Numerical simulation of random curves - lecture 3

Tom Kennedy

Department of Mathematics, University of Arizona

Supported by NSF grant DMS-0501168

http://www.math.arizona.edu/˜tgk

3. Simulation of lattice models

Models which in 2d at critical point are SLE

Simulation of these models not limited to criticality or 2d

All of the algorithms have been around for some time.

So discussion will be rather brief.

For the self-avoiding walk, the fast implementation of the algorithm we discuss is relatively new, so we will spend more time on its description.

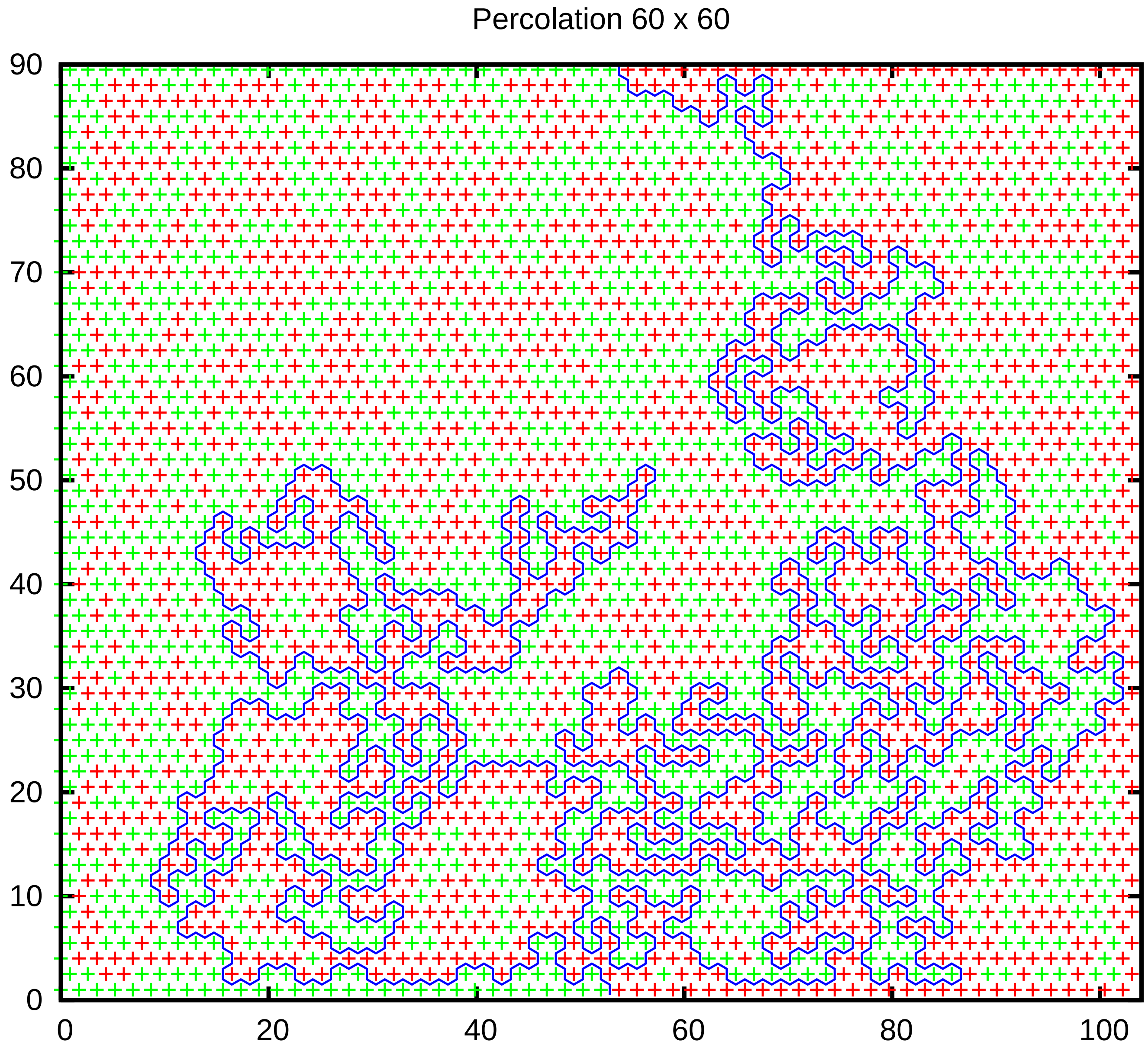
Outline

- Static Monte Carlo (independent samples)

  Percolation and loop-erased random walk (LERW)
- Dynamic Monte Carlo or Markov Chain MC

  Ising and self-avoiding walk (SAW)
- Some practical issues
- Some comparisons of SLE and lattice models
- Problems-projects-homework

3.1.1 Percolation



Percolation

Naive simulation: generate random colors for all the hexagons in a large rectangle.

Construct the exploration path.

Grossly inefficient. Only generate colors as needed.

Must be able to quickly see if a hexagon is already colored.

Giant matrix to store the hexagon colors - memory problem

Use hash function

Hash functions

$h : lattice \to \{1, 2, \cdots, M\}$

M is amount of memory you have.

h is many to one.

Good hash function is “random.” Should send nearby sites in lattice to far apart indices.

hash function for 2d SAW:

$(x, y) \to |(ax + by))\ mod\ M|$

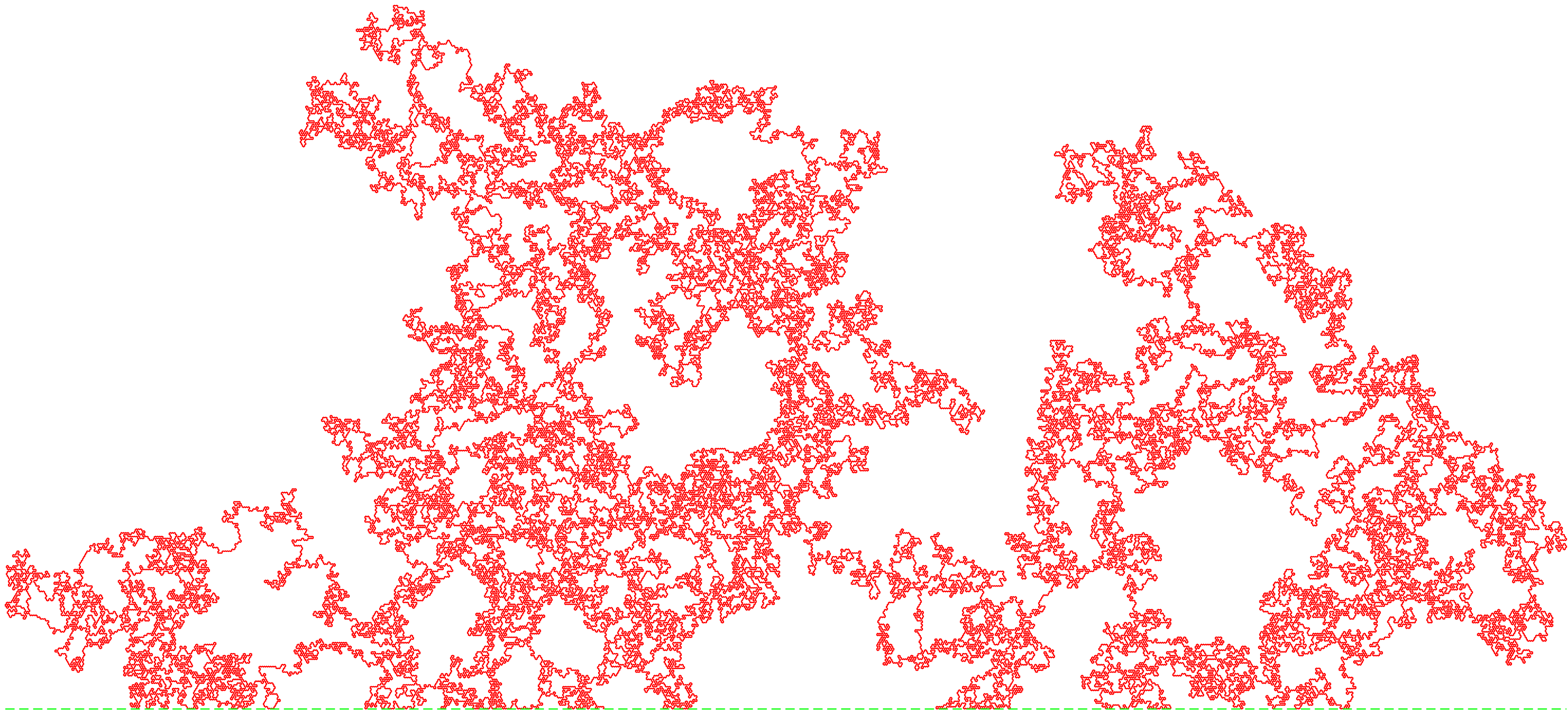
$M = 11001001,\ a = 147,\ b = 23954$

Colisions: what do you do when a lattice site is mapped to a memory location that is already being used?

linear probing

Percolation

100,000 steps



3.1.2 Loop-erased random walk

Take bounded domain D containing origin.

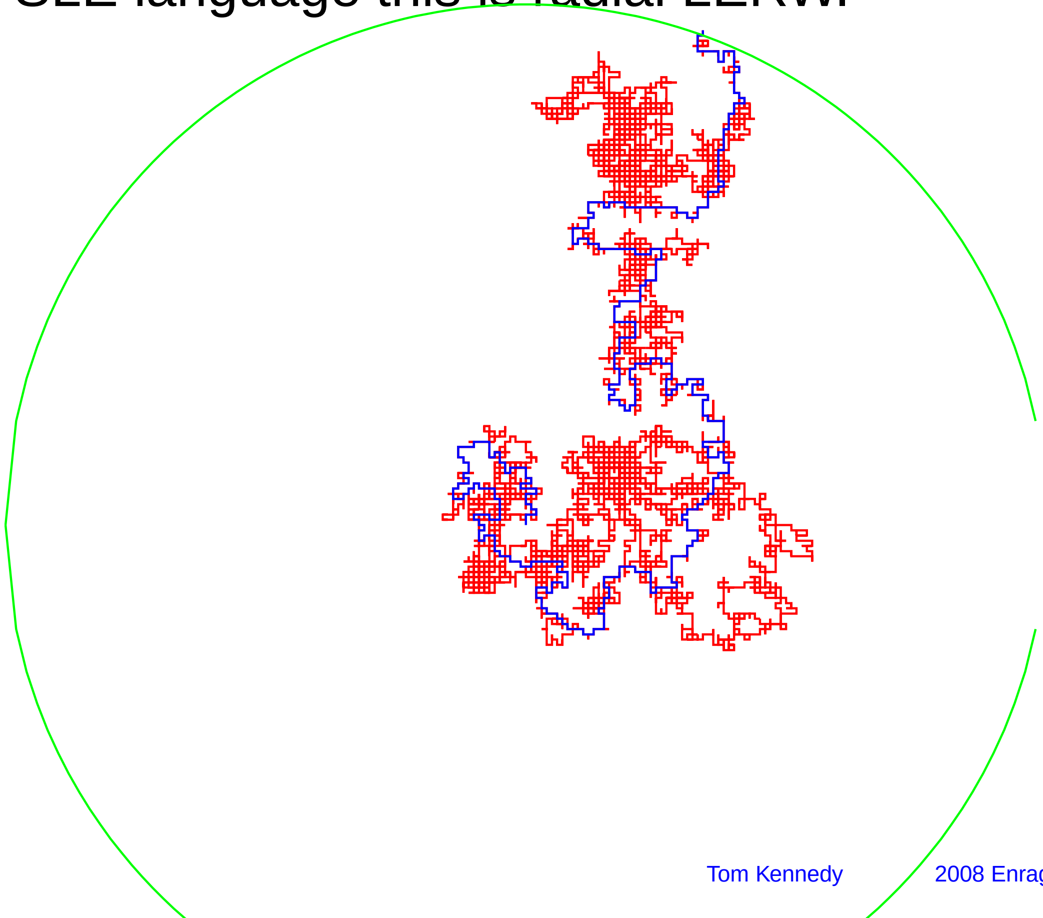
Start ordinary random walk at origin. Run till it leaves D.

Erase loops in chronological order.

Result is a simple random walk from origin to ∂D.

Could condition original random walk to end at a particular site.

In SLE language this is radial LERW.



LERW simulation (radial)

Take D to be a disc centered at 0.

Introduce a lattice with small lattice spacing.

By symmetry, conditioning to end at at fixed point is easy.

Simulation of this LERW is straightforward.

Memory caveats

Erase the loops as you generate the random walk.

Testing for loops : use a hash function

Loop-erased random walk (chordal)

LERW from boundary point to boundary point.

Random walk starts at prescribed boundary point
Condition it to leave D at another prescribed boundary point.

For half plane and 0 to ∞ this is easy.

The ordinary random walk starting at the origin and conditioned to stay in the upper half plane is the half-plane excursion.

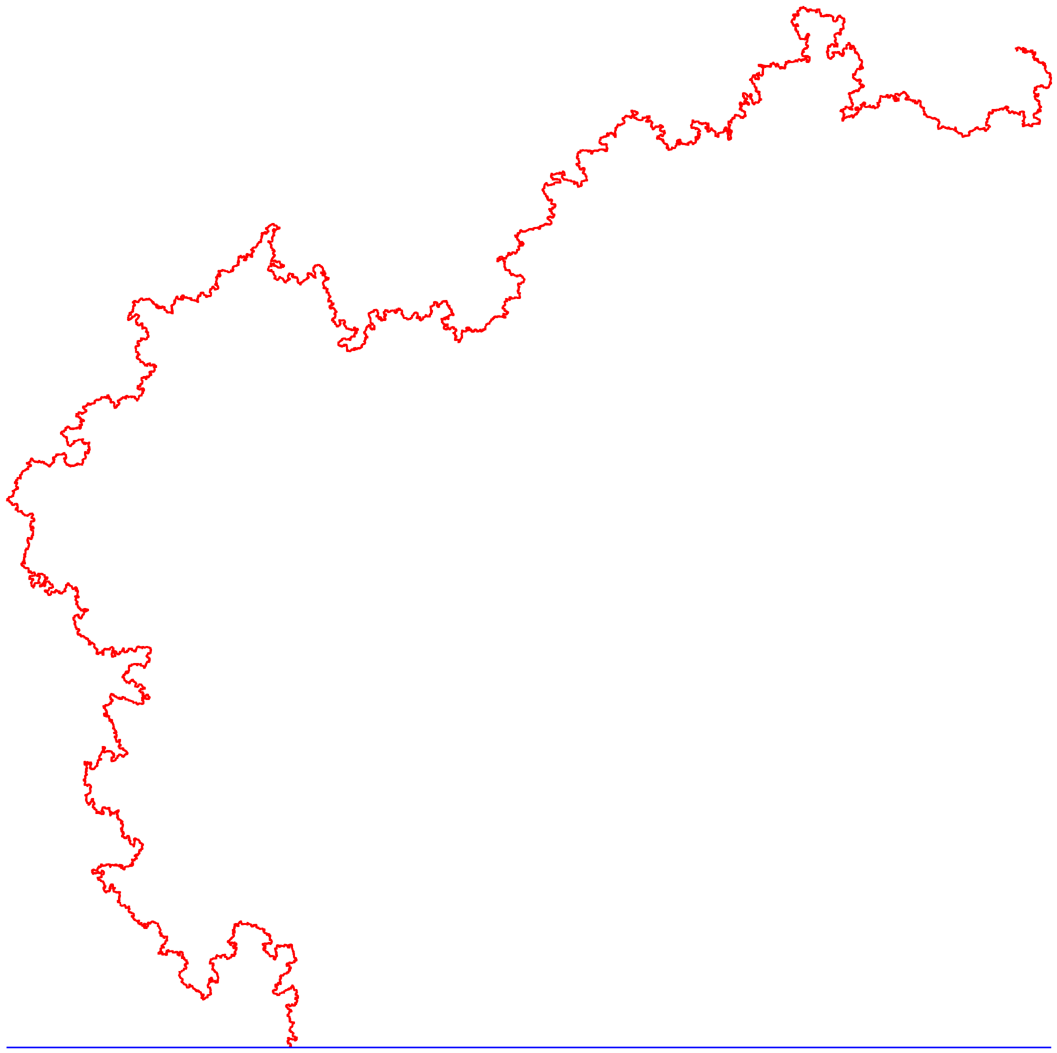
It is just a random walk with transition probabilities

$$\begin{aligned} P(x+yi \to x+(y+1)i) &= \frac{y+1}{4y} \\ P(x+yi \to x+(y-1)i) &= \frac{y-1}{4y} \\ P(x+yi \to x-1+yi) &= P(x+yi \to x+1+yi) = \frac{1}{4} \end{aligned}$$

This process is transient, i.e., the walk will converge to ∞.

LERW in half plane

100,000 steps (in LERW)



3.2.1 MCMC theory

Markov chain: It is specified by

- State space S. For us, a finite set.
- Transition matrix $p_{x,y} = P(x \to y)$. $\sum_y p_{x,y} = 1$.

Irreducible: if you can get from any state to any other through sequence of transitions with $p_{x,y} > 0$.

Aperiodic: If there is a state x such that $p_{x,x} > 0$ then chain is aperiodic. (Not the definition.)

π_x denotes probability measure on S

π is stationary if $\sum_x \pi_x \, p_{x,y} = \pi_y$

π satisfies detailed balance if $\pi_x p_{x,y} = \pi_y p_{y,x}$

Detailed balance $\Rightarrow$ stationary.

MCMC theory - cont.

For a finite chain, finding a stationary distribution is an eigenvector problem. For an irreducible chain it is unique.

Start the chain in some state x_0. Run it and let X_k be state at time k. Stop after n iterations.

Let f be a function on S (RV or observable).

The time average is

$$\frac{1}{n} \sum_{k=1}^{n} f(X_k)$$

Theorem For finite, irreducible, aperiodic chain, the time average of f converges as $n \to \infty$ to

$$\sum_{x} f(x) \pi_x$$

MCMC theory - cont

Think backwards. Given π find transition matrix.

Not just any transition matrix, find a good one.

Good means fast convergence, short auto-correlation time.

Warning: Monte Carlo always converges like $c/\sqrt{n}$.
Fast algorithms have smaller c, but the $1/\sqrt{n}$ is still there.

3.2.2 Ising model

$$H(\sigma) = - \sum_{<i,j>:i,j \in \Lambda} \sigma_i \sigma_j$$

Sum is over nearest neighbor pairs of sites

Define a probability measure by

$$P(\sigma) = \exp(-\beta H(\sigma))/Z$$

where Z is a constant chosen to make this a probability measure. Explicitly,

$$Z = \sum_{\sigma} \exp(-\beta H(\sigma))$$

Need to specify boundary conditions.

Interface in Ising

slides_ising_500.eps

3.2.2 Ising model

State space is set of spin configurations on Λ.

Glauber algorithm

Pick a lattice site at random.

Compute ΔE, change in energy from flipping spin at this site.

If $\Delta E < 0$, accept the flip.

If $\Delta E > 0$, accept the flip with probability $e^{-\beta \Delta E}$.

If flip is accepted, next state is present state with this spin flipped.

If flip is not accepted, next state is present state.

NB: Chain still takes a time step.

Check this satisfies detailed balance.

Cluster algorithms for Ising

Cluster alogrithms flips a (hopefully) large connected cluster of spins at once.

Swendsen-Wang algorithm - based on FK representation

Look at variant - Wolff algorithm

Wolff algorithm

One time step for Markov chain:

1. Choose a seed spin at random from the lattice. Cluster starts with just this spin.

2. Look in turn at each of the neighbors of that spin.

If they point in the same direction as the seed spin, add them to the cluster with probability $p = 1 - \exp(-2\beta)$.

3. For each spin added in last step, examine each of its neighbors to find the ones that are pointing in the same direction and add each of them to the cluster with probability p.

4,5, ... Repeat step 3 until there are no spins left in the cluster whose neighbors have not been considered for inclusion in the cluster.

n. Flip the cluster

Wolff algorithm - continued

As the cluster grows, we may find that some of the neighbors are already members, in which case we do not have to consider adding them again.

Also some of the spins may have been considered for addition before and rejected. In this case they get another chance to be added to the cluster.

Boundary conditions: If the cluster contains a fixed boundary spin, then we do not flip the cluster.

Wolff algorithm - detailed balance

Different implementation :

Consider all directed bonds in lattice. (Each bond appears twice.)

Delete each bond with probability $1-p$.
$\Rightarrow$ random set of directed bonds B.

Given a seed, grow the cluster as before. For x in the cluster, when we consider whether to add its neighbor y to the cluster, add it if the directed bond $x \to y$ is in B.

Let σ, σ' be two configurations, C the set of sites where they differ.

Transitions $\sigma \leftrightarrow \sigma'$ are possible iff C is a connected set and σ, σ' are constant on C.

Transitions $\sigma \to \sigma'$ and $\sigma \to \sigma'$ occur when the cluster generated is exactly C.

Let Γ be Peierls contour (boundary) of C.

Wolff algorithm - detailed balance - cont.

For spin configurations σ and σ', show

(1) $$\frac{P(\sigma \to \sigma')}{P(\sigma' \to \sigma)} = \frac{P(\sigma')}{P(\sigma)}$$

Compare $P(\sigma \to \sigma')$ and $P(\sigma' \to \sigma)$. Look at B which produce cluster=interior or Γ.

Difference comes from directed bonds that cross Γ.

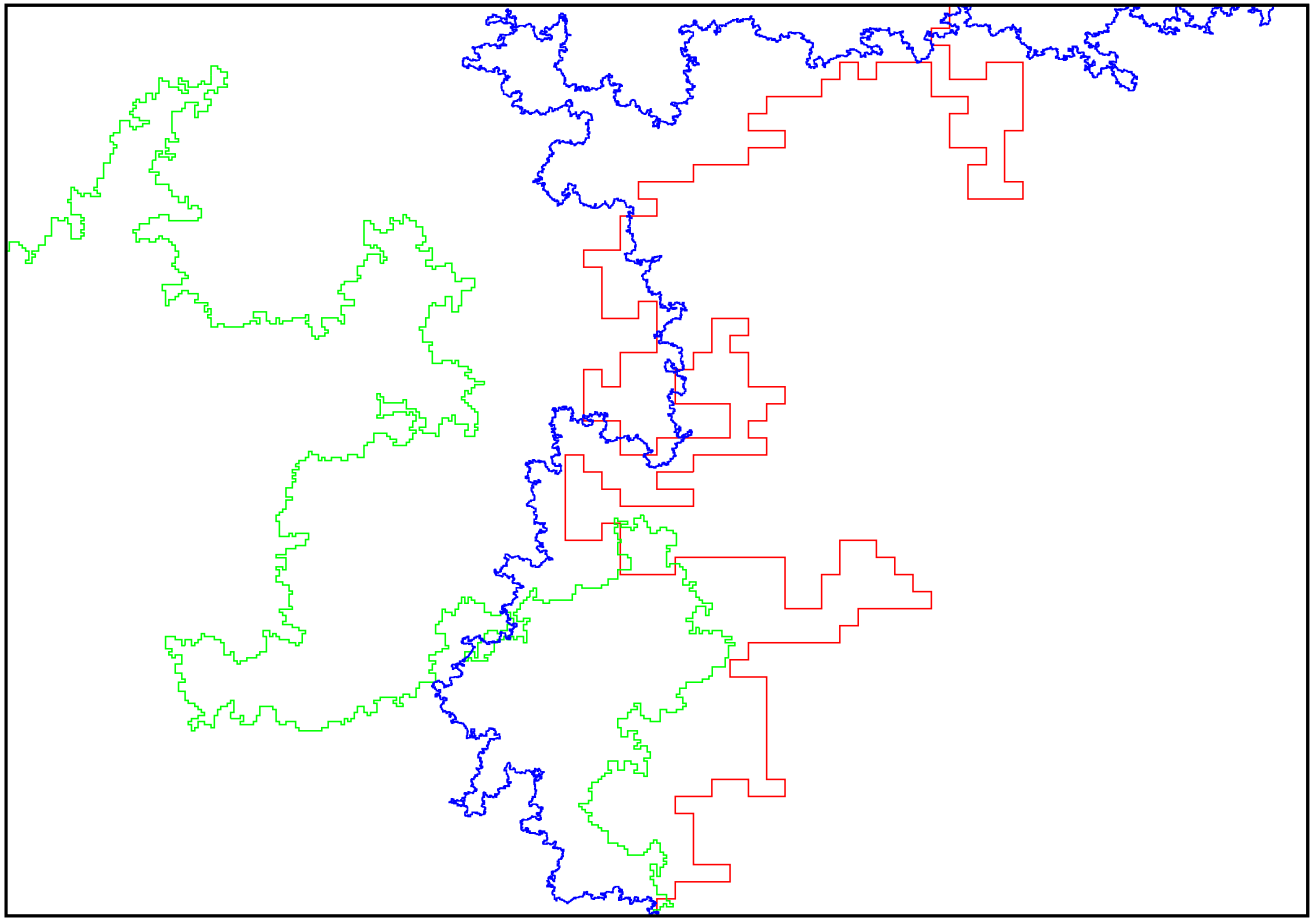
For each such bond you get the same contribution to left and right sides of (1) if $p = 1 - \exp(-2\beta)$.

3.2.3 Monte Carlo for the SAW

Take all N step, nearest neighbor walks in the upper half plane, starting at the origin which do not visit any site more than once.

Give them the uniform probability measure.

Let $N \to \infty$. Then let lattice spacing go to zero.



The pivot algorithm

Fix a length N

State space = $\{$ all SAW of length N starting at 0 $\}$

Pick a site on the walk at random.

(Usually, the sites are given equal probability, but only constraint is that each site has non-zero probability.)

Pick a lattice symmetry at random. (Prob of a symmetry and its inverse must be equal.)

Apply the symmetry w.r.t. the pivot site to the part of the walk “after” the pivot site.

If the new walk is self-avoiding, it is the next state in the chain.

It not, the next state is the state we were already in.

Check detailed balance.

Irreducible, but this takes some work.

How fast is it?

First appears in literature in '69 - Lal

Time to check for a self-intersection is proportional to N. (Use a hash table.)

Probability the pivot is accepted goes as N^{-p}
$p = 0.19$ in $d = 2$, $p = 0.11$ in $d = 3$.

Comprehensive study: Madras and Sokal '88

Look for self intersections starting at the pivot point and working outwards $\Rightarrow$ only take $O(N^{1-p})$ time to check.

So time per accepted pivot is $O(N)$!

In fact you can implement it so time per accepted pivot is $O(N^q)$ with $q < 0.57$ in $d = 2$ and $q < 0.85$ in $d = 3$. (TK, '01).