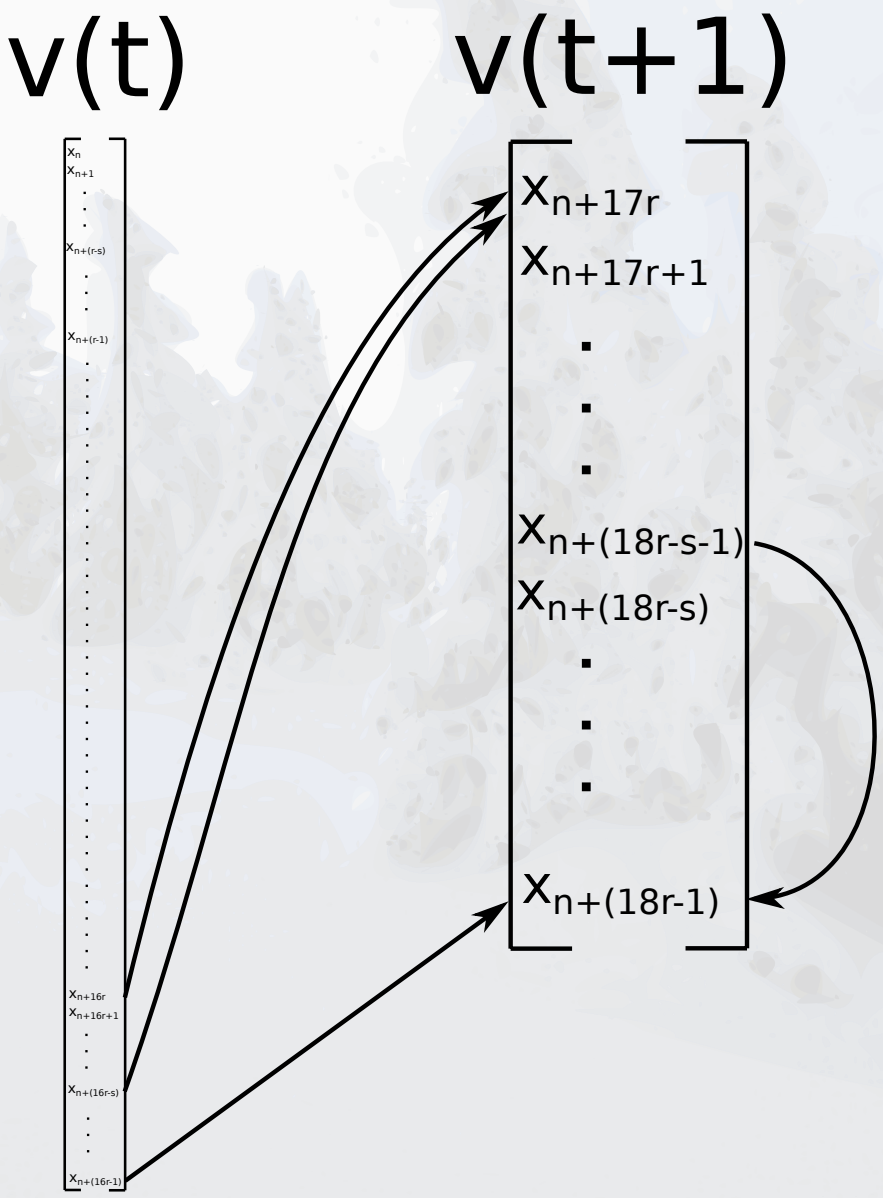
$$\|\vec{u}(t)\| \propto e^{\nu t}$$

$$\nu = r \ln |\lambda_{\max}| = 1.01027..$$

RCARRY is a discretized version of a linear dynamical system which shows chaotic behaviour on large timescales but strong correlation on short timescales.

RANLUX

Principles



Two trajectories decorrelate completely after 17 applications of L^r.
$\rightarrow$ generate long sequences using RCARRY and drop 200 to 400 numbers between draws (luxury level).
Depending on number of dropped sequences, can guarantee complete decorrelation in the sense discussed before.

RANLUX

Properties

(Very?) Good

- in the sense of dynamical systems "demonstrably good decorrelation"
- notion of average distance at sufficient luxury level ensures that different seed values produce truly independent sequences
- passes all statistical tests with sufficient luxury level

Possibly bad

- because sequences are dropped, possibility of violating idea that for any set of seeds all sequences of r-tuples are sampled
  - maybe not important: in practice sample only tiny portion of period anyway, 10^{169} still a very large number, possibly all r-tuples still contained
- what about n-tuples with $n > r$? (a pseudo-spinor, for example...)

"Bad" random numbers in tmLQCD

Weak PRNG vs. RANLUX

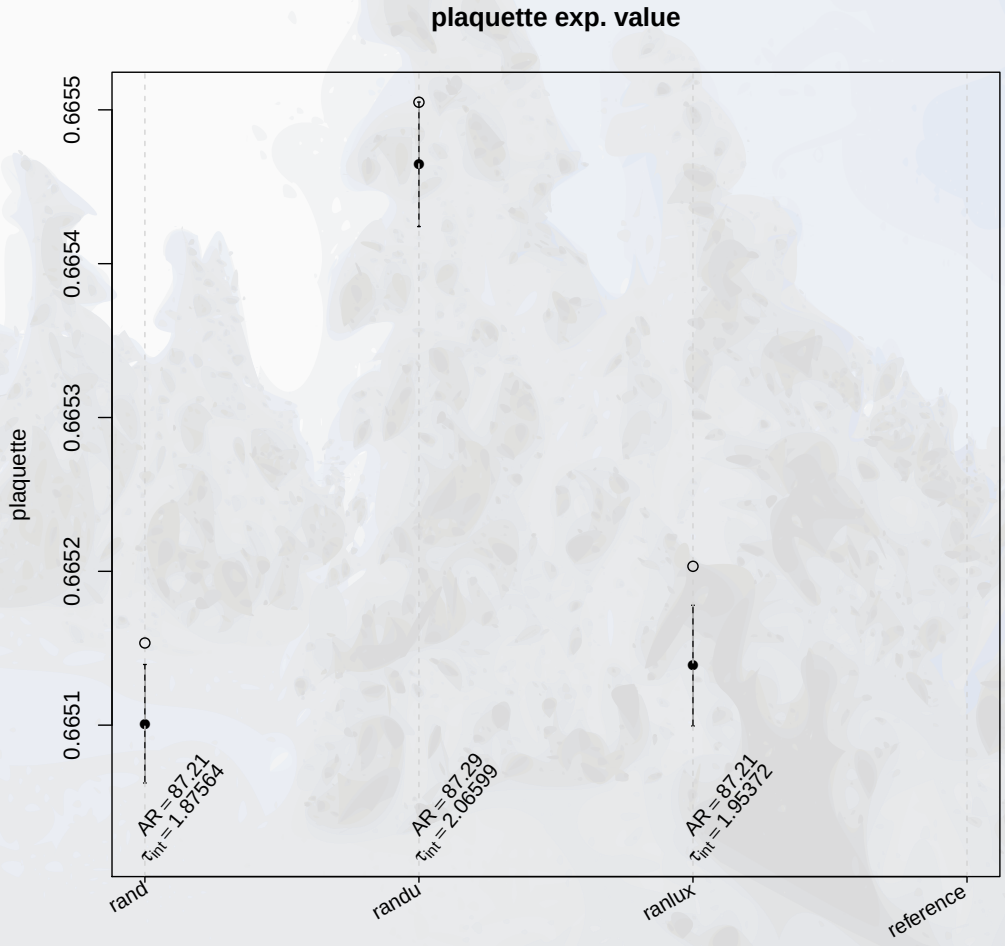


Figure: $N_f = 6$,degenerate,4^4,different RNGs

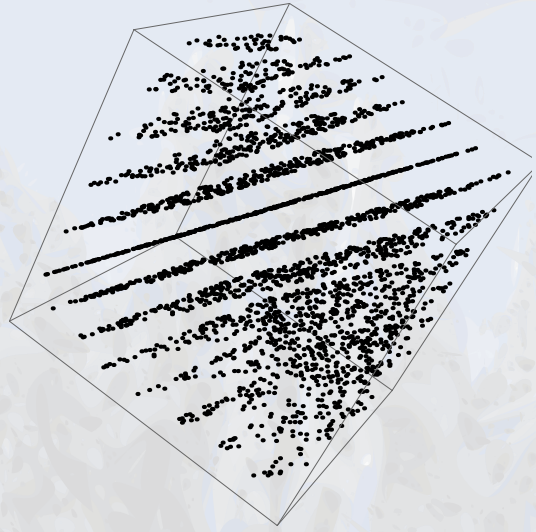


Figure: 3-tuples from RANDU generator highly correlated (15 planes), H. G. Katzgraber, arXiv:1005.4117v1

"Bad" random numbers in tmLQCD

RNG state wrongly set with PHMC (1/3)

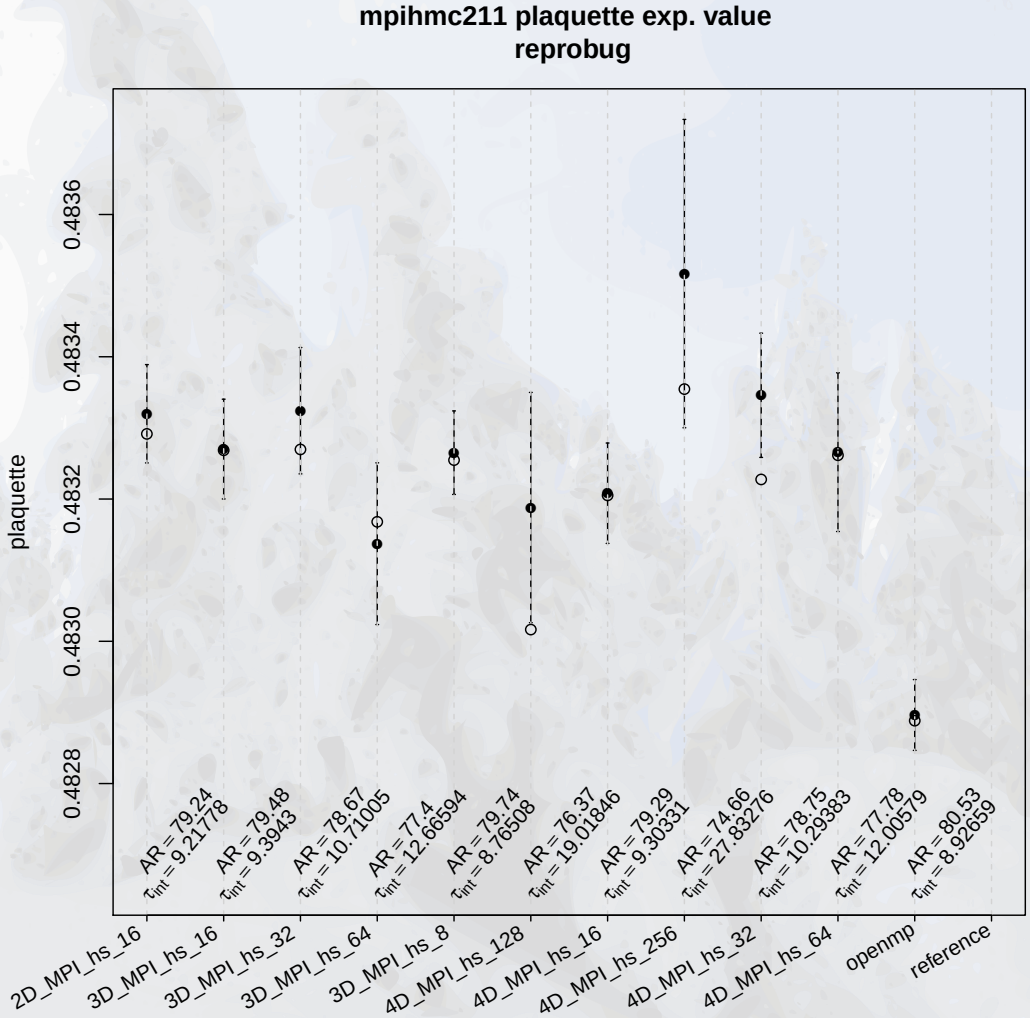


Figure: $N_f = 2 + 1 + 1$, 8^4 volume, 100k trajectories

"Bad" random numbers in tmLQCD

RNG state wrongly set with PHMC (2/3)

Bug caused by switching from parallel to sequential mode and back without reseeding of parallel generators. Process 0 and $n-1$ left in same state, all others reproduce random numbers already used in sequential step. (for instance, $v_1^{(1)}$ would be equivalent to v_2)

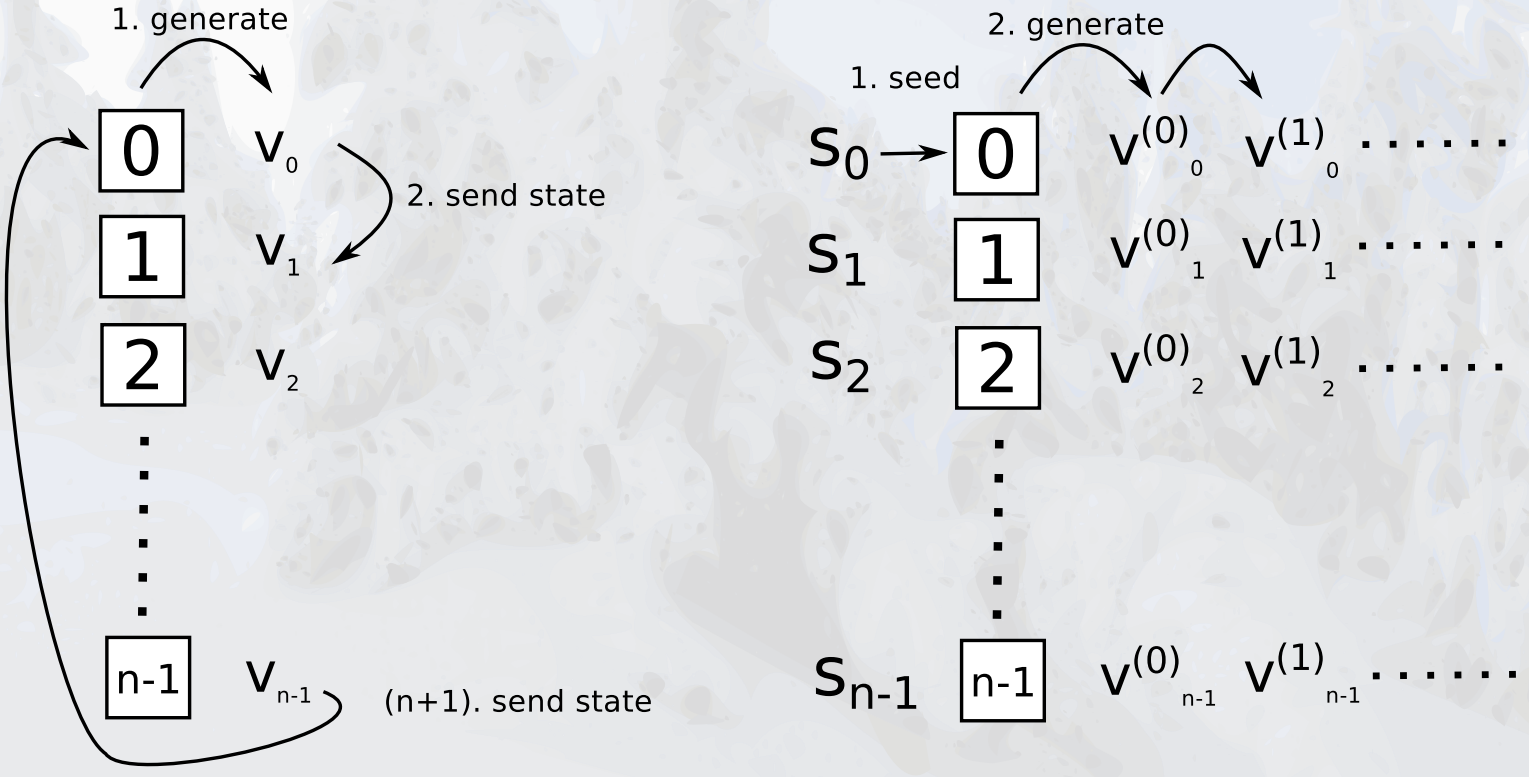


Figure: Sequential (left) and parallel RNG modes in MPI-parallel simulation.

"Bad" random numbers in tmLQCD

RNG state wrongly set with PHMC (3/3)

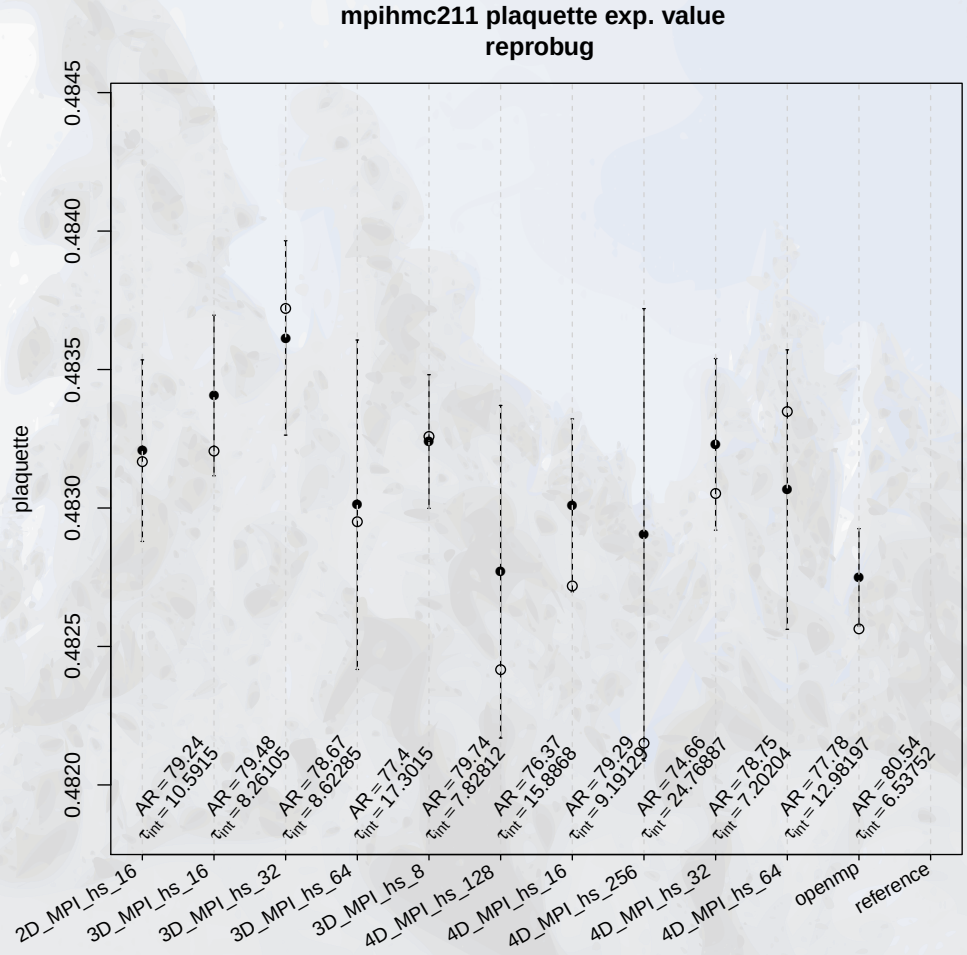


Figure: $N_f = 2 + 1 + 1$, 8^4 volume, 5k trajectories $\rightarrow$ effect difficult to see for some pairs of points (f.ex. passing from 16 to 32 processes)

"Bad" random numbers in tmLQCD

Several MPI processes in same RNG state

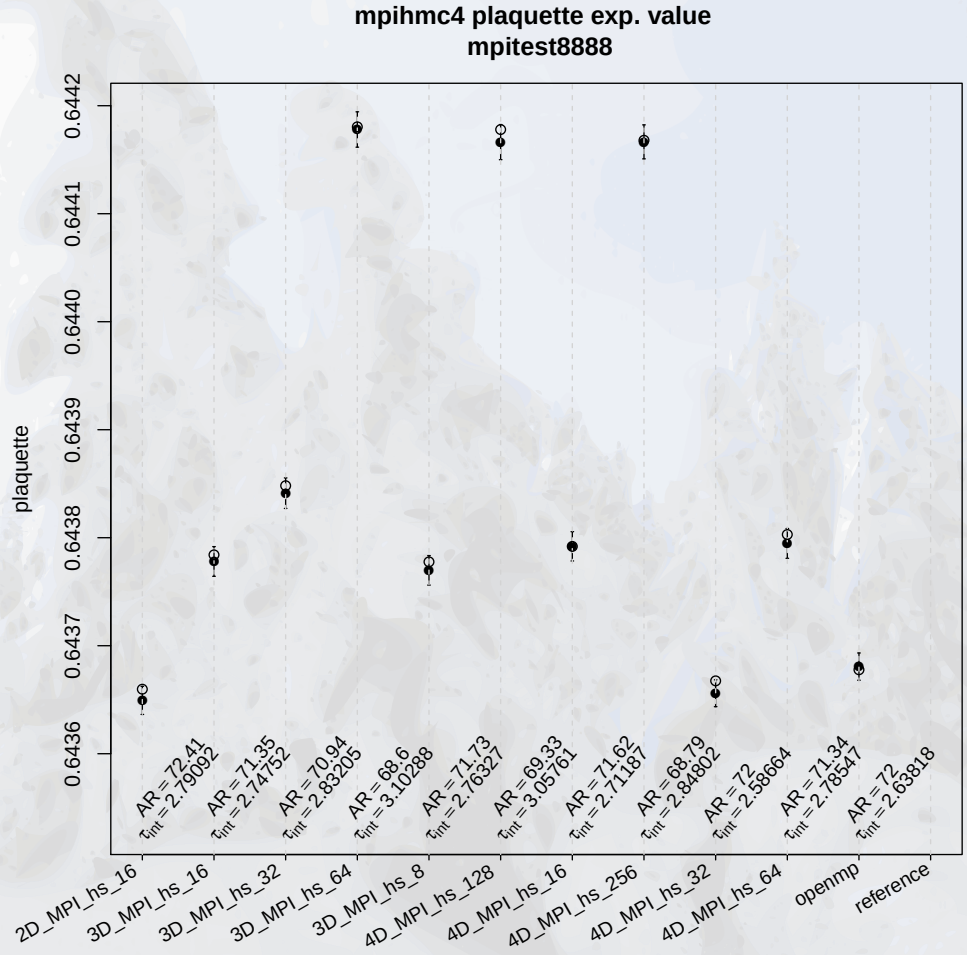


Figure: $N_f = 8$, degenerate, 100k traj., 8^4 volume, two or more processes with same random seed in parallel RNG mode

Conclusions

Simulations

- Make at least one test calculation with multiple PRNGs
- Use well-tested RNGs in their reference implementation if possible
- Repeat one test calculation at least with 2 seeds
- Choose and record a different seed for every simulation
- Perform regular high-statistics tests of your codebase

Analysis

- For stochastic sources adjust your seeds! (Note: this is currently not possible with tmLQCD...)

References

1. H.G. Katzgraber, *Random Numbers in Scientific Computing: An Introduction*, arXiv:1005.4117v1
2. I. Vattulainen, K. Kankaala, J. Saarinen, T. Ala-Nissila, *A Comparative Study of Some Pseudorandom Number Generators*,Computer Physics Communications, Volume 86, Issue 3, 1 May 1995, Pages 209–226
3. I. Vattulainen, *Framework for testing random numbers in parallel calculations*, Phys. Rev. E, Vol. 59, No. 6, June 1999
4. M. Luescher, *A Portable High-Quality Random Number Generator for Lattice Field Theory Simulations*, Computer Physics Communications, Volume 79, Issue 1, February 1994, Pages 100–110
5. Wikipedia articles: *ergodic theory, ergodicity, dynamical systems, mixing (mathematics)*, accessed Jan. 15th, 2013