

The Manufacturing of Randomness

Art or Science?

Stephan Mertens



In an ideal world...

- `log(x)` returns the logarithm of x
- `sin(x)` returns the sine of x
- `time()` returns the wall-clock-time
- `rand()` returns a **random number**

In an ideal world...

- `log(x)` returns the logarithm of x
- `sin(x)` returns the sine of x
- `time()` returns the wall-clock-time
- `rand()` returns a **random number**

In reality:

- `rand()` returns a **pseudo random number**

Any one who considers arithmetic methods of producing random digits is, of course, in a state of sin.
John von Neumann (1953)

The practioner's attitude

- Quality: “This RNG has passed a battery of tests.”
- Tradition: “I got this RNG from my PhD advisor.”
- Ignorance: “RNG? Boring detail.”

Consequences:

- Type of underlying RNG seldom published.
- Simulations rarely cross-checked with different RNGs
- Wrong results remain unnoticed...

The Ferrenberg Affair

- “high quality” RNGs do fail in stat. mech. simulation[†]
- triggered a lot of **empirical research**
- “physical simulations” are added to “battery of tests”

After 40 years of development, one might think that the making of random numbers would be a mature and trouble-free technology, but it seems the creation of unpredictability is ever unpredictable.

Brian Hayes (1993)

[†]Ferrenberg, Landau, Wong *PRL* **69** 3382 (1992)

Randomness as a Resource



1 rand. bit/s

Prob. (same side up) $\simeq 51\%$ [†]

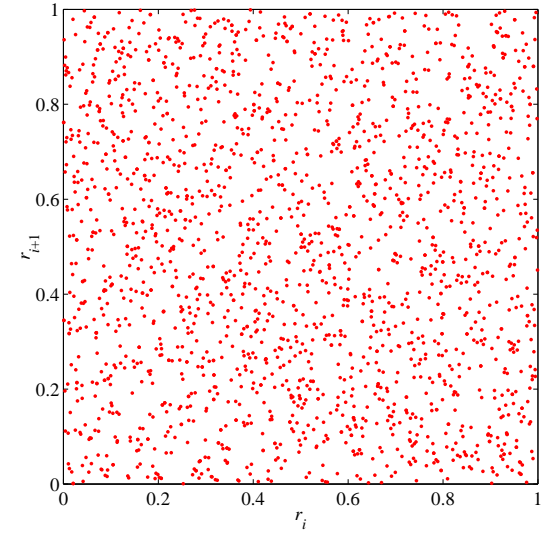
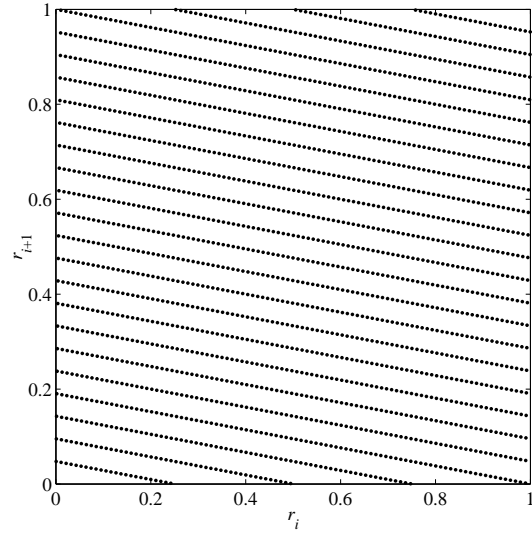
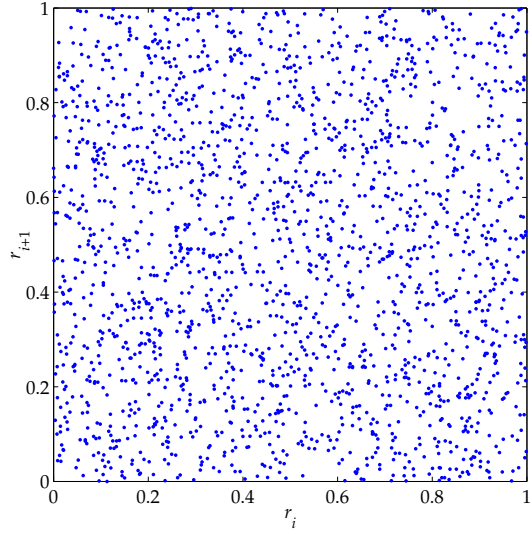
10^{12} random bits/s*
quality is a moving target



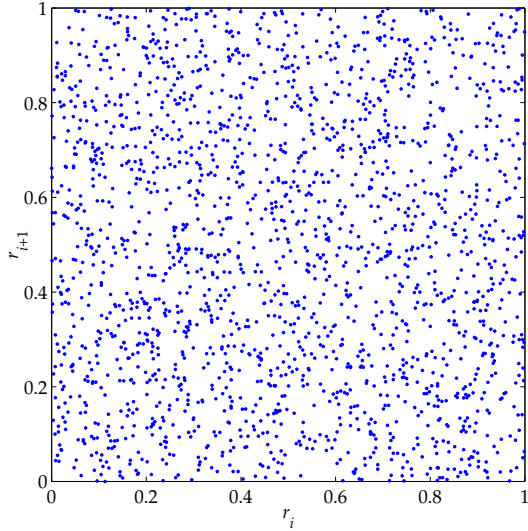
[†] Persi Diaconis, AAAS Conference, Seattle 2004

* <http://tina.nat.uni-magdeburg.de>

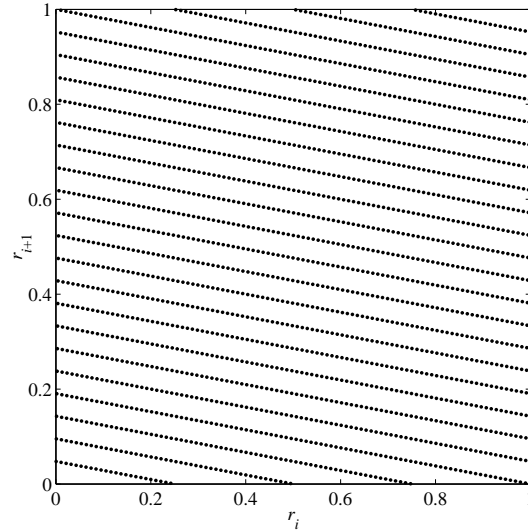
Example



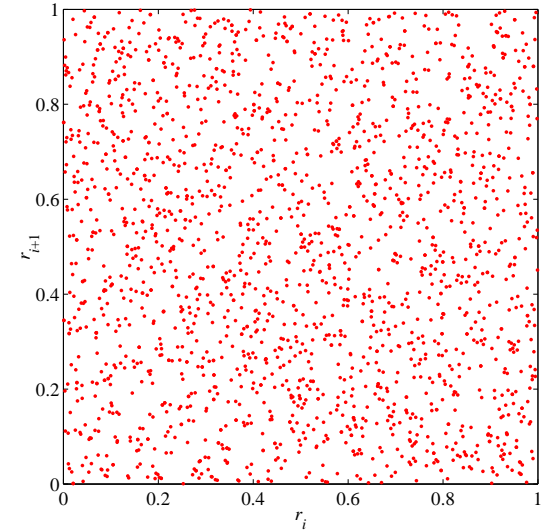
Example



`/dev/random`



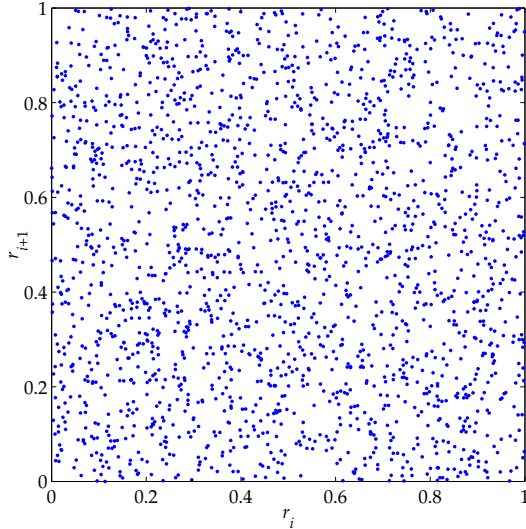
$$r_{i+1} = 95r_i \bmod 1999$$



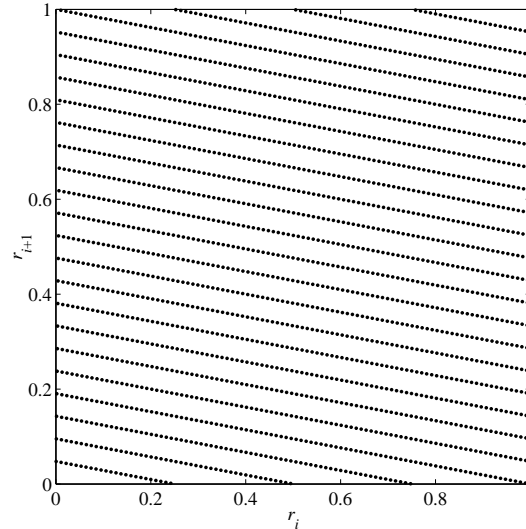
$$x_{i+1} = 95x_i \bmod 1999$$

$$r_i = 1099^{x_i} \bmod 1999$$

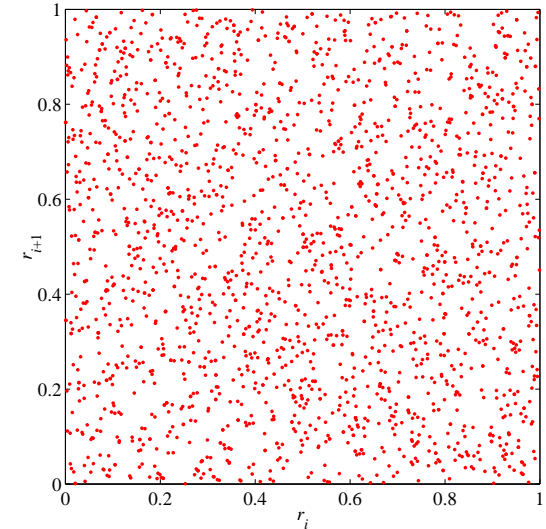
Example



/dev/random



$$r_{i+1} = 95r_i \bmod 1999$$



$$x_{i+1} = 95x_i \bmod 1999$$

$$r_i = 1099^{x_i} \bmod 1999$$

Random numbers should not be generated with a method chosen at random. Some theory should be used.

Donald E. Knuth (1969–1997)

Recursive Randomness

General scheme:

$$x_k = f(x_{k-1}, x_{k-2}, \dots, x_{k-p})$$

Recursive Randomness

General scheme:

$$x_k = f(x_{k-1}, x_{k-2}, \dots, x_{k-p})$$

All periodic sequences on a finite domain are **linear**:

$$x_k = a_1x_{k-1} + a_2x_{k-2} + \dots + a_px_{k-p} \pmod{m}$$

Recursive Randomness

General scheme:

$$x_k = f(x_{k-1}, x_{k-2}, \dots, x_{k-p})$$

All periodic sequences on a finite domain are **linear**:

$$x_k = a_1x_{k-1} + a_2x_{k-2} + \dots + a_px_{k-p} \pmod{m}$$

Linear **F**eedback **S**hift **R**egister sequences

LFSR Sequences

$$x_k = a_1 x_{k-1} + a_2 x_{k-2} + \cdots + a_p x_{k-p} \pmod{m}$$

LFSR Sequences

$$x_k = a_1 x_{k-1} + a_2 x_{k-2} + \cdots + a_p x_{k-p} \pmod{m}$$

If $x^p - a_1 x^{p-1} - \dots - a_p$ is **primitive** mod m then

● Period T is maximized: $T = m^p - 1$

LFSR Sequences

$$x_k = a_1 x_{k-1} + a_2 x_{k-2} + \cdots + a_p x_{k-p} \pmod{m}$$

If $x^p - a_1 x^{p-1} - \cdots - a_p$ is **primitive** mod m then

- Period T is maximized: $T = m^p - 1$
- Tuples $(x_{k+1}, x_{k+2}, \dots, x_{k+w})$ are uniformly distributed for $w \leq p$

LFSR Sequences

$$x_k = a_1 x_{k-1} + a_2 x_{k-2} + \cdots + a_p x_{k-p} \pmod{m}$$

If $x^p - a_1 x^{p-1} - \cdots - a_p$ is **primitive** mod m then

- Period T is maximized: $T = m^p - 1$
- Tuples $(x_{k+1}, x_{k+2}, \dots, x_{k+w})$ are uniformly distributed for $w \leq p$
- **pseudonoise sequences**

Pseudo Coin Tossing



$$x_k = x_{k-p} + x_{k-q} + \dots + x_{k-r} \bmod 2$$

pseudo noise sequence over \mathbb{Z}_2

aka $R(p, q, \dots, r)$

Pseudo Coin Tossing



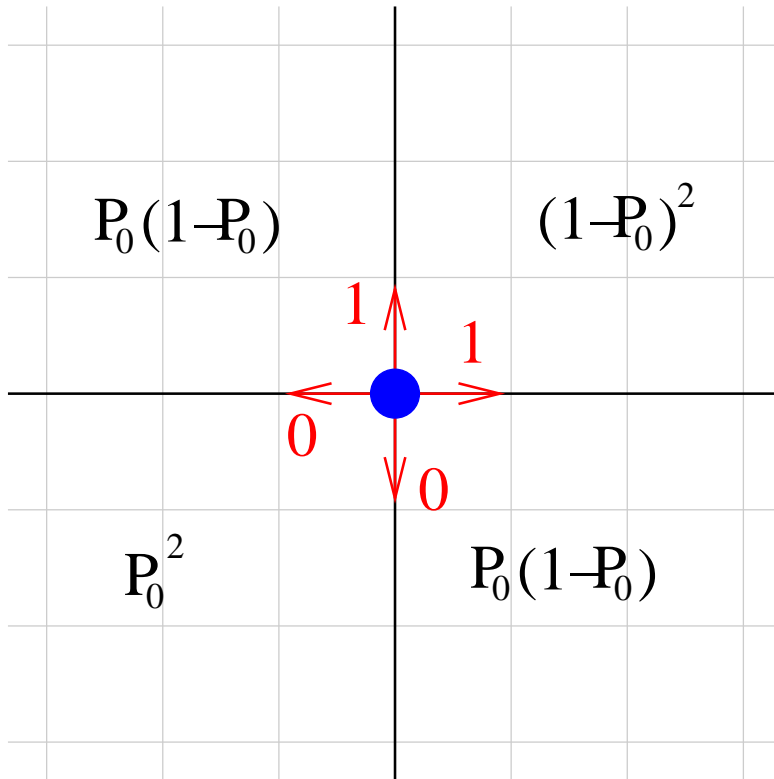
$$x_k = x_{k-p} + x_{k-q} + \cdots + x_{k-r} \bmod 2$$

pseudo noise sequence over \mathbb{Z}_2

aka $R(p, q, \dots, r)$

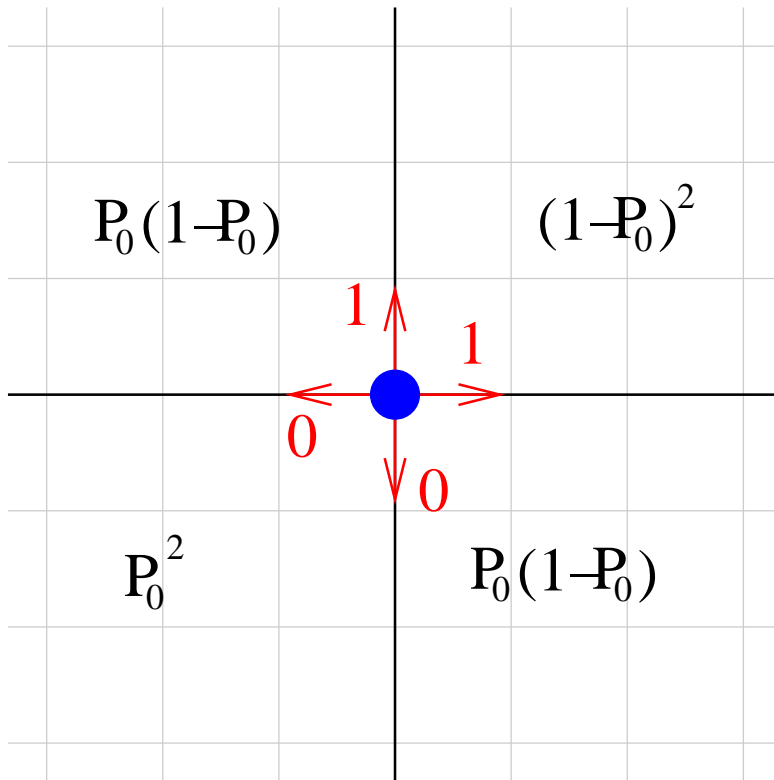
- proposed in 1959 (Green, Smith, Klem)
- popular in physics since 1981 (Kirkpatrick, Stoll)
- “well known instances”:
 $R(250, 103)$, $R(607, 273)$, $R(1279, 418)$.

Tracing the Random Walker



$P_0(w)$: Prob. of having more O 's than 1 's in w -tuple

Tracing the Random Walker



Vattulainen et. al ('94):

R(250, 103) **bad**

R(250, 201, 152, 103) **O.K.**

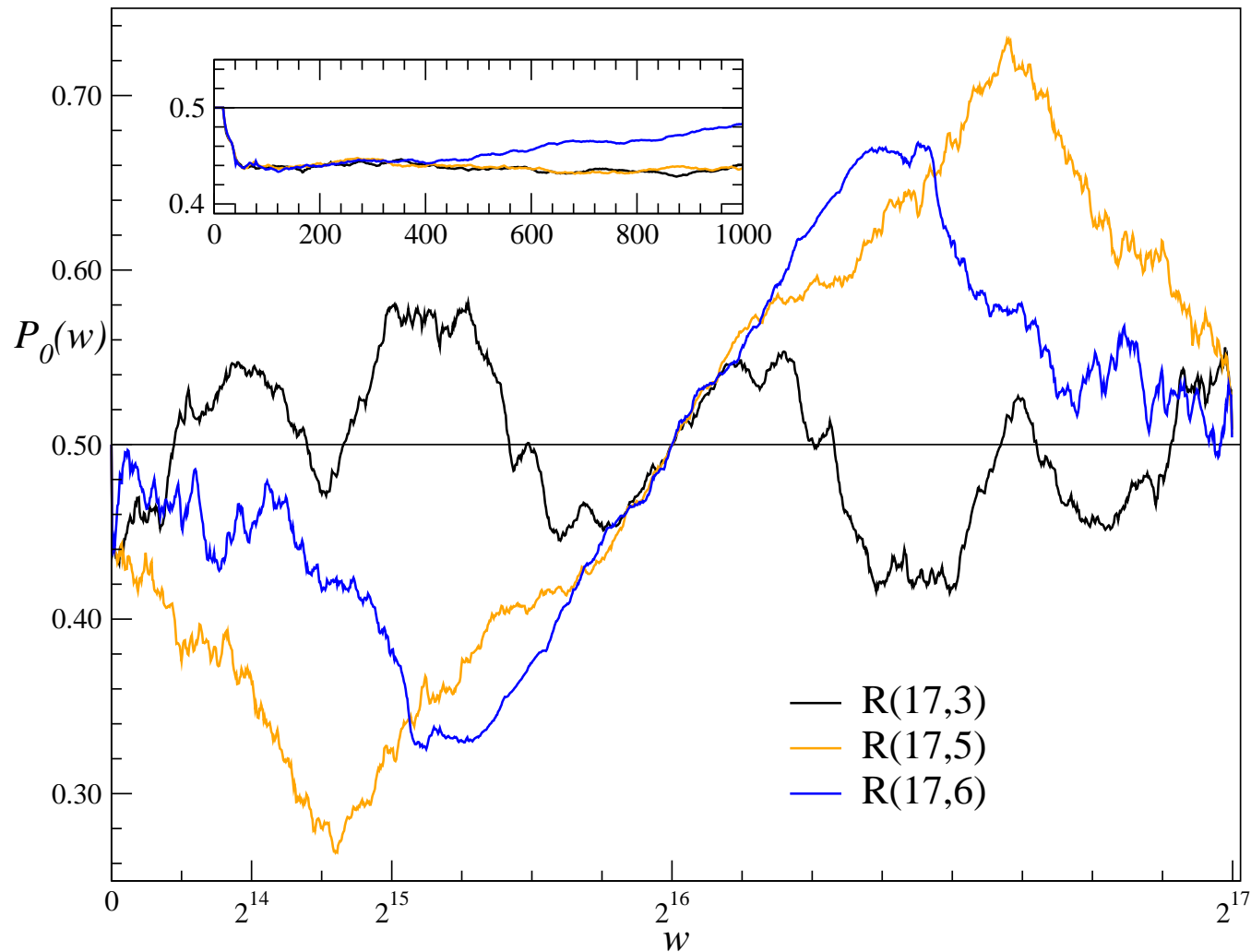
Ziff ('98):

R(9689, 471) **bad**

R(9689, 6988, 1586, 471) **O.K.**

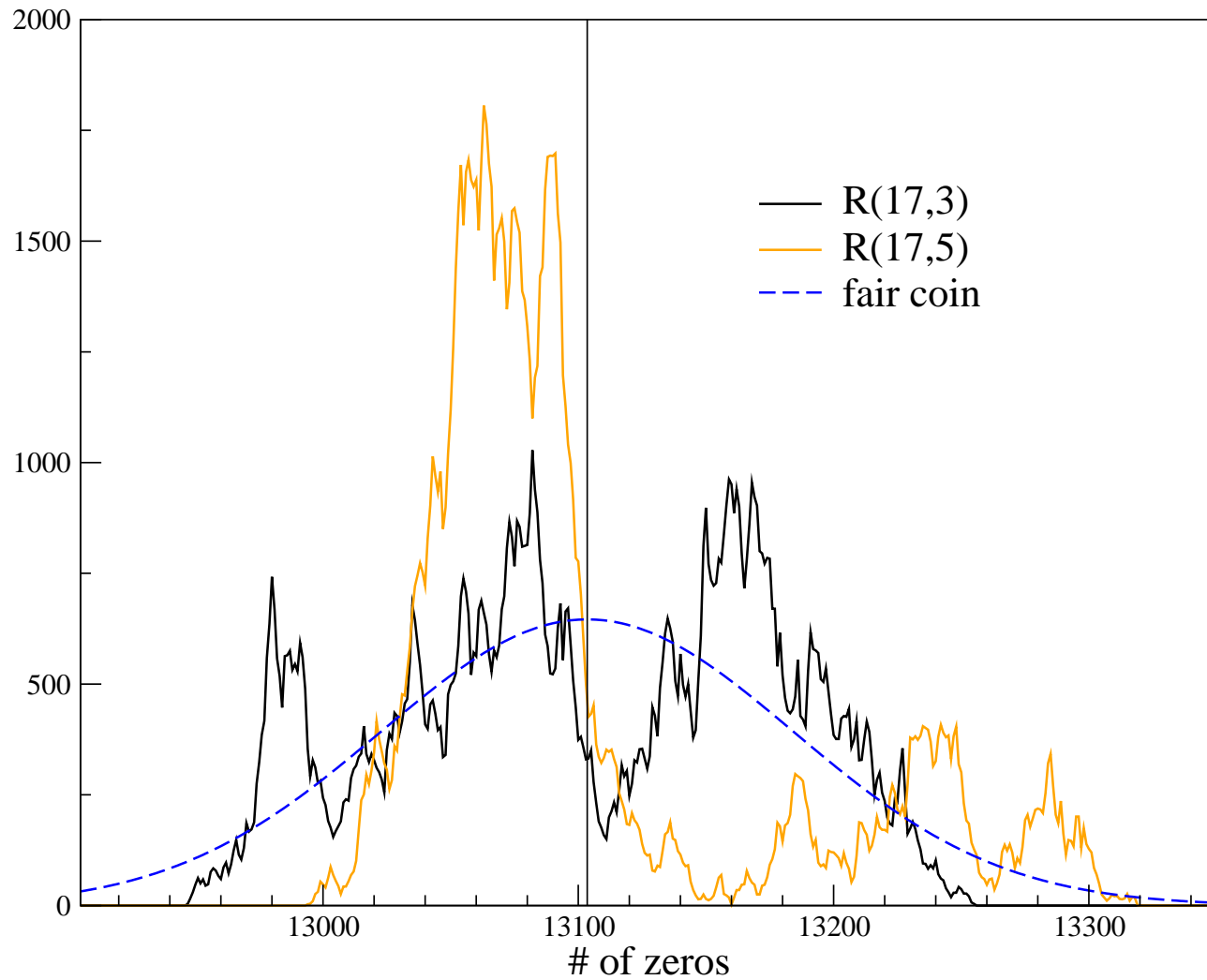
$P_0(w)$: Prob. of having more O 's than 1 's in w -tuple

Heads vs. Tails



Probability to have a majority of 0's in tuples of size w .

Heads vs. Tails



Tuples of size $w = 26207$.

Calculating $P_0(w)$

Generating function: $f_w(z) = \sum_{n=0}^w p_1(w, n) z^n$

|

Prob. of having n 1's in w -tuple

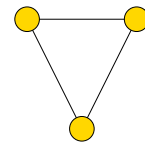
Calculating $P_0(w)$

Generating function: $f_w(z) = \sum_{n=0}^w p_1(w, n) z^n$

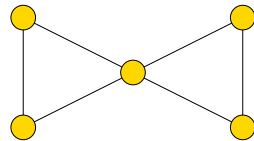
|

Prob. of having n 1's in w -tuple

• $\frac{1+z}{2}$



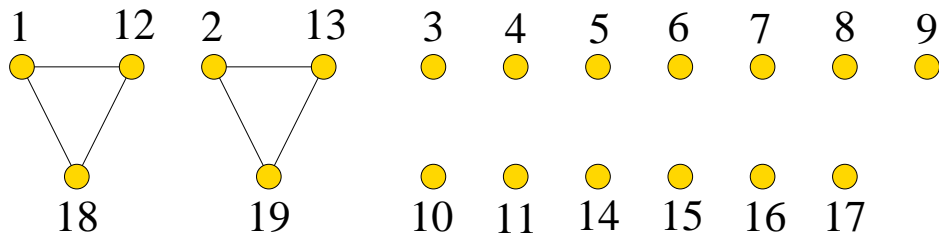
$$\frac{1+3z^2}{4}$$



$$\frac{1+2z^2+4z^3+z^4}{8}$$

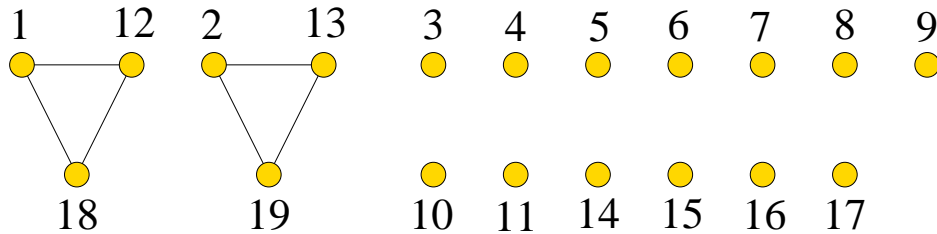
Calculating $P_0(w)$

Example: $P_0(19)$ for $R(17, 6)$



Calculating $P_0(w)$

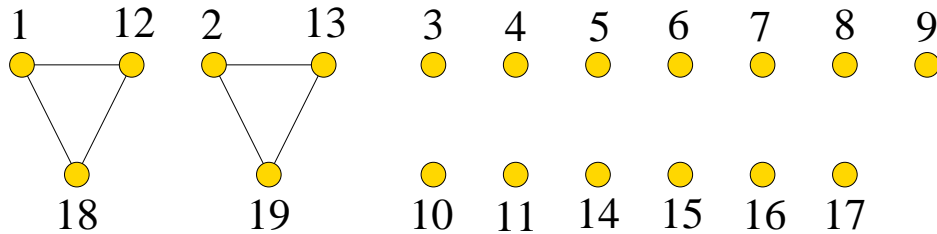
Example: $P_0(19)$ for $R(17, 6)$



$$f_{19}(z) = \left(\frac{1+z}{2}\right)^{13} \left(\frac{1+3z^2}{4}\right)^2 \quad P_0(19) = \frac{32\,053}{65\,536} \approx 0.4891$$

Calculating $P_0(w)$

Example: $P_0(19)$ for $R(17, 6)$

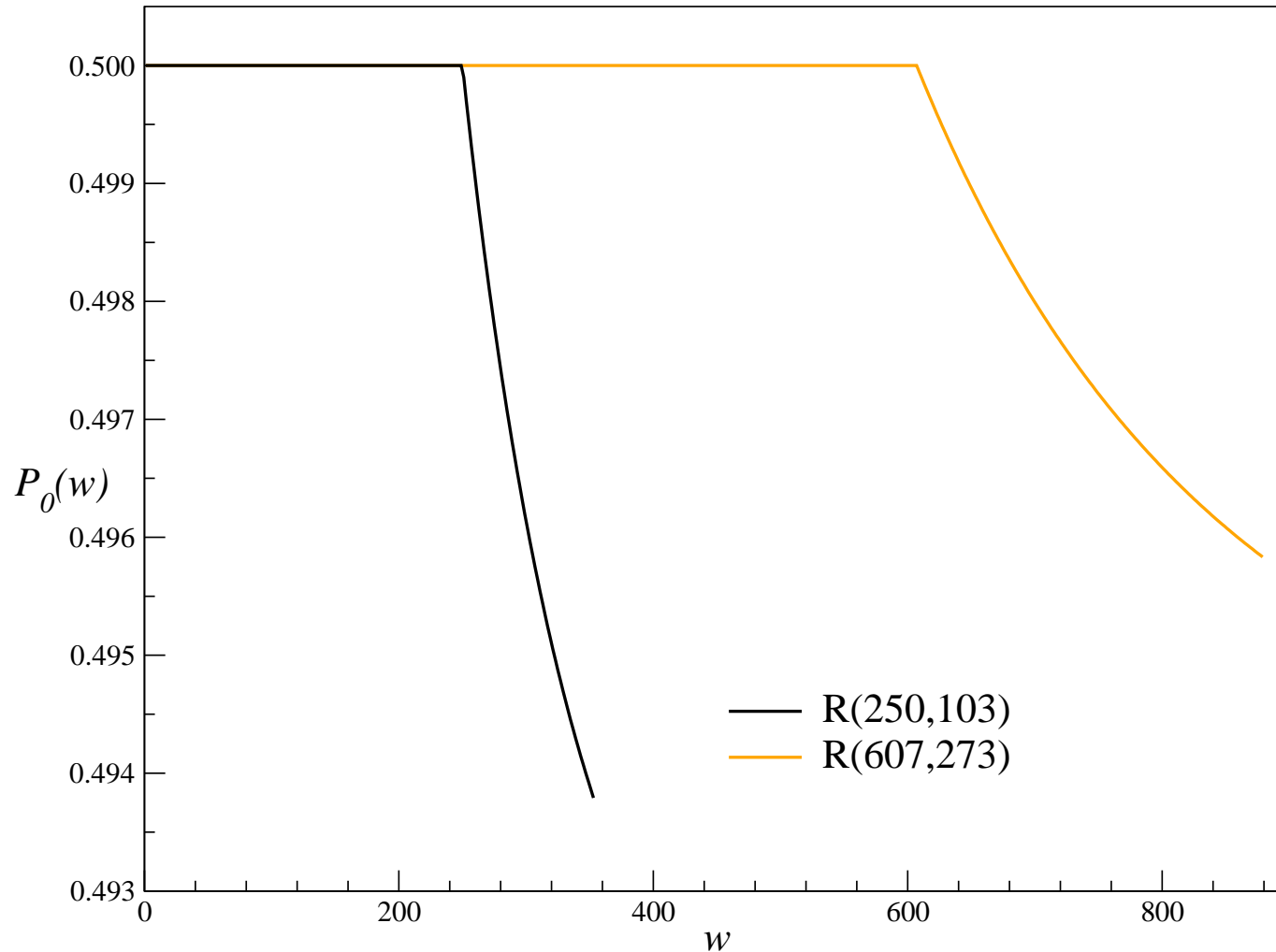


$$f_{19}(z) = \left(\frac{1+z}{2}\right)^{13} \left(\frac{1+3z^2}{4}\right)^2 \quad P_0(19) = \frac{32\,053}{65\,536} \approx 0.4891$$

General case: $R(p, q)$

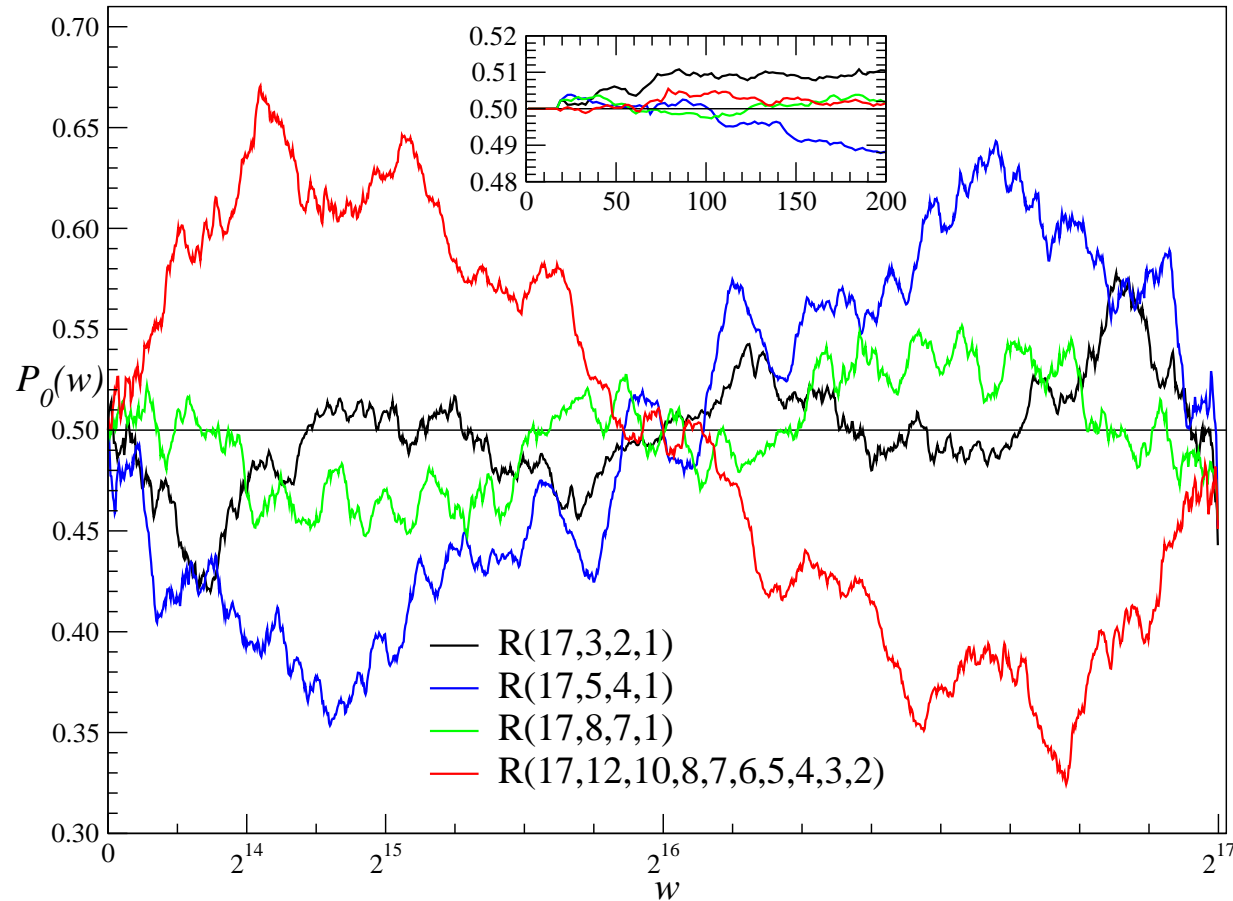
$$P_0(p+1) = \frac{1}{2} - \frac{1}{2^{p+1}(p-1)} \binom{p}{p/2} = \frac{1}{2} - \frac{1}{\sqrt{2\pi p^3}} + \mathcal{O}(p^{-5/2})$$

Heads vs. Tails



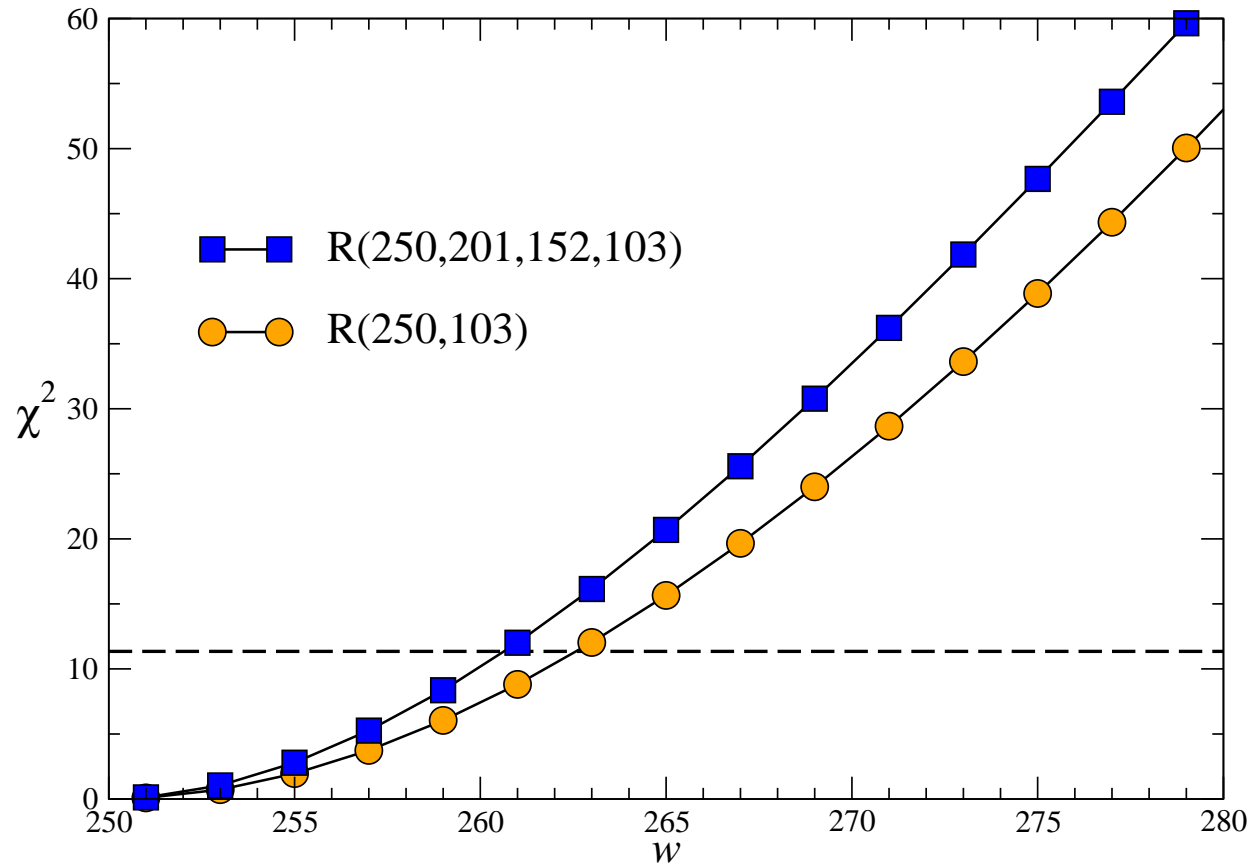
Bias in “industrial sized” random number generators .

More Feedback



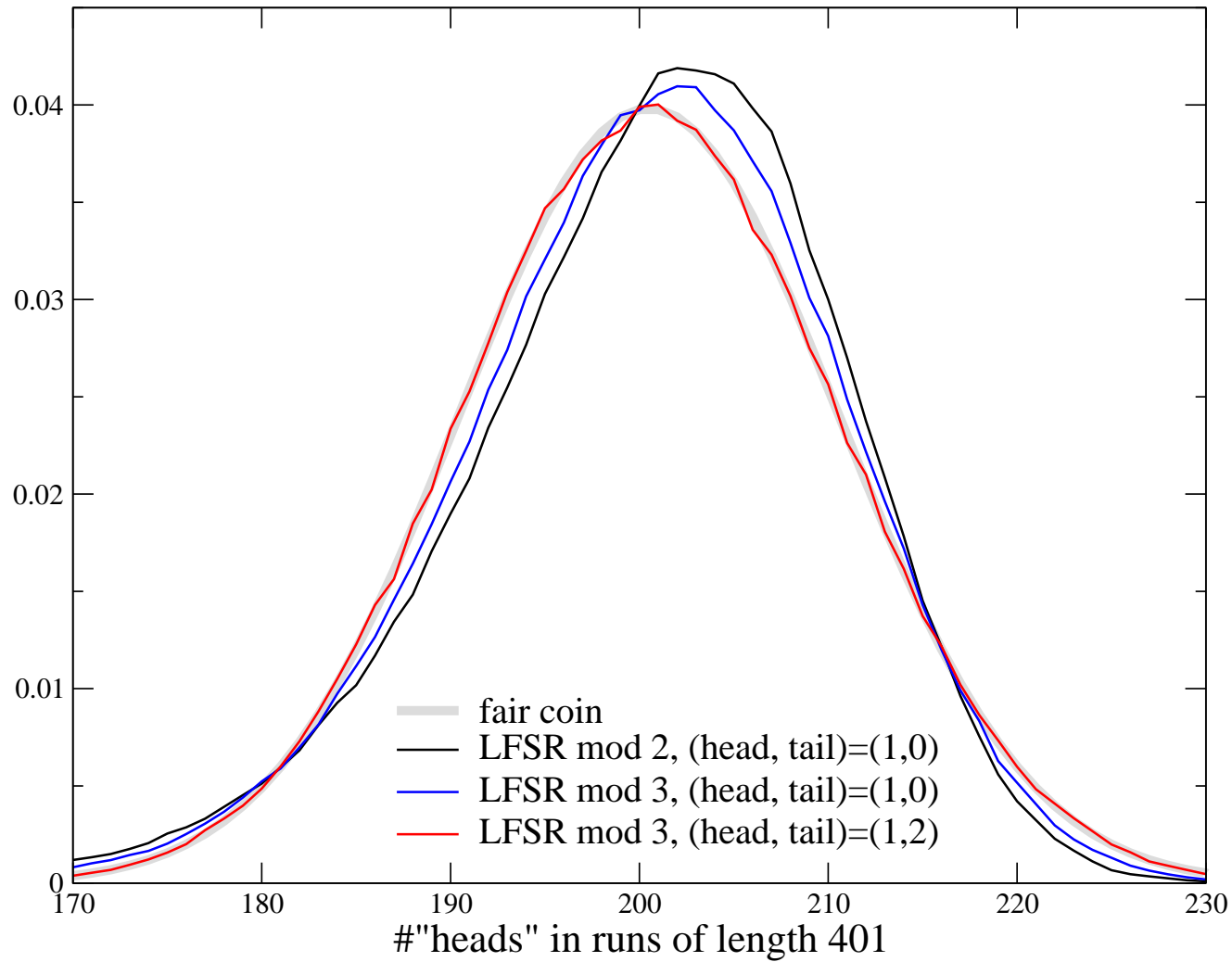
$$P_0(p+1) = \frac{1}{2} + (-1)^{t/2} \frac{(t-1)!!}{\sqrt{2\pi p^{t+1}}} + \mathcal{O}(p^{-\frac{t+3}{2}})$$

Bias in Random Walks



10^6 samples for $R(250, 103)$, 10^{10} for $R(250, 201, 152, 103)$.
empirical quality expires

Avoid the Zeros



The Ferrenberg Drama

Protagonists: Lagged Fibonacci RNGs $F(p, q, \circ)$,

$$x_k = x_{k-q} \circ x_{k-p} \bmod m$$

with properly chosen magic numbers p and q and

$$\circ \in \{\oplus, +, -, \times\}.$$

The Ferrenberg Drama

Protagonists: Lagged Fibonacci RNGs $F(p, q, \circ)$,

$$x_k = x_{k-q} \circ x_{k-p} \bmod m$$

with properly chosen magic numbers p and q and

$$\circ \in \{\oplus, +, -, \times\}.$$

The plot: cluster Monte Carlo simulations in Ising systems.

- $F(p, q, \times)$ o.k.
- $F(p, q, \pm)$ bad
- $F(p, q, \oplus)$ worse

The Wolff Algorithm

Initialize:

- Put a randomly chosen spin in the cluster.
- Put its equally aligned neighbours in the **candidate list**.

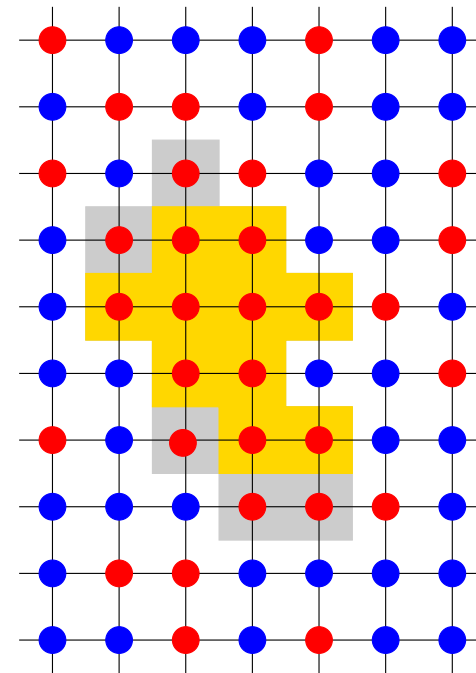
Grow cluster:

While candidate list of spins is not empty

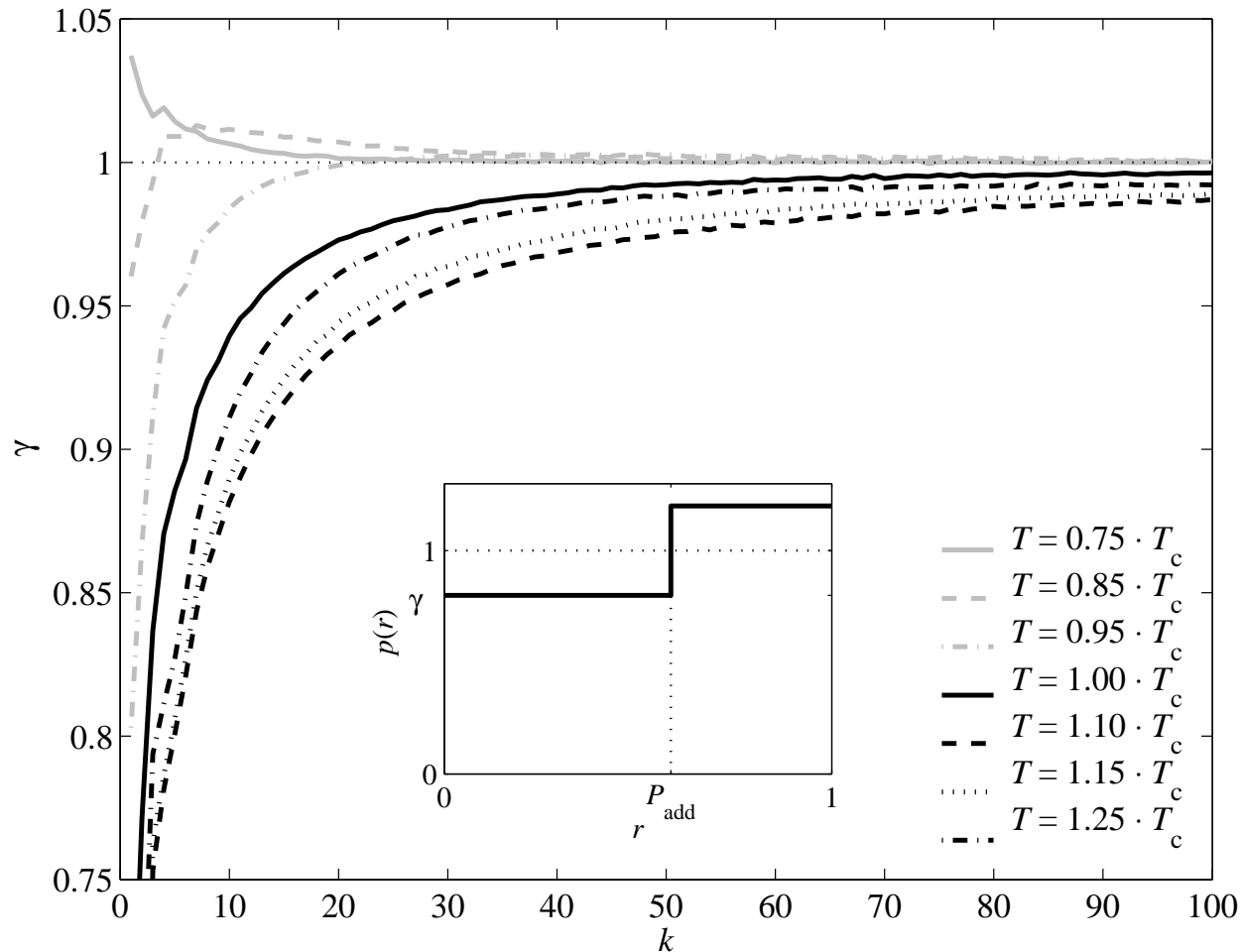
1. **remove** a spin from the **candidate list**
2. with probability $P_{\text{add}} = 1 - e^{-2/T}$:
 - **add** this spin to the cluster
 - **add** its equally aligned neighbors to the **candidate list** if they are not in the cluster

Flip all spins in the cluster.

Repeat

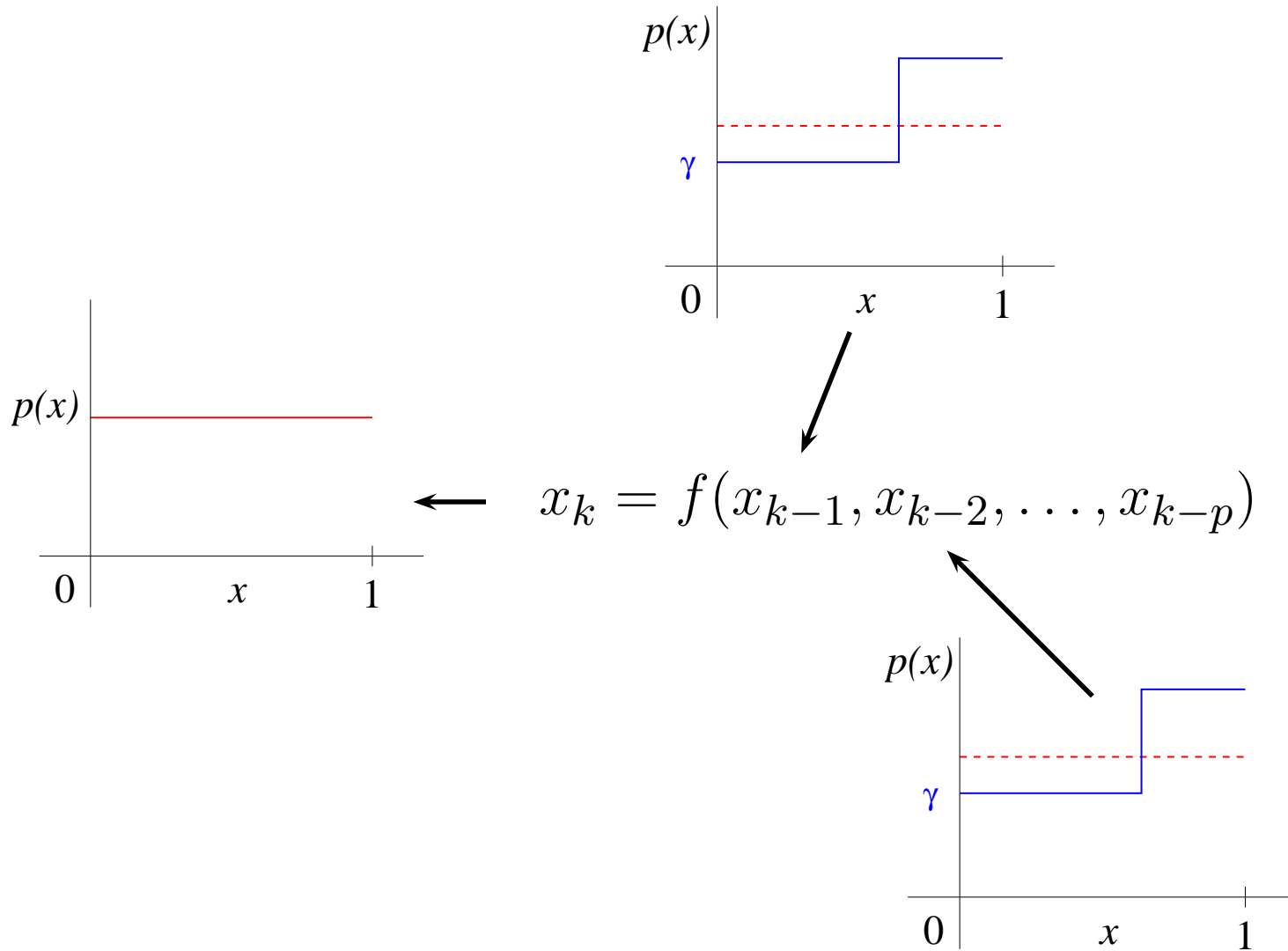


Algorithmically Induced Bias

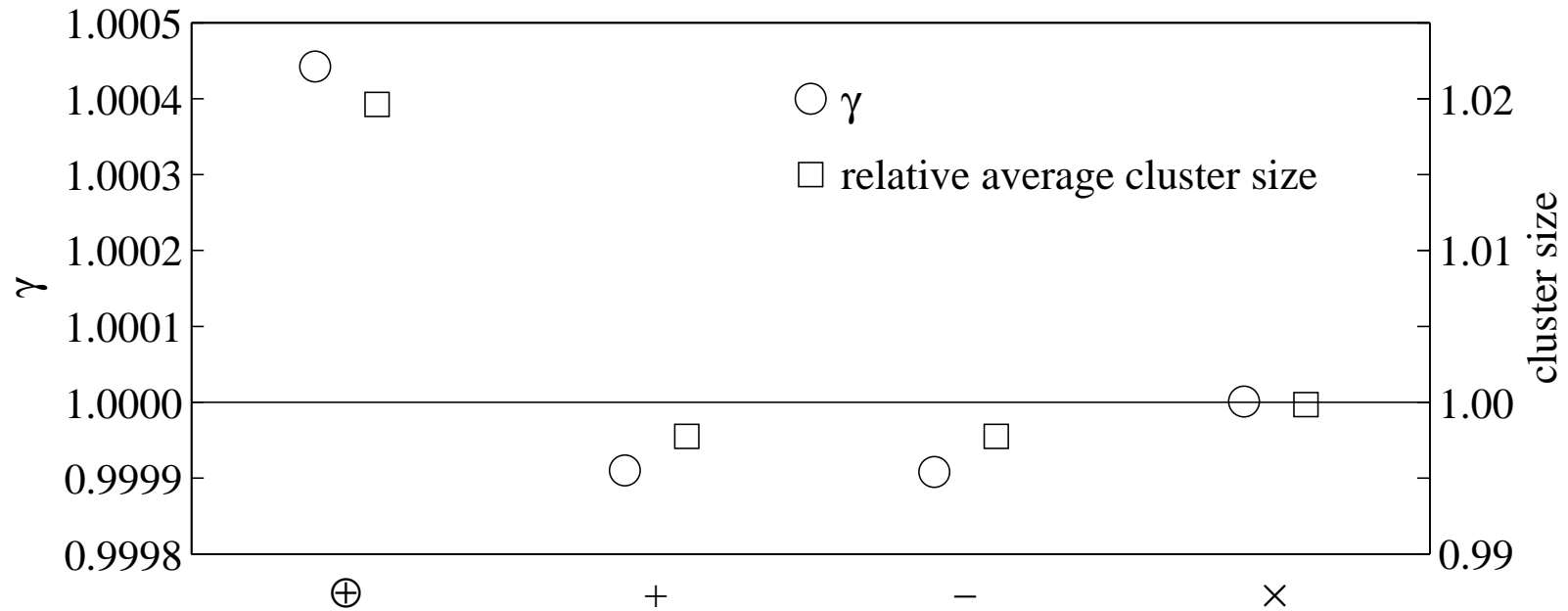


Wolff algorithm on a 24×24 spin Ising model.

Restoring the Flat Measure



Bias and Clustersize



$$\gamma_{\text{in}} = 0.975$$

Simulation: $F(13, 33, \circ)$ on a 16×16 spin system.

A Model RNG

$$r_k = \alpha(r_{k-1} + r_{k-2} + \cdots + r_{k-p}) \bmod 1 \quad r_i \in [0, 1)$$

A Model RNG

$$r_k = \alpha(r_{k-1} + r_{k-2} + \cdots + r_{k-p}) \bmod 1 \quad r_i \in [0, 1)$$

$\rho_k(r)$: pdf. of r_k :

$$\rho_k(r) = \frac{1}{\alpha} \sum_{j=0}^{\lfloor p\alpha \rfloor} \rho_{k-1} \star \rho_{k-2} \star \cdots \star \rho_{k-p} \left(\frac{r + j}{\alpha} \right)$$

A Model RNG

$$r_k = \alpha(r_{k-1} + r_{k-2} + \cdots + r_{k-p}) \bmod 1 \quad r_i \in [0, 1)$$

$\rho_k(r)$: pdf. of r_k :

$$\rho_k(r) = \frac{1}{\alpha} \sum_{j=0}^{\lfloor p\alpha \rfloor} \rho_{k-1} \star \rho_{k-2} \star \cdots \star \rho_{k-p} \left(\frac{r + j}{\alpha} \right)$$

This is the **Riemann sum approximation** of the integral

$$\int \rho_{k-1} \star \cdots \star \rho_{k-p}(x) dx = 1$$

A Model RNG

$$r_k = \alpha(r_{k-1} + r_{k-2} + \cdots + r_{k-p}) \bmod 1 \quad r_i \in [0, 1)$$

$\rho_k(r)$: pdf. of r_k :

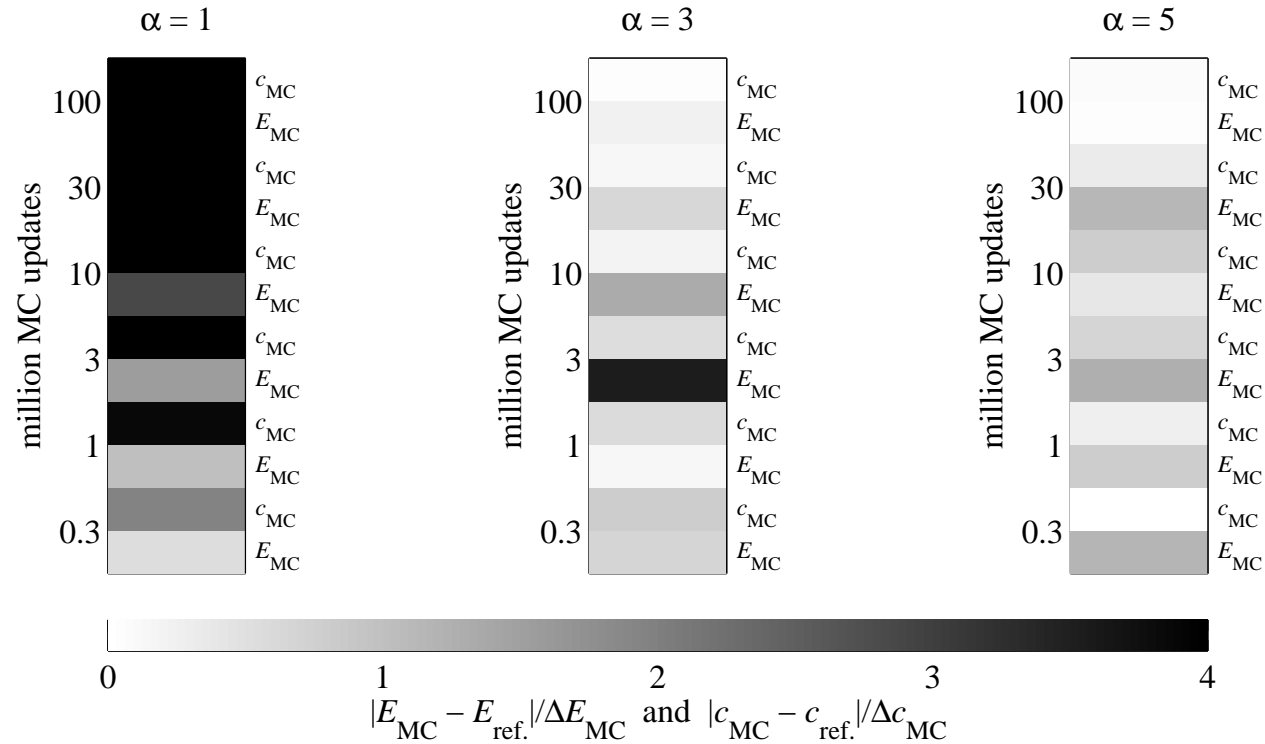
$$\rho_k(r) = \frac{1}{\alpha} \sum_{j=0}^{\lfloor p\alpha \rfloor} \rho_{k-1} \star \rho_{k-2} \star \cdots \star \rho_{k-p} \left(\frac{r + j}{\alpha} \right)$$

This is the **Riemann sum approximation** of the integral

$$\int \rho_{k-1} \star \cdots \star \rho_{k-p}(x) dx = 1$$

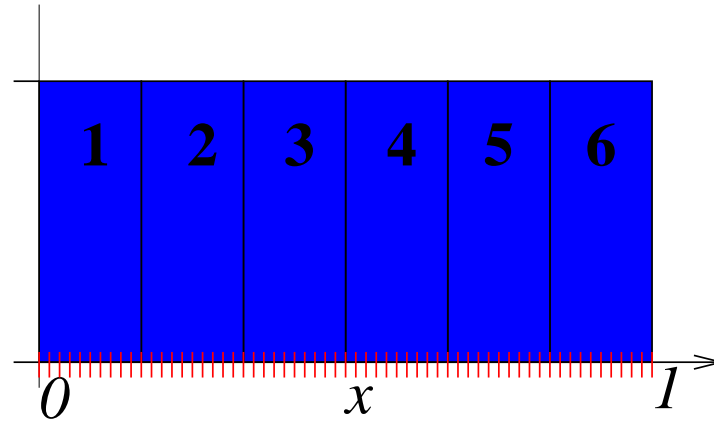
$\lim_{\alpha \rightarrow \infty} \rho_k(r) = 1$ independently of $\rho_{k-1}, \dots, \rho_{k-p}$

Curing the Ferrenberg Syndrome

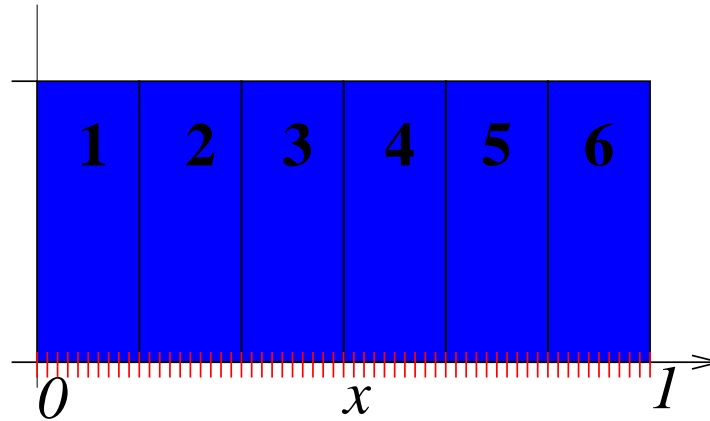


Wolff algorithm on $6 \times 6 \times 6$ Ising model
 RNG: $x_k = \alpha(x_{k-13} + x_{k-33}) \bmod (2^{31} - 1)$

Macrostate Entropy



Macrostate Entropy



2^{32} microstates

$$x_{n+1} = f(x_1, \dots, x_n)$$

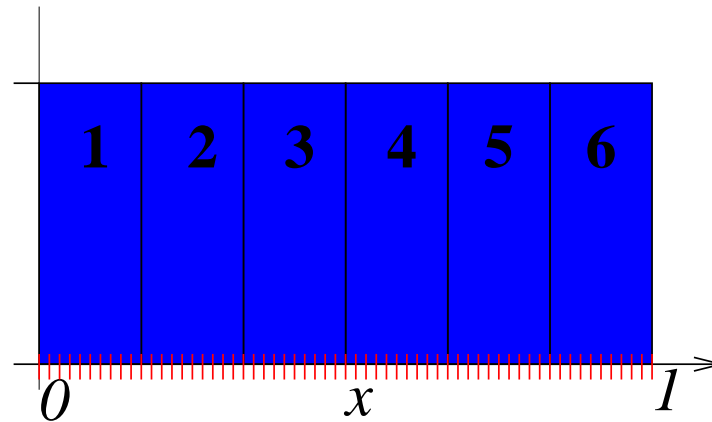
deterministic

M macrostates

$$(m_1, \dots, m_n) \mapsto m_{n+1}$$

random

Macrostate Entropy



2^{32} microstates

$$x_{n+1} = f(x_1, \dots, x_n)$$

deterministic

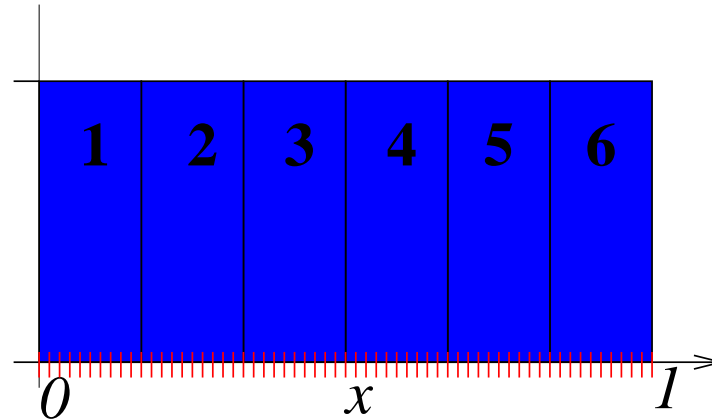
M macrostates

$$(m_1, \dots, m_n) \mapsto m_{n+1}$$

random

$$H = - \sum_{\{m_i\}} \mathbb{P}(m_1, \dots, m_{n+1}) \log_2 \frac{\mathbb{P}(m_1, \dots, m_{n+1})}{\mathbb{P}(m_1, \dots, m_n)} \leq \log_2 M$$

Macrostate Entropy



2^{32} microstates

$$x_{n+1} = f(x_1, \dots, x_n)$$

deterministic

M macrostates

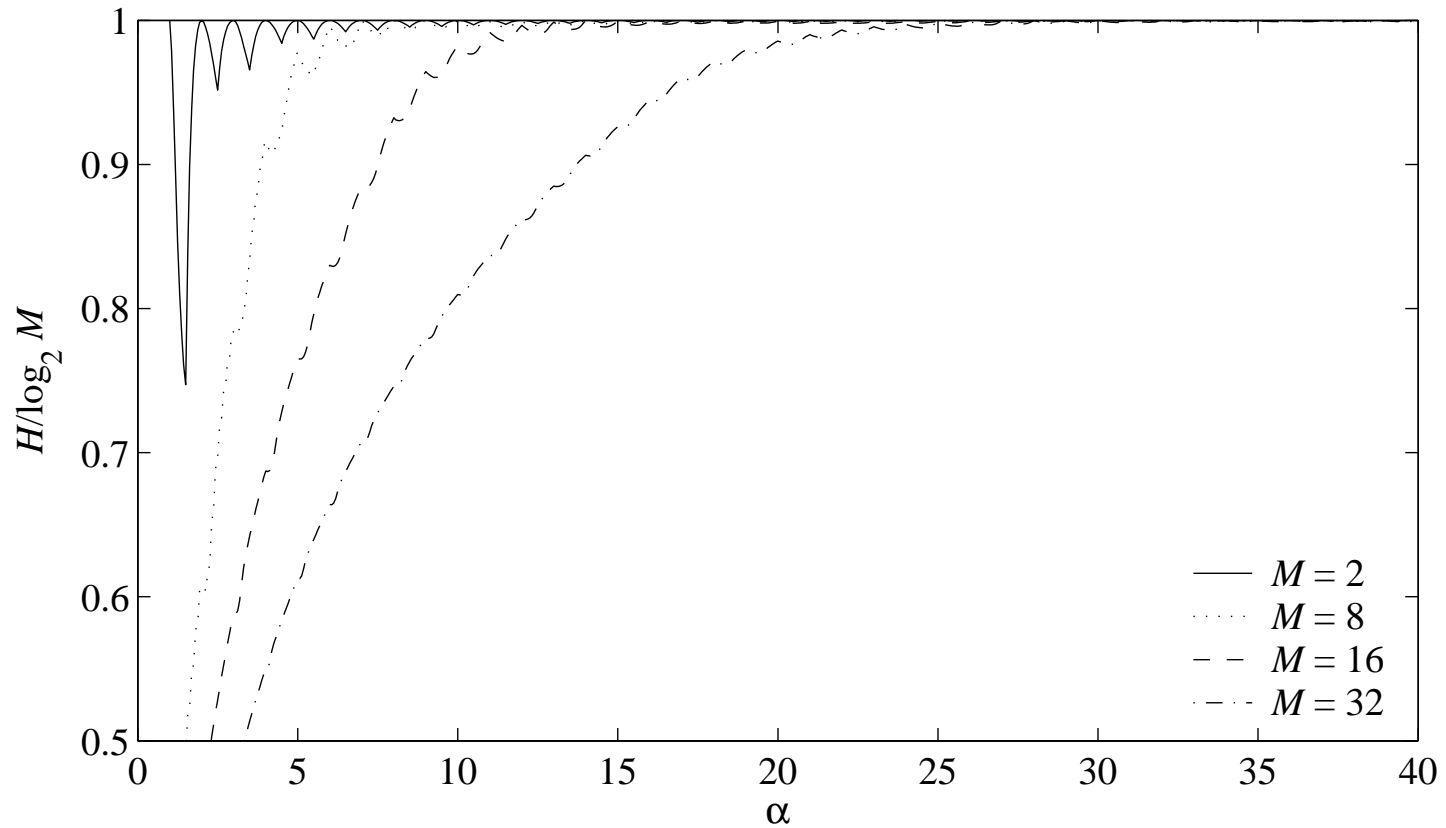
$$(m_1, \dots, m_n) \mapsto m_{n+1}$$

random

$$H = - \sum_{\{m_i\}} \mathbb{P}(m_1, \dots, m_{n+1}) \log_2 \frac{\mathbb{P}(m_1, \dots, m_{n+1})}{\mathbb{P}(m_1, \dots, m_n)} \leq \log_2 M$$

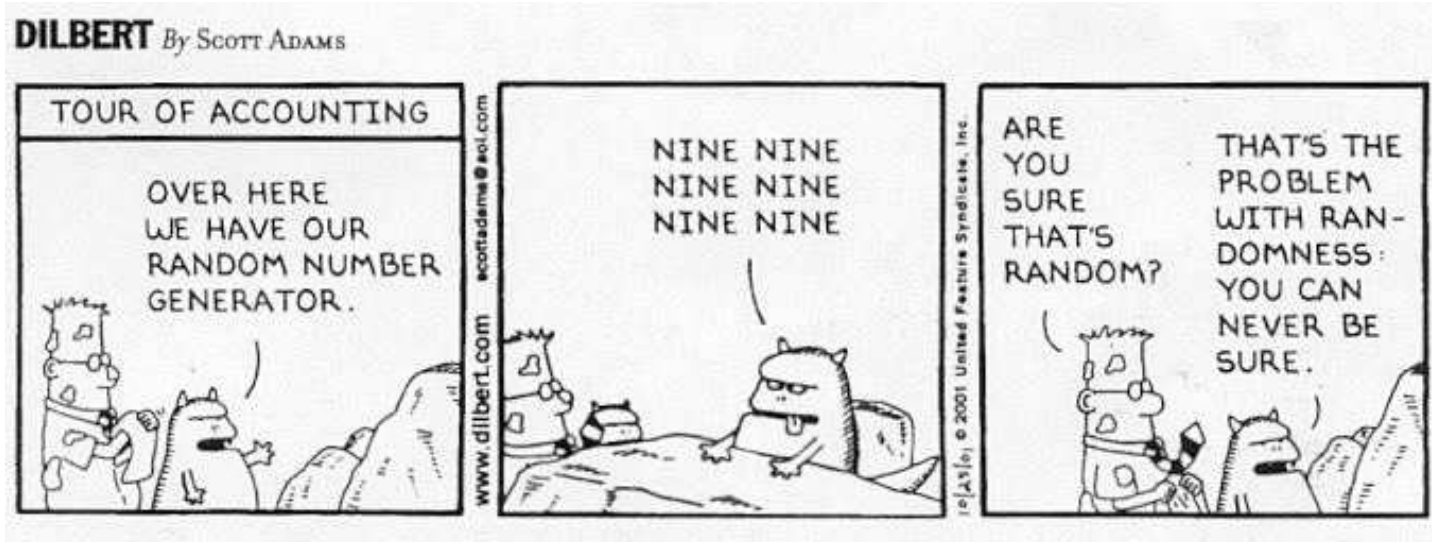
property of **production rule** f (RNG) and **macrostates** M (application) only

Macrostate Entropy



$$r_i = \alpha(r_{i-p} + r_{i-q}) \bmod 1$$

Finis



- Pseudo Random Coins Show More Heads Than Tails

H. Bauke, S.M., *J. Stat. Phys.* **114** 1149-1169 (2004)

- Entropy of Pseudo Random Number Generators

S.M., H. Bauke, *Phys. Rev. E* **69** 055702(R) (2004)

- <http://www.uni-magdeburg.de/mertens>

- <http://tina.nat.uni-magdeburg.de/intern/software/trng>