Lower and Upper Bounds for the Time Constant of First-Passage Percolation

Sven Erick Alm Robert Parviainen

June 20, 2001

Abstract

We present improved lower and upper bounds for the time constant of first-passage percolation on the square lattice. For the case of lower bounds, a new method, using the idea of a transition matrix, has been used. Numerical results for the exponential and uniform distributions are presented. A simulation study is included, which results in new estimates and improved upper confidence limits of the time constants.

1 Introduction

1.1 Percolation

The percolation process was introduced as a mathematical model for the spread of a fluid through a random medium by Broadbent and Hammersley [2]. The term fluid has a broad interpretation and can for instance mean a liquid, an epidemic or a particle. The medium is represented by a connected graph, with a (possible finite) countable set of *vertices (sites)* and *edges (bonds)* joining the vertices.

Broadbent and Hammersley considered bond and site percolation, where each edge or vertex is open or closed for the fluid, with given probability.

In *first–passage percolation*, introduced by Hammersley and Welsh [3], each edge is open, and associated with a random variable, representing the time for the fluid to pass the bond.

1.2 First-passage percolation

We will study the time constant for first-passage percolation on the graph given by the square lattice. The vertices are the points $(x, y) \in \mathbb{Z}^2$. The edges are the lines of length 1 joining adjacent points, and with each edge e we associate a random variable X_e. We assume the variables X_e to be non-negative, independent, identically distributed with finite mean. We will be interested in the first-passage time to the line $x = n$, starting from the origin. (We will, with a slight abuse of notation, denote the line $\{(x, y) \in \mathbb{Z}^2 : x = n\}$ by $x = n$.)

A walk on the lattice is an alternating sequence $v_0, e_1, v_1, e_2, \ldots, e_n, v_n$ of vertices and edges. The walk is self-avoiding if all vertices are distinct. $(x(\gamma), y(\gamma))$ will be used to denote v_n, the endpoint for some walk γ, and $|\gamma|$ its length. We will need some notation for sets of walks. Γ is the set of all self-avoiding walks starting from the origin. There are some subsets of Γ that will be of use; Γ_n is the subset consisting of all self-avoiding walks of length n, $\Gamma(n,m)$ consists of all self-avoiding walks that end in (n,m), or more general $\Gamma(R)$ for walks that end in some non-empty subset $R \subset \mathbb{Z}^2$. $F(n) = |\Gamma_n|$ will denote the number of self-avoiding walks of length n.

For a given walk γ we define the passage time $S_\gamma = \sum_{e\in\gamma} X_e$. We also define the first-passage times $T(R) = \inf_{\gamma\in\Gamma(R)} S_\gamma$, the first-passage time to the set R, starting at the origin, and $T_G = \inf_{\gamma\in G} S_\gamma$, the first-passage time over the subset $G \subset \Gamma$.

We will on occasion use the term *infected* for sets of edges and vertices that may be reached from the origin in a given time, that is, the edge e or vertex v is infected at time t if there exists a path γ with $v_0 = (0,0)$, and $S_\gamma \leq t, e \in \gamma$ or $v \in \gamma$.

1.3 The time constant

The time constant of first-passage percolation is defined as the limit of the, by n, normalized first-passage time from the origin to the line $x = n$. However, one can show (see the book by Smythe and Wierman [5]) that the same limit also arises if we only consider cylinder restricted walks. This allows us to use subadditivity, which will be important for the upper bounds.

Let C_{mn} be the subset of Γ consisting of the walks with end-vertex on the line $x = n - m$, and with all other vertices inside the cylinder $0 \leq x < n - m$.

Now, define s_{mn} as the first-passage time from $(m,0)$ to the line $x = n$ over walks in C_{mn}. Note that $s_{mn} \overset{d}{=} T_{C_{mn}}(x = n - m)$. The time constant τ is defined as

$$\tau = \lim_{n\to\infty} \frac{s_{0n}}{n}.$$

The function Es_{0n} is subadditive (for a proof see [5]), which implies, for $n \geq 1$,

$$\tau = \inf_{k\geq 1} \frac{Es_{0k}}{k} \leq \frac{Es_{0n}}{n},$$

a key fact for our upper bounds.

The time constant measures the speed with which the fluid spreads, and is unknown for all non-trivial distributions. Previous lower and upper bounds are given by Janson [4] and Smythe and Wierman [5].

2 Lower bounds

In [4], Janson derives a method for calculating lower bounds for the time constant. The method is based on counting finite self-avoiding walks. The basic idea is to consider too many infinite walks, which is done by joining short self-avoiding walks at their endpoints.

Time constants may be defined in general directions as well, and Janson's method treats all directions simultaneously. Define the set

$$N^* = \{(a,b) \in \mathbb{R}^2 | \lim_{(m,n)\to\infty} P(T(m,n) \leq am + bn) = 0\}.$$

The (horizontal) time constant τ may then be defined by

$$\tau = \sup_{a \in \mathbb{R}^+} \{(a, 0) \in N^*\}.$$

In the following theorem from [4], we use the generating function

$$F_n(s,t) = \sum_{\gamma \in \Gamma_n} s^{x(\gamma)} t^{y(\gamma)},$$

and the moment generating function for X_e,

$$\psi(\nu) = E(e^{-\nu X_e}),$$

to achieve criterias for a to belong to N^*, and thus lower bounds for the time constant τ.

Theorem 2.1 (Janson). *If* $F_n\left(e^{a\nu}, e^{b\nu}\right)^{\frac{1}{n}} < \frac{1}{\psi(\nu)}$ *for some* $\nu > 0$ *and* $n \geq 1$, *then* $(a, b) \in N^*$.

For a proof, see [4]. Since $F_n(e^{a\nu}, 1)$ can be computed for a given n, theorem 2.1 allows us to compute lower bounds for τ, by finding a and ν such that the assumptions are satisfied.

In a note in [4], Janson points out that theorem 2.1 may be improved by realizing that F_n may be substituted by a smaller sum, namely the original sum, minus, for each x-coordinate, the walks that starts in that direction so that the resulting sum is maximized. This comes from the fact that when joining two walks, for the resulting walk to be self-avoiding there are at most three possible directions for the first step of the second walk.

However, as we shall see, it is possible to join short self-avoiding walks in a more clever may, to reduce the number of non-self-avoiding walks in the limit, thereby improving the lower bounds.

2.1 An improved method

In [1], Alm uses a version of a method introduced by Wakefield [6] to find upper bounds for the connective constant of self-avoiding walks. A modified version of this method can be used to improve the bounds for the time constant as well.

Again, we count the number of self-avoiding walks of a given length n, but this time we also remember the m first and last steps, as well as the m:th last x-coordinate. This gives us a $F(m) \times F(m)$ matrix $\mathbf{B}$, see below. Each element in the matrix is the sum of two polynomials, one in s and one in s^{-1}, both of degree at most $n - m$. The largest eigenvalue of a matrix $\mathbf{A}$ will be denoted by $\lambda_1(\mathbf{A})$. $\mathbf{1}$ will denote a row vector of suitable length. We will use the norm $\|\mathbf{A}\| = \sum_i \sum_j a_{ij} = \mathbf{1}\mathbf{A}\mathbf{1}'$.

Let m be fixed, let the walks of Γ_m be denoted $\gamma_1, \gamma_2, \ldots, \gamma_{F(m)}$, and let $a_{ij}^{(n)}(k)$ be the number of self-avoiding walks of length n that starts with γ_i and ends with a translation of γ_j, $|\gamma_i| = |\gamma_j| = m$, and has m:th last x-coordinate k. Define the matrix $\mathbf{A}$ by

$$\mathbf{A}^{(n)}(k) = \left(a_{ij}^{(n)}(k)\right), 1 \leq i, j \leq F(m),$$

and the matrix $\mathbf{B}$ by

$$\mathbf{B}^{(n)}(s) = \sum_{k=-n+m}^{n-m} \mathbf{A}^{(n)}(k)s^k = \left(b_{ij}^{(n)}(s)\right), 1 \leq i,j \leq F(m).$$

Now, every self-avoiding walk of length $2n-m$ that starts with γ_i and ends with a translation of γ_j, having m:th last x-coordinate k may be constructed by joining two self-avoiding walks of length n, the first starting with γ_i, and ending with a translation of γ_l, with m:th last x-coordinate r, the second starting with γ_l, and ending with a translation of γ_j, with m:th last x-coordinate $k-r$. Their composition $\gamma_1 \circ \gamma_2$ will then have m:th last x-coordinate k. Therefore

$$a_{ij}^{(2n-m)}(k) \leq \sum_{r=-n+m}^{n-m} \sum_{l=1}^{F(m)} a_{il}^{(n)}(r)a_{lj}^{(n)}(k-r) = \sum_{r=-n+m}^{n-m} \left(\mathbf{A}^{(n)}(r)\mathbf{A}^{(n)}(k-r)\right)_{ij},$$

and, for $\mathbf{B}$,

$$\begin{aligned}
b_{ij}^{(2n-m)}(s) &= \sum_{k=-2n+2m}^{2n-2m} a_{ij}^{(2n-m)}(k)s^k \\
&\leq \sum_{k=-2n+2m}^{2n-2m} s^k \sum_{r=-n+m}^{n-m} \sum_{l=1}^{F(m)} a_{il}^{(n)}(r)a_{lj}^{(n)}(k-r) \\
&= \sum_{k=-2n+2m}^{2n-2m} \sum_{r=-n+m}^{n-m} \sum_{l=1}^{F(m)} a_{il}^{(n)}(r)s^r a_{lj}^{(n)}(k-r)s^{k-r} \\
&= \sum_{l=1}^{F(m)} \sum_{r=-n+m}^{n-m} a_{il}^{(n)}(r)s^r \left(\sum_{k=-2n+2m}^{2n-2m} a_{lj}^{(n)}(k-r)s^{k-r}\right) \\
&= \sum_{l=1}^{F(m)} b_{il}^{(n)}(s)b_{lj}^{(n)}(s) = \left(\mathbf{B}^{(n)}(s)\mathbf{B}^{(n)}(s)\right)_{ij}.
\end{aligned}$$

In the same way we get

$$b_{ij}^{(k(n-m)+m)}(s) \leq \left(\mathbf{B}^{(n)}(s)\right)_{ij}^{k}. \tag{1}$$

Let the $F(m) \times 1$ column vector $\mathbf{R}^{(m)}$ be defined by the relation

$$F_{2n-m}(s,1) = \sum_{\gamma \in \Gamma_{2n-m}} s^{x(\gamma)} = \mathbf{1}\mathbf{B}^{(2n-m)}(s)\mathbf{R}^{(m)}.$$

$\mathbf{R}^{(m)}$ is just a correction vector, depending only on m, with elements from the set $\{s^{-m}, s^{-(m-1)}, \ldots, s^{m-1}, s^m\}$, due to the fact that we are using the m:th last x-coordinate. By (1), we then get, for all $k \geq 1$,

$$F_{k(n-m)+m}(s,1) = \mathbf{1}\mathbf{B}^{(k(n-m)+m)}(s)\mathbf{R}^{(m)} \leq \mathbf{1}\left(\mathbf{B}^{(n)}(s)\right)^k \mathbf{R}^{(m)}. \tag{2}$$

For $s > 1$ (we will use $s = e^{a\nu}$ in Theorem 2.2), we have $\max_{r\in\mathbf{R}^{(m)}} \mathbf{R}^{(m)} \leq s^m$ and $\min_{r\in\mathbf{R}^{(m)}} \mathbf{R}^{(m)} \geq s^{-m}$, and

$$\mathbf{1}\left(\mathbf{B}^{(n)}(s)\right)^k \mathbf{R}^{(m)} \leq \mathbf{1}\left(\mathbf{B}^{(n)}(s)\right)^k \mathbf{1}' s^m = \| \left(\mathbf{B}^{(n)}(s)\right)^k \| s^m$$
$$\mathbf{1}\left(\mathbf{B}^{(n)}(s)\right)^k \mathbf{R}^{(m)} \geq \mathbf{1}\left(\mathbf{B}^{(n)}(s)\right)^k \mathbf{1}' s^{-m} = \| \left(\mathbf{B}^{(n)}(s)\right)^k \| s^{-m}.$$

Now let $k \to \infty$. Since

$$\lim_{k\to\infty} \left(s^{-m}\right)^{\frac{1}{k(n-m)+m}} = \lim_{k\to\infty} \left(s^{m}\right)^{\frac{1}{k(n-m)+m}} = 1$$

we get, by the Power method,

$$\begin{aligned} \lim_{k\to\infty} \left(1\left(\mathbf{B}^{(n)}(s)\right)^k \mathbf{R}^{(m)}\right)^{\frac{1}{k(n-m)+m}} &= \lim_{k\to\infty} \left(\| \left(\mathbf{B}^{(n)}(s)\right)^k \|\right)^{\frac{1}{k(n-m)+m}} \\ &= \left(\lambda_1\left(\mathbf{B}^{(n)}(s)\right)\right)^{\frac{1}{n-m}}. \end{aligned}$$

And finally, by (2),

$$\left(F_{k(n-m)+m}(s,1)\right)^{\frac{1}{k(n-m)+m}} \leq \left(\lambda_1\left(\mathbf{B}^{(n)}(s)\right)\right)^{\frac{1}{n-m}}.$$

With the help of theorem 2.1 we have thus proved

Theorem 2.2. *If* $\left(\lambda_1\left(\mathbf{B}^{(n)}(e^{a\nu})\right)\right)^{\frac{1}{n-m}} < \frac{1}{\psi(\nu)}$ *for some* $\nu > 0$*, then* $(a, 0) \in N^*$.

We thus have a criteria for lower bounds. If the largest eigenvalue of the matrix $\mathbf{B}$ in the point $(e^{a\nu}, 1)$, to the power $(n-m)^{-1}$, is less than $\frac{1}{\psi(\nu)}$ then a is a lower bound for the time constant.

2.1.1 Reducing B

The matrix $\mathbf{B}$ is actually unnecessary large. We can use a reduced $K(m)\times K(m)$ matrix $\tilde{\mathbf{B}} = (\tilde{b}_{ij})$, where $K(m)$ is the number of equivalence classes of walks of length m. We consider two walks equivalent if one can be mapped on the other by reflection in the x-axis. Every walk except those two that only uses horizontal steps (to $(m, 0)$ and $(-m, 0)$) has exactly one equivalent walk, so that $K(m) = \frac{F(m)}{2} + 1$. Denote the walks in the equivalence class $\tilde{\gamma}_1, \tilde{\gamma}_2, \ldots, \tilde{\gamma}_{K(m)}$. Let $\tilde{\gamma}_1$ and $\tilde{\gamma}_{K(m)}$ be the walks of length m that goes straight along the x-axis to $(m, 0)$ and $(-m, 0)$. We define $\tilde{b}_{ij} = b_{ij}$ for $j = 1$ and $j = K(m)$, and $\tilde{b}_{ij} = b_{ij} + b_{ij'}$, where γ_j and $\gamma_{j'}$ are equivalent, otherwise. The following theorem shows that we can use $\tilde{\mathbf{B}}$ instead of $\mathbf{B}$.

Theorem 2.3. $\lambda_1(\tilde{\mathbf{B}}) = \lambda_1(\mathbf{B})$

Proof. Let $\tilde{\lambda}_1 = \lambda_1(\tilde{\mathbf{B}})$, with corresponding right eigenvector $\tilde{\mathbf{h}}$. Define $\mathbf{h}_{F(m)\times 1}$ by

$$h_j = \tilde{h}_s \text{ if } \gamma_j \text{ is equal to or equivalent to } \tilde{\gamma}_s.$$

If γ_i is equivalent to $\tilde{\gamma}_r$, then

$$\sum_{j=1}^{F(m)} b_{ij}h_j = b_{i1}\tilde{h}_1 + \sum_{s=2}^{K(m)-1} (b_{is} + b_{is'})\tilde{h}_s + b_{iK(m)}\tilde{h}_{K(m)} = \sum_{s=1}^{K(m)} \tilde{b}_{rs}\tilde{h}_s = \tilde{\lambda}_1\tilde{h}_r,$$

so $\tilde{\lambda}_1$ is an eigenvalue for $\mathbf{B}$. It remains to show that this is the largest eigenvalue, which is easily done by the Power method. In the recursion $\mathbf{v}^{(n)} = \frac{\mathbf{B}\mathbf{v}^{(n-1)}}{c_n}$, choose $\mathbf{v}(0) = \mathbf{h}$, and we get $c_n = \lambda_1(\tilde{\mathbf{B}})$ for all $n \geq 1$. Therefore $\lambda_1(\mathbf{B}) = \lim_{n\to\infty} c_n = \lambda_1(\tilde{\mathbf{B}})$. □

m / n	0	1	2	3	4	5	6	7
2	0.286787							
3	0.289423	0.298253						
4	0.292680	0.299266	0.299631					
5	0.293828	0.299473	0.299789					
6	0.295207	0.299685	0.299968	0.300186				
7	0.295900	0.299780	0.300025	0.300201				
8	0.296518	0.299860	0.300077	0.300223	0.300245			
9	0.296934	0.299913	0.300106	0.300233	0.300252			
10	0.297292	0.299955	0.300130	0.300242	0.300258	0.300272		
11	0.297561	0.299988	0.300147	0.300247	0.300261	0.300274		
12	0.297794	0.300015	0.300161	0.300251	0.300264	0.300275	0.300279	
13	0.297984	0.300037	0.300172	0.300254	0.300266	0.300276	0.300279	
14	0.298150	0.300056	0.300181	0.300257	0.300267	0.300277	0.300280	0.300282
15	0.298291	0.300072	0.300189	0.300259	0.300268	0.300277	0.300280	0.300282
16	0.298416	0.300086	0.300196	0.300261	0.300269	0.300277	0.300281	0.300282
17	0.298524	0.300098	0.300202	0.300263	0.300270	0.300278	0.300281	0.300282
18	0.298621	0.300109	0.300205	0.300265	0.300271	0.300278	0.300281	0.300282
19	0.298708	0.300119	0.300207	0.300266	0.300272	0.300279	0.300281	0.300282
20	0.298786	0.300127	0.300214	0.300267	0.300273	0.300279	0.300281	0.300282
21	0.298851	0.300135	0.300217	0.300268	0.300274	0.300280	0.300281	0.300282
22	0.298921	0.300142	0.300222	0.300268	0.300275	0.300280	0.300281	0.300282

Table 1: Lower bounds for the time constant, exponential distribution

2.2 Numerical results

The results are summarized in Tables 1 and 2. In Table 1, the entries for $m = 0$ correspond to the improved version of theorem 2.1. The previous lower bounds, given in [4], where 0.29842 for the exponential distribution, by the improved version of theorem 2.1 with $n = 16$, and 0.24294 for the uniform, by a result not included here, which only uses the moment generating function $\psi(\nu)$. We improve these bounds already with $n = 4, m = 1$. The best lower bounds obtained here are 0.300282 and 0.243666, respectively.

The limitation in n is the time available. When going from n to $n + 1$ the time needed increases roughly by a factor 3. In m, it is the amount of available internal computer memory that is the limiting factor, increasing by a factor 9

m / n	1	2	3	4	5	6	7
3	0.242941						
4	0.243325	0.243479					
5	0.243399	0.243518					
6	0.243468	0.243572	0.243643				
7	0.243500	0.243589	0.243647				
8	0.243526	0.243604	0.243653	0.243658			
9	0.243543	0.243613	0.243655	0.243660			
10	0.243557	0.243620	0.243657	0.243661	0.243664		
11	0.243568	0.243625	0.243658	0.243662	0.243665		
12	0.243577	0.243629	0.243659	0.243662	0.243665	0.243666	
13	0.243584	0.243633	0.243659	0.243663	0.243665	0.243666	
14	0.243591	0.243635	0.243659	0.243663	0.243665	0.243666	0.243666
15	0.243596	0.243638	0.243659	0.243663	0.243665	0.243666	0.243666
16	0.243601	0.243640	0.243661	0.243663	0.243665	0.243666	0.243666
17	0.243605	0.243641	0.243661	0.243663	0.243665	0.243666	0.243666
18	0.243608	0.243643	0.243661	0.243664	0.243666	0.243666	0.243666
19	0.243614	0.243644	0.243662	0.243664	0.243666	0.243666	0.243666
20	0.243614	0.243646	0.243662	0.243664	0.243666	0.243666	0.243666
21	0.243619	0.243647	0.243662	0.243664	0.243666	0.243666	0.243666
22	0.243619	0.243648	0.243662	0.243664	0.243666	0.243666	0.243666

Table 2: Lower bounds for the time constant, uniform distribution

for each step in m. The time needed also increases in m, due to the fact that we must find the largest eigenvalue for a matrix that is roughly three times as wide. For $m = 7$ (the largest m we used), we need around 180 MB of RAM, and the computations for the largest n took a couple of days on a standard microcomputer. However, as can be seen in the Tables 1 and 2, there is little gained by doing larger calculations, especially by increasing n.

3 Upper bounds

3.1 The method

In principle, it is easy to find upper bounds. We only have to calculate the expected first-passage time for some small subset of walks. We formalize this in the following proposition. Let C_{0n}, as before, be the subset of Γ, consisting of walks with its endpoint on the line $x = n$, and with all other vertices inside the cylinder $0 \leq x < n$.

Proposition 3.1. *$\tau \leq \frac{E(T_G(x=n))}{n}$, where $T_G(x = n)$ is the first-passage time from the origin to the line $x = n$ over walks in $G \subset C_{0n}$.*

Proof. First, if $G_1 \subset G_2$, then

$$T_{G_1} = \inf_{\gamma \in G_1} S_\gamma \geq \inf_{\gamma \in G_2} S_\gamma = T_{G_2},$$

so that,

$$\tau \leq \frac{E(s_{0n})}{n} \leq \frac{E(T_G(x=n))}{n}.$$

□

Therefore, if we can compute $t = \frac{E(T_G(x=n))}{n}$ for some set G, t will be an upper bound for the time constant τ.

3.2 The exponential distribution

The nice properties of the exponential distribution makes it easy to compute upper bounds. We will consider rectangular subsets, with the origin on the left side. If the rectangle has M lines in the y-direction, the origin is placed at (the integer part of) $M/2$.

We start with only the origin infected. The expected first-passage time is then rewritten with the law of total probability, conditioning on the first step, giving an expression in terms of expected first-passage times with two infected vertices. All uninfected edges, adjacent to some infected vertex, now have, by the lack of memory property, still the exponential mean 1 distribution, and we can continue in this way, successively conditioning on the next step, until we get explicit expressions for the expected first-passage times. Let $T_{k_1 \ldots k_l}$ denote the first-passage time with $l+1$ infected vertices, with k_1 as the first edge used, k_2 as the second, and so on. In the same way, let $N_{k_1 \ldots k_l}$ be the number of uninfected edges, adjacent to some infected vertex. Finally, let $X_{[1]}$ denote the minimum of $\{X_1, \ldots, X_N\}$. Then,

$$\begin{aligned}
E(T) &= \sum_{k_1=1}^{N} E(T|X_{k_1} = X_{[1]}) P(X_{k_1} = X_{[1]}) \\
&= \sum_{k_1=1}^{N} E(X_{k_1} + T_{k_1}|X_{k_1} = X_{[1]}) P(X_{k_1} = X_{[1]}) \\
&= P(X_1 = X_{[1]}) \left[\sum_{k_1=1}^{N} E(X_{k_1}|X_{k_1} = X_{[1]}) + E(T_{k_1}|X_{k_1} = X_{[1]}) \right] \\
&= \frac{1}{N} + \frac{1}{N} \sum_{k_1=1}^{N} E(T_{k_1}), \\
E(T_{k_1}) &= \frac{1}{N_{k_1}} + \frac{1}{N_{k_1}} \sum_{k_2=1}^{N_{k_1}} E(T_{k_1 k_2}),
\end{aligned}$$

and so on, until every vertex but those on the right borderline are infected. Then the expected first-passage time is $1/M$. By back substitution, we then find the wanted expected first-passage time. Note that the expected first-passage time only depends on the set of infected vertices. The order of infection, and the presence or not of edges between two infected vertices does not matter. This can be used to significantly reduce the number of cases by equating all configurations with the same set of infected vertices.

To line M	x=2	x=3	x=4	x=5	x=6	x=7
2	0.722222	0.709684	0.703017	0.698968	0.696262	0.694328
3	0.606463	0.597846	0.594009	0.591976	0.590729	0.589879
4	0.582480	0.567714	0.559863	0.555199	0.552170	0.550056
5	0.562535	0.544247	0.534522	0.528867	0.525309	0.522908
6	0.558398	0.537586	0.525706	0.518352	0.513503	0.510129
7	0.554433	0.531414	0.517790	0.509168	0.503425	
8	0.553678	0.529793	0.515191	0.505626		
9	0.552931	0.528208	0.512692			
10	0.552801	0.527828				

Table 3: Upper bounds, exponential distribution.

A PASCAL program was written with the purpose to automatically find these equations. Since the rational numbers involved soon have very large numerators and denominators, they where rounded to double precision reals, but *always upward*, to assure us that we really get an upper bound. The results are summarized in Table 3. The best upper bound found here is 0.503425. Smythe and Wierman [5] calculated the expected first-passage time to the line $x = 1$, which gives the upper bound 0.59726.

3.3 The uniform distribution

Without the lack of memory property, things are now a bit more complicated. We must keep track of the order of infection, and the time of each infection. However, the same approach still works, but in smaller scale.

Assume that l edges has been infected, in the order $k_1, k_2, \ldots, k_l$, with associated edge variables $X_{k_i} = x_{k_i}$. Also assume that there are n uninfected edges incident to at least one infected vertex. Only edges whose sole incident infected vertex is the last infected will be $U(0,1)$ distributed. The other edges distributions will still be uniform, but on $(0, z_i)$, where $z_i = 1 - \sum_{j \in S_i} x_{k_j}$, and S_i is a subset of $\{1, \ldots, l\}$. For a given configuration, the subsets S_i can be found by inspection, by considering the order of infection.

Let $f_{k_1 k_2 \ldots k_l}$ denote the joint density function for the n uninfected edges. The joint conditional density, given that the next edge infected is k_{l+1}, is denoted by $f_{k_1 \ldots k_l | k_1 \ldots k_{l+1}}$. We can express this in terms of the unconditional densities as $\frac{f_{k_1 \ldots k_l}}{f_{k_1 \ldots k_{l+1}}}$.

Let $T_{k_1 k_2 \ldots k_l}$ denote the first-passage time if we start with these edges infected, and $H_{k_1 k_2 \ldots k_l}$ the expected time until the next vertex gets infected. So $H_{k_1 k_2 \ldots k_l}$ is the expectation of the minimum of the m edges that are incident to *exactly* one infected vertex,

$$H_{k_1 \ldots k_l} = \mathrm{E}(\min\{X_1, X_2, \ldots, X_m\}),$$

where $X_i \in \mathrm{U}(0, z_i)$.

The expected first-passage time may be decomposed as

$$E(T_{k_1 \ldots k_l}) = H_{k_1 \ldots k_l} + \sum_{k_{l+1}} \int_0^a E(T_{k_1 \ldots k_{l+1}}) \frac{f_{k_1 \ldots k_l}}{f_{k_1 \ldots k_{l+1}}} dx_{k_{l+1}},$$

where the sum is over the m edges that are incident to exactly one infected vertex, and $a = \min\{z_1, \ldots, z_m\}$. An important fact is that every integrand will be a polynomial in the variable of integration. Otherwise the number of integrals necessary to compute, even for very small areas, would make the calculations intractable. Still, we have so far only been able to perform the calculations for the rectangle bounded by the lines $x = 0, x = 2, y = 2, y = -1$. This gives the upper bound 0.403141. The best upper bound in Smythe and Wierman [5] is 0.425 in this case.

It is possible to implement the rules for these equations as a computer program, which generate as output the equations as a Maple file, which is read and processed (in Maple). The calculations has been cross-checked by simulations, and by other methods for smaller areas.

4 Simulation

A simulation study of the cylinder restricted process s_{0n} has been performed. The purpose of this study is to estimate the time constant, and to obtain upper confidence limits for the time constant. We also use the simulation program to cross-check the computations for the upper bounds. The simulation program generates a number of independent realizations of the process, and outputs means of first-passage times to each line from $x = 1$ up to some predetermined line $x = n$, as well as the y-coordinate for the first hit on the line $x = n$ for each simulation. To avoid unnecessary and time consuming programming the walks are restricted in y-direction as well. This is not a problem since it is easy to choose the restriction such that the probability that a walk will be restricted in the y-direction is virtually zero, which can in part be confirmed by the data of hitting points on the line $x = n$.

4.1 Upper confidence limits

The first-passage times to the line $x = n$ are used to compute upper confidence limits for the time constant. Since

$$\tau \leq E\left(\frac{s_{0n}}{n}\right),$$

an upper confidence limit for $E\left(\frac{s_{on}}{n}\right)$ will also be an upper confidence limit for τ, with higher level of confidence. The confidence limits are constructed in the usual way, using the Gaussian 95% quantile.

4.1.1 Numerical results

For the exponential distribution, 5000 realizations of the process $s_{0,500}$ were generated. The estimated expected first-passage time to the line $x = 500$ was 204.786, with standard deviation 2.99526, giving an 95% upper confidence limit of 0.409711.

Since the simulations for the uniform distribution are less time consuming than for the exponential distribution, a sample of 1000 realizations of the process $s_{0,750}$ could be generated. The estimated expected first-passage time to the line $x = 750$ was 236.629, with standard deviation 2.23546, giving an 95% upper confidence limit of 0.315660.

Figure 1: Estimating τ, uniform distribution

4.2 Estimates

The same data used for the confidence limits was also used to estimate the time constant. From the first-passage times we try to extrapolate towards infinity to get estimates. Here we use a simple method. We plot the (by n) normalized first-passage times against $\frac{1}{\sqrt{n}}$, and fit a regression line. The intercept will then estimate the time constant. The choice of $\frac{1}{\sqrt{n}}$ as the predictor is based only on data exploration, and not on theoretical ideas.

The removal of some data points corresponding to the lines closest to the origin results in slightly higher estimates than using the whole data set. Further removal and fitting polynomials with higher degree had in most cases only minor or no influence on the estimates. Examples are found in Figures 1 and 2. The chosen estimates are 0.402 for the exponential distribution, and 0.312 for the uniform distribution.

	Exponential	Uniform
Upper bound	0.503425	0.403141
Upper confidence limit	0.409711	0.315660
Estimate	0.402	0.312
Lower bound	0.300282	0.243666

Table 4: Summary

5 Algorithms

5.1 Lower bounds

The general algorithm implemented consists of three steps.

Figure 2: Estimating τ, exponential distribution

1. *Generate an enumeration of the $K(m)$ self-avoiding walks of length m.* This is done recursively, by starting with the walk that goes straight to $(m, 0)$, and successively altering more and more of the end of the walk, enumerating all walks as we pass them, up to equivalence.

2. *Generate the $K(m) \times K(m)$ matrix $\tilde{\mathbf{B}}$.* For each enumerated walk γ_i, we cycle through all walks that start with γ_i. If a walk ends with a translation of γ_j with m:th last x-coordinate z we increment the element (i, j, z) in the three dimensional array representing $\tilde{\mathbf{B}}$.

3. *Look for a good lower bound.* We start with values of a and ν that we know fulfills the criteria. Then a good ν is found and held fixed. a is then increased to the best possible value. This is repeated a few times (typically two or three times).

The program was written in C++. The time-consuming parts are step 2 (for large n) and step 3 (for large m). Good start values of a and ν are critical for fast performance, but this is easily accomplished by extrapolation from smaller n or m. Also, ν is quite stable for varying n and m.

5.2 Simulations

We start with a list (implemented as a doubly linked list) of the vertices that can be infected, that is, they are adjacent to some already infected vertex. In the first step the list contains the points $(1, 0), (0, -1)$ and $(1, 0)$. The list is sorted with respect to the times of the forthcoming infection. The first vertex in the list (the next vertex to be infected) is then removed from the list, and the list is expanded with eventual new vertices that are adjacent to the vertex being removed, but not yet infected. Of course only vertices inside the cylinder may be in the list. We then continue removing vertices until the infection has reached the line $x = n$, and we are done.

5.3 Upper bounds, exponential distribution

The calculations outlined in Section 3.2 were implemented in a recursive PASCAL program, where all calculation were rounded upward to double precision. All calculated expectations were saved. As the number of possible configurations in some of the cases was very large ($M = 7$, $x = 6$ gives 2^{42} configurations), memory was used dynamically. The computing times was several days for the larger cases, with close to 1 GB of memory used.

5.4 Upper bounds, uniform distribution

We see each configuration as a node in a rooted tree, with the case of only the origin infected as the root, and where each (tree) node's children are the configurations with one more edge infected than the parent. The algorithm makes a postorder traversal of this tree, where in each step the equation for the expectation for this configuration is determined. The processing of a given configuration is rather involved, due to the somewhat complex rules for the set of edges that should be considered, and the difficulties to implement these in an efficient way.

References

[1] S. E. Alm. Upper bounds for the connective constant of self–avoiding walks. *Combinatorics, Probability and Computing*, 2:115–136, 1993.

[2] S. R. Broadbent and J. M. Hammersley. Percolation processes I. Crystals and mazes. *Proceedings of the Cambridge Philosophical Society*, 53:629–641, 1957.

[3] J. M. Hammersley and J. D. Welsh. First–passage percolation, subadditive processes, stochastic networks, and generalized renewal theory. In *Bernoulli-Bayes-Laplace Anniversary Volume*, pages 61–110. Springer-Verlag, 1965.

[4] S. Janson. An upper bound for the velocity of first–passage percolation. *Journal of Applied Probability*, 18:256–262, 1981.

[5] R. T. Smythe and J. C. Wierman. *First-Passage Percolation on the Square Lattice*. Lecture notes in Mathematics 671. Springer-Verlag, 1978.

[6] A. J. Wakefield. Statistics of the simple cubic lattice. II. *Proceedings of the Cambridge Philosophical Society*, 47:799–810, 1951.