

# Rotor-routing, smoothing kernels, and reduction of variance: breaking the $O(1/n)$ barrier

Jim Propp (UMass Lowell, visiting UC Berkeley)

MSRI Research Seminar  
January 24, 2012

Slides for this talk are on-line at

<http://jamespropp.org/msri12a.pdf>

# Acknowledgments

This talk describes work that evolved from conversations and collaborations with David Einstein, Ander Holroyd, and Lionel Levine, as well as answers I received to questions I posted on MathOverflow (see <http://mathoverflow.net>).

# I. Introduction

## Monte Carlo simulation (the two-minute course)

Consider a sequence of identically distributed r.v.'s  $X_1, X_2, \dots$  with  $\text{Law}(X_n) = \text{Law}(X)$  for all  $n$ .

Let  $S_n := X_1 + \dots + X_n$ .

$\text{Exp}(S_n/n) = \text{Exp}(X)$ , and if the  $X_n$ 's are independent, the LLN says  $S_n/n \rightarrow \text{Exp}(X)$  almost surely.

## Monte Carlo simulation (the two-minute course)

When it's easy to generate IID samples from  $\text{Law}(X)$  but hard to compute  $\text{Exp}(X)$ , this gives a way to estimate  $\text{Exp}(X)$  empirically (the “Monte Carlo method”):

Estimate  $\text{Exp}(X)$  by  $S_n/n \sim \text{Exp}(X)$  for some large  $n$ .

If  $\text{Var}(X)$  is finite, then  $\text{Var}(S_n/n) = \text{Var}(S_n)/n^2 = n \text{Var}(X)/n^2 = \text{Var}(X)/n$ , so the standard deviation  $\sigma(S_n/n)$  of our estimate is typically  $O(1/\sqrt{n})$ .

## Quasi-Monte Carlo, aka “Casablanca simulation” (Customer: “Are you sure this place is honest?”)

Idea: We make  $\sigma(S_n/n)$  smaller than  $O(1/\sqrt{n})$  by “cheating”, i.e., by using NON-independent, identically distributed r.v.’s.

Here we are working in the space of all joinings  $\text{Law}(X_1, X_2, \dots)$  whose  $n$ th marginal is  $\text{Law}(X)$  for all  $n$ , and we are trying to simultaneously minimize the functionals  $\text{Var}(S_n/n)$ , or equivalently minimize the functionals  $\text{Var}(S_n)$ .

Note that  $S_n/n$  is automatically an unbiased estimator of  $\text{Exp}(X)$ .

## The Bernoulli case is special

For most r.v.'s, these goals conflict.

E.g., when  $\text{Law}(X)$  is uniform in  $[0, 1]$ ,  
there are joinings for which  $\text{Var}(S_2)$  is zero (easy),  
and other joinings for which  $\text{Var}(S_3)$  is zero (a fun puzzle),  
but there's no joining that achieves both at the same time.

But when  $\text{Law}(X)$  is Bernoulli( $p$ ), there's a law for  $X_1, \dots, X_n$   
that simultaneously minimizes all of the  $\text{Var}(S_n)$ 's.

## The first two-thirds of this talk

Such “maximally anticorrelated Bernoulli sequences” can be used as components of networks (“randomized rotor-router networks”) that compute anticorrelated sequences of r.v.’s of various kinds.

I’ll apply this to a random variable associated with absorbing Markov chains (namely the indicator r.v. of absorption in a particular state) and show that with randomized rotor-routers we get  $\text{Var}(S_n/n) = O((\log n)/n^2)$  (cf.  $\text{Var}(S_n/n) = O(1/n)$  for IID).

I’ll show that for a different but related scheme we get  $\text{Var}(S_n/n) = O(1/n^2)$ .

That is, the typical difference between  $S_n/n$  and  $\text{Exp}(X)$  is  $O(1/n)$ .

Moreover, I’ll show that the tail of the distribution of  $S_n - \text{Exp}(X)$  isn’t just small; it vanishes beyond a certain point.

## The last third of this talk

The preceding result is clearly best possible;  $\text{Var}(S_n)$  can't be  $o(1)$ , since  $S_{n-1}$  and  $S_n$  differ by an instance of  $X$ .

So  $\text{Var}(S_n/n)$  can't be  $o(1/n^2)$  and  $\sigma(S_n/n)$  can't be  $o(1/n)$ .

But in the last part of this talk, I'll describe how to use  $X_1, \dots, X_n$  to get estimates of  $\text{Exp}(X)$  with typical error  $o(1/n)$ .

## II. Rotor-routing

## “MAID” (Maximally Anticorrelated, Identically Distributed) Bernoulli sequences

With  $0 < p < 1$ , and  $U$  uniform in  $[0, 1)$ , let

$X_n = \lfloor U + np \rfloor - \lfloor U + (n-1)p \rfloor \in \{0, 1\}$  for all  $n \geq 1$ .

Then  $\text{Law}(X_n)$  is Bernoulli( $p$ ), and

$$S_n = \lfloor U + np \rfloor - \lfloor U \rfloor \in \{\lfloor np \rfloor, \lceil np \rceil\},$$

so  $S_n$  has variance as small as it can be, subject to  $\text{Exp}(S_n) = np$ .

$(X_1, X_2, \dots)$  is a random Sturmian sequence of density  $p$ .

## A physical model

We have a spinner consisting of an arrow pinned to a disk; the center of the arrow is pinned to the center of a disk and the arrow is free to rotate. The disk has a black sector occupying a proportion  $0 < p < 1$  of the disk.

If we were to repeatedly randomize the arrow, outputting a 1 or a 0 according to whether the arrow pointed into the black sector or not, we would get an IID Bernoulli( $p$ ) process.

But instead we randomize the arrow just once at the start, and on subsequent turns we merely rotate it counterclockwise  $p$  turns around the circle, so that our sequence of 1's and 0's is a MAID Bernoulli( $p$ ) process.

## Irrational $p$ vs. rational $p$

When  $p$  is irrational, there is a measure-preserving almost-bijection between the circle  $\mathbf{R}/\mathbf{Z}$  and the set of Sturmian sequences of density  $p$ .

When  $p$  is rational, say  $p = a/b$  in lowest terms, then there are just  $b$  Sturmian sequences of density  $p$ , and the MAID Bernoulli( $p$ ) process gives them equal likelihood.

E.g., for  $p = 2/5$ , the law of  $(X_1, X_2, \dots)$  is supported on five infinite sequences of period 5, each of which has probability  $1/5$ :

$(1, 0, 1, 0, 0, \dots)$

$(0, 1, 0, 1, 0, \dots)$

$(0, 0, 1, 0, 1, \dots)$

$(1, 0, 0, 1, 0, \dots)$

$(0, 1, 0, 0, 1, \dots)$

# Markov chains and hitting probabilities

Consider a finite-state absorbing Markov chain with a designated source-state  $s$  and absorbing states (“target-states”)  $t_1, \dots, t_m$ , such that  $\text{Prob}(\text{the chain eventually enters } \{t_1, \dots, t_m\} \mid \text{the chain starts at state } s) = 1$ .

Let  $t^*$  be one of the target states, and let  $p^*$  be the probability that the chain eventually enters state  $t^*$ .

Then  $p^* = \text{Exp}(X)$ , where  $X$  is 1 or 0 according to whether a run of the Markov chain leads to absorption at  $t^*$  or absorption elsewhere.

## Particles on a directed graph

It's convenient to imagine that the states of the chain are vertices in a directed graph, and that the evolution of the chain corresponds to the trajectory of a particle that travels from the  $i$ th vertex to the  $j$ th vertex when the Markov chain goes from state  $i$  to state  $j$ .

## Gambler's ruin

For simplicity, assume that each state  $i$  of the Markov chain has just two successors, with probabilities  $p_i$  and  $1 - p_i$ .

We'll focus on the Markov chain with state-space  $\{0, 1, 2, 3\}$  where 1 is the source and 0 and 3 are the targets, with all allowed transitions of the form  $i \rightarrow i + 1$  and  $i \rightarrow i - 1$ , with  $\text{Prob}(1 \rightarrow 2) = \text{Prob}(2 \rightarrow 3) = p$ .

(This corresponds to the purse-size of a gambler who starts out with \$1 and makes a succession of bets, gaining \$1 with probability  $p$  and losing \$1 with probability  $1 - p$ , until he either has \$3 and leaves happy or has \$0 and leaves broke.)

Let  $t^* = 3$ . It's easy to prove that  $p^* = p^2 / (1 - p + p^2)$ . But what if we want to estimate  $p^*$  empirically by running the chain?

## Monte Carlo made complicated

To simulate the gambler's ruin Markov chain, we use random variables  $U_{i,t}$  ( $i \in \{1, 2\}$ ,  $t \in \{1, 2, 3, \dots\}$ ) where each  $U_{i,t}$  is  $U(0, 1)$ .

If the chain is in non-absorbing state  $i$  at time  $t$ , let its state at time  $t + 1$  be  $i + 1$  if  $U_{i,t} \in [0, p)$  and  $i - 1$  otherwise.

## Monte Carlo made even more complicated

To simulate the gambler's ruin Markov chain repeatedly (in “supertime”) we use random variables  $U_{i,t,k}$  ( $i \in \{1, 2\}$ ,  $t \in \{1, 2, 3, \dots\}$ ,  $k \in \{1, 2, 3, \dots\}$ ) where  $U_{i,t,k}$  is the source of randomness we use when, for the  $k$ th supertime, a run of the Markov chain has put us in state  $i$  at time  $t$ .

All the  $U_{i,t,k}$ 's are independent of one another.

(It may be helpful to imagine a spinner at  $(i, t)$  that we spin each supertime the Markov chain is in state  $i$  at time  $t$ .)

When we reach a target state, we start again from the source state.

Each run will result in absorption at 0 or absorption at 3, outputting a Bernoulli( $p^*$ ) bit.

The successive Bernoulli( $p^*$ ) bits will be IID.

# Anticorrelated quasi-Monte Carlo

Instead of having each sequence  $U_{i,t,1}, U_{i,t,2}, \dots$  be IID, have each sequence be MAID.

That is,  $U_{i,t,k+1} = U_{i,t,k} + p \pmod{1}$ .

Once again there is a spinner at  $(i, t)$ , but instead of randomizing it each time we use it, we rotate it by  $p$  turns (i.e., by an angle of  $2\pi p$ ).

We call this kind of spinner a rotor.

## Rotor-routing

(see Holroyd, Levine, Mészáros, Peres, P., and Wilson, “Chip-Firing and Rotor-Routing on Directed Graphs,” [arXiv:0801.3306](#))

The particle starts at  $i_1 = s$ .

For  $t = 1, 2, \dots$  in succession, we update the rotor at  $(i_t, t)$  (incrementing it by  $p_{i_t} \pmod{1}$ ) and use it to decide which  $(i_{t+1}, t + 1)$  the particle should go to, until the particle gets absorbed at a target.

Rotors can be used for general finite-state Markov chains (even when states have more than two successors), and indeed for some infinite-state Markov chains; see Holroyd and Propp, “Rotor Walks and Markov Chains”, [arXiv:0904.4507](#).

# Stationarity

Theorem (P.): The operation of updating the rotors by sending a particle through the network (incrementing some of the rotors (mod 1) and leaving the rest alone) is a bijection from  $\mathbf{R}/\mathbf{Z} \times \mathbf{R}/\mathbf{Z} \times \cdots$  to itself that preserves product measure.

Corollary: If the initial setting of the rotors is governed by product measure, the outcome of the  $k$ th run (1 if the  $k$ th run leads to  $t^*$ , 0 otherwise) has law  $\text{Bernoulli}(p^*)$ .

That is, the sequence of bits arising from anticorrelated Monte Carlo with “spacetime rotors” will be a sequence of (identically distributed)  $\text{Bernoulli}(p^*)$  bits.

# Confluence

Imagine that instead of just one particle in the directed graph we have many; at each step we can advance only one particle (by updating the rotor at the vertex it occupies and then moving the particle in the direction indicated by the rotor), but we get to choose which particle to move, until all particles have been absorbed at target vertices.

Confluence Property (aka abelian property): The number of particles absorbed at  $t_i$  does not depend on the choices we make.

## Parallel rotor-routing

So, when sending  $n$  particles through the network, instead of sending one particle through the network at a time (“sequential rotor-routing”), we may start with all  $n$  particles at the source  $s$ , advance each particle one step, advance each not-yet-absorbed particle another step, advance each not-yet-absorbed particle another step, etc.

The number of particles absorbed at  $t^*$  will be the same for sequential rotor-routing and parallel rotor-routing.

## Spacetime rotors achieve logarithmic discrepancy

Theorem (P.): If one sends  $n$  particles through a finite network of spacetime rotors, the number that get absorbed at  $t^*$  differs from  $np^*$  by  $O(\log n)$  (so that the number of particles that get absorbed at  $t^*$ , divided by  $n$ , differs from  $p^*$  by  $O((\log n)/n)$  on average).

## Illustration of proof in a simple case

Consider \$3 gambler's ruin with  $p = 1/2$ ,  $p^* = 1/3$ .

Use parallel rotor-routing.

The sum of the positions of the particles is initially  $n \times 1 = n$ .

When there are an even number of particles at  $(i, t)$ , half go to  $(i - 1, t)$  and half go to  $(i + 1, t)$ , so the sum of the positions isn't changed.

When there are an odd number of particles at  $(i, t)$ , the sum of the positions changes by  $\pm 1$ .

So at each stage  $t$  the sum of the positions changes by  $\pm 2$  (since there are two sites  $i$ ).

## Punchline of proof

After  $O(\log n)$  stages, all particles have been absorbed at 0 or 3, so

$$A \times 0 + B \times 3 = n \pm O(\log n)$$

where  $A$  and  $B$  denote the number of particles absorbed at 0 and 3 respectively; hence

$$B = (1/3)n + O(\log n).$$

(The general case involves extra technology and in particular uses harmonic functions on directed graphs, but no new ideas are required.)

### III. A different kind of rotor-routing

## Monte Carlo revisited

Instead of having an IID process  $U_{i,t,1}, U_{i,t,2}, \dots$   
at each spacetime location  $(i, t)$ ,

have an IID process  $U_{i,1}, U_{i,2}, \dots$   
at each space location  $i$  (that is, at each vertex),  
so that a rotor may get used more than once during a run.

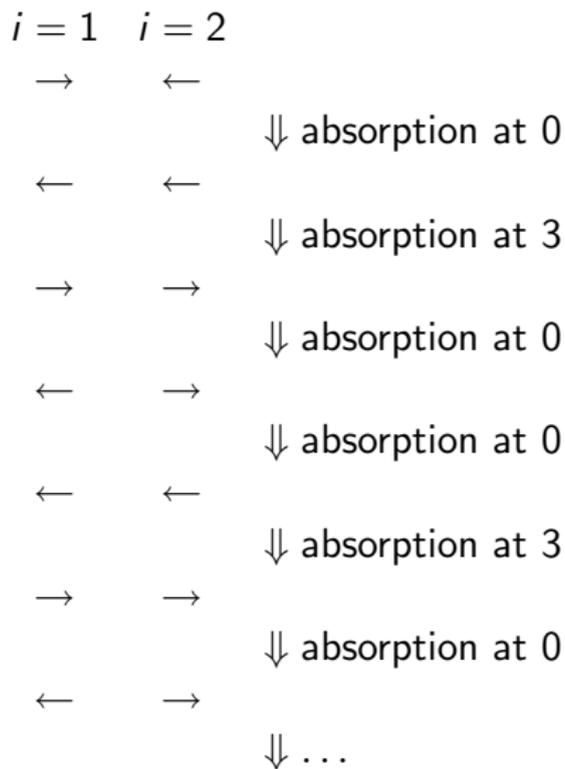
## Casablanca revisited

Replace each IID process by a MAID process.

Example: Consider \$3 gambler's ruin with  $p = 1/2$ .

1st particle:  $1 \rightarrow 0$   
2nd particle:  $1 \rightarrow 2 \rightarrow 3$   
3rd particle:  $1 \rightarrow 0$   
4th particle:  $1 \rightarrow 2 \rightarrow 1 \rightarrow 0$   
5th particle:  $1 \rightarrow 2 \rightarrow 3$   
6th particle:  $1 \rightarrow 0$   
7th particle:  $1 \rightarrow 2 \rightarrow 1 \rightarrow 0$   
etc. (with period 3)

# “Let’s see that again in configuration space”



## Cycles go away

Note that initially the rotors at 1 and 2 form a 2-cycle (with each pointing toward the other), but that thereafter the graph formed by the rotors is acyclic.

More generally, for rotor-routing on any finite graph, the initial configuration of the rotors may contain cycles, but eventually all the cycles go away.

## Stationarity revisited

Theorem (P.): The operation of updating the rotors by sending a particle through the network preserves the restriction of product measure to the set of “acyclic rotor configurations”.

(Note: It's easy to sample from this conditional distribution; use Wilson's partial rejection sampling scheme, aka cycle-popping, whereby you repeatedly rerandomize rotors that participate in cycles until there aren't any.)

## ID-ness revisited

Corollary: If the initial setting of the rotors is governed by the conditional measure on acyclic configurations, then the successive runs are identically distributed.

## Confluence revisited

If we send  $n$  particles through the network until all have been absorbed, the number of particles absorbed at  $t^*$  does not depend on the order in which we choose to move the particles.

# Hey, wait a minute

The successive runs don't look like runs of the original Markov chain (e.g., the time-to-absorption is never more than 3)!

## No, don't wait a minute

But if we just look at which absorbing state the Markov chain eventually enters (i.e., which target-vertex the particle gets absorbed at), the Casablanca process is governed by the same distribution as the Monte Carlo process (see Holroyd and Propp, “Rotor Walks and Markov Chains”, [arXiv:0904.4507](#)).

That is, if we let  $X_n$  be 1 if the  $n$ th particle gets absorbed at  $t^*$  and 0 otherwise, then  $\text{Law}(X_n) = \text{Bernoulli}(p^*)$ .

## Space rotors achieve constant discrepancy

Theorem (Holroyd-P.): If one sends  $n$  particles through a finite network of space rotors, the number that get absorbed at  $t^*$  differs from  $np^*$  by  $O(1)$  (so that the number of particles that get absorbed at  $t^*$ , divided by  $n$ , differs from  $p^*$  by  $O(1/n)$ ).

Compare this with our result on spacetime rotors: not only is the bound better (by a log factor), but we don't have to say "on average"!

## Illustration of proof in a simple case

Consider \$3 gambler's ruin with  $p = 1/2$ ,  $p^* = 1/3$ .

Use parallel rotor-routing.

The sum of the positions of the particles is initially  $n \times 1 = n$ .

We repeat the analysis we used for spacetime rotors, but we now note that if at some stage vertex  $i$  introduces a change of  $+1$  in the sum of the positions, then the next time there are an odd number of particles at vertex  $i$ , the sum of the positions will change by  $-1$ , and vice versa.

## Punchline of proof

When all the particles have been absorbed at 0 or 3,

$$A \times 0 + B \times 3 = n \pm 2$$

where  $A$  and  $B$  denote the number of particles absorbed at 0 and 3 respectively; hence

$$B = (1/3)n \pm 2/3.$$

## IV. Kernel smoothing

## The $O(1/n)$ barrier

$\text{Var}(S_n)$  can't be  $o(1)$ , since  $S_{n-1}$  and  $S_n$  differ by an instance of  $X$ .

So  $\sigma(S_n/n)$  can't be  $o(1/n)$ .

## Breaking the $O(1/n)$ barrier

To get past the barrier, we consider weighted sums

$$S_n^* = (a_1 X_1 + a_2 X_2 + \cdots + a_n X_n) / (a_1 + a_2 + \cdots + a_n)$$

with the  $a_i$ 's not equal to one another (and depending on  $n$ ).

(If the  $X_i$ 's were IID, this wouldn't help at all!)

Note that such a weighted average is automatically an unbiased estimator of  $\text{Exp}(X)$ .

I'll call the family of weights  $a_{n,i}$  ( $n \geq 1$ ,  $1 \leq i \leq n$ ) a smoothing kernel, since a similar device in nonparametric estimation goes by that name.

## Kernel smoothing: a digression and advertisement

Kernel smoothing can be successfully applied to estimating quantities in analysis

(the mean value of an almost periodic function),

geometry

(the density of a point-set with discrete spectrum),

and number theory

(the asymptotic average of an arithmetic function);

see the slides for my talk “How well can you see the slope of a digital line? (and other applications of averaging kernels)”,

<http://jamespropp.org/Slope.cdf>.

But I haven't got a general idea for when kernel smoothing should work and how much improvement it should yield; there doesn't seem to be literature on it.

## Parabolic weights

A good choice of weights for many applications, and rotor-routing in particular, is

$$a_{n,i} = i(n + 1 - i),$$

which I'll adopt hereafter.

The resulting weighted average is the slope of the least-squares regression line through the points

$$(1, S_1), (2, S_2), \dots, (n, S_n)$$

where  $S_j = X_1 + X_2 + \dots + X_j$ .

## Rational $p^*$

Example: Suppose all transition probabilities in the Markov chain are rational, so that the sequence  $X_1, X_2, \dots$  is periodic, and  $p^*$  is rational.

Theorem (Einstein): The difference between the weighted sum  $S_n^*$  and the absorption probability  $p^*$  is  $O(1/n^2)$ .

(Note: The constant implicit in the  $O(1/n^2)$  depends on the Markov chain.)

## Random $p$

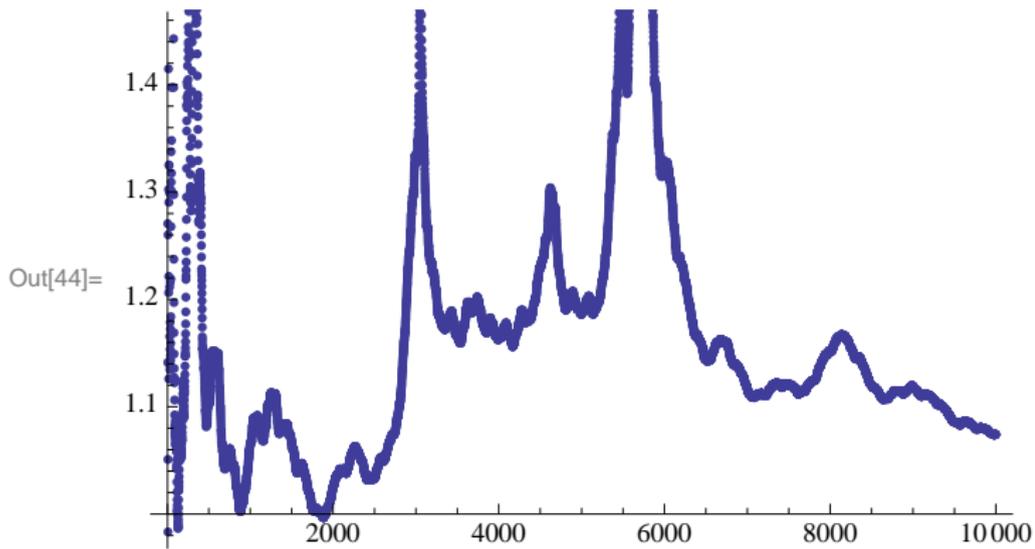
Theorem (P.): If  $X_1, X_2, \dots$  is a random Sturmian sequence of density  $p$ , where  $p$  is chosen uniformly at random in  $[0, 1]$ , then the standard deviation of the difference between the weighted sum  $S_n^*$  and the density  $p$  is  $O(1/n^{3/2})$ .

## Does kernel smoothing help rotor-routing?

The figure on the next slide shows what we get when we do  $10^4$  rounds of Casablanca simulation of the \$3 gambler's ruin process with  $p = 1/\pi$  with randomized **spacetime** rotors.

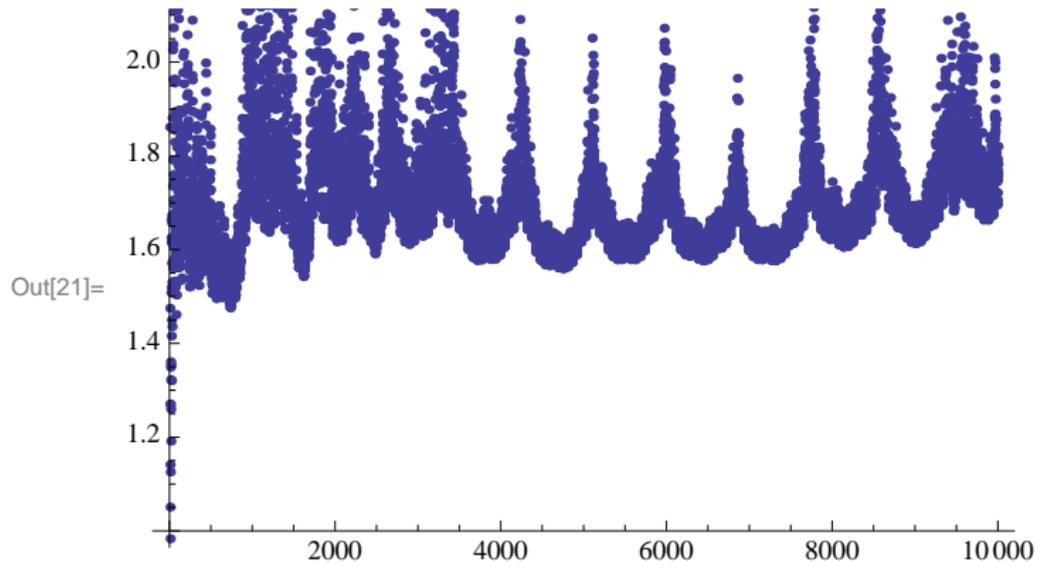
The horizontal axis records  $n$ ,  
and the vertical axis records the exponent  $\alpha$  such that the discrepancy between  $S_n^*$  and  $p^*$  equals  $1/n^\alpha$ .

The line  $\alpha = 1$  represents the  $O(1/n)$  barrier; we want to go above it.



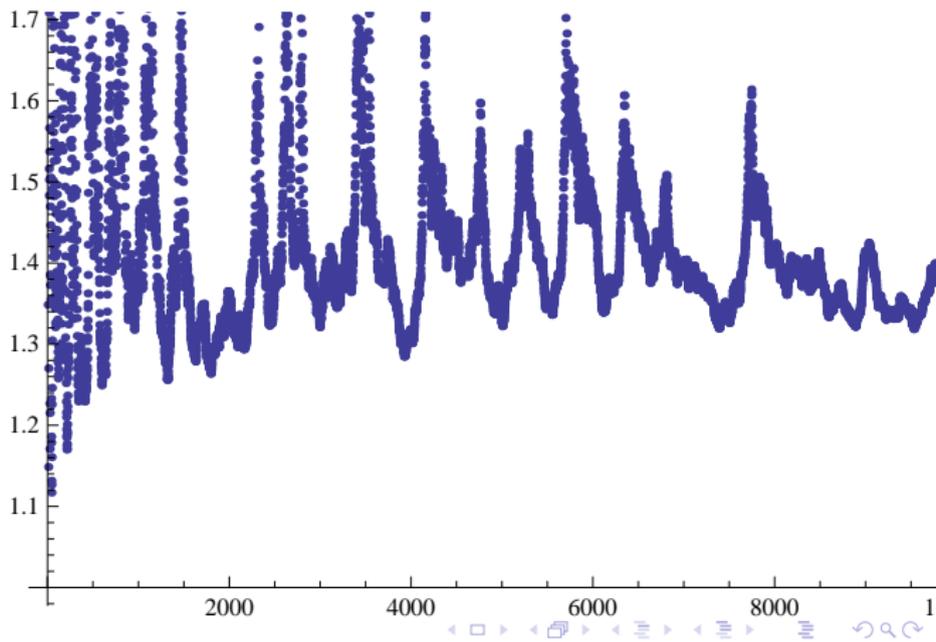
## Does kernel smoothing help rotor-routing?

Empirically, here's what we get when we do  $10^4$  rounds of Casablanca simulation of the \$3 gambler's ruin process with  $p = 1/\pi$  with randomized **space** rotors:



## Does kernel smoothing help rotor-routing?

Empirically, here's what we get when we do  $10^4$  rounds of Casablanca simulation of unbiased random walk on  $\mathbf{Z} \times \mathbf{Z}$  with source  $s = (0, 0)$  and targets  $t_1 = (0, 0)$  and  $t_2 = t^* = (1, 1)$  and  $\rho^* = \pi/8$ , with **non-randomized** space rotors initialized in the “fylfot” configuration (see Holroyd and Propp, “Rotor Walks and Markov Chains”, [arXiv:0904.4507](https://arxiv.org/abs/0904.4507) for an explanation):



## Non-randomized rotors?

Note that in the preceding case we are no longer doing probability theory; everything is deterministic.

However, it seems likely that probabilistic intuitions and even probabilistic theorems can be applied to the analysis of the asymptotic ( $n \rightarrow \infty$ ) behavior of these systems, and be used to rigorously prove that kernel smoothing applied to rotor-routing breaks the  $O(1/n)$  barrier.

(See Levine and Peres, “Strong Spherical Asymptotics for Rotor-Router Aggregation and the Divisible Sandpile”, [arXiv:0704.0688](https://arxiv.org/abs/0704.0688) for an example of how to use probabilistic theorems to prove results about a deterministic rotor-router process.)

# V. Conclusion

Derandomization of Monte Carlo with rotor-routers (for finite Markov chains) improves discrepancy from  $O(1/\sqrt{n})$  to  $O(1/n)$ .

(I've discussed this in the context of absorption probabilities; for applications to absorption times, steady-state probabilities, etc., see Holroyd and Propp, "Rotor Walks and Markov Chains", [arXiv:0904.4507](https://arxiv.org/abs/0904.4507).)

The use of kernel smoothing appears to give discrepancy  $o(1/n)$  in many situations, but I don't have proofs of this outside the "instant absorption" case (Markov chains in which absorption always occurs on the first step) and a few other easy-to-analyze special cases. Nor do I understand what the relevant asymptotic for the kernel-smoothed discrepancy should be, even heuristically.

# The problem

How can we get actual proofs that kernel smoothing breaks the  $O(1/n)$  barrier for generic finite-state Markov chains?

Note that prior results (Holroyd and P.) bounding the discrepancy between the absorption probability  $p^*$  and the ordinary average  $S_n/n$  (i.e, the relative frequency of absorption at target  $t^*$ ) all required the confluence property.

However, the confluence property can't be used here, since we can't compute  $S_n^*$  without knowing the order in which the respective absorptions at, and not at,  $t^*$  occur.

# Challenges

Find a method that will let us prove things about the estimate for  $p^*$  obtained by doing a rotor-router derandomized simulation of the absorption process and applying kernel smoothing to the resulting sequence of 0's and 1's.

A brand new idea is needed!

(Maybe some sort of nonabelian version of the confluence property?)

Also: Might there be some (nonlinear) method of estimating  $p^*$  that's better than (linear) kernel-smoothing?

Geometry of numbers says you can't hope (generically) to do better than  $O(1/n^2)$ , but David Einstein has almost proved (?) that for the seeing-the-slope-of-a-line problem you can get within a power-of-log factor of this bound.