Critical exponents for self-avoiding walks from a fast implementation of the pivot algorithm

Nathan Clisby (MASCOS, University of Melbourne)

December 2, 2008

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:

$$c_n \sim A\, n^{\gamma-1} \mu^n \left[1 + \text{corrections}\right]$$

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

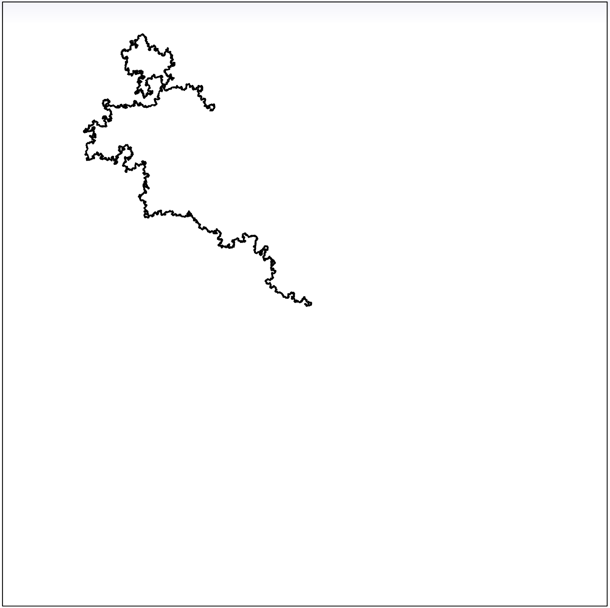
$$\langle R_e^2 \rangle = D n^{2\nu} \left[1 + \text{corrections}\right]$$

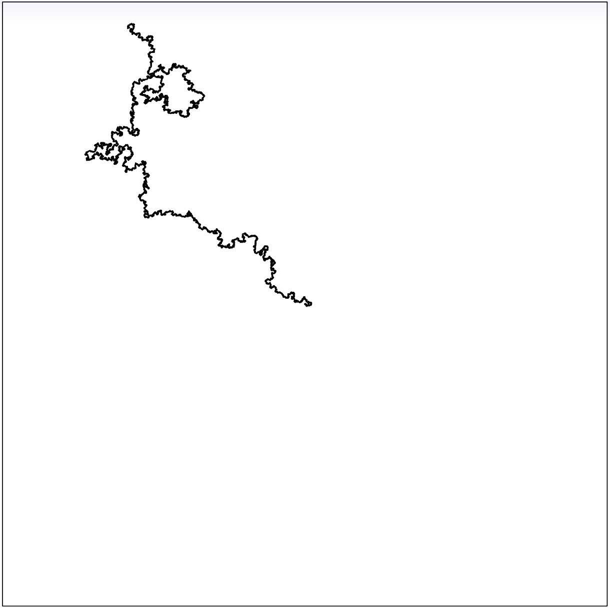
- We wish to determine γ and ν as accurately as possible.

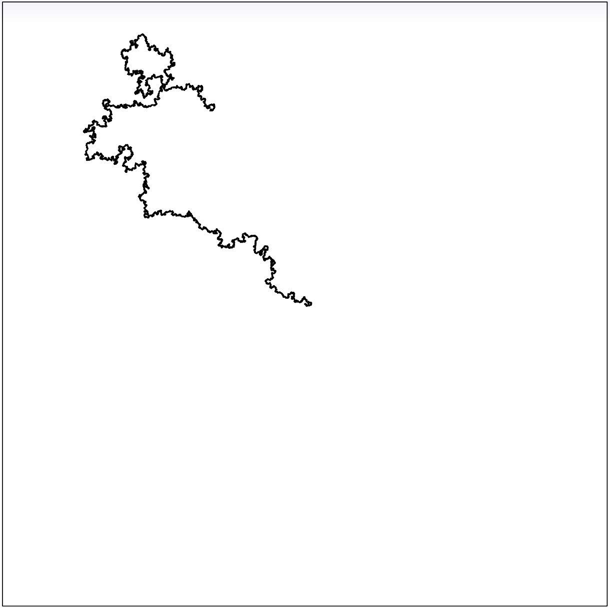
Pivot algorithm

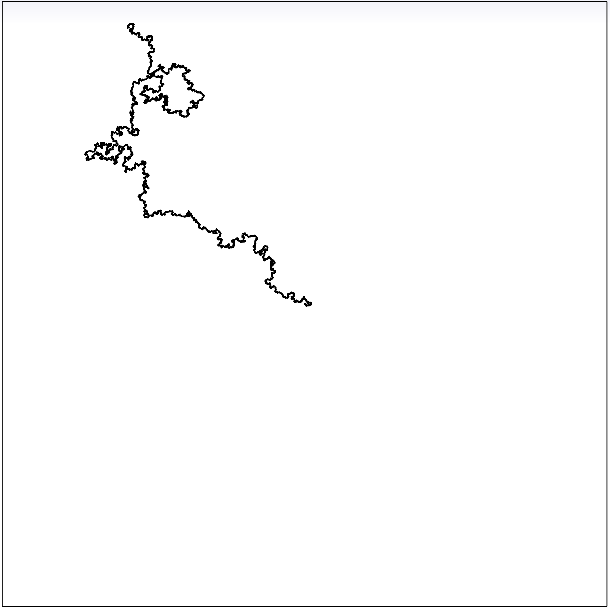
- Want to sample from the set of SAWs of a particular length.
- Set up a Markov chain as follows:
  - Randomly select a pivot site on the current SAW configuration.
  - Randomly choose a lattice symmetry q (rotation or reflection).
  - Apply this symmetry to one of the two subwalks created by splitting 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.

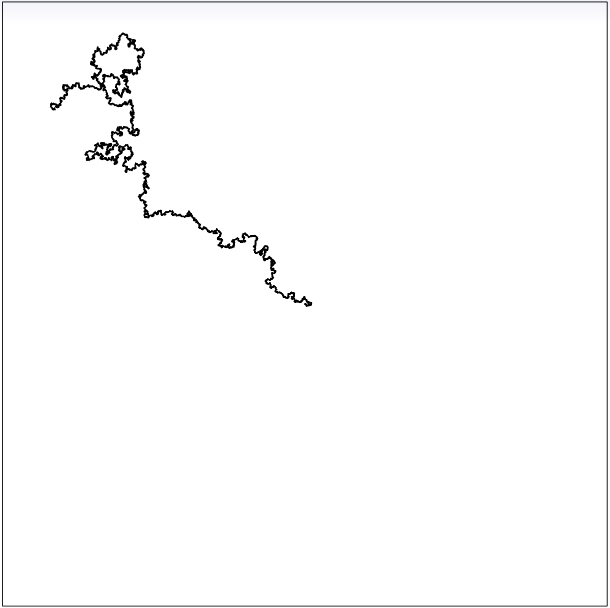
Will now show a sequence of *successful* pivots applied to an $n = 65536$ site SAW.

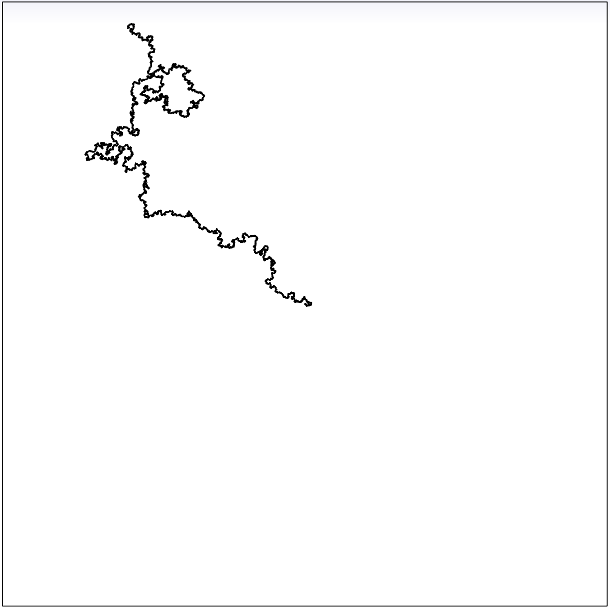


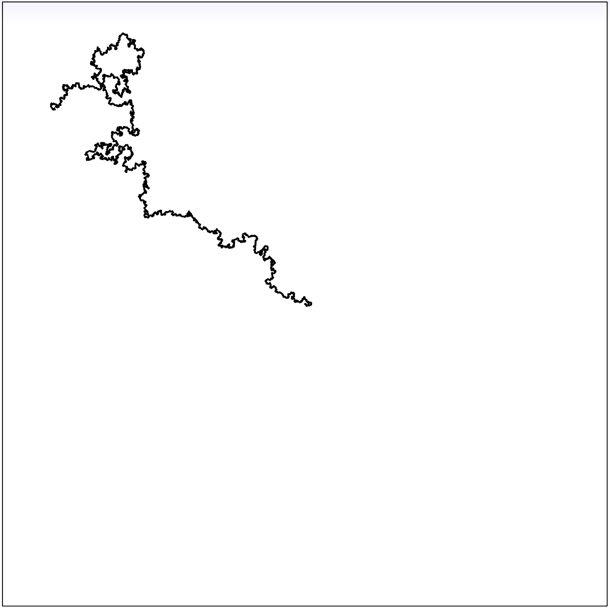


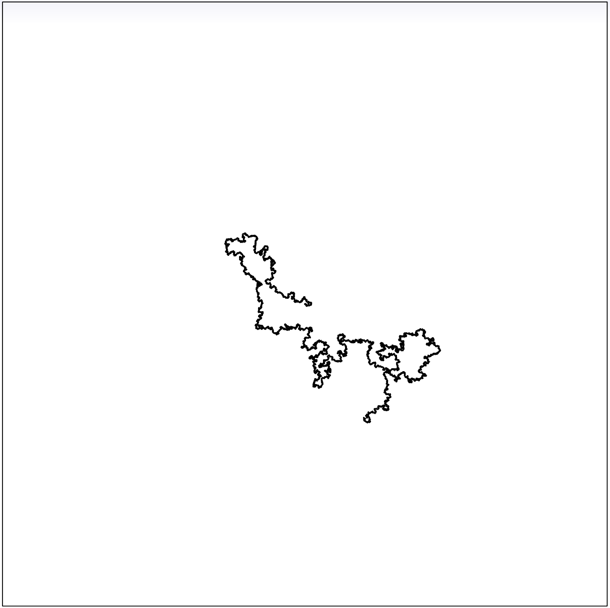


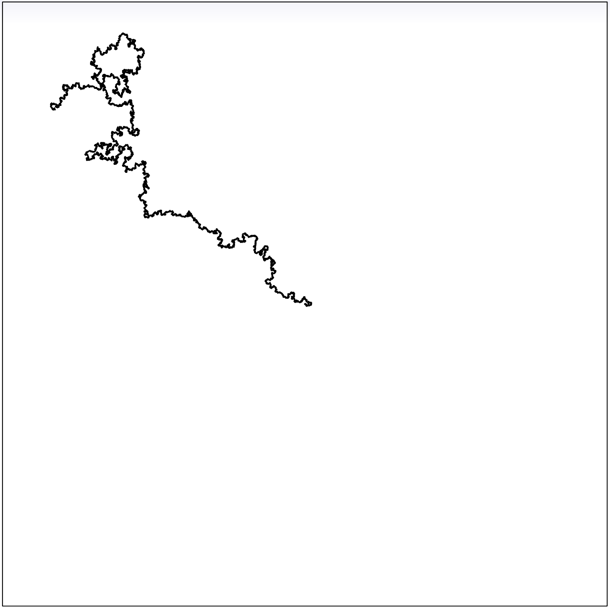


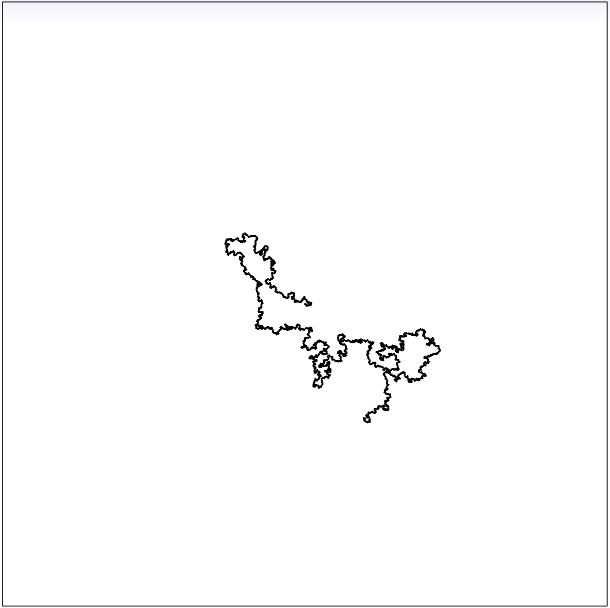


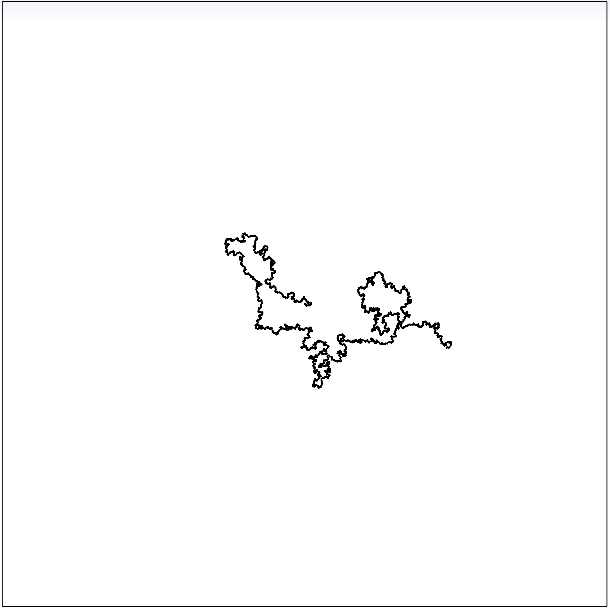


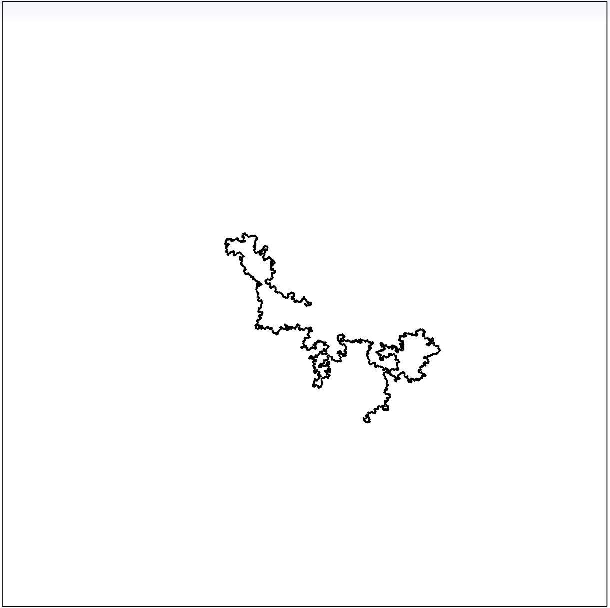


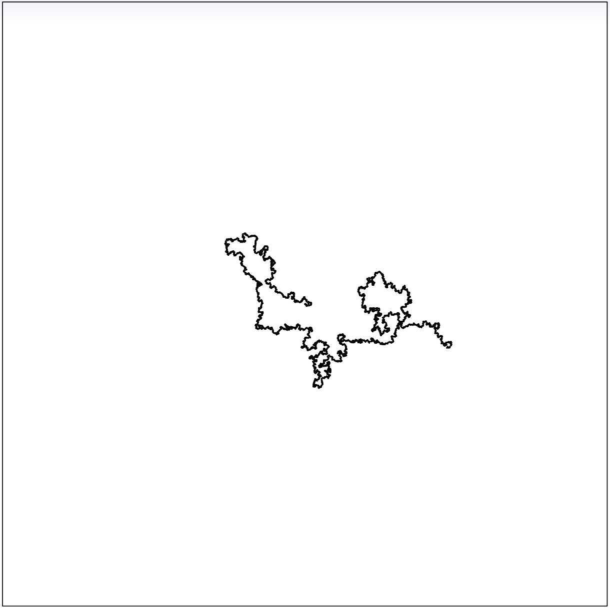


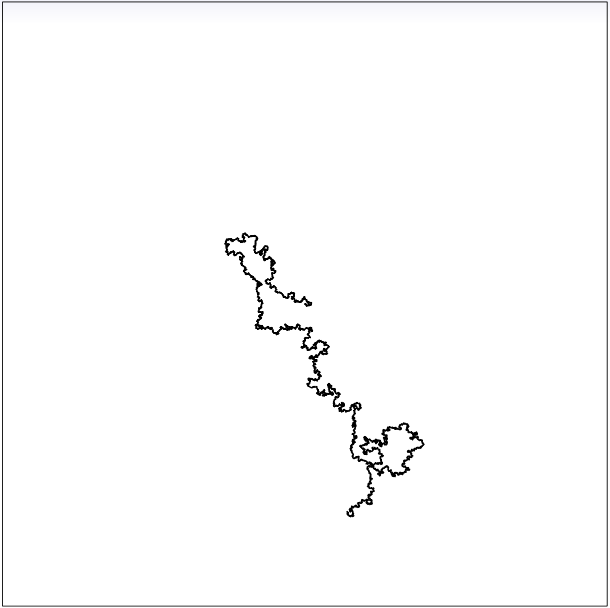


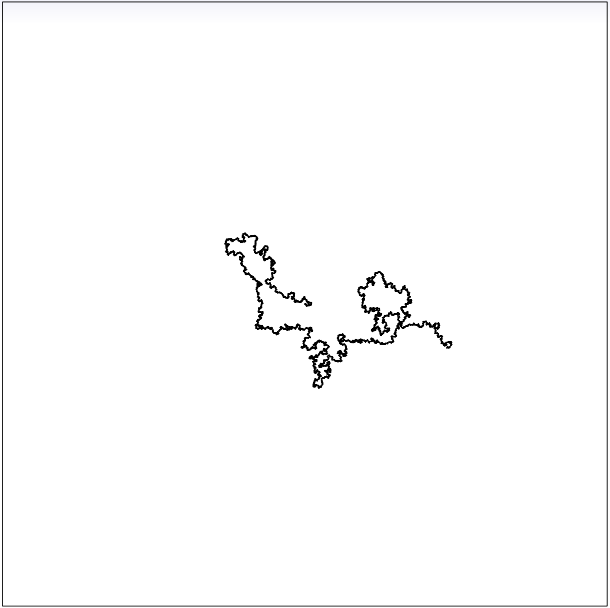


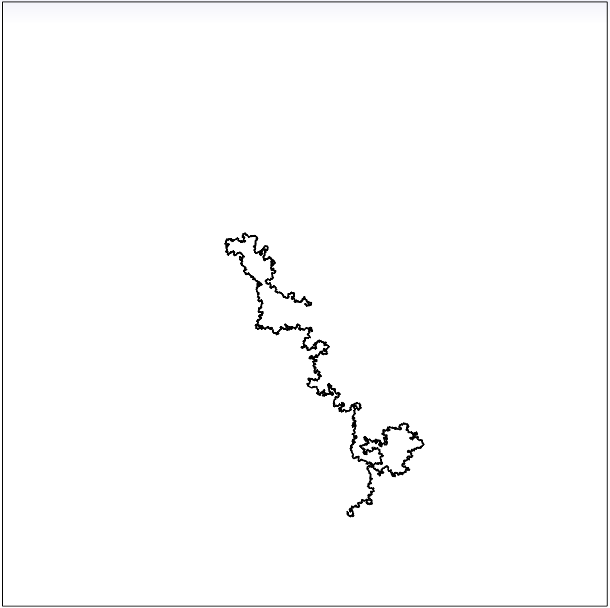


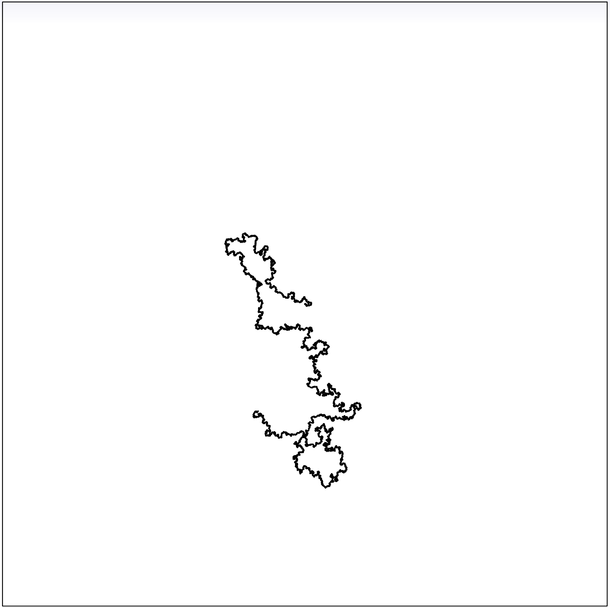


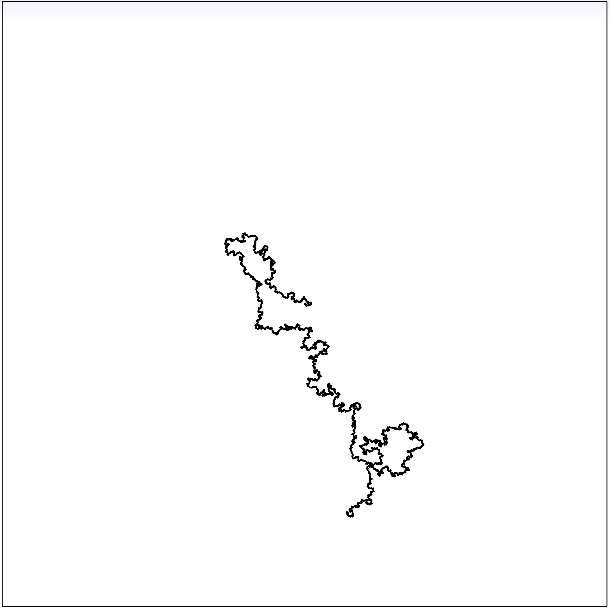


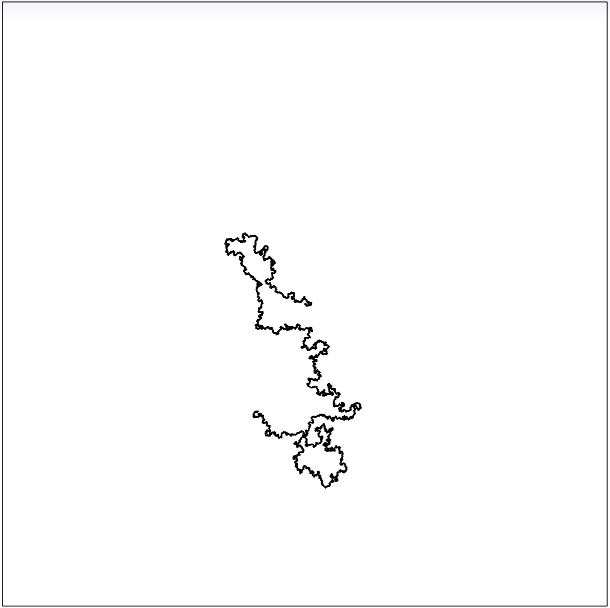


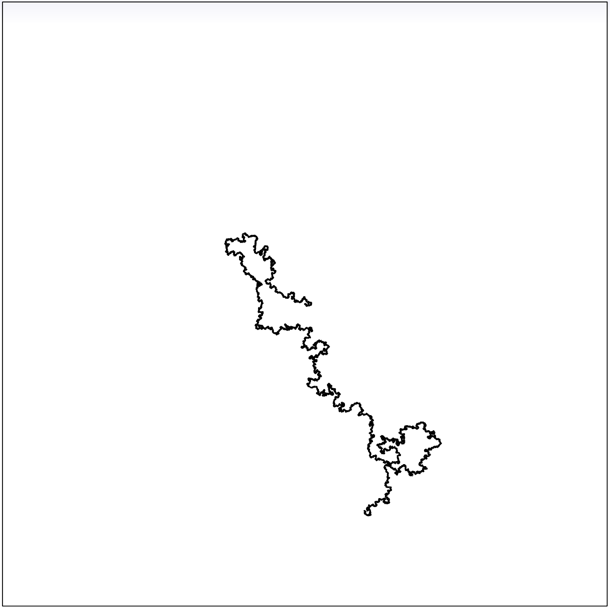


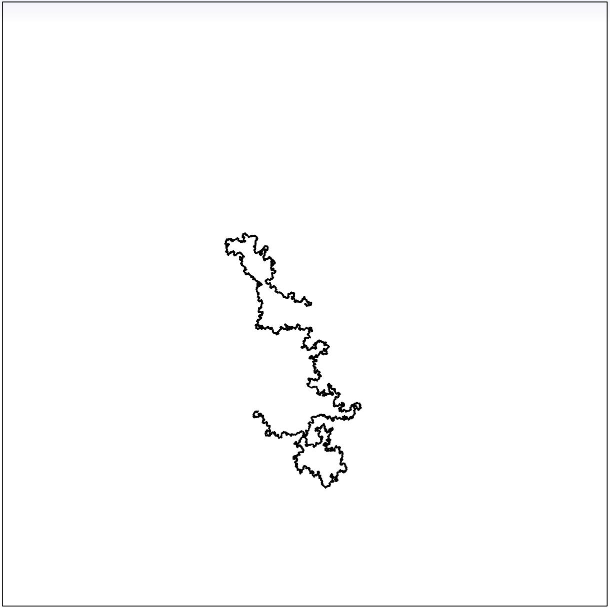


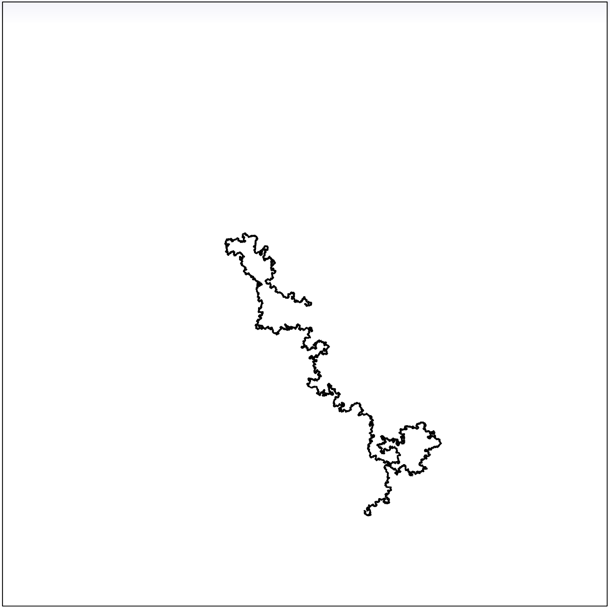


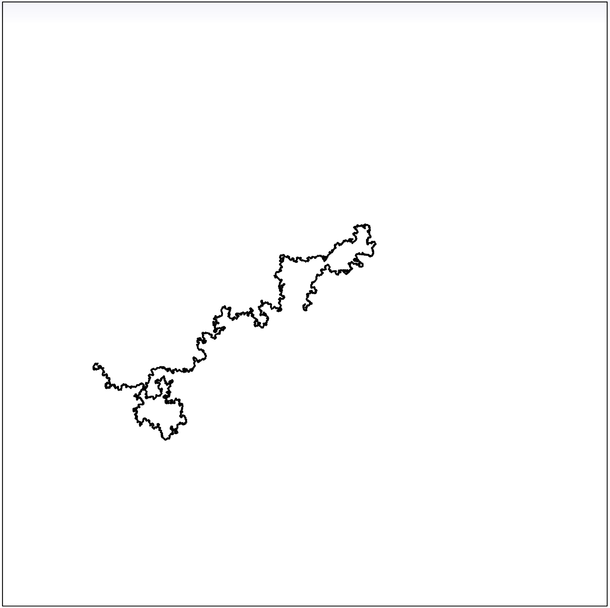


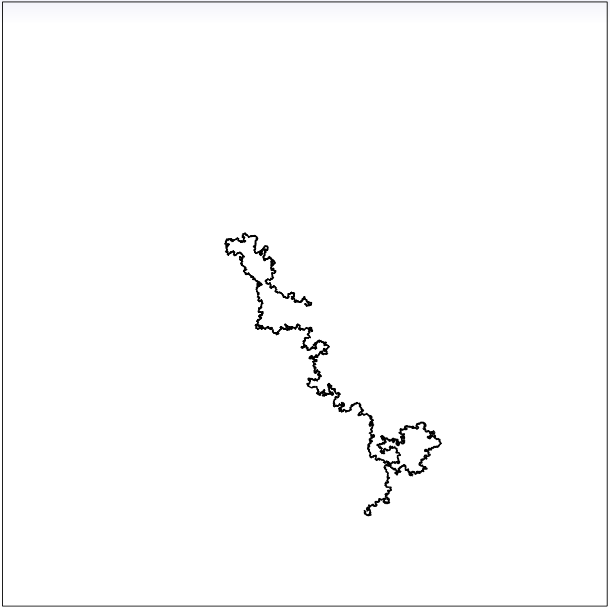


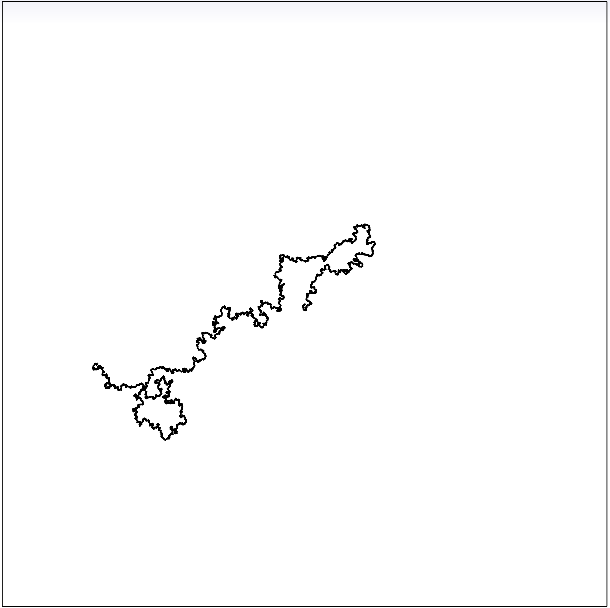


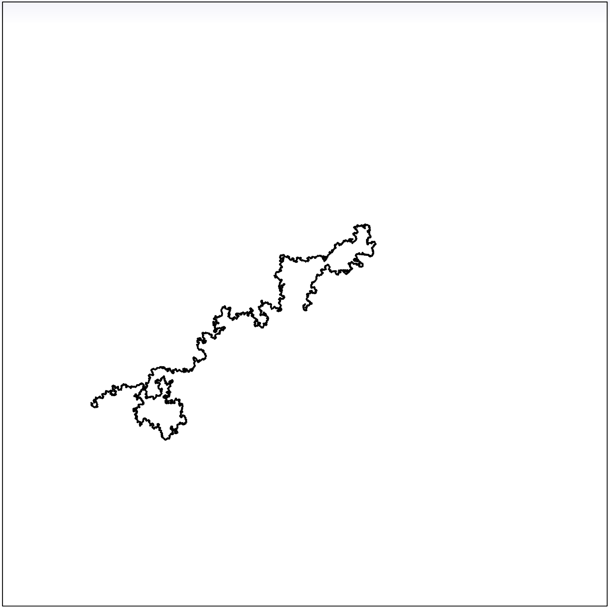


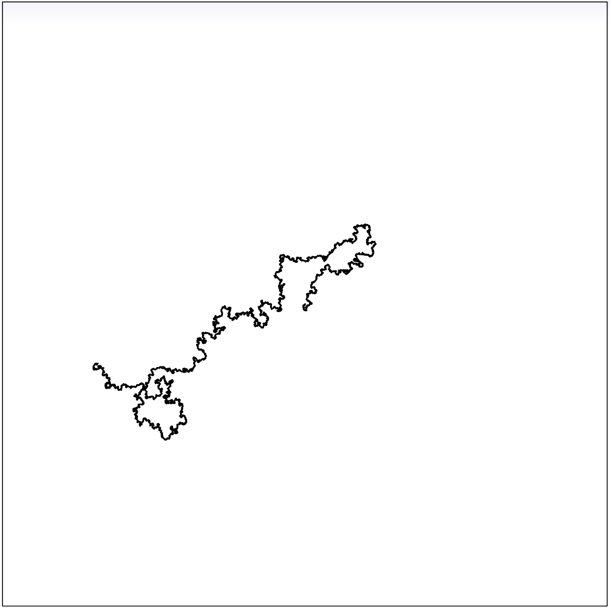


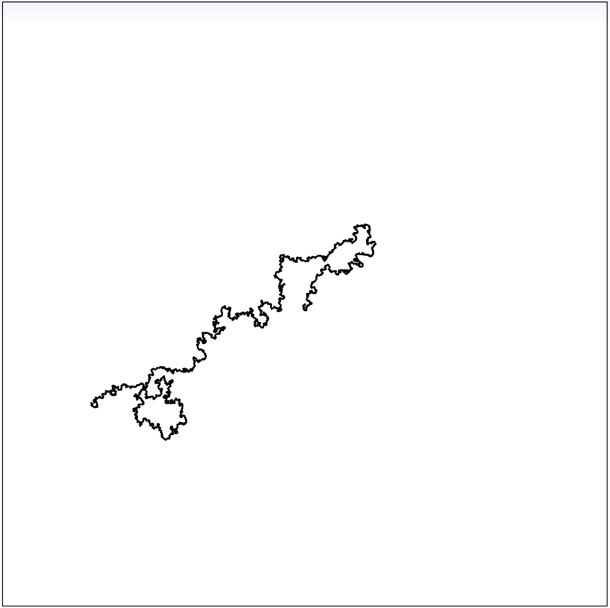


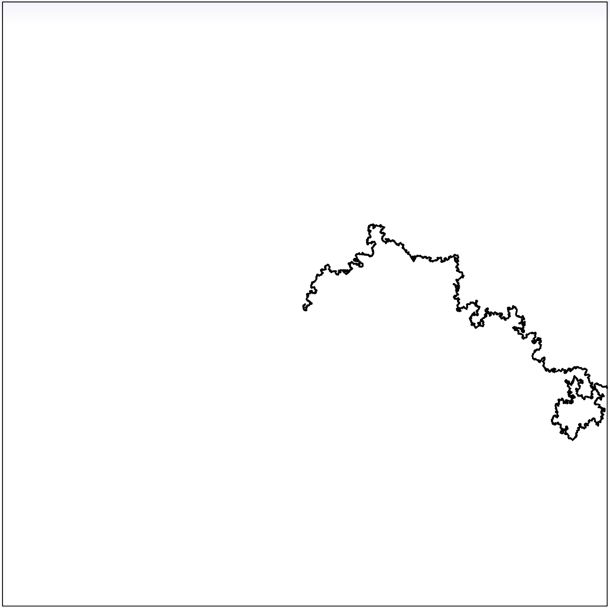


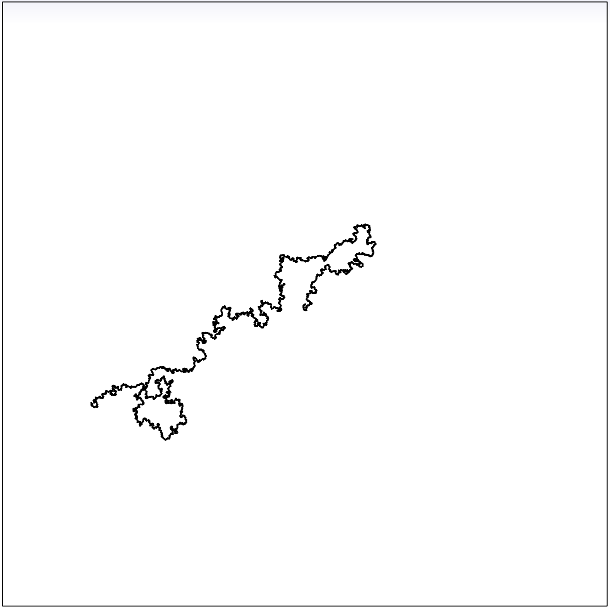


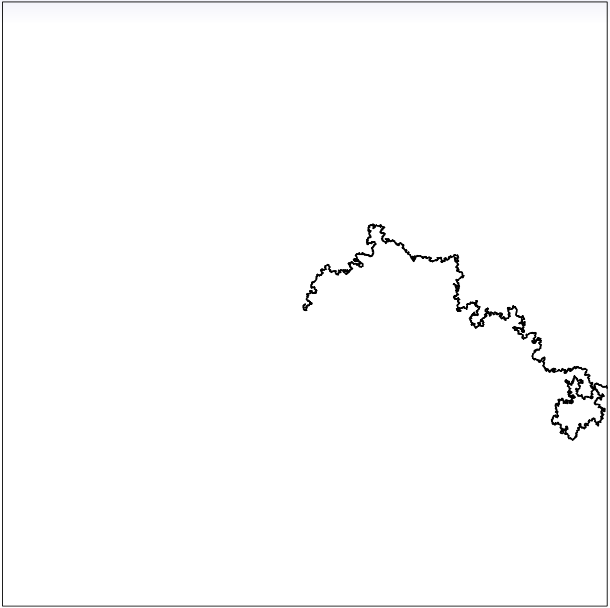


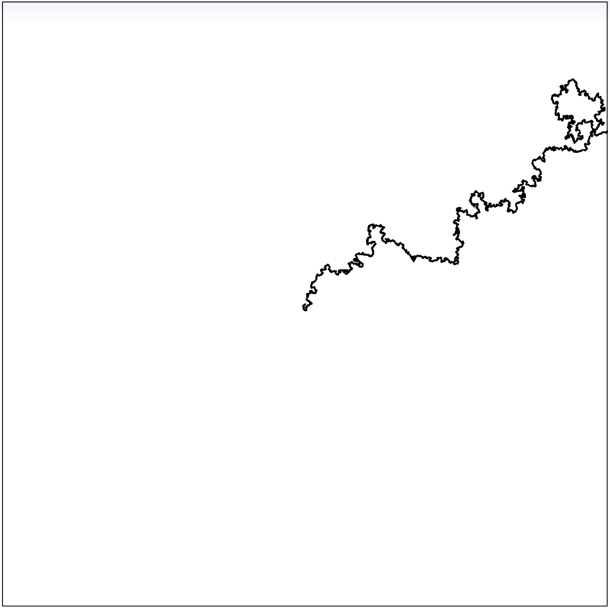


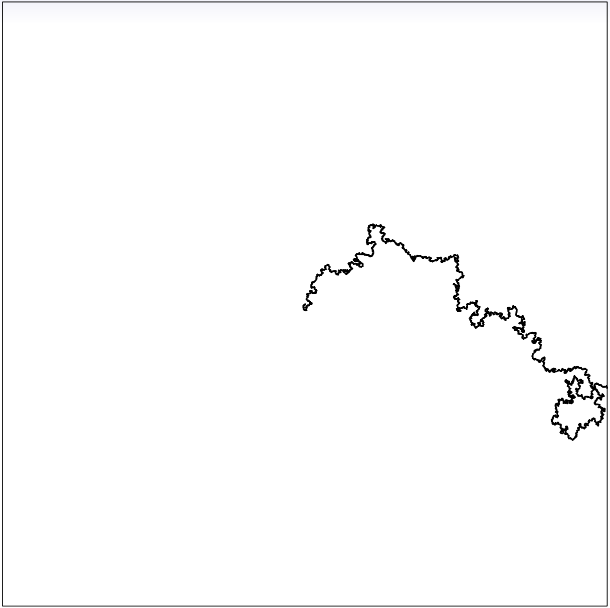


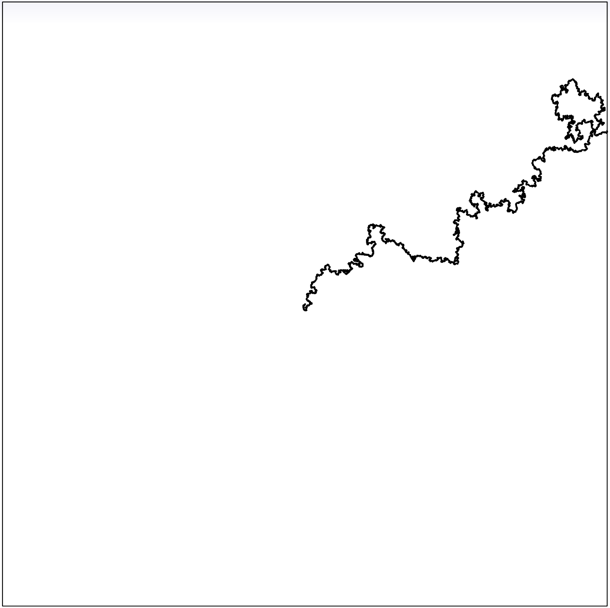


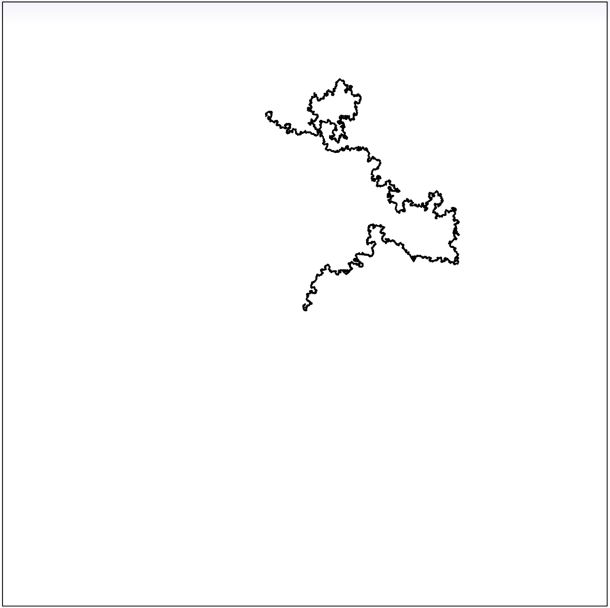


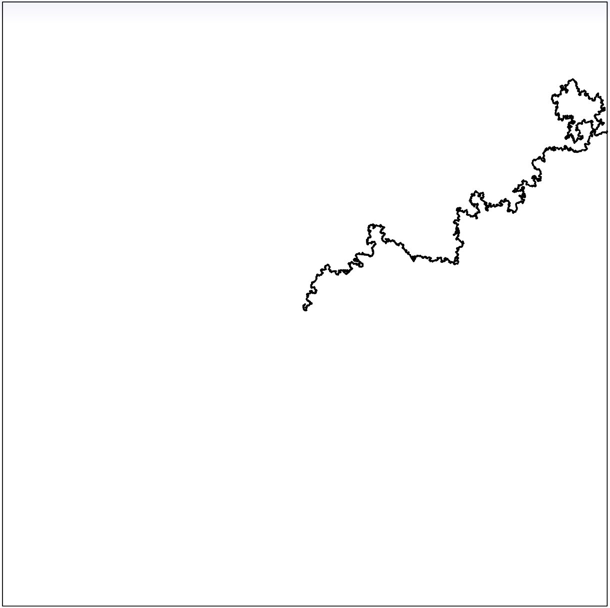


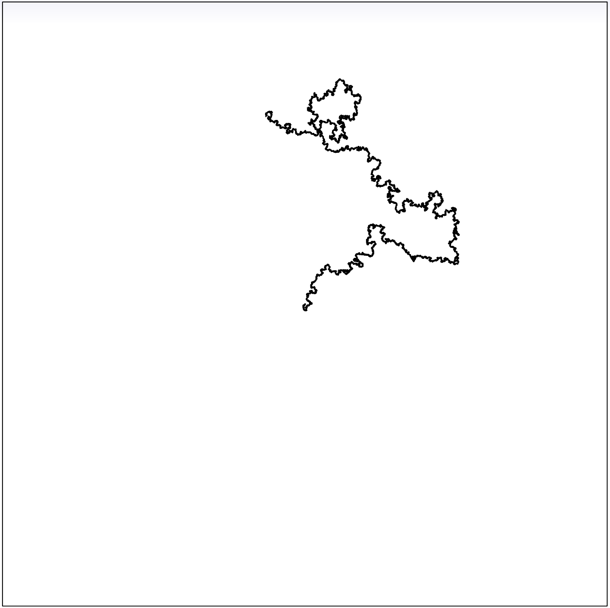


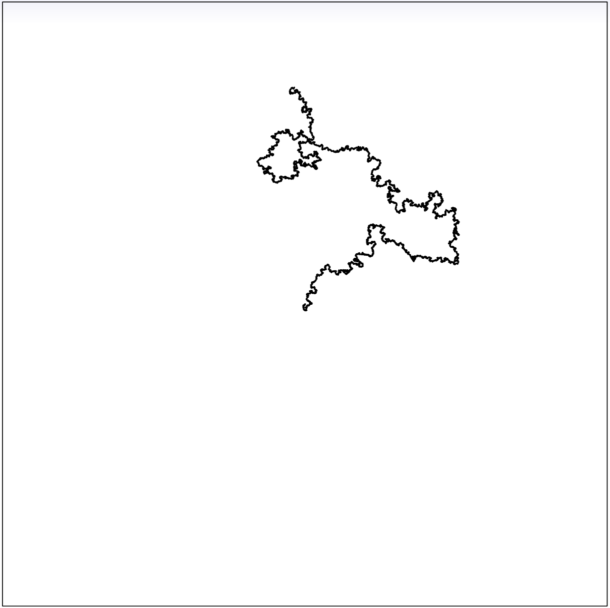


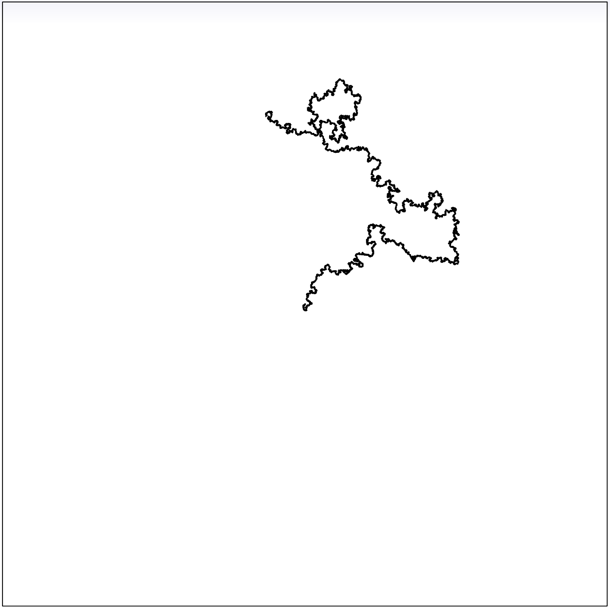


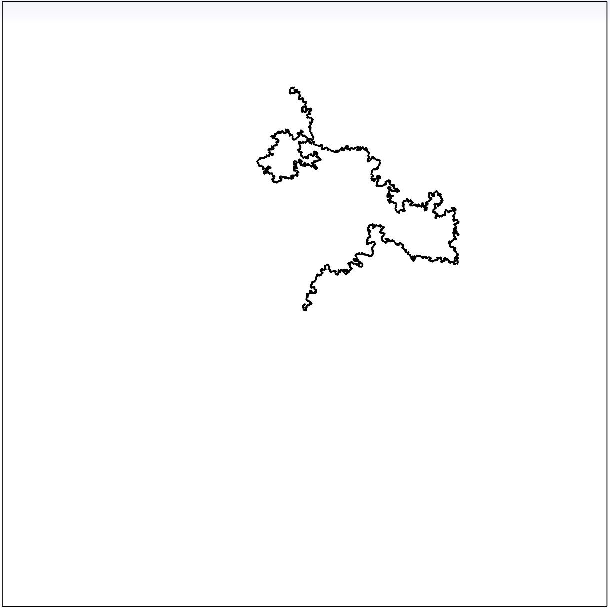


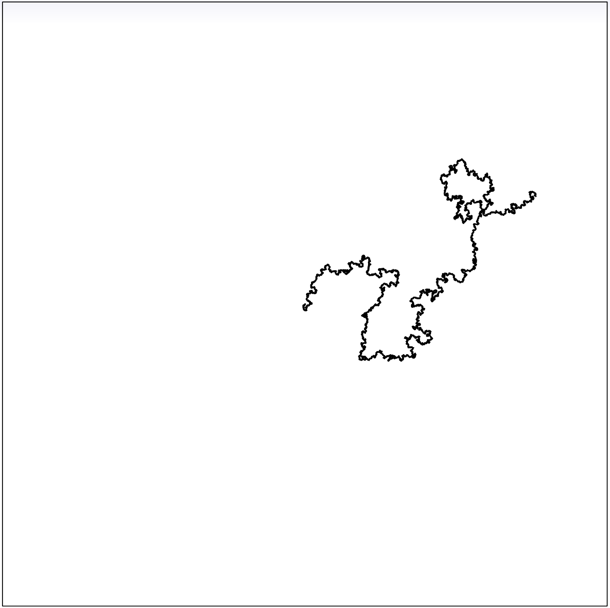


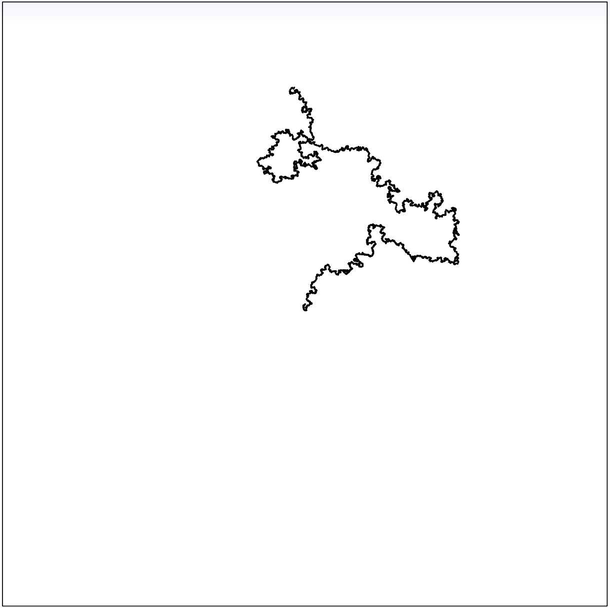


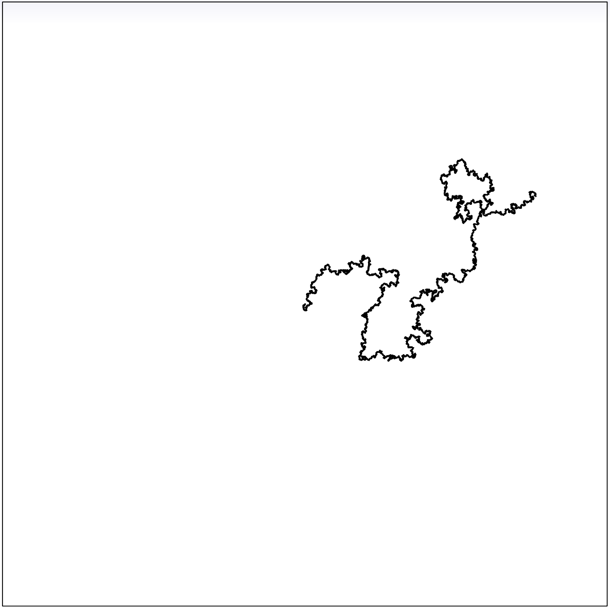


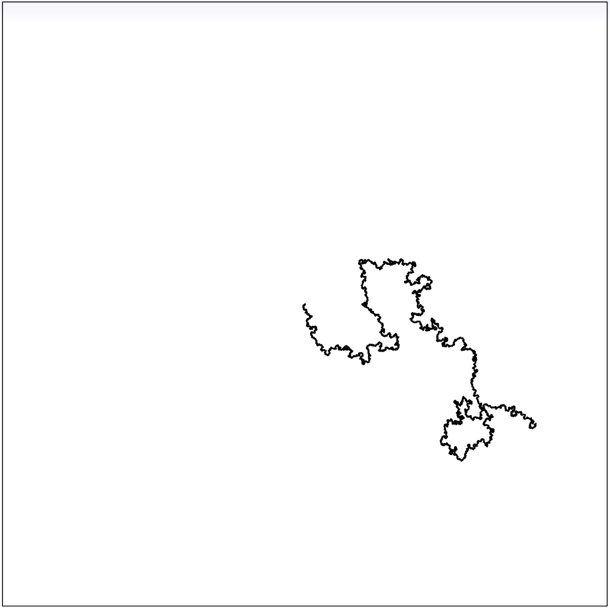


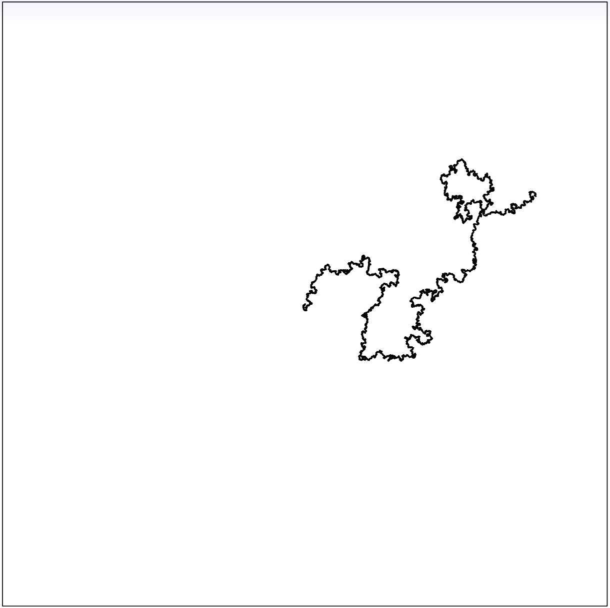


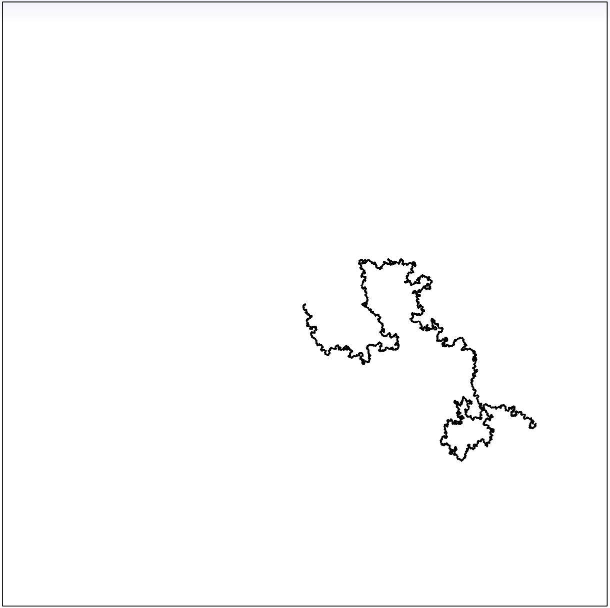


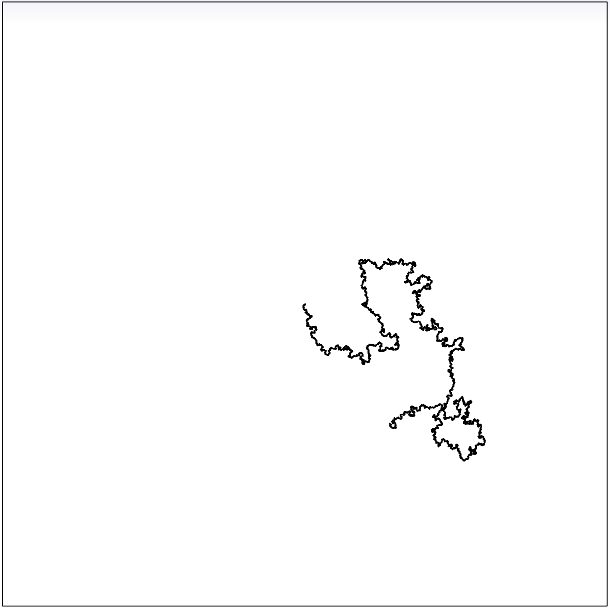


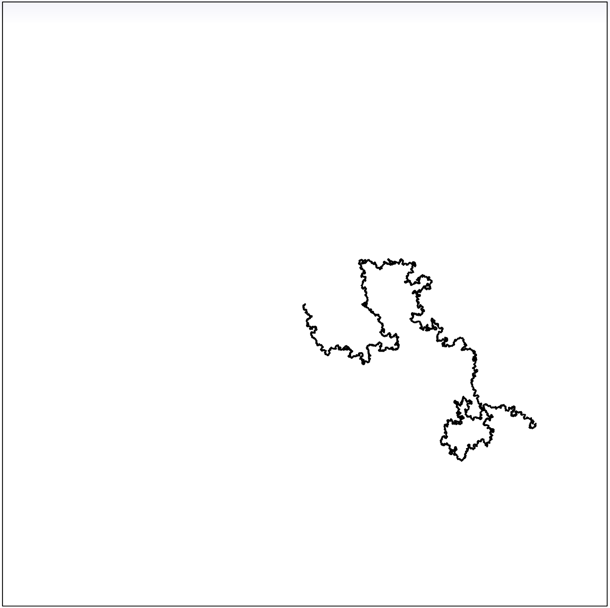


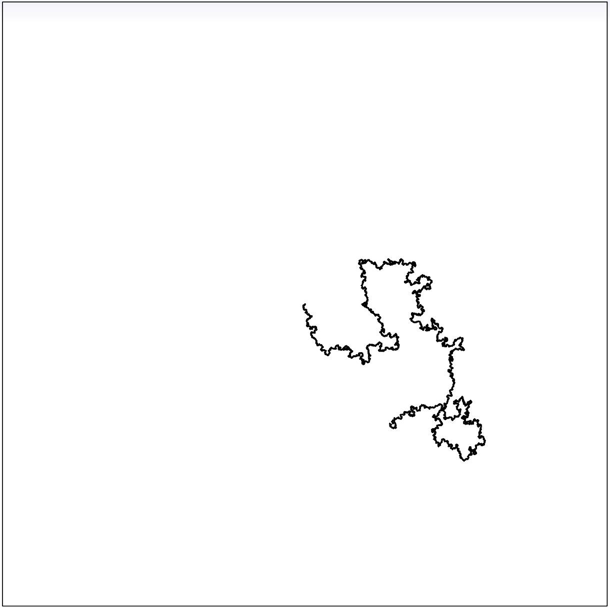


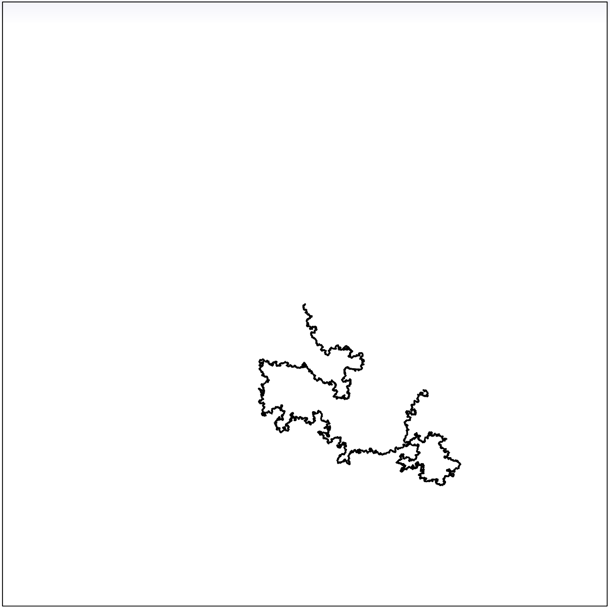


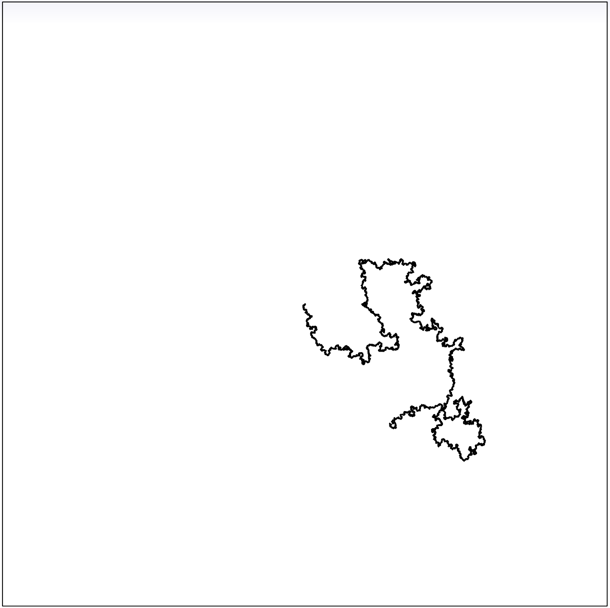


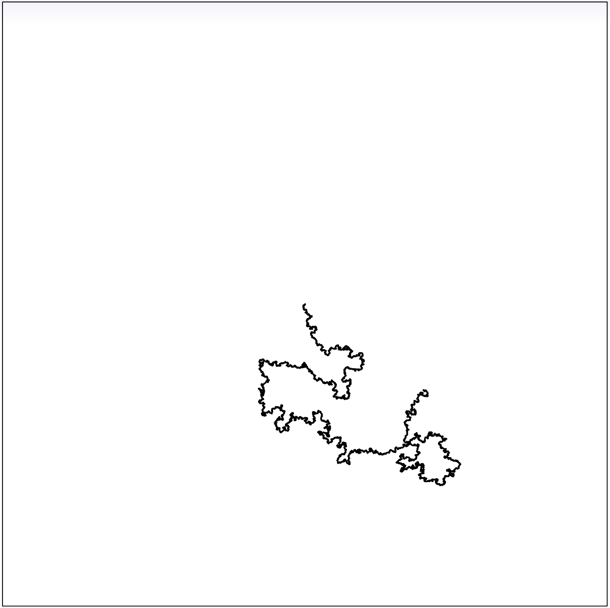


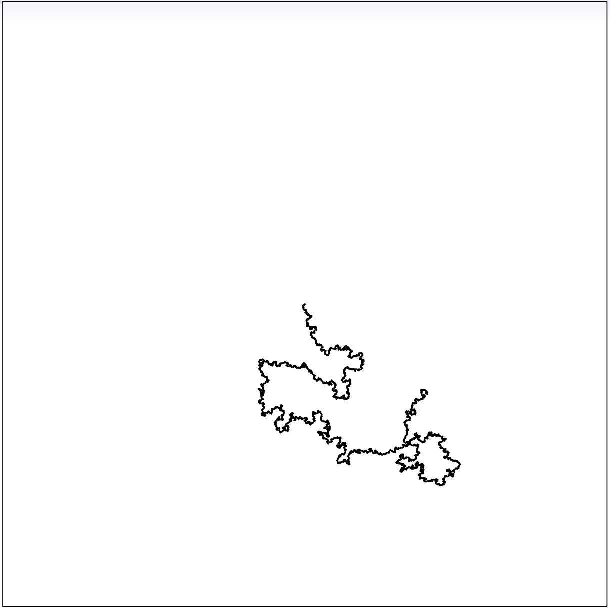


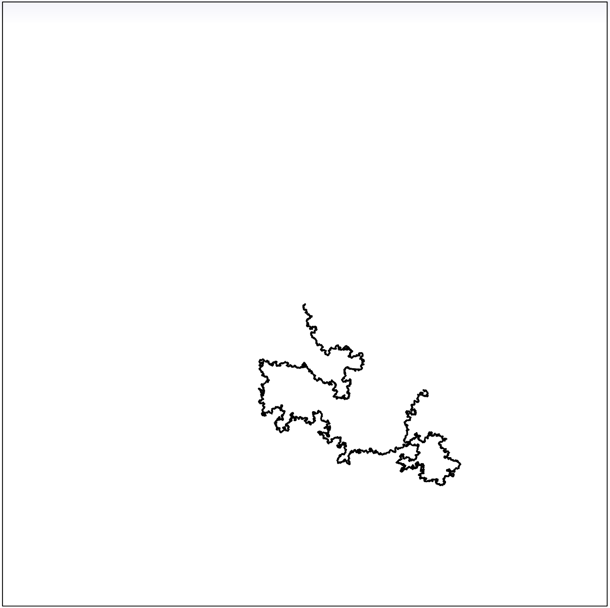


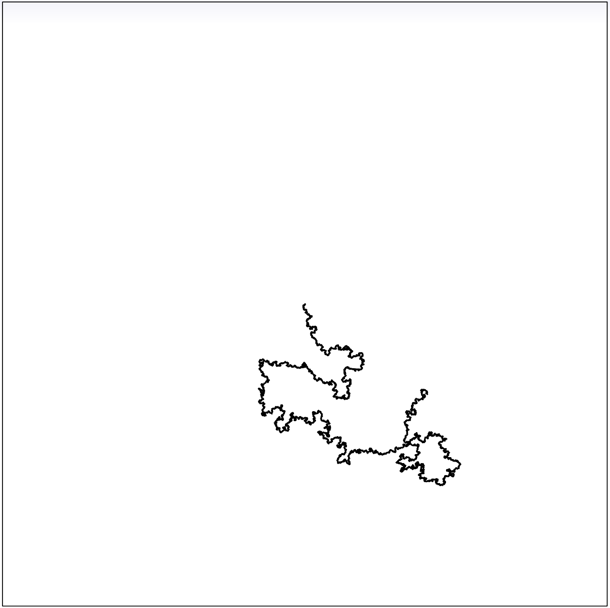


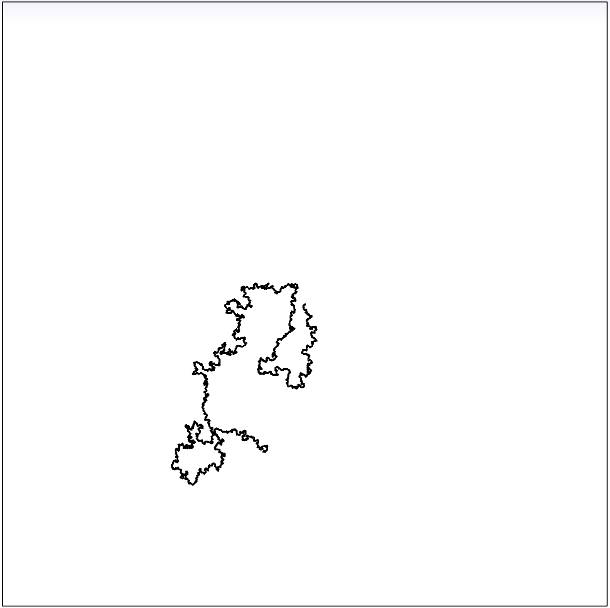


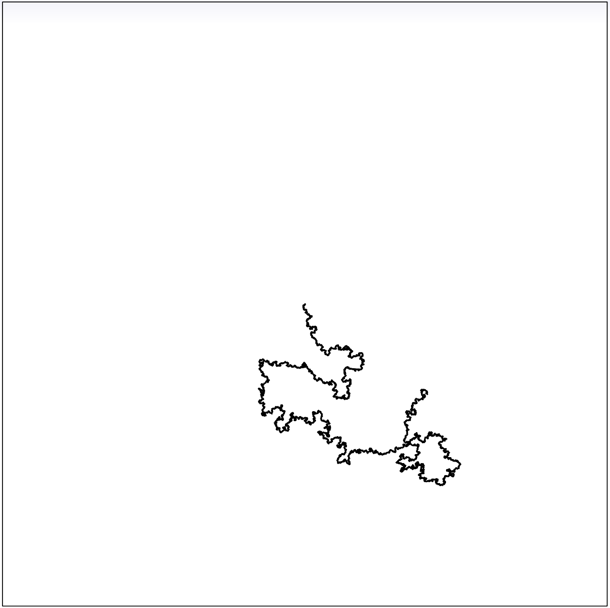


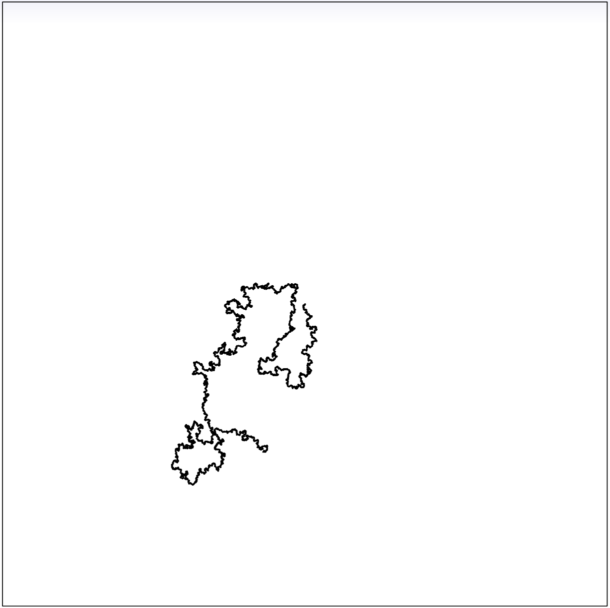




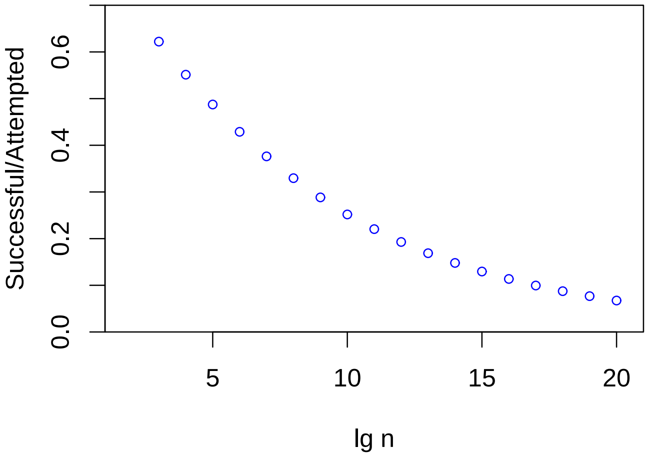




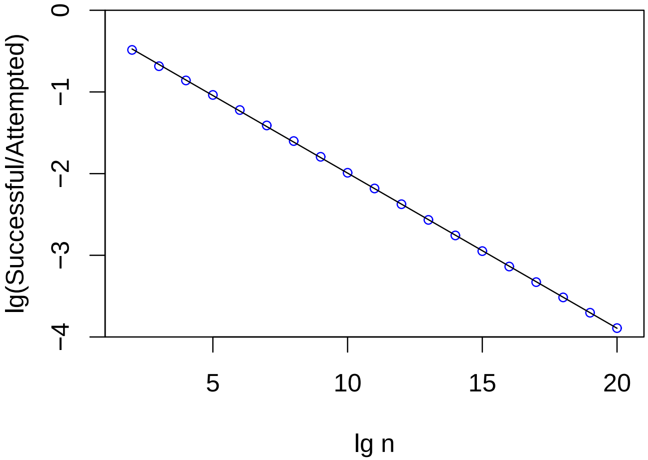




How often are pivots successful?



How often are pivots successful?



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. R_e^2 are (almost) uncorrelated.
- Integrated autocorrelation time for global observables $O(n^p)$, $p \approx 0.19$ for square lattice, $p \approx 0.11$ for cubic.
- Monte Carlo methods which rely on *local* rather than *global* moves and typically take much more CPU time to generate an effectively independent configuration.

Implementation

Madras and Sokal, 1988:

- Use a hash table to test for self-intersections.
- When a pivot is attempted, build up new configuration incrementally, starting at pivot site.
- If resulting configuration is:
  - Not self-avoiding ($Prob \sim 1$), then intersection will typically be found in time $O(n^{1-p})$.
  - Self-avoiding ($Prob \sim n^{-p}$), must generate whole walk, time $O(n)$.
- Overall, $O(n^{1-p})$ time per attempted pivot.
- If we update the whole data structure after each successful update, then this is the best possible implementation

Kennedy, 2002:

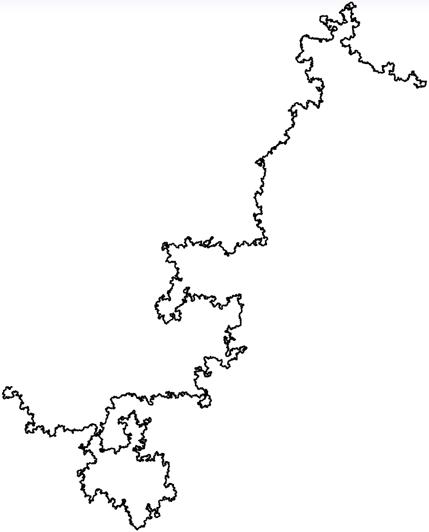
- Complicated implementation, $O(n^{\alpha})$ ($\alpha < 1 - p$) version of the pivot algorithm! Don't need to update the whole data structure after each successful update.

Fast pivot algorithm

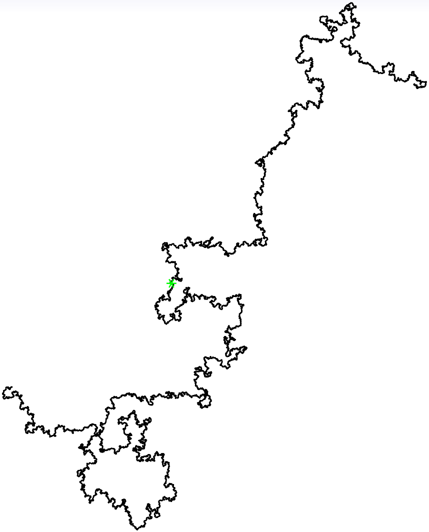
- Ingredients for a fast implementation of the pivot algorithm:
  - a fast test for intersections for a proposed pivot move;
  - a fast update operation to change walk if pivot is accepted.
- Key observation: a SAW can be decomposed into two (equal) subwalks, with a symmetry operation concatenating the two subwalks.
- Results in a natural *binary tree* structure.
- State of a walk, which includes information on global observables such as R_e^2, only depends on the states of its two subwalks.
- R_e^2, R_g^2, R_m^2 can all be calculated in this way.

Intersection testing

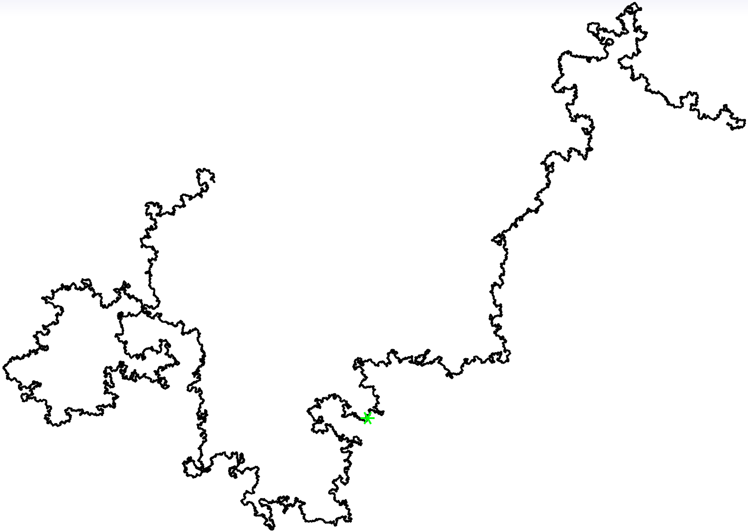
- How much information do we need about a walk in order to decide whether it is self-avoiding?
- Imagine looking at a walk with a magnifying glass.
  - For parts of the walk which are far apart we can easily see that there are no intersections.
  - Whenever parts of the walk approach each other we need to examine the walk closely using magnifying glass.
  - Degree of magnification depends on how closely they approach.
- This idea captured by storing *bounding box* information.



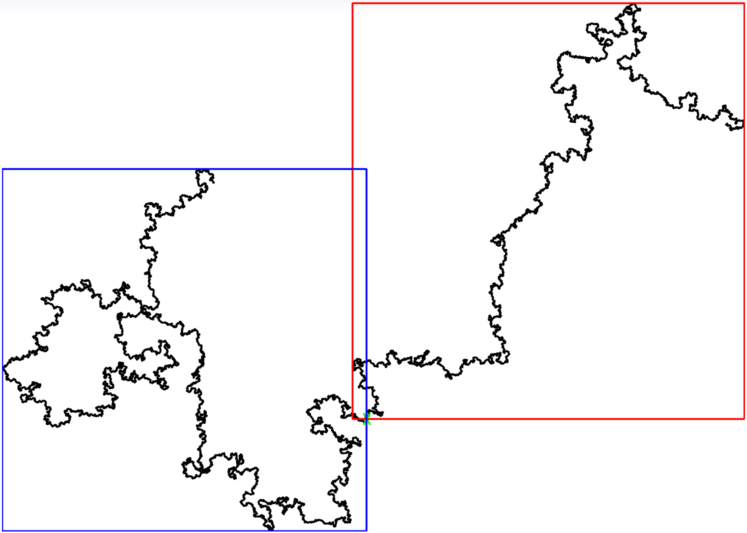
A 65536 site SAW.



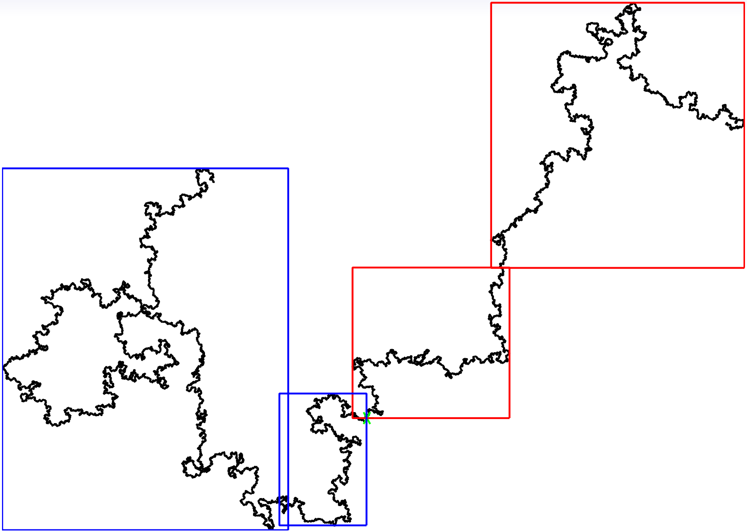
Attempt to pivot (rotate 90°) around green star.



How does the intersection testing algorithm determine if the new configuration is self-avoiding?

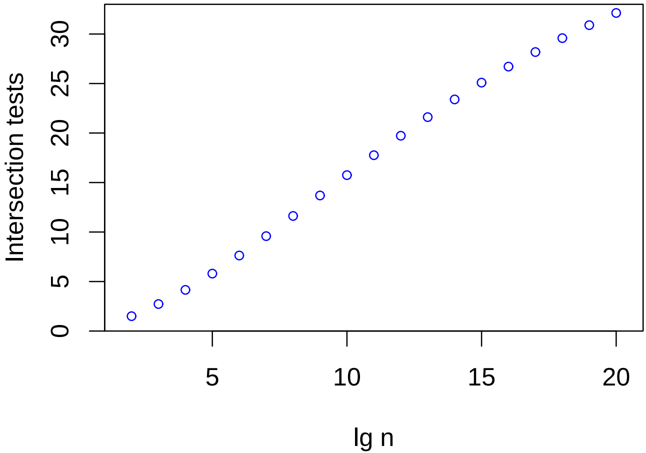


Bounding boxes intersect, and so two halves of the walk may intersect.



Refine bounding boxes; intersection can only occur between subwalks whose bounding boxes intersect.

Mean number of intersection tests necessary for a pivot attempt (successful and unsuccessful pivots).



- CPU time per attempted pivot, for a SAW of length n:

	Madras and Sokal	Kennedy	New method
Square	$n^{0.81}$	$n^{0.38}$	$\log n$
Cubic	$n^{0.89}$	$n^{0.74}$	$\log n$

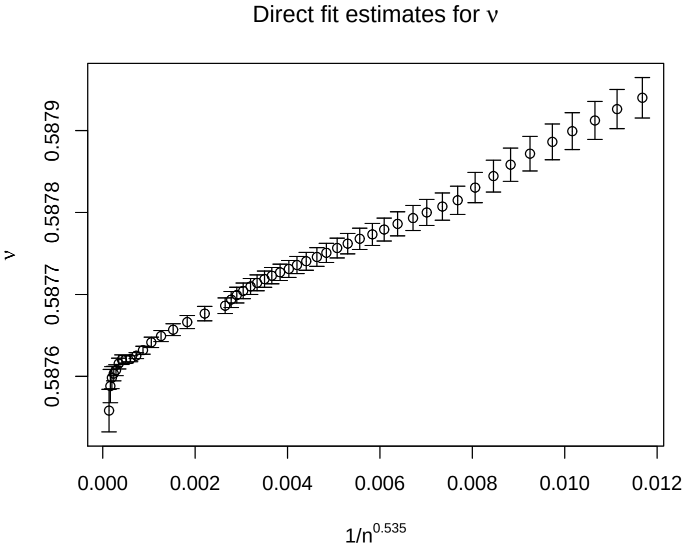
- Approximate CPU time per attempted pivot for $n = 10^6$:

	Madras and Sokal	Kennedy	New method
Square	1240	15.5	1
Cubic	1410	202	1

Critical exponents: ν

- Pivot algorithm samples uniformly from within set of SAWs with fixed length, calculate $\langle R_e^2 \rangle$, $\langle R_g^2 \rangle$, $\langle R_m^2 \rangle$, and hence estimate ν.
- Lengths ranged from 500 to 33 million, 1.9×10^{13} configurations in total (16500 CPU hours).
- Fit coefficients in asymptotic form:

$$\langle R_e^2 \rangle = Dn^{2\nu} \left[1 + \text{corrections} \right].$$

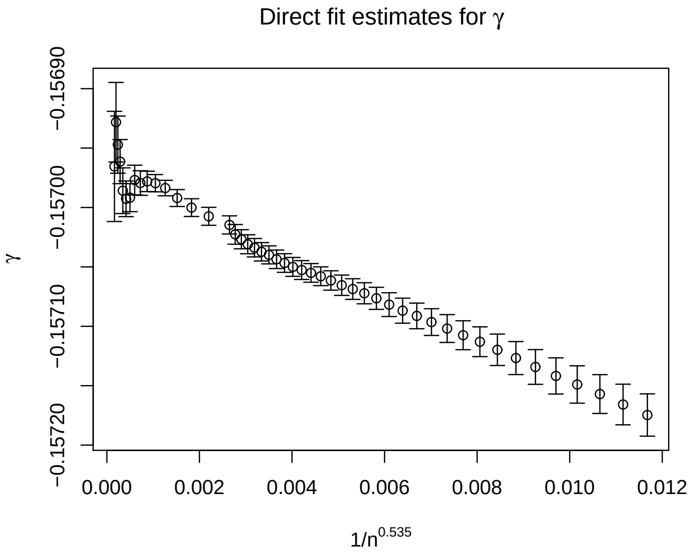

Direct fit estimates for ν
ν
0.5876
0.5877
0.5878
0.5879
0.000
0.002
0.004
0.006
0.008
0.010
0.012
1/n^0.535

Critical exponents: γ

- Two Markov chains of n-step SAWs generated via the pivot algorithm.
- Probability that can concatenate the two SAWs to form a valid $2n$-step walk is:

$$\begin{aligned} P_n &= \frac{c_{2n}}{c_n^2} \approx \frac{A\mu^{2n}(2n)^{\gamma-1}}{(A\mu^n n^{\gamma-1})^2} \\ &\approx \frac{2^{\gamma-1}}{An^{\gamma-1}} \end{aligned}$$

- Note: preferentially choose pivot sites near the ends being concatenated.
- Estimate γ from P_n.
- Lengths ranged from 500 to 33 million, 0.8×10^{13} configurations in total (12500 CPU hours).


Direct fit estimates for γ
γ
−0.15690
−0.15700
−0.15710
−0.15720
0.000
0.002
0.004
0.006
0.008
0.010
0.012
1/n^0.535

Conclusion

- Preliminary estimates are $\nu = 0.587596(10)$ and $\gamma = 1.156954(15)$. These values are an order of magnitude more accurate than any other estimates in the literature.
- Up to 10^9 steps possible.
- Method can be easily extended to continuum (off lattice) walks.
- Can be applied to other models, e.g. polymers in a confined space, polymers with short range attraction (hydrogen bonding).
- Other applications? e.g. pivot algorithm can be used to study protein conformations, ultra long polymers for industrial purposes.

SAP on cubic lattice with $n = 92672$.