Self-avoiding walks

Rob Bisseling

Mathematical Institute, Utrecht University

Mathematics colloquium, Utrecht
April 19, 2012

Universiteit Utrecht

Joint work

- 

Gerard Barkema (Theoretical Physics)
- 
Raoul Schram (student mathematics/physics)

Contents

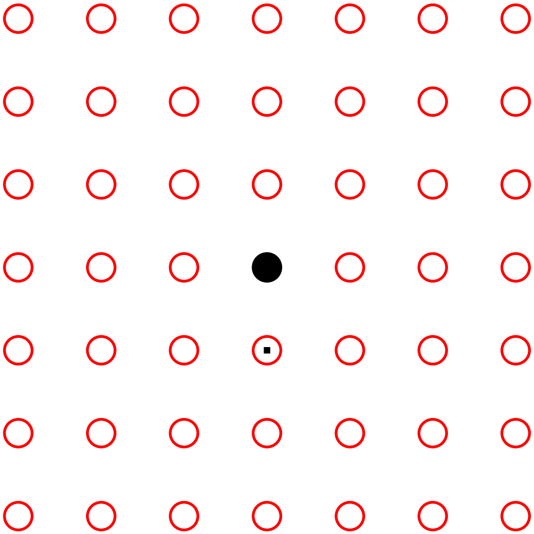
Curiosity-driven walks

Source: my-new-york.com, nyc-architecture.com

Definition self-avoiding walks

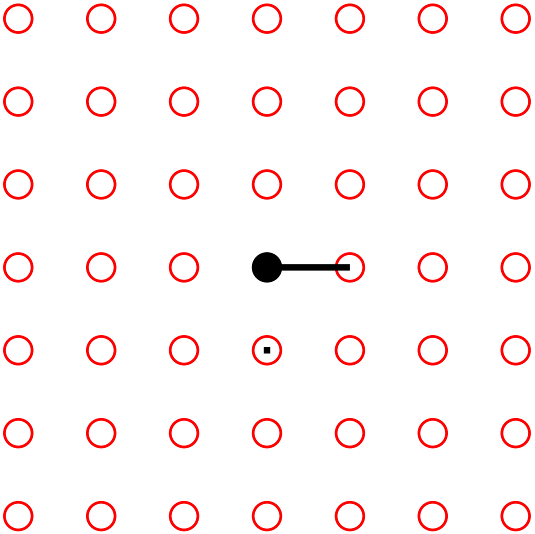
- A self-avoiding walk (SAW) is a walk on a regular lattice that never returns to a position already visited.
- We start in the origin.
- The length of a walk is the number of steps, N.

A self-avoiding walk of length 0 in 2D

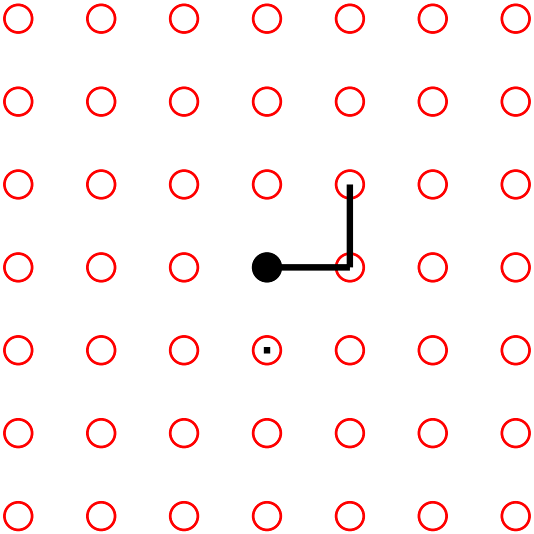


Universiteit Utrecht

A self-avoiding walk of length 1 in 2D

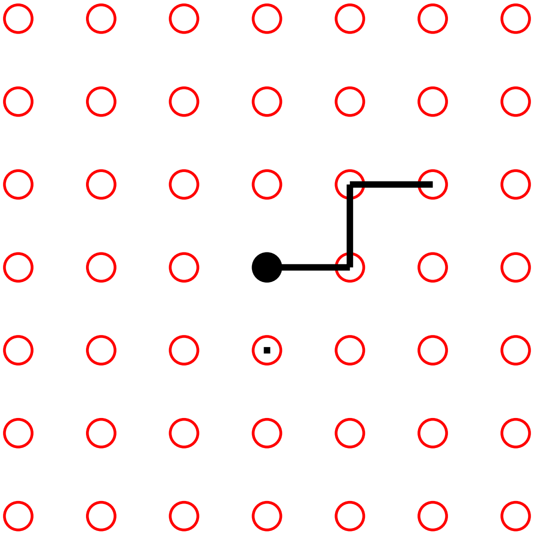


A self-avoiding walk of length 2 in 2D

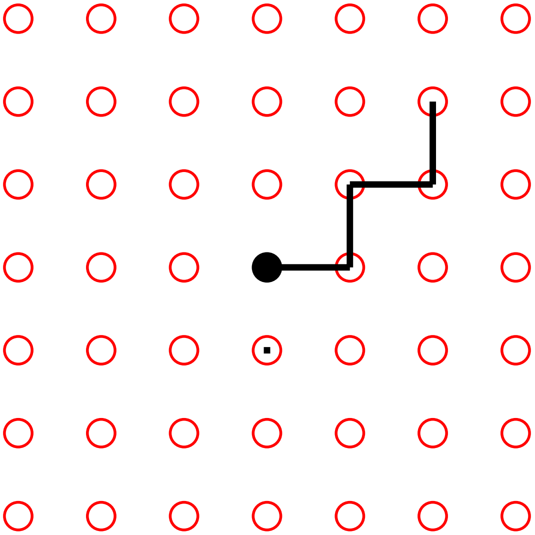


Universiteit Utrecht

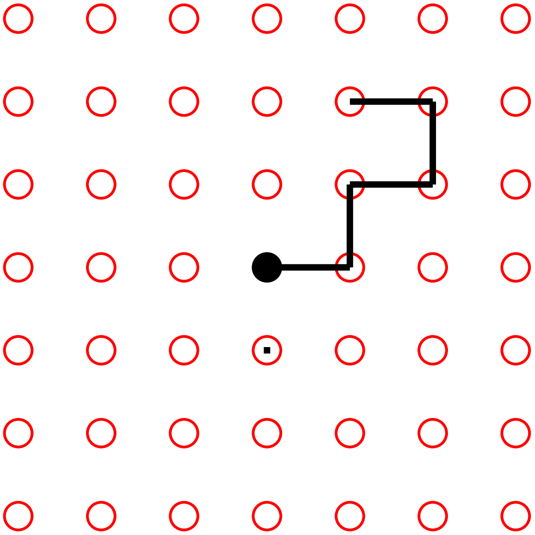
A self-avoiding walk of length 3 in 2D



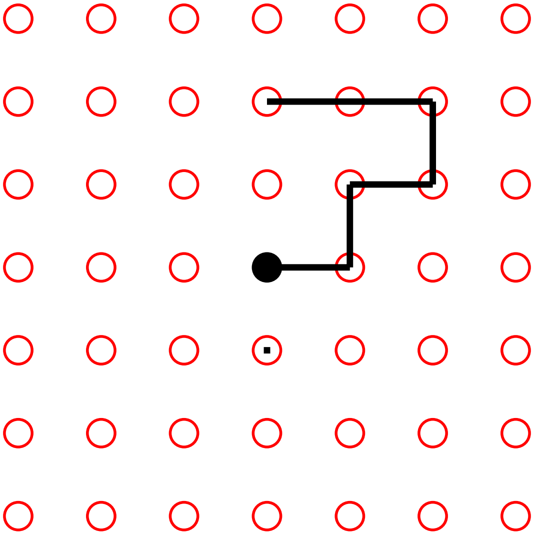
A self-avoiding walk of length 4 in 2D



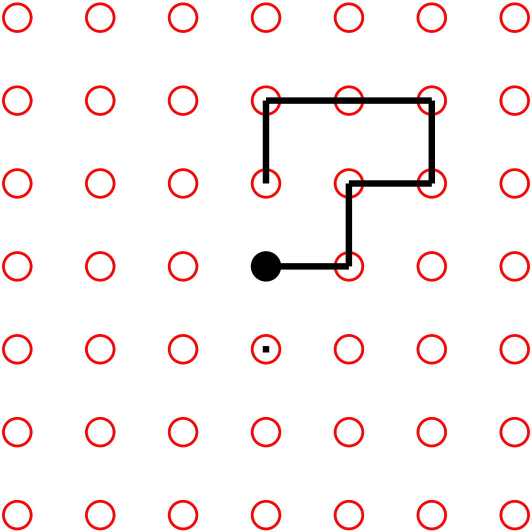
A self-avoiding walk of length 5 in 2D



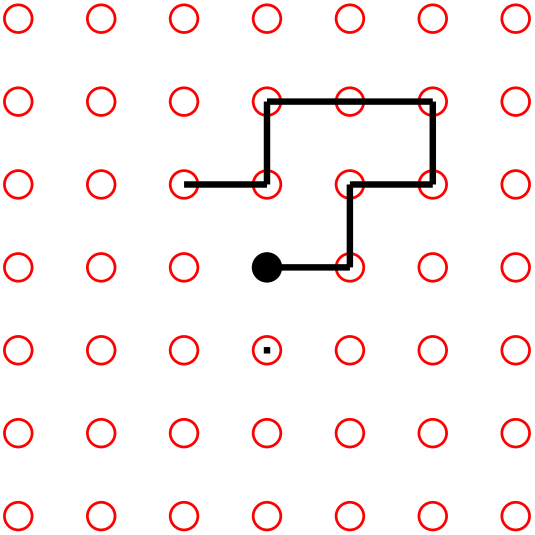
A self-avoiding walk of length 6 in 2D



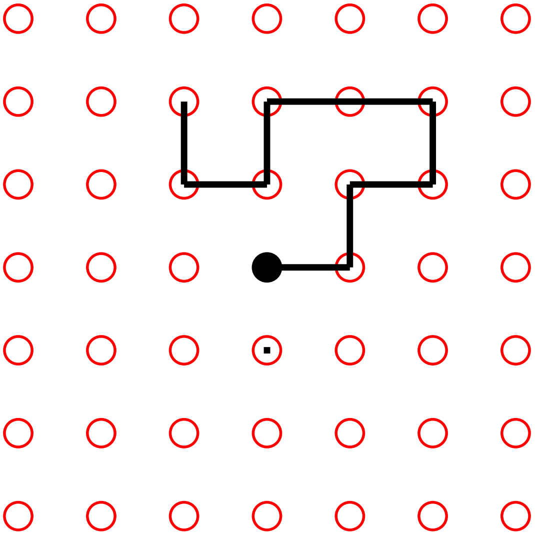
A self-avoiding walk of length 7 in 2D



A self-avoiding walk of length 8 in 2D

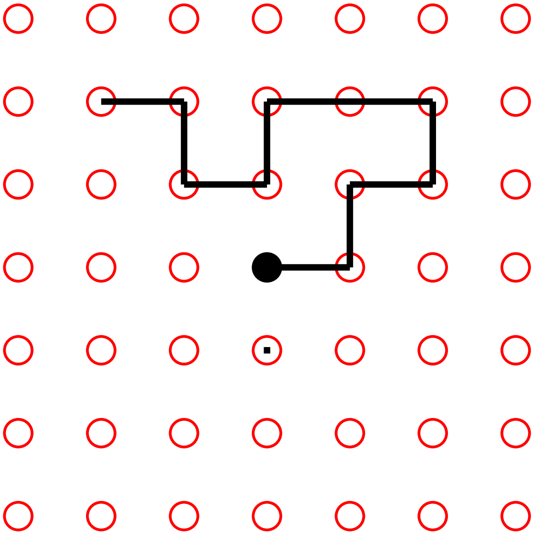


A self-avoiding walk of length 9 in 2D



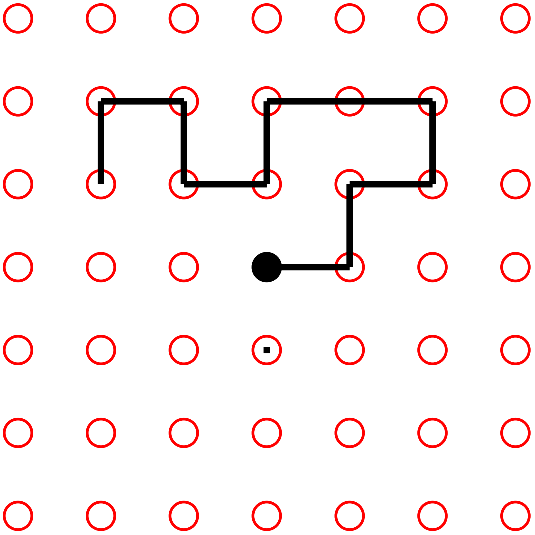
Universiteit Utrecht

A self-avoiding walk of length 10 in 2D

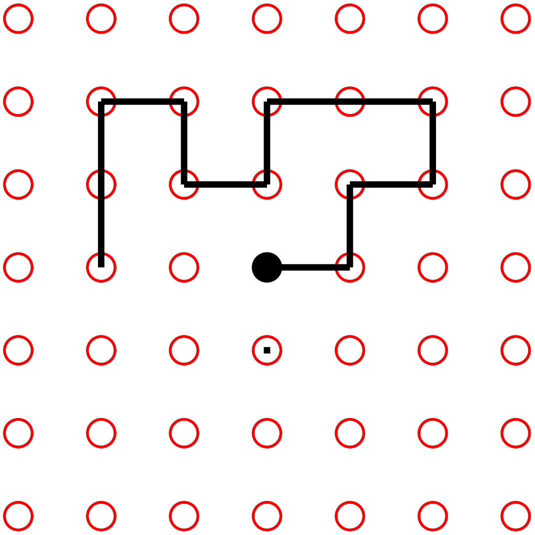


Universiteit Utrecht

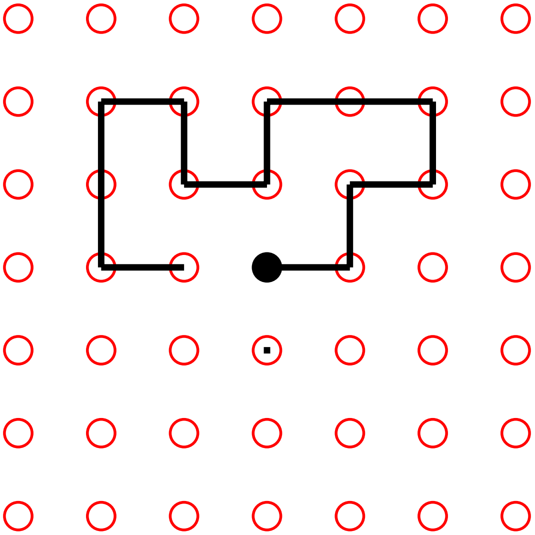
A self-avoiding walk of length 11 in 2D



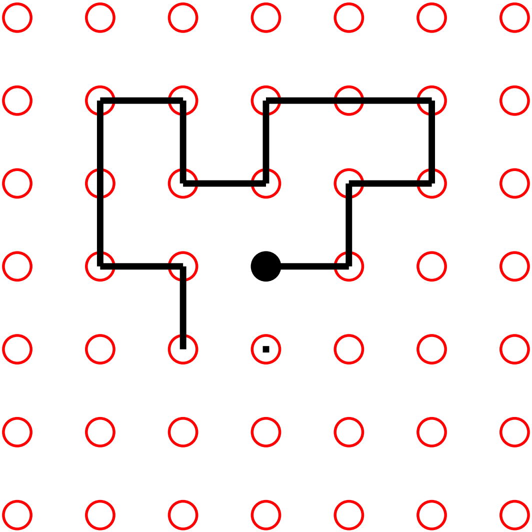
A self-avoiding walk of length 12 in 2D



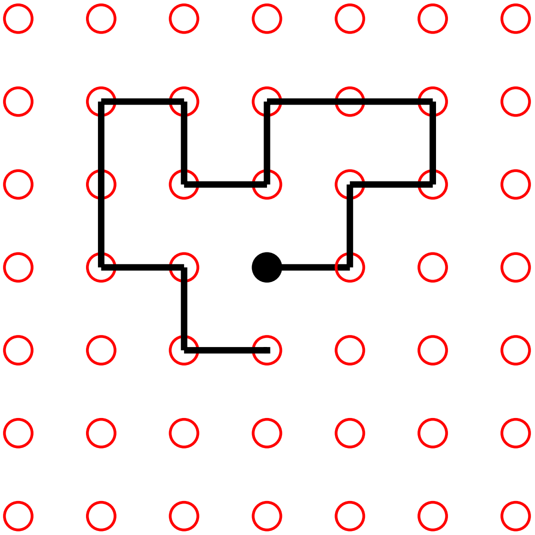
A self-avoiding walk of length 13 in 2D



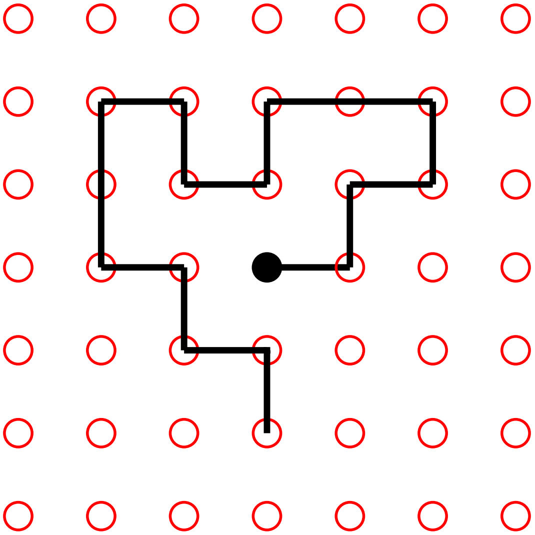
A self-avoiding walk of length 14 in 2D



A self-avoiding walk of length 15 in 2D

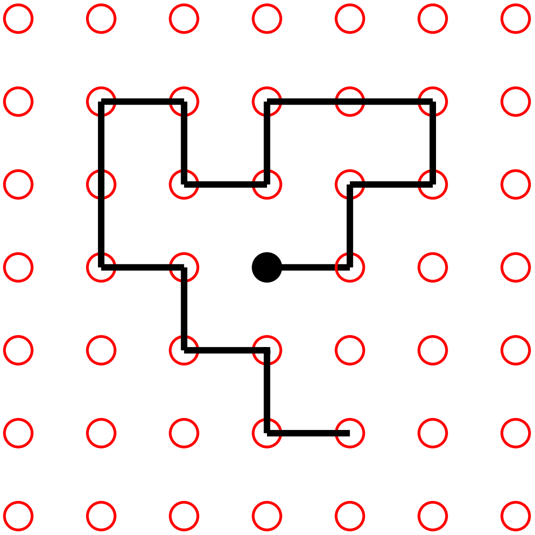


A self-avoiding walk of length 16 in 2D



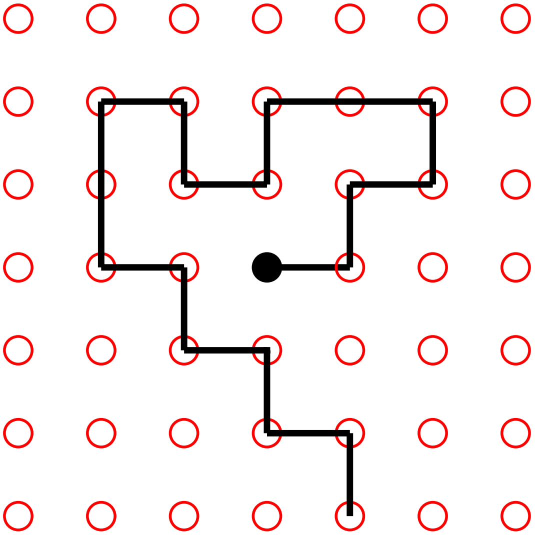
Universiteit Utrecht

A self-avoiding walk of length 17 in 2D

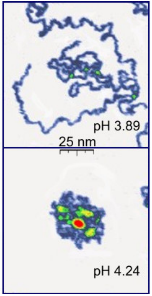


Universiteit Utrecht

A self-avoiding walk of length 18 in 2D



Why are self-avoiding walks useful?

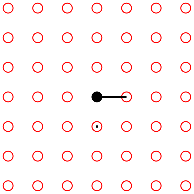


(Roiter and Minko 2007)

Poly(2-vinylpyridine) observed by Atomic Force Microscope.

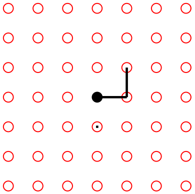
- The walk models a polymer, a long molecule, based on a carbon chain C–C–C–C$\cdots$C.
- A prime motivation is the DNA polymer.
- Self-avoiding because 2 carbon atoms cannot exist at the same location (the excluded-volume property).

How many self-avoiding walks are there?



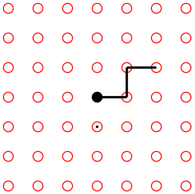
- In 2D: $Z_1 = 4$ walks.

How many self-avoiding walks of length 2?



- $Z_2 = 4 \times 3 = 12$ walks.

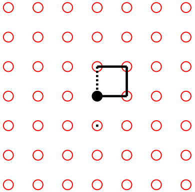
How many self-avoiding walks of length 3?



- $Z_3 = 4 \times 3 \times 3 = 36$ walks.

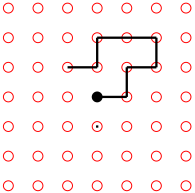
Universiteit Utrecht

How many self-avoiding walks of length 4?



- There are $4 \times 3 \times 3 \times 3 = 108$ possible walks, not all self-avoiding.
- In 8 cases, we return to the origin. So $Z_4 = 108 - 8 = 100$.
- Question: can you give an upper bound for Z_8?

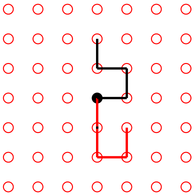
How many self-avoiding walks of length 8?



- There are $4 \times 3^7 = 8748$ possible walks, not all self-avoiding. So $Z_8 \leq 8748$.
- In general:

$$2^N \leq Z_N \leq 4 \times 3^{N-1}.$$

How many self-avoiding walks of length 8?



- We can concatenate two self-avoiding walks of length 4:

$$Z_8 \leq Z_4^2 = 10000.$$

- A sharper upper bound: the first red step cannot be the reverse of the last black step:

$$Z_8 \leq \frac{3}{4} Z_4^2 = 7500.$$

- $Z_8 = 5916$.

Recursive 2D SAW algorithm

$\text{SAW}(i, N)$
i = number of steps made, $0 \le i \le N$
N = desired length of the walk.
$(x_0, y_0), (x_1, y_1), \ldots, (x_{i-1}, y_{i-1})$ is self-avoiding.

if not visited (x_i, y_i) **then**
  if $i = N$ **then**
    print "$(x_0, y_0), \ldots (x_N, y_N)$ is a SAW"
  else
    visited(x_i, y_i) = **true**;
    $x_{i+1} = x_i + 1$; $y_{i+1} = y_i$; $\text{SAW}(i+1, N)$;
    $x_{i+1} = x_i - 1$; $y_{i+1} = y_i$; $\text{SAW}(i+1, N)$;
    $x_{i+1} = x_i$; $y_{i+1} = y_i + 1$; $\text{SAW}(i+1, N)$;
    $x_{i+1} = x_i$; $y_{i+1} = y_i - 1$; $\text{SAW}(i+1, N)$;
    visited(x_i, y_i) = **false**;

Bound for Z_{M+N}

- A self-avoiding walk of length $M+N$ can be cut into walks of lengths M and N, so

$$Z_{M+N} \leq Z_M \cdot Z_N.$$

- For $M = N$, we get $Z_{2N} \leq (Z_N)^2$.
- So $Z_N \geq (Z_{2N})^{1/2}$ for all N, giving

$$Z_1 \geq (Z_2)^{1/2} \geq (Z_4)^{1/4} \geq (Z_8)^{1/8} \geq \cdots$$

Convergence in 2D

- In the limit case for the 2D square lattice:

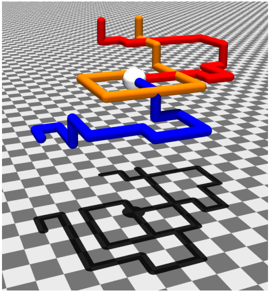
$$\lim_{N\to\infty}(Z_N)^{1/N} = \mu \approx 2.638, \qquad \text{so } Z_N \sim \mu^N$$

- $Z_{71} = 4,190,893,020,903,935,054,619,120,005,916$ (Jensen 2004).
- For 2D hexagonal lattice, $\mu = \sqrt{2+\sqrt{2}} \approx 1.848$ (Duminil-Copin and Smirnov 2010).

The world is 3D

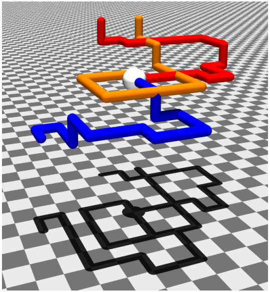
- Clisby, Liang, Slade (2007):
  $Z_{30} = 270,569,905,525,454,674,614$
- Nathan Clisby's animation of a self-avoiding walk of length $N = 1,048,575$.

Three self-avoiding walks of length 18 in 3D



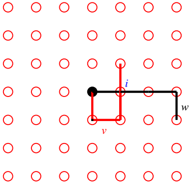
- Self-avoiding walks of length 18:
  red, orange, blue.
- How many pairs of self-avoiding walks can be glued together to give a self-avoiding walk of length 36?

Counting method based on intersection sets



- Intersections $a = (2, 0, 0)$, $b = (2, 3, 1)$: red/orange.
- Intersection $c = (0, -2, 0)$: blue/orange.
- There are 3 pairs of walks v/w with $v \neq w$.
- There are 3 intersections: remove the corresponding pair.
- Correct for over-removal: red/orange was removed twice, so 3-3+1 $=$ 1 pair remains, blue/red.

Counting pairs of walks



- A_i = set of pairs of self-avoiding walks (v, w) of length N that both pass through lattice point i.
- The lattice points have been numbered (excluding 0).
- The set $\bigcup_{i=1}^{n} A_i$ contains all pairs that intersect.

Length doubling

- There is a bijection between:
  - the self-avoiding walks of length $2N$
  - the non-intersecting pairs of walks of length N

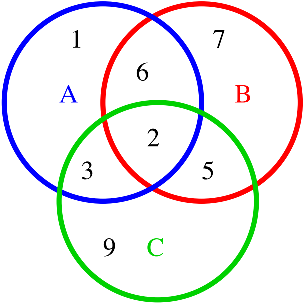
  because we can concatenate two walks.
- So we have:

$$Z_{2N} = Z_N^2 - \left| \bigcup_i A_i \right| .$$

- We can compute Z_{2N} efficiently by looking only at walks of length N.

Principle of inclusion–exclusion

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{i} |A_i| - \sum_{i<j} |A_i \cap A_j| + \sum_{i<j<k} |A_i \cap A_j \cap A_k| + \cdots$$
$$\cdots + (-1)^{n+1} |A_1 \cap A_2 \cdots \cap A_n|,$$



Length-doubling formula

- We obtain

$$Z_{2N} = Z_N^2 + \sum_{S \neq \emptyset} (-1)^{|S|} Z_N^2(S).$$

- $Z_N(S)$ is the number of self-avoiding walks of length N that pass through a subset S of lattice sites.

Computational complexity

- To compute $Z_N(S)$, we create all walks of length N.
- For each walk, we create all 2^N subsets of its N lattice sites and add 1 to their counter in a global data structure.
- Overall complexity

$$\mathcal{O}(2^N \cdot Z_N) = \mathcal{O}(2^N \mu^N) = \mathcal{O}((2\mu)^N).$$

  Much less than $\mathcal{O}(\mu^{2N}) = \mathcal{O}((\mu^2)^N)$, provided $\mu > 2$.
- 3D cubic lattice: $\mu = 4.68$, for $2N = 36$ savings of factor $(\mu/2)^{18} \approx 4.4 \times 10^6$.

Universiteit Utrecht

Tree data structure

- Walk $\{1, 7, 12, 49\}$ is stored along a path in the tree, where 1 is a child of the root and 49 is a leaf.
- The tree is defoliated, one layer of nodes with the same site number at a time.
- A layer s can be included so that $s \in S$, or excluded.
- Good site numbering (by increasing distance from 0) gives narrower trees.

Exploiting 48-fold symmetry of cubic lattice

- 8 reflections, such as $(x, y, z) \to (-x, y, z)$.
- 6 rotations, such as $(x, y, z) \to (y, z, x)$.
- Hence symmetry group of 48 operations.
- We use this through the numbering of the lattice.
- All ≤ 48 symmetrically equivalent lattice points get site numbers in the same range $[48t, 48t + 47]$.
- Hence, $s \equiv s' \Leftrightarrow \lfloor s/48 \rfloor = \lfloor s'/48 \rfloor$.

Split the computations

- Split computations for sets S into two:
  1. Sets $S = \{s_1, \ldots, s_k\}$ with $s_1 < s_2 < \cdots < s_k$, where $s_i \not\equiv s_k$ for all $i < k$.
  2. All other sets, i.e., those with at least one $s_i \equiv s_k$, where $i < k$.
- Case 1: only one highest site s_k from each equivalence class needs to be handled, saving a factor of up to 48.
- We choose s_k with $48 | s_k$: no equivalent $s_i < s_k$ in its walk, so no need to check equivalences.
- Case 2: fewer walks, since walk must pass through at least one other equivalent of the highest site.

Universiteit Utrecht

National supercomputer

- National supercomputer Huygens named after Christiaan Huygens (1629–1695).
- Located at SARA in Amsterdam.
- It has 3456 cores, with 2 cores per processor.
- Each core has a clock speed of 4.7 GHz.

Computing time

- Total computing speed 60 Teraflop/s $= 60 \times 10^{12}$ floating-point operations per second. Total electricity consumption 552 kW (excluding cooling).
- We used up to 192 cores, during 10 days, in total 50,000 CPU hours in Oct/Nov 2010.
- Estimated electricity bill: 5000 euro.

Parallelisation

$$Z_{2N} = Z_N^2 + \sum_{S \neq \emptyset} (-1)^{|S|} Z_N^2(S).$$

- We can split the work by size of the set S, computing one correction term for each size $|S|$.
- We can also split by the highest site s_k occurring in a set S.
- Or a larger subset $T \subset S$ that must occur.
- We used separate jobs, communicating with sockets, thus masquerading as a parallel program (and preventing some I/O as well).
- Fault tolerance is important, so various checks of results.

Number of self-avoiding walks in 3D

N	Z_N	Year Author
1	6	
2	30	
3	150	
4	726	
5	3 534	
6	16 926	1947 Orr, Univ. Glasgow
7	81 390	
8	387 966	
9	1 853 886	1959 Fisher, Sykes, King's College London
10	8 809 878	
11	41 934 150	
12	198 842 742	
13	943 974 510	
14	4 468 911 678	
15	21 175 146 054	
16	100 121 875 974	
17	473 730 252 102	
18	2 237 723 684 094	

Number of self-avoiding walks in 3D

N	Z_N	Year Author
19	10 576 033 219 614	
20	49 917 327 838 734	1987 Guttmann, Univ. Melbourne
21	235 710 090 502 158	1989 Guttmann
22	1 111 781 983 442 406	
23	5 245 988 215 191 414	1992 MacDonald et al, Nova Scotia
24	24 730 180 885 580 790	
25	116 618 841 700 433 358	
26	549 493 796 867 100 942	2000 MacDonald et al
27	2 589 874 864 863 200 574	
28	12 198 184 788 179 866 902	
29	57 466 913 094 951 837 030	
30	270 569 905 525 454 674 614	2007 Clisby, Liang, Slade, Univ Melbourn
31	1 274 191 064 726 416 905 966	
32	5 997 359 460 809 616 886 494	
33	28 233 744 272 563 685 150 118	
34	132 853 629 626 823 234 210 582	
35	625 248 129 452 557 974 777 990	
36	2 941 370 856 334 701 726 560 670	2011 Schram, Barkema, Bisseling

Publication

Journal of Statistical Mechanics: Theory and Experiment
An IOP and SISSA journal

Exact enumeration of self-avoiding walks

R D Schram[1,2], **G T Barkema**[1] **and R H Bisseling**[2]

[1] Institute for Theoretical Physics, Utrecht University, PO Box 80195, 3508 TD Utrecht, The Netherlands
[2] Mathematical Institute, Utrecht University, PO Box 80010, 3508 TA Utrecht, The Netherlands
E-mail: raouldschram@gmail.com, g.t.barkema@uu.nl and R.H.Bisseling@uu.nl

Received 12 April 2011
Accepted 9 June 2011
Published 27 June 2011

Online at stacks.iop.org/JSTAT/2011/P06019
doi:10.1088/1742-5468/2011/06/P06019

Abstract. A prototypical problem on which techniques for exact enumeration are tested and compared is the enumeration of self-avoiding walks. Here, we show an advance in the methodology of enumeration, making the process thousands or millions of times faster. This allowed us to enumerate self-avoiding walks on the simple cubic lattice up to a length of 36 steps.

Keywords: loop models and polymers, critical exponents and amplitudes (theory), exact results

J. Stat. Mech. (2011) P06

Universiteit Utrecht

Possible appplication

- Biopolymers like DNA, proteins are the fundaments of life.
- Polymers are of great industrial importance: plastics (DSM), synthetic fibres (Akzo).
- Insight into polymer behaviour:
    - viscosity
    - mean squared distance

$$P_N/Z_N \sim N^{2\nu}.$$

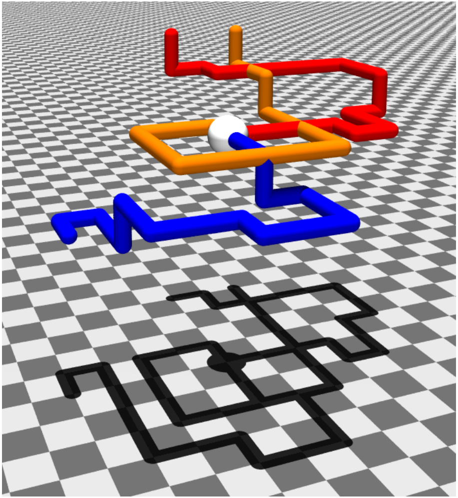
The value $\nu \approx 0.588$ can be computed with the simplest possible lattice model, SAWs on a cubic lattice.

Conclusion and outlook

- Our new enumeration method, length doubling, reduces the asymptotic complexity of counting self-avoiding walks from 4.68^N to 3.06^N.
- We improved the current world record from 30 to 36 steps, using symmetry, parallel computing, and a special lattice numbering scheme.
- Length doubling can be used for all kinds of problems:
  - body-centred cubic lattice
  - 4D hypercubic lattice
  - self-avoiding polygons
- Software package Sawdoubler to be released soon.

Universiteit Utrecht

Thanks



Thank you!