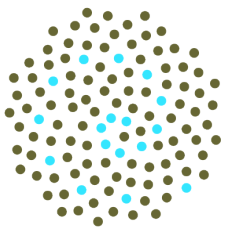
Subtleties in the Monte Carlo simulation of lattice polymers

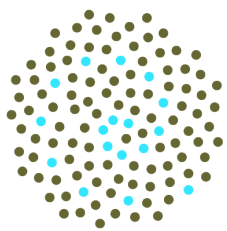
Nathan Clisby
MASCOS, University of Melbourne

July 9, 2010



Outline

- Critical exponents for SAWs.
- The pivot algorithm.
- Improving implementation of the pivot algorithm.
- Calculating ν (in the canonical ensemble).
- How can we efficiently calculate γ and μ?
- Lessons for other problems: confined polymers, knots.



Critical phenomena

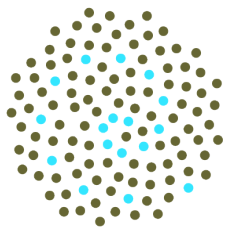
- The number of SAWs of length N, c_N, tells us about how many conformations are available to SAWs of a particular length:

$$c_N \sim A\, N^{\gamma-1}\, \mu^N\, [1 + \text{corrections}]$$

- Mean square end to end distance tells us about the size of a typical SAW:

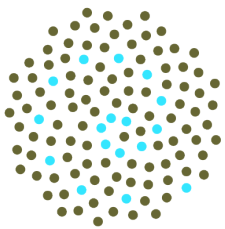
$$\langle R_e^2 \rangle_N \sim D_e N^{2\nu}\, [1 + \text{corrections}]$$

- We wish to determine γ, ν, and μ as accurately as possible for SAWs on $\mathbb{Z}^3$.



Markov chain Monte Carlo

- Sample from a probability distribution.
- Generate a new configuration from current one.
- Ensure that chain samples uniformly from whole set of configurations.
- Efficiency for calculating observable A determined by degree of correlation in the time series A_i. In particular, the integrated autocorrelation time τ of the chain.



Markov chain Monte Carlo

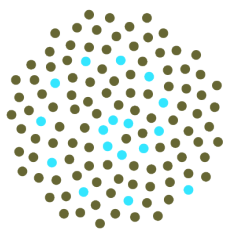
- $N \equiv$ *system size*.
- Run Markov chain for T time steps,

$$\text{Error}(A_N) = \sqrt{\frac{\text{var}(A_N)\tau_N}{T}}.$$

- Neglects CPU time taken to execute a step; real error is limited by computer time $t = T\ \text{CPU}_N$:

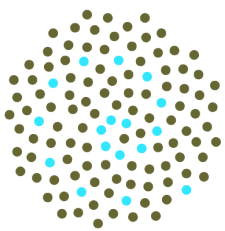
$$\text{Error}(A_N) = \sqrt{\frac{\text{var}(A_N)\tau_N \text{CPU}_N}{t}}.$$

- To decrease error
  Plan A: solve model exactly.
  Plan B:
  - Choose another observable to reduce $\text{var}(A_N)$.
  - Choose your moves wisely to reduce τ_N (pivots).
  - Design better data structures to reduce CPU_N.
  - Buy a GPU cluster to increase t.



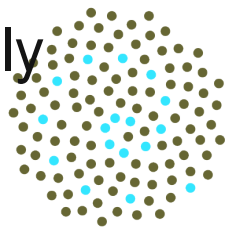
Pivot algorithm

- Invented in 1969 by Lal, reinvented in 1985 by MacDonald et al. and then Madras.
- The power of the method only realised since influential paper by Madras and Sokal in 1988 (over 500 citations).
- Monte Carlo method of choice for studying SAWs and similar models when the length of the walk is fixed.

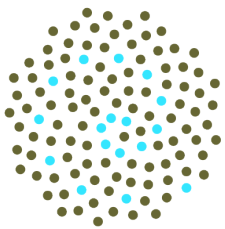


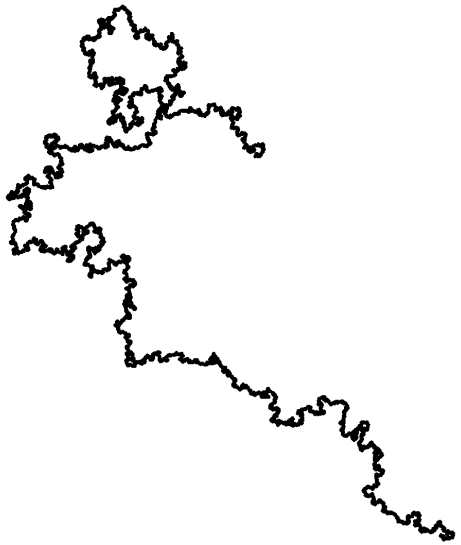
Pivot algorithm

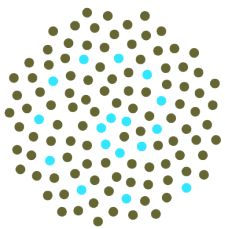
- Want to sample from the set of SAWs of a particular length.
- Set up a Markov chain as follows:
  - Randomly select a pivot site on the current SAW configuration.
  - Randomly choose a lattice symmetry q (rotation or reflection).
  - Apply this symmetry to one of the two subwalks created by splitting the walk at the pivot site.
  - If walk is self-avoiding: *accept* the pivot and update the configuration.
  - If walk is not self-avoiding: *reject* the pivot and keep the old configuration.
- The pivot algorithm is ergodic, and satisfies detailed balance which ensures that SAWs are sampled uniformly at random.

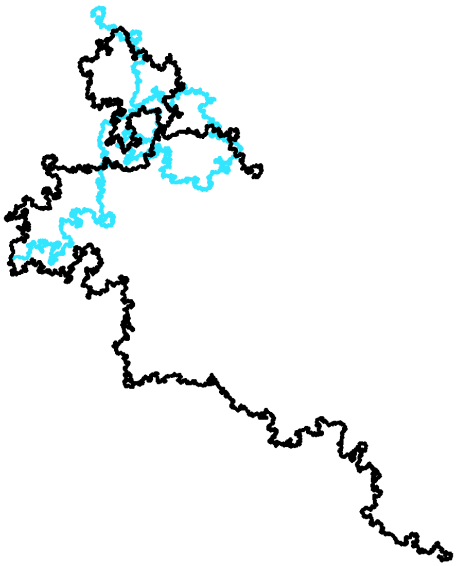


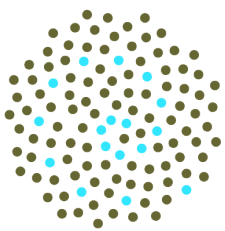
Will now show a sequence of *successful* pivots applied to an $n = 65536$ site SAW on the square lattice.

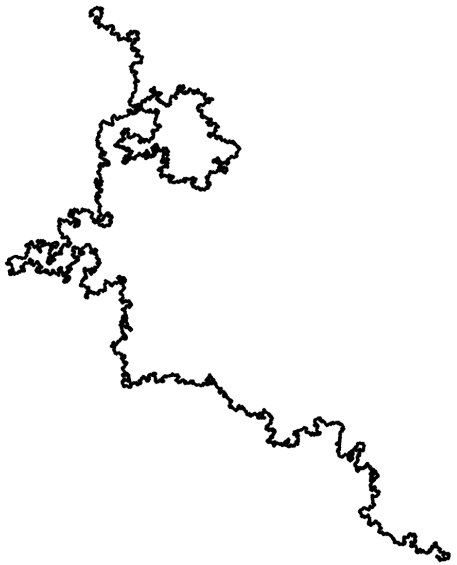


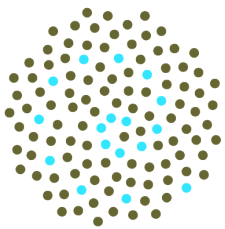


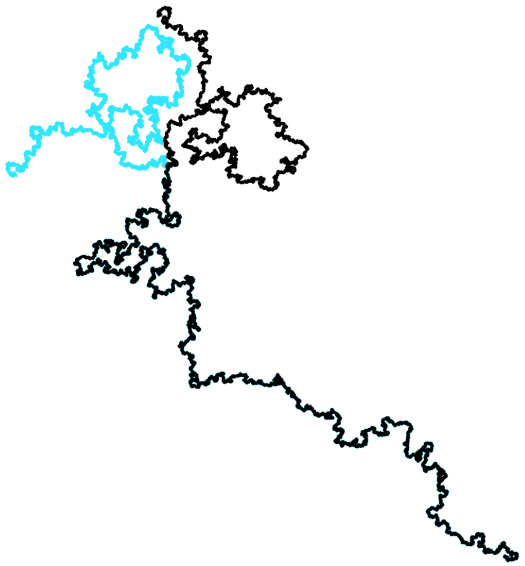


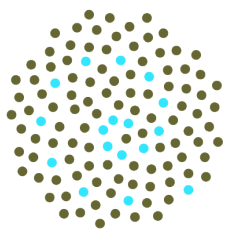


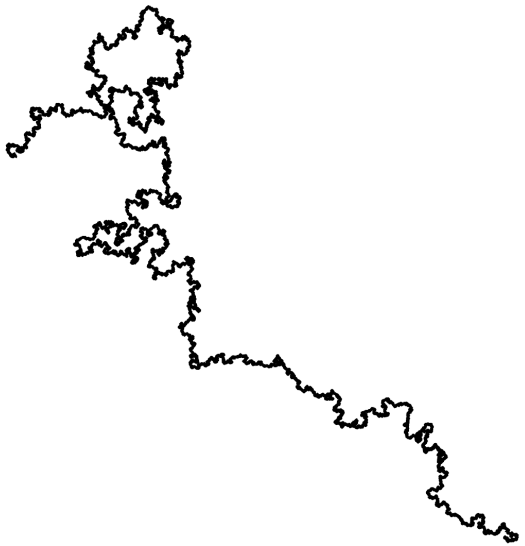


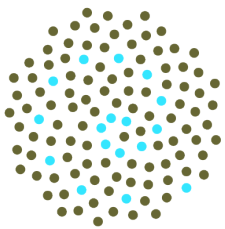


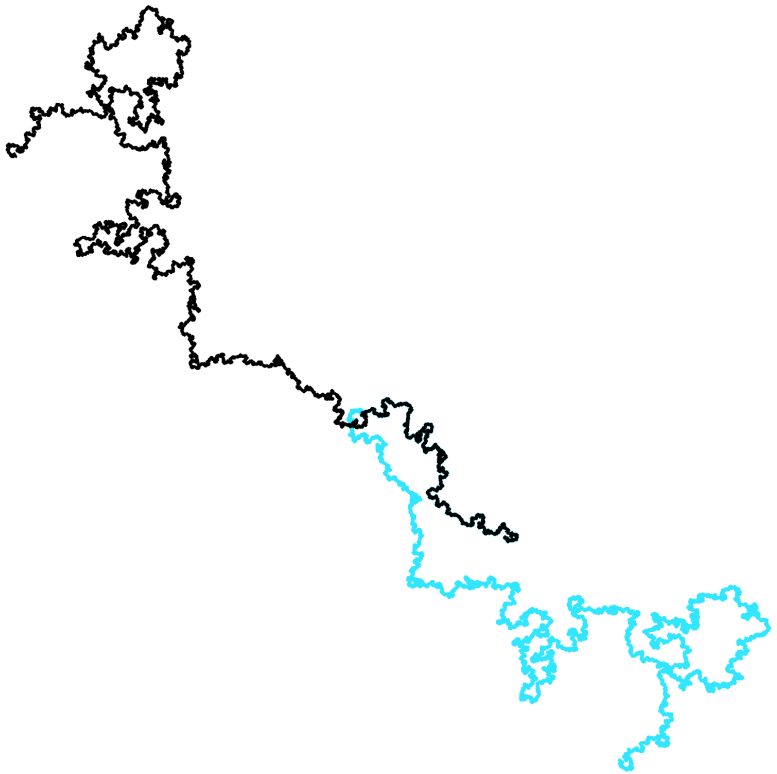


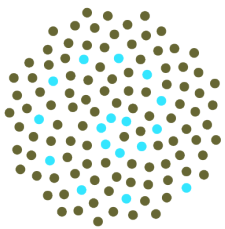


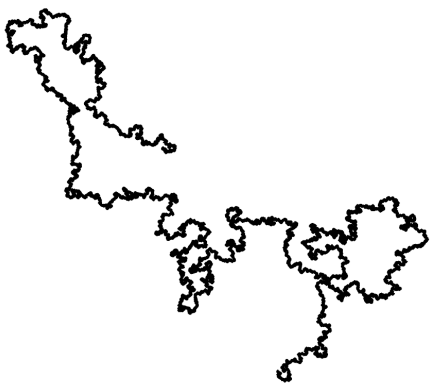


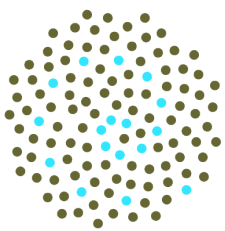


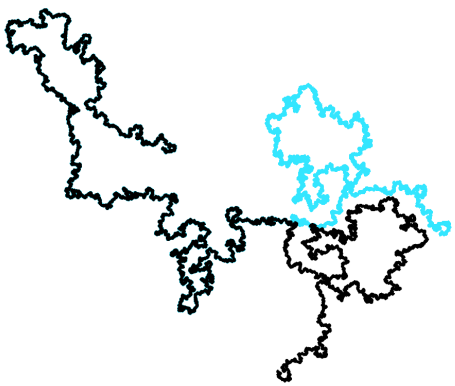


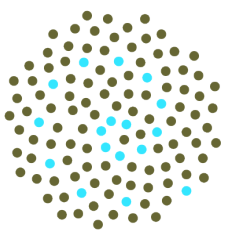


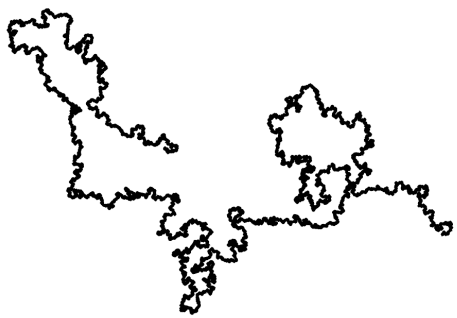


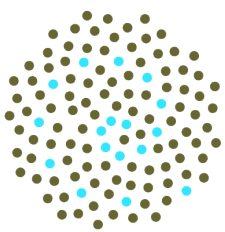


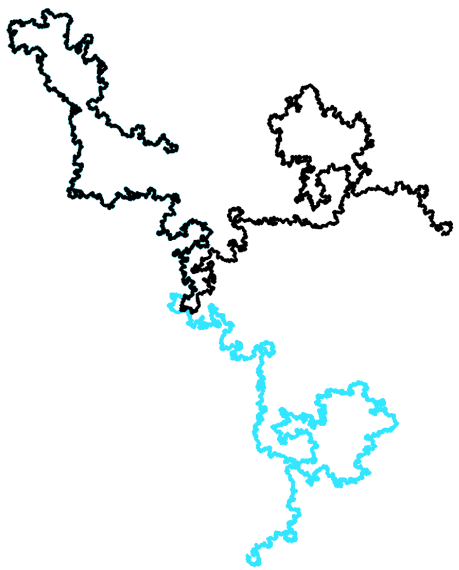


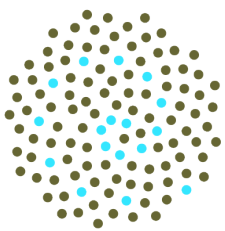


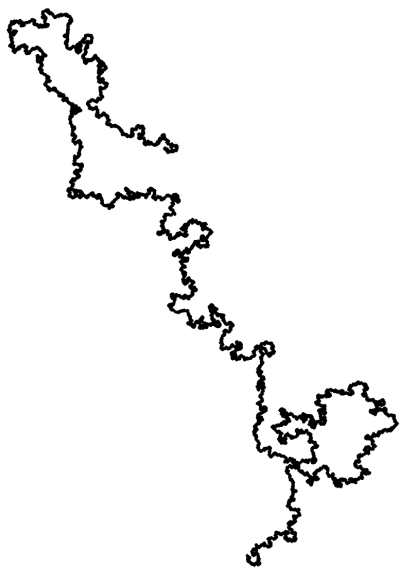


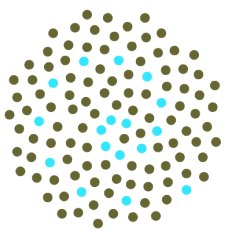


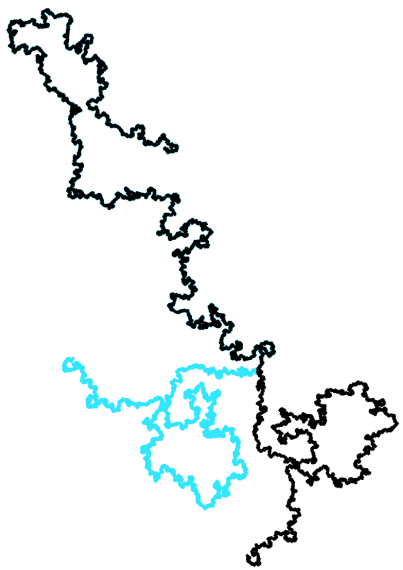


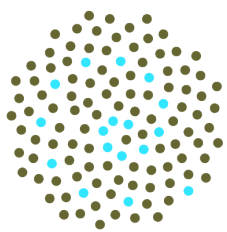


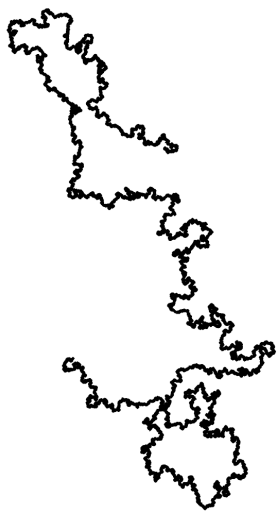


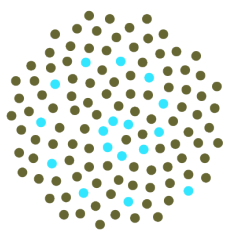


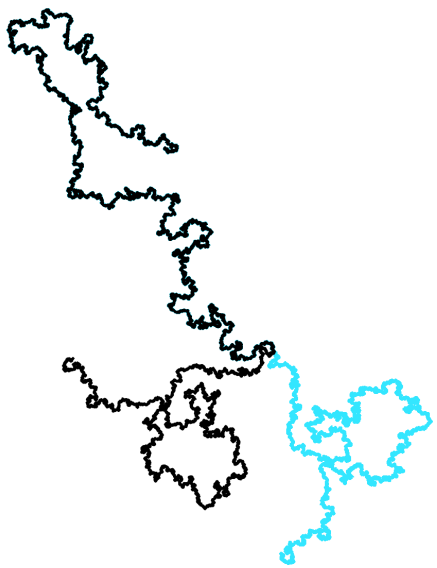


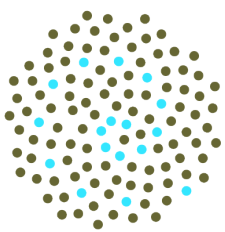


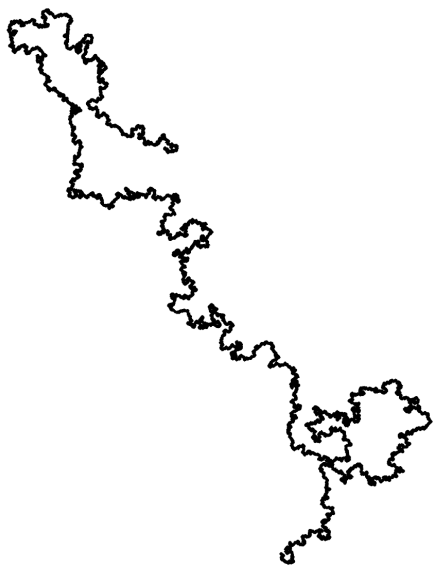


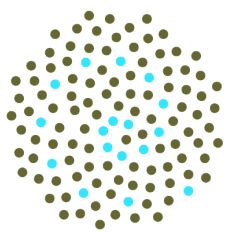


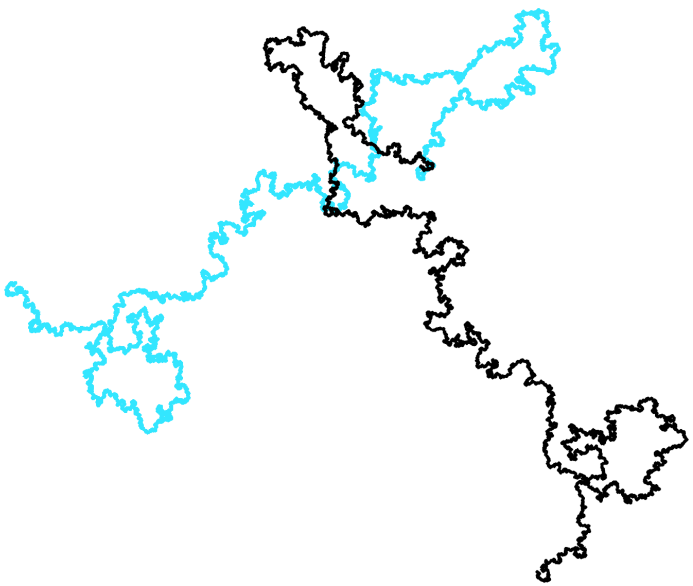


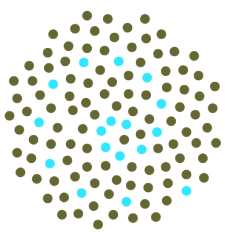


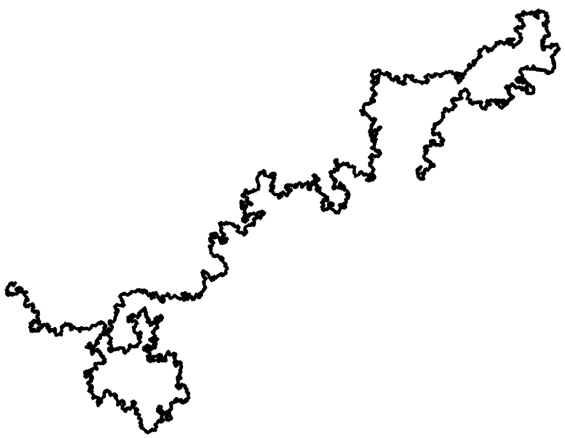


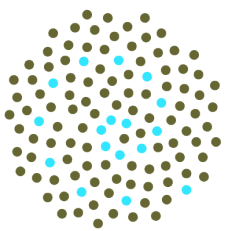


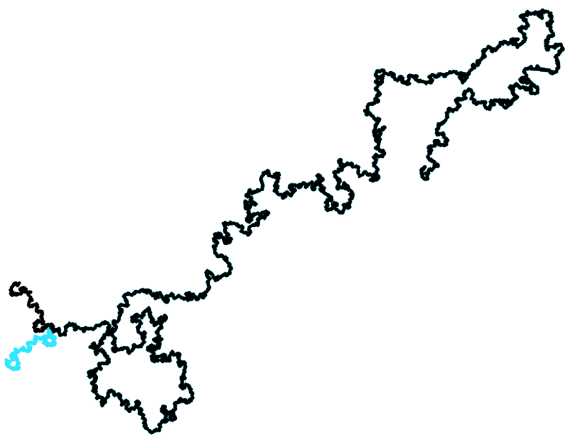


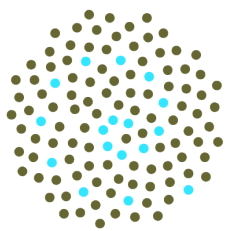


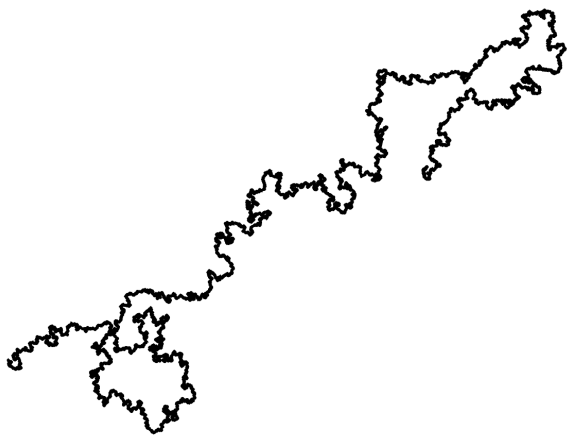


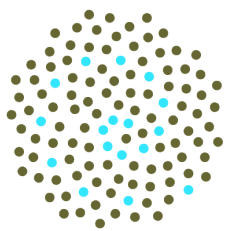


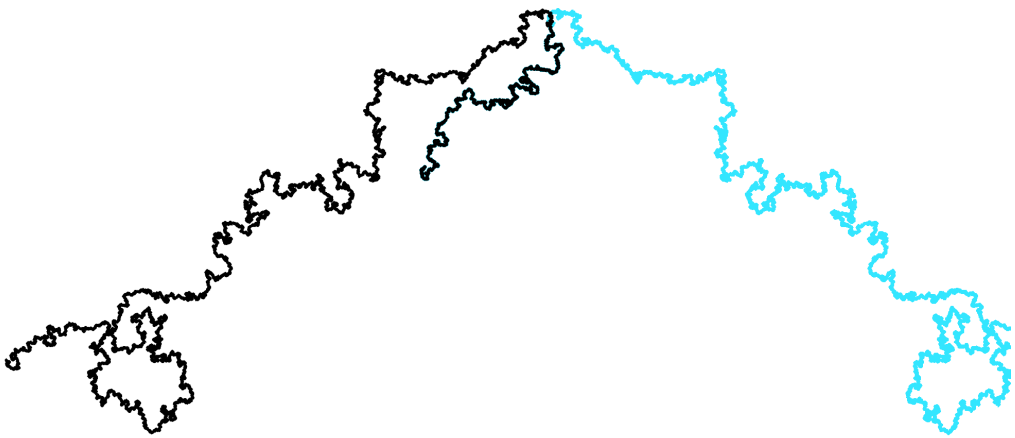


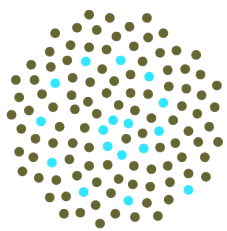


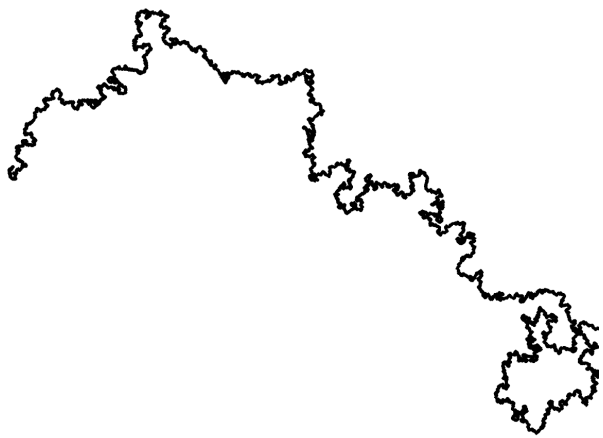


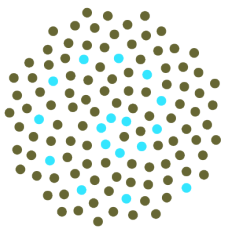


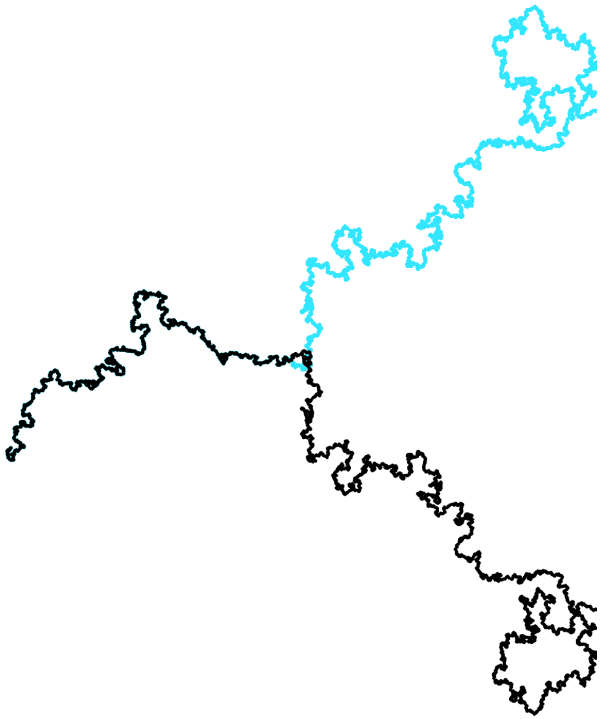


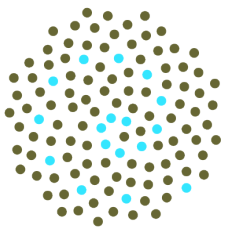


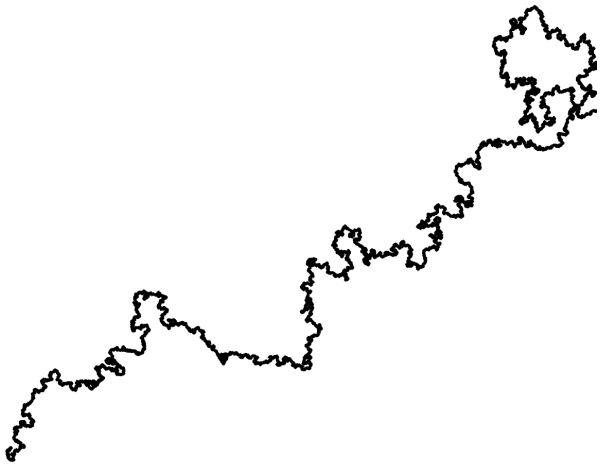


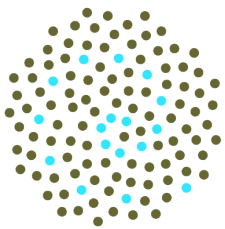


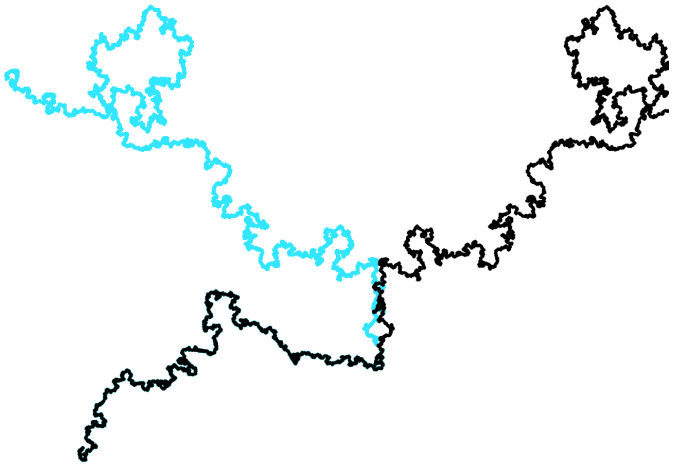


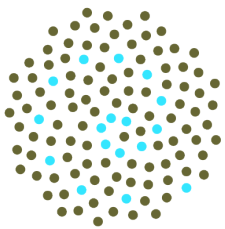


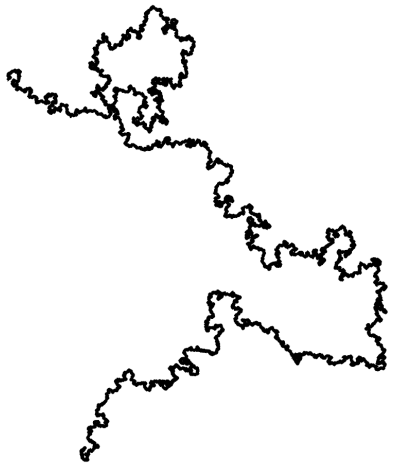


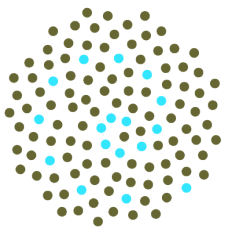


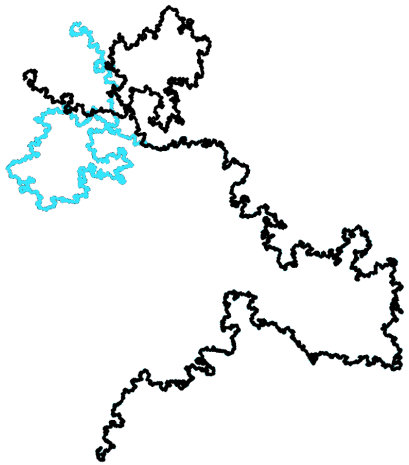


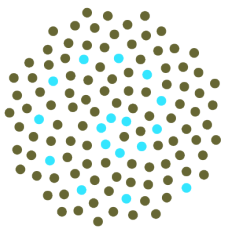


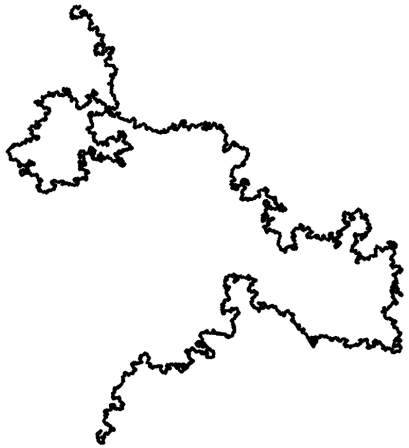


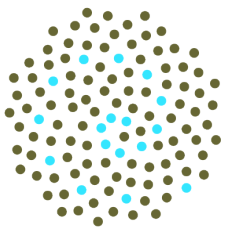


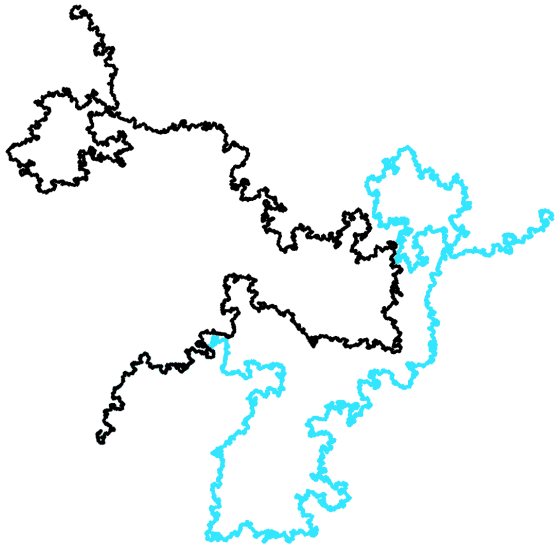


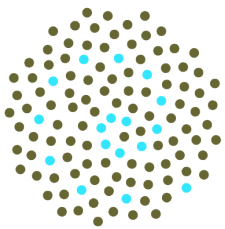


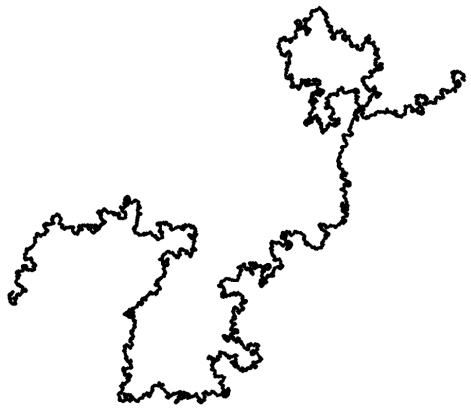


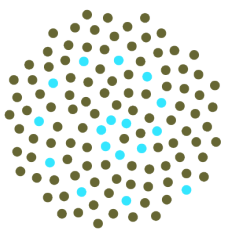


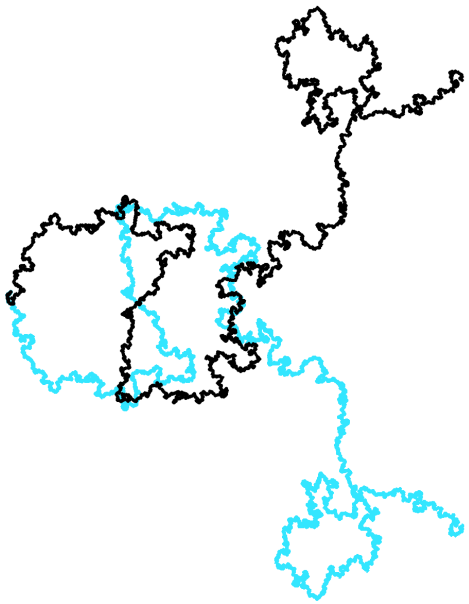


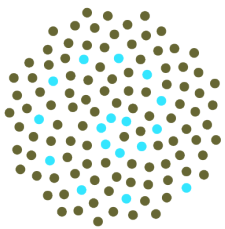


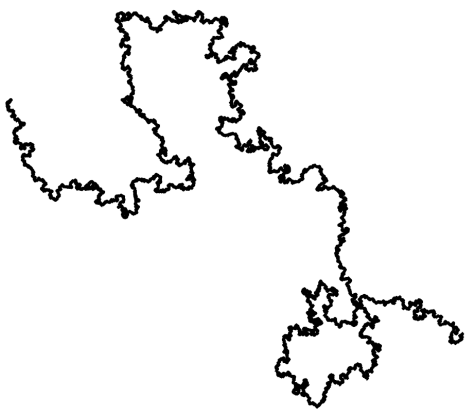


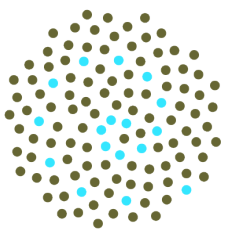


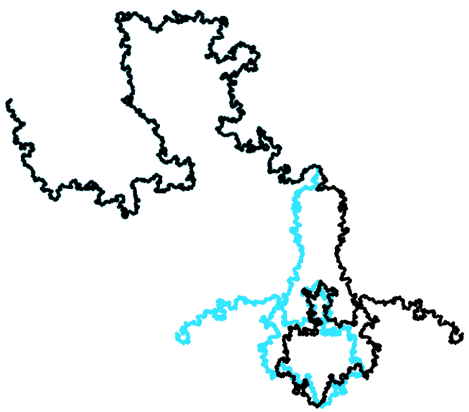


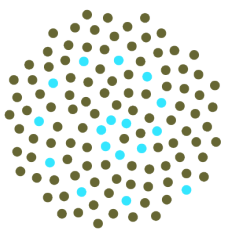


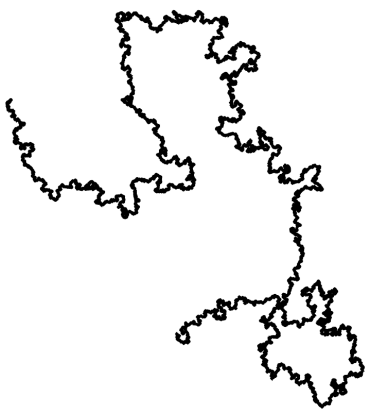


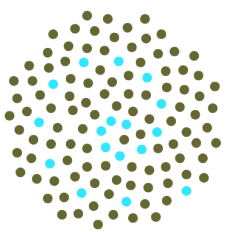


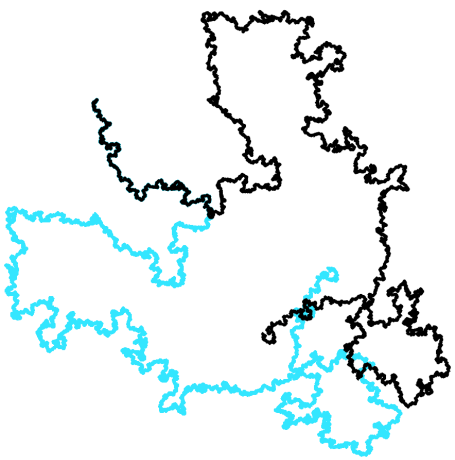


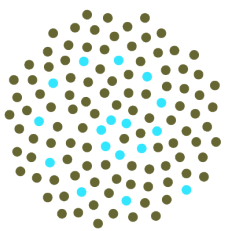


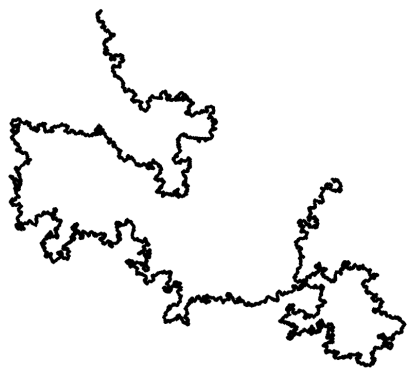


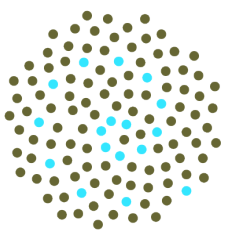


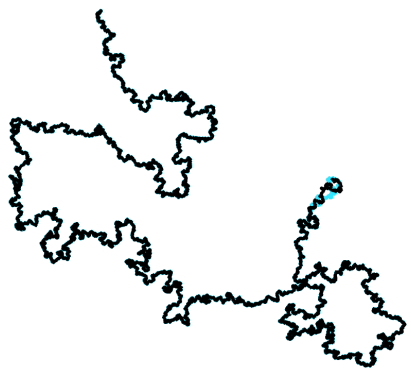


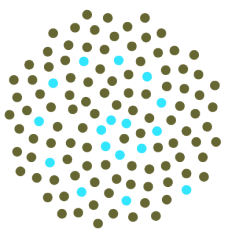


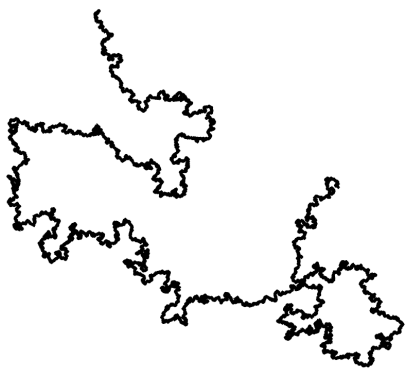


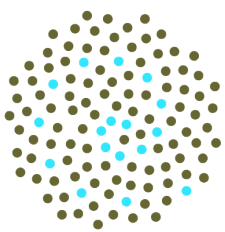


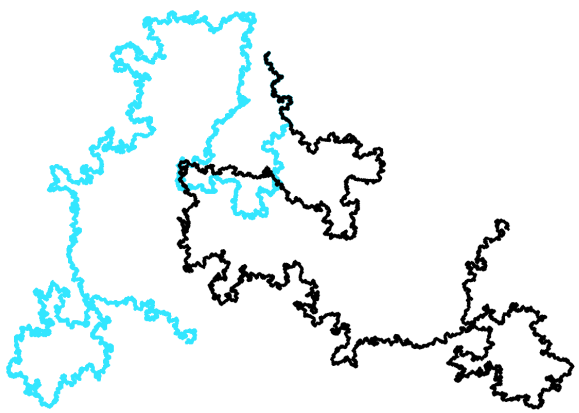


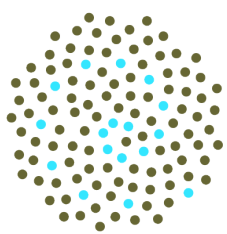


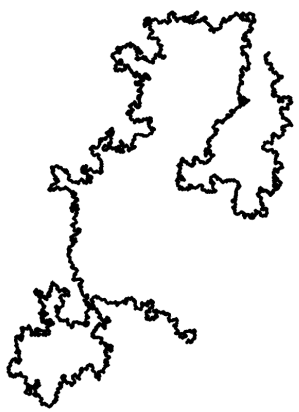


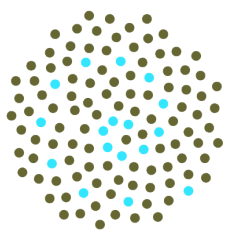






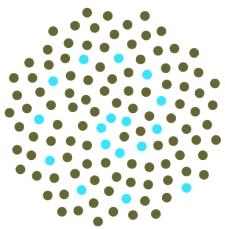






Why is it so effective?

- Pivots are rarely successful.
- Every time a pivot attempt *is* successful there is a large change in global observables.
- After each successful pivot, the successive values of global properties e.g. mean square end-to-end distance $\langle R_e^2 \rangle_N$, are (almost) uncorrelated.
- However *local* observables require $O(N)$ successful pivots.

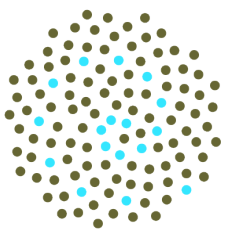


A better implementation

- Madras and Sokal (1988): implementation CPU time of approximately $O(N^{0.89})$ per attempted pivot for $\mathbb{Z}^3$.
- Kennedy (2002): implementation which is approximately $O(N^{0.74})$.

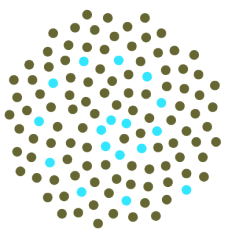
New "SAW-tree" implementation exploits geometric properties of walks.

- Represent the SAW as binary trees, with "bounding box" information for sub-walks.
- Other things to keep track of, so results in a complicated data structure.
- Most operations take $O(\log N)$, including intersection testing after a pivot attempt.



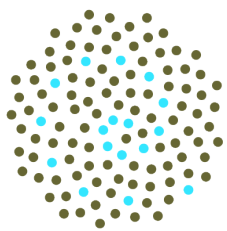
CPU time per attempted pivot, for SAWs of length N:

Lattice	Madras and Sokal	Kennedy	SAW-tree
Square	$O(N^{0.81})$	$O(N^{0.38})$	$o(\log N)$
Cubic	$O(N^{0.89})$	$O(N^{0.74})$	$O(\log N)$



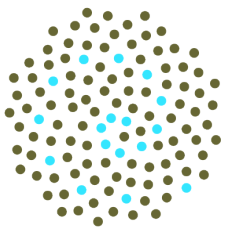
CPU time per attempted pivot, for SAWs of length N:

	$\mathbb{Z}^2$			$\mathbb{Z}^3$		
N	S-t (μs)	M&S/S-t	K/S-t	S-t (μs)	M&S/S-t	K/S-t
31	0.41	0.894	1.06	0.59	0.981	1.37
1023	0.87	5.15	1.90	1.71	6.31	3.75
32767	1.27	68.6	4.92	3.36	79.2	21.5
1048575	2.91	2510	32.2	7.53	3830	385
33554431	4.57	35200	134	12.58	61700	7130



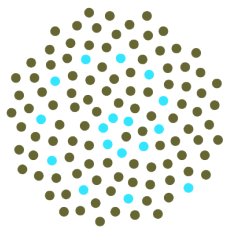
Calculating ν

- We sampled SAWs in the canonical ensemble, calculating metric properties at each pivot attempt.
- N up to 33×10^6, about 2×10^{13} pivot attempts, 16500 CPU hours.
- Estimated $\nu = 0.587597(7)$.
- Previous estimates are 0.5874(2) (MC, Prellberg, 2001) and 0.58756(5) (MCRG, Belohorec, 1997).



What to do with our new hammer?

- γ and μ are calculated naturally in the grand canonical ensemble of variable length walks.
- Our new hammer works in the canonical ensemble.
- Methods I will use are straightforward and obvious - suggested by Madras and Sokal (1988).
- Gives some insight which may be more widely useful.

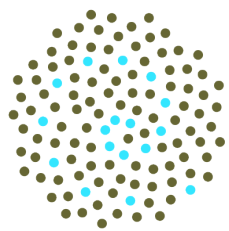


- Most straightforward method to calculate γ: dimerization, i.e. concatenating two SAWs to see if they form a longer SAW.
- Probability of successful concatenation is our observable, and

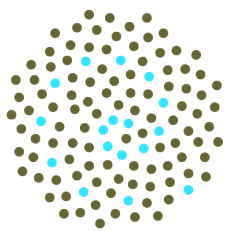
$$\begin{aligned} p_N &= \frac{c_{2N-1}}{c_{N-1}^2} \\ &\sim \frac{A\mu^{2N-1}(2N-1)^{\gamma-1}}{A^2\mu^{2N-2}(N-1)^{2\gamma-2}} \\ &\sim \frac{2^{\gamma-1}\mu}{A} N^{1-\gamma}\,[1 + \text{corrections}] \end{aligned}$$

- Generate two Markov chains of $N-1$ step walks using pivot algorithm.
- How can we minimize τ_N?

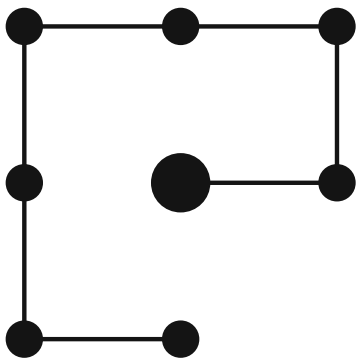
$$\text{Error}(p_N) = \sqrt{\frac{\text{var}(p_N)\tau_N \text{CPU}_N}{t}}.$$



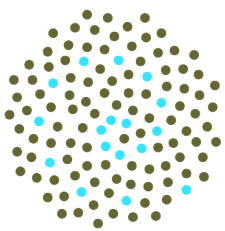
- How many pivots must be completed before two walks are essentially new configurations with respect to observable p_N?
- Shape of walks close to the joint clearly important.
- Simple argument suggests mean distance from joint where first intersection occurs is $O(N^{2-\gamma})$.
- Definition: the k-atmosphere of a walk is the number of ways of extending the walk by k steps so that it is still self avoiding.



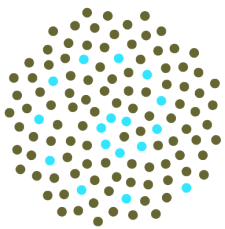
- Trapped SAWs (with 0 endpoint atmosphere) have small but positive density in set of SAWs for $\mathbb{Z}^2$ and $\mathbb{Z}^3$ (calculated for $\mathbb{Z}^2$ by Owczarek and Prellberg, 2008).
- Shortest trapped end-pattern: these configurations have positive density in set of SAWs.



- Trapped SAWs cannot be concatenated no matter what the rest of the walk is like.



- Choose $\Pr(i) \propto \frac{1}{i}$.
- Sites chosen at all length scales with equal probability. Probability that $i \in [L, 2L]$ is $O(1/\log N)$.
- Plausible upper bound for τ_N is the time necessary for successful pivots to be achieved at all possible length scales, i.e. $O(\log N)$ successful pivots.



Autocorrelation function

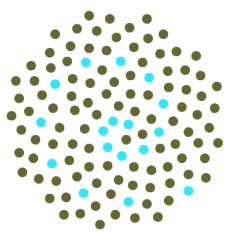
- Variance of an observable, $\mathrm{var}(A) = \langle A^2 \rangle - \langle A \rangle^2$.
- The autocorrelation function for the time series measurement of an observable A is

$$\rho_{AA}(t) = \frac{\langle A_s A_{s+t} \rangle - \langle A \rangle^2}{\mathrm{var}(A)}.$$

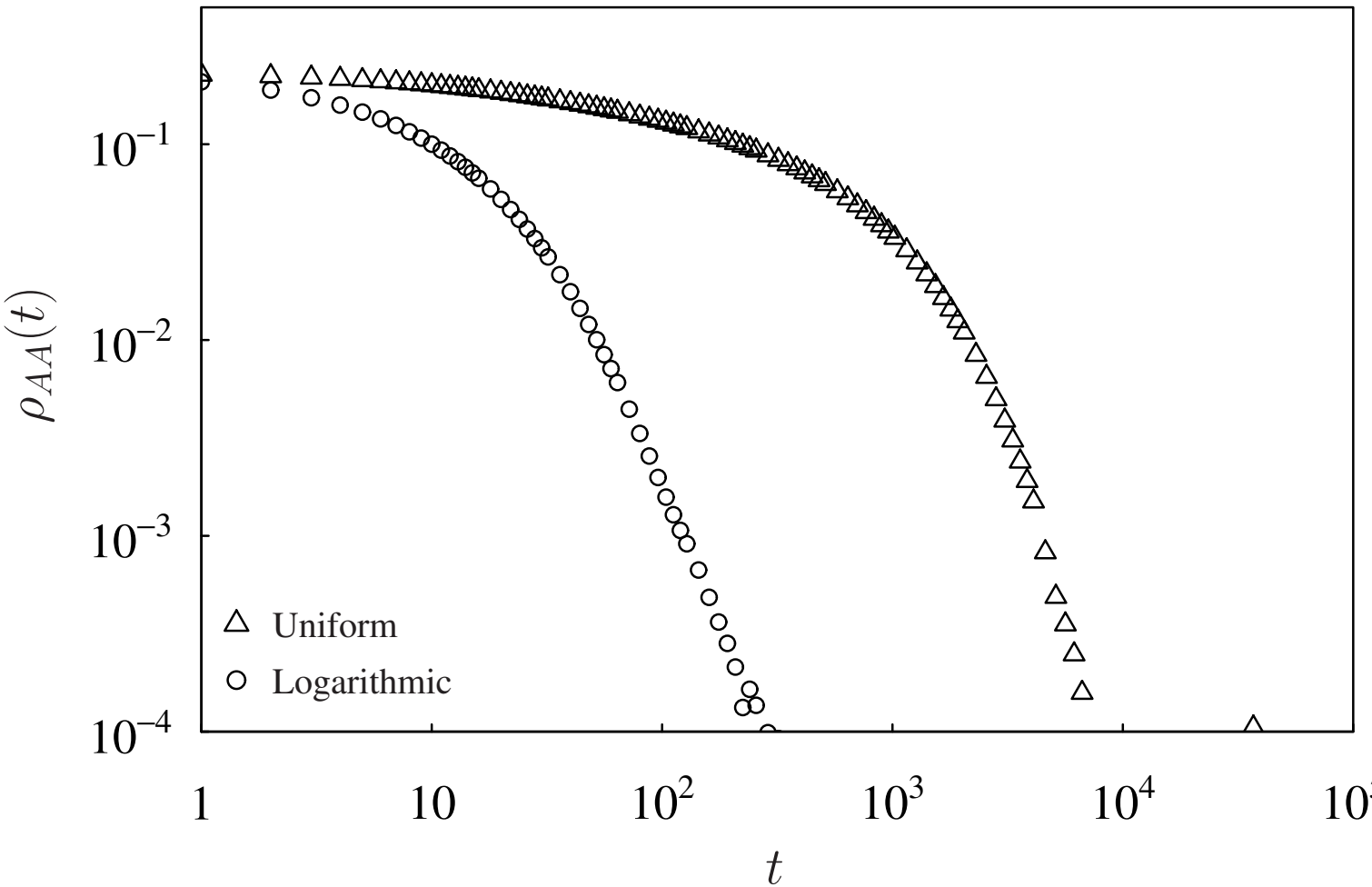
- Integrated autocorrelation time then:

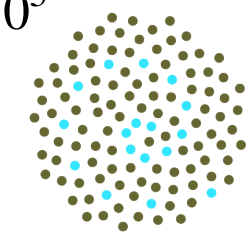
$$\tau_{\mathrm{int}}(A) = \frac{1}{2} + \sum_{t=1}^{\infty} \rho_{AA}(t)$$

- What is τ_{int} for p_N?

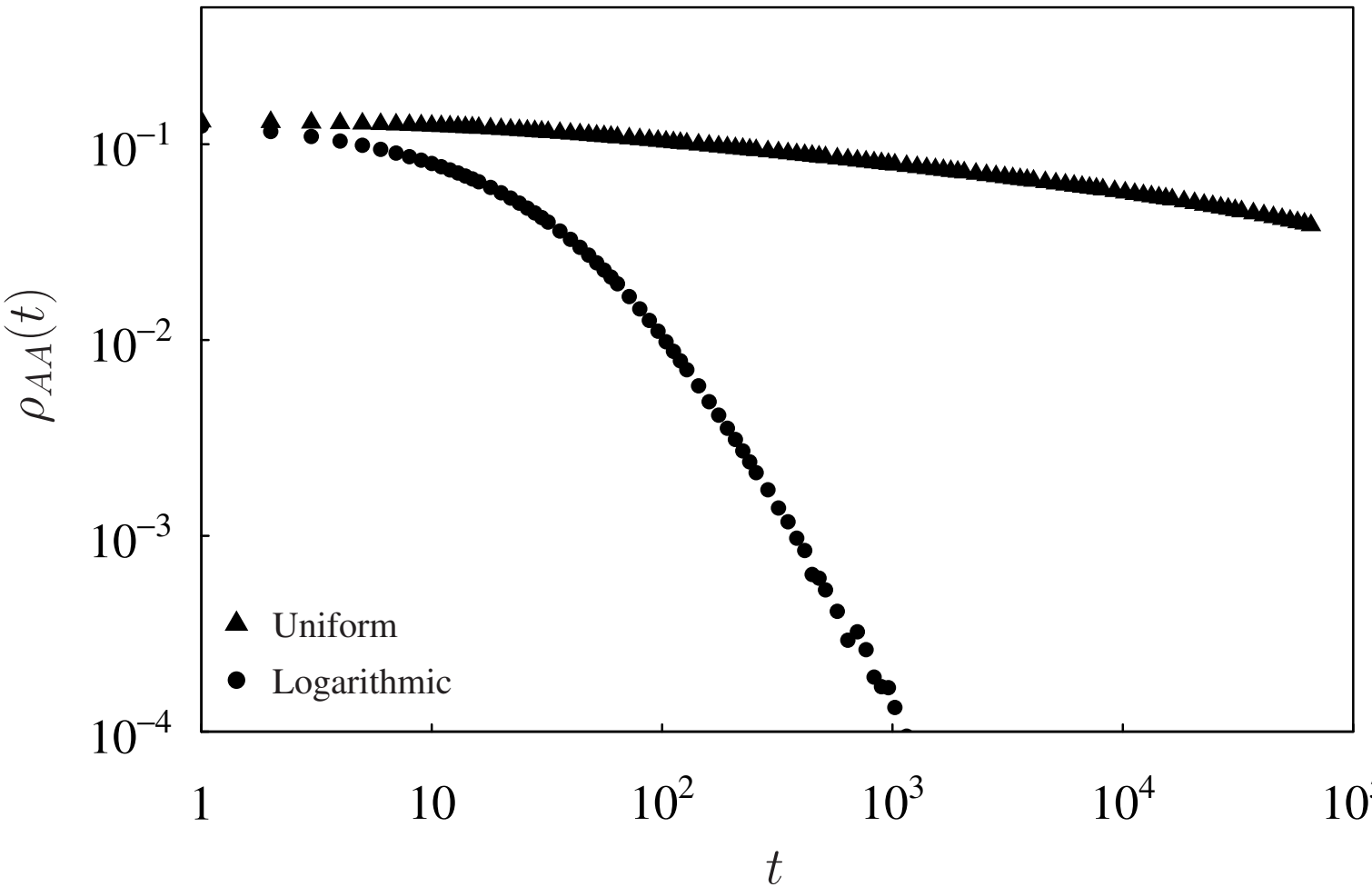


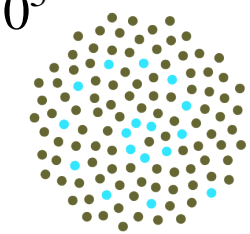
Autocorrelation function for p_N on $\mathbb{Z}^3$, $N = 1023$



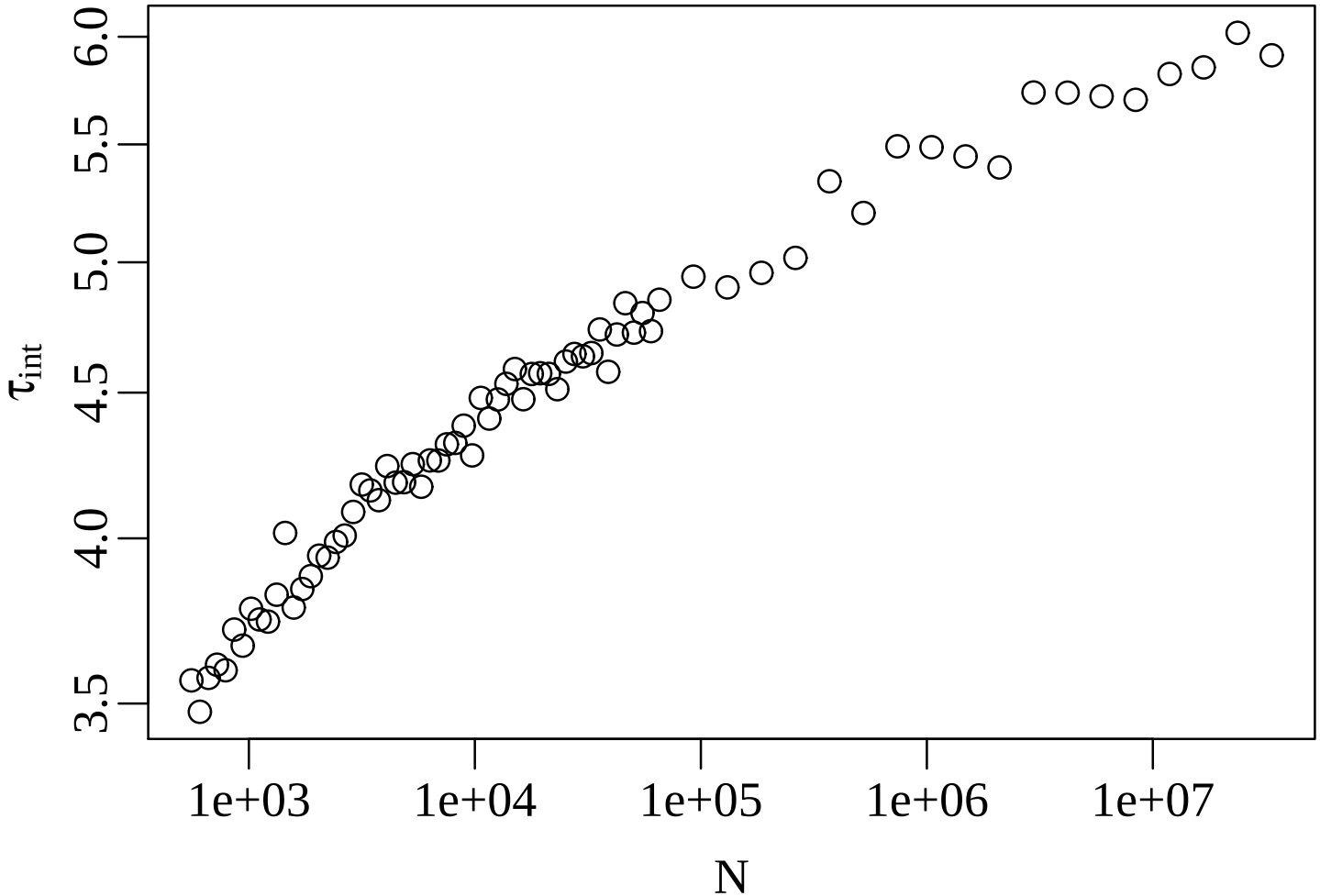


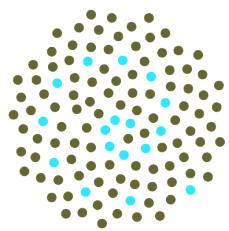
Autocorrelation function for p_N on $\mathbb{Z}^3$, $N = 1048575$





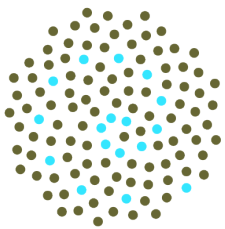
Integrated autocorrelation time





Comparison of parameter estimates.

- From $\langle p_N \rangle$, accurate calculation of the critical exponent γ for $d = 3$, $\gamma = 1.156957(9)$.
- Compare with 1.1573(2) (MC, Hsu & Grassberger, 2004), 1.1575(6) (MC, Caracciolo et al. 1998), and 1.1569(6) (enumeration, Clisby et al. 2007).

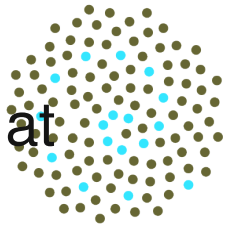


Calculation of μ

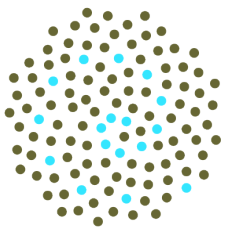
- Calculate by sampling SAWs uniformly at random via pivot algorithm; attempt to extend one of the ends using walks of k steps with no immediate returns (slightly different in our case).
- Observable is the probability of the resulting walk being self-avoiding:

$$\begin{aligned} O_N &= \frac{c_{N+k}}{(2d-1)^k c_N} \\ &\sim \frac{A\mu^{N+k}(N+k)^{\gamma-1}}{(2d-1)^k A\mu^N N^{\gamma-1}} \\ &\sim \frac{\mu^k}{(2d-1)^k}\left[1+\frac{k(\gamma-1)}{N}+\text{corrections}\right] \end{aligned}$$

- Again, shape near joint is crucial. We choose uniformly at random on log-transformed distance from joint.

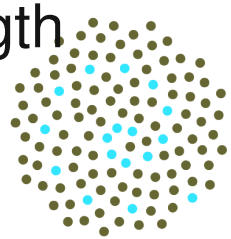


- Estimate $\mu = 4.6840398(9)$ in 6500 CPU hours (data run not yet complete).
- Compare with 4.684038(6) (MC, Hsu & Grassberger, 2004), 4.684043(12) (enumeration, Clisby et al., 2007).
- In 60×10^6 CPU years I could beat Iwan Jensen's estimate for $\mathbb{Z}^2$ from SAP series!
- (There should be a better way of calculating μ.)



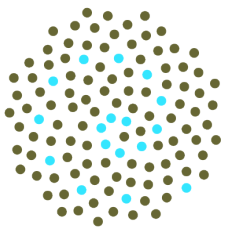
Confined polymers

- Length scales introduced by putting polymer in a confined region, e.g. between two parallel plates, or in a tube.
- Perform “moves” on sub-walks.
- If we select endpoints of sub-walks uniformly from log(distance), we guarantee that all length scales will be accounted for.
- Potential move set:
  - Two-point moves, endpoints fixed (Madras 1990).
  - Two-point moves, endpoints free.
  - Cut-and-paste moves, e.g. swap two sub-walks.
  - One-point pivots. (Free endpoints, one of the points is the end of walk).
- These moves may well have different characteristic length scales; automatic tuning by selecting site separation uniformly from log(distance).



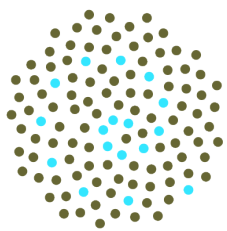
Knots

- Knots have been found to be localized on polygons.
- Pivot algorithm, with two-point moves (Madras et al., 1990, and Janse van Rensburg, Whittington & Madras, 1990), is very effective at sampling polygon configurations.
- Probability of success decreases with distance between points; could improve by selecting log(distance) uniformly at random. Will improve acceptance probability, and therefore decrease autocorrelation time for knot type.



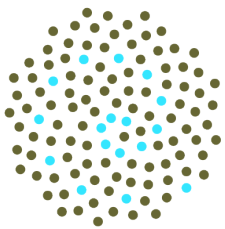
Other thoughts

- Star polymers / branched polymers (distance to branch point)
- Polymers tethered to a surface.
- Cluster moves for hard spheres? (select particles in a disc / sphere, relevant length is size of cluster)
- Sampling of lattice models? (equivalent to multi-grid Monte Carlo?)



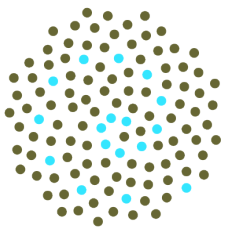
Summary

- Exploiting geometry of problem via SAW-tree data structure has resulted in much faster implementation of pivot algorithm.



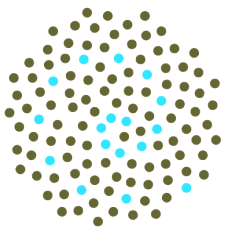
Summary

- Exploiting geometry of problem via SAW-tree data structure has resulted in much faster implementation of pivot algorithm.
- For problems where there are multiple length scales $\cdots$



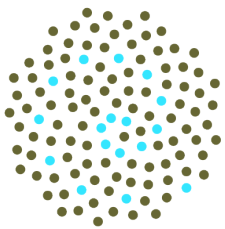
Summary

- Exploiting geometry of problem via SAW-tree data structure has resulted in much faster implementation of pivot algorithm.
- For problems where there are multiple length scales $\cdots$
- and Monte Carlo moves on all length scales are equally computationally cheap $\cdots$



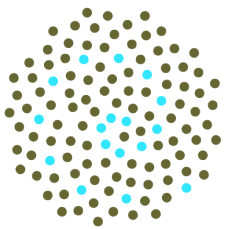
Summary

- Exploiting geometry of problem via SAW-tree data structure has resulted in much faster implementation of pivot algorithm.
- For problems where there are multiple length scales $\cdots$
- and Monte Carlo moves on all length scales are equally computationally cheap $\cdots$
- choose Monte Carlo moves so that all length scales of the system are probed.



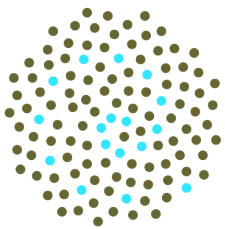
Summary

- Exploiting geometry of problem via SAW-tree data structure has resulted in much faster implementation of pivot algorithm.
- For problems where there are multiple length scales $\cdots$
- and Monte Carlo moves on all length scales are equally computationally cheap $\cdots$
- choose Monte Carlo moves so that all length scales of the system are probed.
- Generalisation: spend same amount of CPU time attempting moves on all length scales.



Summary

- Exploiting geometry of problem via SAW-tree data structure has resulted in much faster implementation of pivot algorithm.
- For problems where there are multiple length scales $\cdots$
- and Monte Carlo moves on all length scales are equally computationally cheap $\cdots$
- choose Monte Carlo moves so that all length scales of the system are probed.
- Generalisation: spend same amount of CPU time attempting moves on all length scales.
- Either way, at most modest log N penalty; if combined with uniform sampling performance is at most a factor of 2 worse, and gain can be much greater than 2.



- Nathan Clisby, *Accurate estimate of the critical exponent ν for self-avoiding walks via a fast implementation of the pivot algorithm*, Phys. Rev. Lett. **104** (2010), 055702.
- N. Clisby, *Efficient implementation of the pivot algorithm for self-avoiding walks*, J. Stat. Phys. **140** (2010), 349–392.
- N. Clisby, *Accurate estimate of the critical exponent γ for three-dimensional self-avoiding walks*, in preparation.
- N. Clisby, *Determination of the connective constant for self-avoiding walks via the pivot algorithm and the method of atmospheres*, in preparation.