Monte Carlo simulations of periodic self-avoiding walks

Bachelor's thesis

Sebastian Mohr

Supervisor:
Prof. Dr. Wolfhard Janke
Dr. Stefan Schnabel

UNIVERSITÄT
LEIPZIG

March 31, 2020

Contents

1 Introduction

The self-avoiding walk (SAW) is one of the simplest basic problems in statistical physics. In short a self-avoiding walk is a sequence of moves on a lattice or in other words a path. These paths are not allowed to cross themself i.e. they do not visit a site more than once. These paths are used to describe polymers with excluded volume or other chain-like entities. Despite that, not much about these paths is known in a rigorous mathematical fashion. Therefore, physicists have provided numerous conjectures that are believed to be true and are strongly supported by numerical simulations.

The most intuitive approach would be to simply enumerate all possible self-avoiding walks. This was done as early as 1992 and up to a length of 39 steps [1]. With more advanced and efficient algorithms, one can archive even longer walks up to a length of 71 steps [2].

Due to the small lengths of these enumerated walks compared to real polymers[1], it is chosen to rely on random sampling to obtain numerical results i.e. Monte Carlo methods and in particular the pivot algorithm.

Even as early as 1988, using this pivot algorithm walks of up to 10000 steps could be generated [3]. And later with an even more efficient implementation up to 268 million steps [4].

As can be seen, computer simulations play a big role at obtaining knowledge about the behavior of this model and there was quite some work done in the last 50 years in addition to the ones mentioned before.

Let us restrict the definition of self-avoiding walks a bit and make it even more interesting by adding periodic boundary conditions. One extends the original self-avoiding walks by adding a copy of them at there respective start- and endpoints.

Afterwards, if these new periodic walks are still self-avoiding what can one say about changes with regard to the normal self-avoiding walks? Which conjectures change or even which stay the same, compared to normal self-avoiding walks?

In the following thesis I am going to look at some of the main proportionalities of self-avoiding walks and compare them to periodic SAWs. Such as the critical exponent ν and connective constant μ. The observed statistical quantities are than compared to exact results obtained via enumeration and the normal self-avoiding case, for which a lot of reference data can be found.

[1] Typical polymers such as HDPE can be composed of 10.000 monomers/steps.

2 Theoretical background

In general a N-step self-avoiding walk ω, on a d-dimensional hypercubic lattice[2] $\mathbb{Z}^{\mathrm{d}}$, is defined as a sequence of coordinate points $w(i) \in \mathbb{Z}^d$ or also called sites. The walk

$$\omega = (\omega(0), \omega(1), ..., \omega(N)) \tag{2.1}$$

consists of $N+1$ sites and has to satisfy the condition $|\omega(j+1) - \omega(j)| = a$ i.e. the step length is always a, whereby a is the lattice constant for the hypercubic lattice.

In the following we only look at two dimensions, hence the 2-dimensional hypercubic lattice. In other words this lattice is the square lattice. The lattice constant is in the following proposed to be one unit long.

The walk is not allowed to visit a site more than once or in other words there exist no two equal sites in the sequence. One could say as condition for this self-avoidance $|w(j) - w(i)| \neq 0$ for all pairwise distinct $i, j \in [0, N]$.

A walk is defined to always start at the coordinate origin i.e. $\omega(0) = \vec{0}$ and one can define the length of such a walk as

$$N = |\omega| = \sum_{i=0}^{N-1} |\omega(i+1) - \omega(i)| \ . \tag{2.2}$$

The length is always equal to the number of bonds N.

If one counts all the possible walks c_N for a given length N, one expects to find the behaviour of c_N to be

$$c_N \approx A\mu^N N^{\gamma-1} \tag{2.3}$$

where A, μ, γ are dimension-dependent positive constants, which I will further contemplate in the following.

2.1 Connective constant

If this proportionality (2.3) holds true, taking the limit gives

$$\mu = \lim_{N->\infty} c_N^{1/N} \tag{2.4}$$

[2]Considering crystal structure, a d-dimensional hypercubic lattice has a simple cubic crystal structure in d-dimensions.

and hence there must exist a limit for the connective constant μ, by

$$\mu \leq c_N^{1/N} \ . \tag{2.5}$$

This constant describes the average number of possible step directions (connections) a walk has at every single site. For example, a normal random walk on the 2-dimensional hypercubic lattice $\mathbb{Z}^2$ has the connectivity constant of $\mu_{\text{rw}} = 4$. Therefore, there are four equally as likely step directions for every single site and hence there are 4^N possible walks.

The exact value of μ is not known for most lattices[3] but is considered to be $\mu_{\text{sq}} = 2,63815853031(3)$ [5] for the square lattice.

As always if one considers the periodic case that will be defined later, it is not known if any of this still holds. However, it is expected to hold and should approach a limit as well.

2.2 End to end distance

The distance from the start of the walk i.e the origin $\omega(0)$ to the end of a walk $\omega(N)$ is called end to end distance.

For one considered length N it is necessary to find the average end to end distance over multiple/all walks. The average distance (squared) for all possible walks of one length N is given by the mean-square displacement

$$\langle |\omega(N)|^2 \rangle = \frac{1}{c_N} \sum_{\omega:|\omega|=N} |\omega(N)|^2 \ . \tag{2.6}$$

The sum is taken over all considered walks with the length $|\omega| = N$, whereby c_N is the number of considered walks. As a reminder the walks have N bonds but $N + 1$ sites.

It is believed that the mean-square displacement will not always be linear in the number of step. The average end to end distance $\langle |\omega(N)|^2 \rangle$ behaves proportional to the length of the walk N.

$$\langle |\omega(N)|^2 \rangle \propto N^{2\nu} \tag{2.7}$$

with ν as dimension-dependent positive constants.

[3]The only exact values are known for the hexagonal lattice and the (3.12^2) lattice.

2.3 Generating functions

In discrete mathematics, generating functions are a powerful tool to efficiently represent sequences. In the case of self-avoiding walks they can be used to describe all possible walks on a given lattice.

For our use case generating function are defined as the power series

$$\chi(z) = \sum_{N=0}^{\infty} c_N z^N \tag{2.8}$$

whereby the coefficient c_N tells us how many walks are possible for the length N.

This series can be constructed if one knows all coefficients c_N, but there is no way of calculating them in a mathematical rigorous fashion[4]. As such physicists have come up with another way of obtaining them. For very small lengths it is possible to count these coefficients by hand, but that gets tedious quite fast. Another way would be to use computers to do this enumeration, which is explained later on in section 3.1.

There are some advanced methods which use generating functions. These methods can be used to more efficiently enumerate all possible walks as can be seen in [6], but usually the transfer matrix method used is to do enumeration to a greater extend.

In a following section I am going to exactly enumerate the coefficients c_N up to a length of $N = 18$ for both normal and periodic self-avoiding walks.

These coefficients can be used to construct an approximation for the generating function and can be compared with the lower bound

$$\chi(z) \geq \sum_{N=0}^{\infty} \mu^N z^N = \frac{1}{1-\mu z} \,. \tag{2.9}$$

This follows from (2.5) and can be seen at greater length in [7, p. 13].

2.4 Gyration radius

The radius of gyration is normaly used to describe the dimension of a polymer chain i.e. a self-avoiding walk.

[4] At least it is not possible for the square lattice.

As before in the end to end distance case one looks at the average over all considered walks, therefore the mean-square radius of gyration is defined as

$$\langle |S(N)|^2 \rangle = \frac{1}{c_N} \sum_{\omega} |S(N)|^2 \tag{2.10}$$

with

$$|S(N)|^2 = \frac{1}{N+1} \sum_{i=0}^{N} (\omega(i) - \omega_{\text{mean}})^2 \tag{2.11}$$

$$= \frac{1}{N+1} (\sum_{i=0}^{N} \omega(i) - (\frac{1}{N+1} \sum_{j=0}^{N} \omega(j)))^2. \tag{2.12}$$

In other words it is defined as the average squared distance from any site of the walk to its center of mass.

Very little is proven about this quantity, especially in the periodic case. It is believed to have the asymptotic behavior

$$\langle |S(N)|^2 \rangle \propto N^{2\nu} \ . \tag{2.13}$$

This is the same proportionality as for the end to end distance (2.7), except for a different proportionality constant or in other words except a different amplitude. Later it will be seen, that this proportionality constant is way smaller than the end to end distance constant.

2.5 Correction to scaling

Since only relativly small N are taken into account for the enumeration, I will get into why later, one should consider the correction to scaling exponents Δ_i which alters our anticipated scaling relation (2.3) to

$$c_N \approx A\mu^N N^{\gamma-1}(1 + B_1 N^{-\Delta_1} + B_2 N^{-\Delta_2} + ...) \ . \tag{2.14}$$

Only the leading term was used in the following analysis of the enumeration, since the terms after it are believed to be a lot smaller and not of statistical relevance. For the enumeration of the critical exponent ν, one modifies (2.7) with the correction-to-scaling term to obtain

$$\langle |w(N)|^2 \rangle \approx DN^{2\nu}(1 + EN^{-\Delta_1}) \ . \tag{2.15}$$

A more detailed explaination and appealing evidence, that the leading correction term is $\Delta_1 = 3/2$, can be found in [8].

2.6 Periodic boundary conditions

In the following we only accept walks if they suffice periodicity. There are multiple ways to define such periodicity.

One could define a finite lattice and add periodic boundaries at its edges. This is done a lot for spin models such as the potts model. However, for the case of random walks on a lattice this would restrict our visitable sites and one would have to scale the lattice according to the walks length, which could be proven to be quite troublesome.

The chosen other approach is done without restricting the walks to a finite lattice. Consider G to be the group of orthogonal transformations and the symmetry operation $g \in G$, which will be discussed a bit later or as seen in [3]. Let ω be our original self-avoiding walk with length N and ω' the original walk transformed by a symmetry operation g_i and translated by a vector $b \in \mathbb{Z}^2$ i. e.

$$\omega' = g_i\omega + b \; . \tag{2.16}$$

Now to obtain a periodic self-avoiding walk one adds two of such transformed walks to the original walk. The first transformed walk ω'_- is translated such that the end point is at the start point of the original walk, i.e. $\omega'_- = \omega g_i - w(N)$. The second walk ω'_+ is translated in such a way, that its start point is at the end of the original walk, or in other words $\omega'_+ = \omega g_i + w(N)$. The combined walk is than defined as

$$\Omega = \omega'_- \circ \omega \circ \omega'_+ \tag{2.17}$$

and if this combined walk is still a valid self-avoiding walk, the walk ω is considered periodic with respect to the chosen symmetry operation g_i.

As defined above the original walk always starts at the coordinate origin and an example for this method can be seen in figure 1.

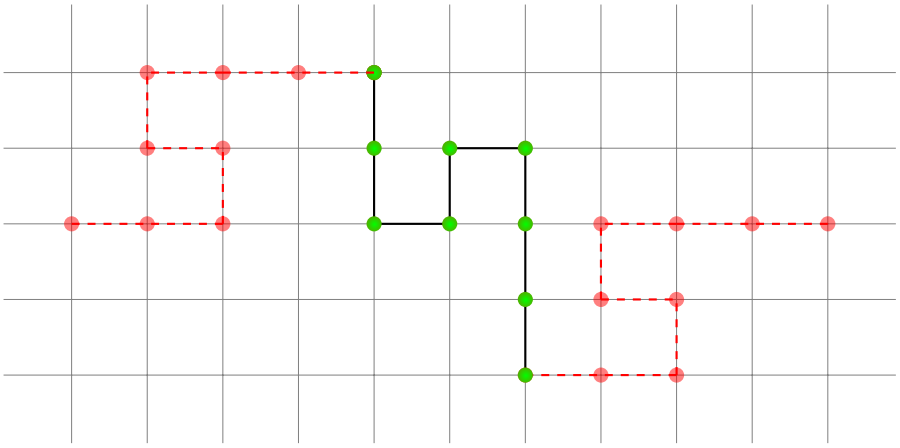


Figure 1: Sample image of a periodic self avoiding walk with a rotation by $\pi/2$. Where the original walk is green and the rotated ones are red.

To make the transformation and translation of the walks easier on myself and hopefully the simulations as well, I defined a walk as a sequence of step directions instead of a sequence of sites i.e up, down, left, right.

A bond walk now called γ of a length N [5] is expressed as sequence of step directions. The only allowed steps are the unit vectors $\pm\vec{e_k}$ on the observed 2-dimensional hypercubic lattice $\mathbb{Z}^2$ and every step is still one unit long. In other words the walk γ is defined as

$$\gamma = (\gamma(0), ..., \gamma(N-1)) \tag{2.18}$$

whereby

$$\gamma(i) = \pm \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \pm \begin{pmatrix} 0 \\ 1 \end{pmatrix} \ . \tag{2.19}$$

If one again looks at the construction of a periodic walk but now considering the walk γ instead of ω, it is not necessary anymore to translate the transformed walks γ' to the end point or start point of the original walk. The sequence is translation independent i.e for every $b \in \mathbb{R}^2$

$$\gamma = \gamma + b \tag{2.20}$$

holds true.

I hope to achieve better computation times because of this small change, since it is not necessary to keep track of the position vectors anymore. The pivot operations should also be a bit more efficient. On the other hand, it is

[5] The length N is equivalent to the monomere walks ω.

a bit more troublesome to verify the self-avoidance of the walks. To do that one computes every position and checks if the position is occupied more than once, since it is necessary to retrospectively compute the positions to check the self-avoidance, one could expect worse performance for this validation.

In the considered two dimensions, G is the dihedral group and the transformed walks $\gamma' = g_i\gamma$ are formed with one of the 8 following operations g_i $i \in [0, 7]$. The operations can all be formed with a transformation matrix as well but since we use bond walks, it should be way more efficient to do the symmetry operations according to figure 2.

g_0 Identity:

There is no change in the walks step directions.

$g_{1,2}$ Rotation by $\pm\frac{\pi}{2}$:

We check every direction the walk takes and change it according to figure 2a, we could do all of these operations with a transformation matrix as well, but the chosen approach should be more efficient for the simulation.

g_3 Rotation by π:

We have to change every *up* step to a *down* step and every *left* step to a *right* step and vice versa as can be seen in figure 2b.

$g_{4,5}$ Axis reflections:

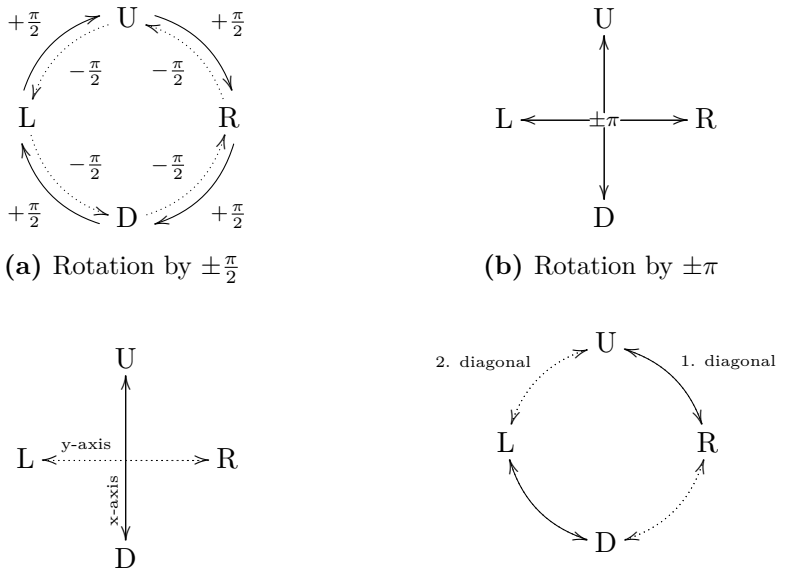
By reflecting our walk on the x-axis we change every *up* step to a *down* step and every *down* step to an *up* step. The y-axis reflection is done the same way with the *left* and *right* steps which can be seen in figure 2c.

$g_{6,7}$ Diagonal reflection:

By reflection of our walk on the diagonal lines $x = y$ and $x = -y$ we have to change the step directions according to figure 2d.

As a short note in three dimensions, there are 36 different symmetry operations, in contrast to the 8 ones in two dimensions.

It is easy to see, that every periodic walk, which has to suffice a rotation by π is not valid per definition! One can see this with relative ease, since the end and start sites will always overlap. An example for this can be seen in figure 3.



(a) Rotation by $\pm\frac{\pi}{2}$

(b) Rotation by $\pm\pi$

(c) Reflection on the x or y axis

(d) Reflection on the lines $x = y$ or $x = -y$

Figure 2: Change in step direction by applying a symmetry operation g_i to a walk γ.

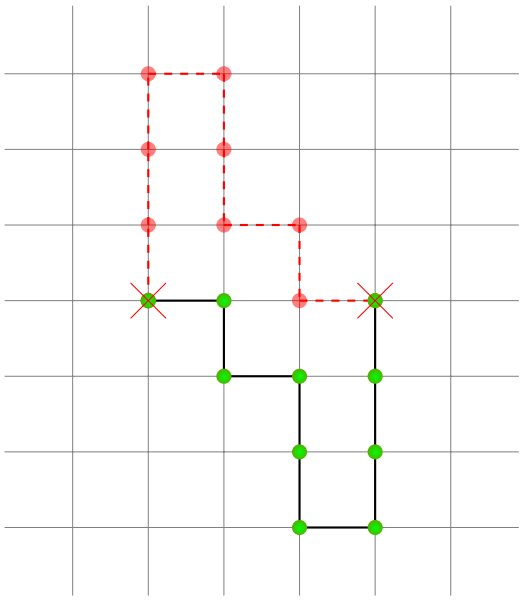


Figure 3: Sample image of an invalid periodic self avoiding with a rotation by π, where the original walk is green and the rotated ones are red.

Since the rotation by $\pm\frac{\pi}{2}$ and the four reflections are pairwise symmetric one also expects to get the same pairwise results for these operations. This will hold true and will be seen easier on the basis of the enumeration later. One could also look at the mathematical definition of the dihedral group G in two dimensions (klein four-group) and see this symmetric behaviour with relative ease.

2.7 Classification in categories

To achieve a better comparison between these different symmetric operations, it could be useful to arrange them into different categories. I want to introduce three different categories K_i. They were chosen roughly by the likelyhood of obtaining a valid periodic walk γ after applying a symmetry operation g_i to a walk.

These categories accommodate bond walks γ, if the combined walk

$$\Gamma = (g_i\gamma \circ \gamma \circ g_i\gamma) \tag{2.21}$$

is a valid self-avoiding walk. As explained above the transformed bond walks $g_i\gamma$ do not have to be translated.

The first category K_1 houses all walks, that are valid with the exact copy of the original walk added at the start- and the endpoint. The second one K_2 is a subset of K_1 and accommodates all walks that additionally fulfill periodic boundaries with respect to a rotation by $+\frac{\pi}{2}$ or $-\frac{\pi}{2}$. K_3 is again a subset of K_2, which further restricts the accepted walks by periodicity with mutual respect to one axis reflection and one diagonal reflection.

Or in short words, the walk γ has to be a valid periodic self-avoiding walk, which fulfills different symmetric operations. Depending on the different symmetric operations the original walk gets sorted into different categories. The different symmetric operations and their corresponding categories are shown in table 1.

	g_0	g_1 or g_2	g_4 or g_5	g_6 or g_7
K_1	x			
K_2	x	x		
K_3	x	x	x	x

Table 1: Classifications of the three different considered categories.

Since the symmetric operations are pairwise symmetric and yield the same global observables, one can always only consider one of the pairwise sym-

metric operations i.e rotation, axis reflection and diagonal reflection. Hence, these pairwise symmetric operations are put into one column in table 1.

3 Methods

All results obtained are either from exact enumeration or Monte Carlo methods i.e. the pivot algorithm. Monte Carlo methods are in the following used to get a statistical estimate of the connective constant μ and the critical exponent ν. All results are later compared to the estimates obtained from the exact enumeration.

3.1 Exact enumeration

To obtain exact results for walks of length N, one has to get every possible paths, that the walks can occupy on the chosen lattice. This can be done via „bruteforce“ iterating over every possible step the walks can take, which is explained later to greater extend. One could also use the transfer matrix method as can be seen in [9] and later in [10]. The transfer matrix method is more efficient, but since in this case the enumeration is only done to get reference values, the easier but less efficient way of bruteforce iterating should be sufficient.

I used the following recursive algorithm i.e. „bruteforce“ iterating for the periodic case:

1. Do a step in every direction.
2. Check if step was valid, if not terminate the walk.
3. Check if walk has reached the desired length and if that is the case, break the recursion. Also check if the walk fulfills any of the desired periodic boundary conditions, if that is the case, add the walk to the list of viable walks.
4. If walk has not reached the desired length, do another step in every direction (goto 1.).

The exact enumeration is very useful at obtaining results for small lengths, but as N approaches bigger and bigger numbers the computing time necessary to perform the enumeration exceeds the feasible. E.g. it took about 40 computing hours for the enumeration of length $N = 18$. As a short note it was possible to get way longer walks[6] for the normal case without periodic boundary conditions, hence the periodicity check took quite some computing power.

[6]Normal self-avoiding walks were enumerated up to a length of $N = 25$.

Overall, a length of $N = 18$ is not enough to conclude limiting behavior i.e. the critical exponent or the connective constant. Therefore, one additionally uses a different approach.

3.2 Pivot algorithm

The pivot algorithm is a dynamic Monte Carlo algorithm, which generate self-avoiding walks in a canonical ensemble (fixed number of steps N). A good starting point to read up on Monte Carlo methods is [11]. The pivot algorithm was replicated and slightly modified from [3].

It is used in the following to generate new valid walks with periodic boundary conditions. The pivot algorithm is done as follows:

1. First a normal self-avoiding walk is created at random. (One could also start with an easy case e.g. a rod, which was done in the following.)
2. A random pivot point k along the walk ($0 \leq k \leq N-1$) is chosen according to any preset strictly positive probabilities $p_0, ..., p_{N-1}$. In this bachelor thesis we consider the uniform distribution ($p_k = 1/N$ for all k).
3. We chose a symmetry operation $g_j \in G$ again according to any preset strictly positive probability. Now we apply this operation to the walk from the pivot point k onwards, i.e $g_j(\gamma_k, \gamma_{k+1}, ..., \gamma_N)$. Since we have only 8 possible operations in 2 dimensions and the operation g_0 (identity operation) can be ignored our probability for any operation is again the uniform distribution $p_g = 1/7$.
4. Now we check if our generated walk is still valid. If it is, the properties interesting to us are recorded. Otherwise the walk is discarded and we start again from 2., with the last valid walk.
5. At last if the walk is valid, we check if it is also valid for every single periodic boundary condition defined above. The walks are added to a list accordingly. After that we start again at 2. with the new walk. The validation for the periodic boundary condition can be easily done in parallel.

We record any global or local properties that are interesting for our studies. Here that is the acceptance fraction f, the end to to end distance $|w_N|^2$ and the gyration radius $|S_N|^2$.

Since the pivot moves are very radical, after a few successful steps the global conformation of the walk should have reached an essentially new state. We expect the same for the global properties of the walk, e.g squared end-to-end distance $|w_N|^2$ or the squared gyration radius $|S_N|^2$. One can easily confirm this fact by validating the autocorrelation time or simply looking at an autocorrelation plot for the selected properties. This was done later on and can be seen in figure 8 and figure 9.

Since random numbers or better said pseudo random numbers play a big role in Monte Carlo algorithms i.e. picking a pivot point and choosing a symmetry operation, one should look precisely at the generation of such. The simulation programs for the pivot algorithm used the mersenne twister pseudo-random-number generator, which is known to generate good random numbers [12].

3.3 Weighted least squares

In the following analysis for the enumeration and the pivot algorithm some regression has to be done e.g. for the critical exponent ν. The following explanations can be seen at greater extend in [7, p. 292-296].

In short one takes the equation (2.7) or (2.13), which gives a two-parameter family of curves

$$Y = AN^{2\nu} \; . \tag{3.1}$$

Or with considerations to the dominant correction term, as can be seen in section (2.5), gives the four-parameter family

$$Y = AN^{2\nu}(1 + BN^{-\Delta}) \; . \tag{3.2}$$

To fit these curves one uses the method of least squares regression. Linear regression functions are the easiest to work with, so by taking the logarithm of (3.1) and (3.2), we obtain

$$\log Y = \log A + 2\nu \log N \tag{3.3}$$

or again by taking the correction to scaling into account

$$\log Y = \log A + 2\nu \log N + BN^{-\Delta} \; . \tag{3.4}$$

The last part of (3.4) was obtained by the approximation $\log(x+1) \approx x$ for x near 0.

For now only considering (3.3), normal least squares are done by minimizing the sum of square

$$\sum_{i=1}^{m} (\log \hat{Y}_i - \log A - 2\nu \log N_i)^2 \; . \tag{3.5}$$

Sadly this is not sufficient for our case, since the variance of $\log \hat{Y}_i$ is not the same for every i. Instead one uses *weighted least squares*, weighting each term to the inverse of its (estimated) variance, so that the Y_i's, in which one has more confidence are considered, to a greater extend.

For our case considering the variance v_i^2, then the least squares estimates $\hat{\log A}$ and $\hat{\nu}$ are the values of $\log A$ and ν that minimize the weighted sum of square

$$\sum_{i=1}^{m} \frac{1}{v_i^2} (\log \hat{Y}_i - \log A - \nu \cdot 2 \log N_i)^2 \ . \tag{3.6}$$

Whereby the minimized values are

$$\hat{\nu} = \frac{\sum v_i^{-2} \sum v_i^{-2} 2 \log N_i \log \hat{Y}_i - \sum v_i^{-2} 2 \log N_i \sum v_i^{-2} \log \hat{Y}_i}{\sum v_i^{-2} \sum v_i^{-2} (2 \log N_i)^2 - (\sum v_i^{-2} 2 \log N_i)^2} \tag{3.7}$$

and

$$\hat{\log A} = \frac{\sum v_i^{-2} \log \hat{Y}_i - \hat{\nu} \sum v_i^{-2} 2 \log N_i}{\sum v_i^{-2}} \ . \tag{3.8}$$

This looks quite complicated, but is actually not that hard to implement and is done by a lot of different libraries e.g. *scipy* for python or *GNU Scientific Library* for C/C++.

In the following the *scipy* library was used to do the weighted least square regressions.

4 Enumeration

The enumeration of all SAWs with PBC was done with the method explained above and up to a length of $N = 18$. The simulation was written in C++ and analysed with python using mainly the scipy library [13].

4.1 Generating functions

As defined in equation (2.8) one can determine the coefficients of the generating function by recursively counting (enumerating) all possible walks for each length N.

The coefficients obtained from the enumeration are given in table 2. They are sorted by the different categories as proposed in section 2.6 and the normal self-avoiding walk was also added as reference. The percentage values are noted down for an easier comparison between the normal self-avoiding walks and the periodic walks.

	c_N	c_{NK1}		c_{NK2}		c_{NK3}	
1	4	4	100.0000%	4	100.0000%	2	50.0000%
2	12	12	100.0000%	4	33.3333%	2	16.6667%
3	36	28	77.7778%	12	33.3333%	4	11.1111%
4	100	76	76.0000%	28	28.0000%	12	12.0000%
5	284	204	71.8310%	60	21.1268%	22	7.7465%
6	780	540	69.2308%	140	17.9487%	56	7.1795%
7	2172	1404	64.6409%	332	15.2855%	124	5.7090%
8	5916	3724	62.9479%	852	14.4016%	310	5.2400%
9	16268	9748	59.9213%	2068	12.7121%	744	4.5734%
10	44100	25772	58.4399%	5236	11.8730%	1826	4.1406%
11	120292	67940	56.4792%	13044	10.8436%	4468	3.7143%
12	324932	179068	55.1094%	33220	10.2237%	11168	3.4370%
13	881500	472580	53.6109%	83356	9.4562%	27688	3.1410%
14	2374444	1245620	52.4594%	213292	8.9828%	69768	2.9383%
15	6416596	3286308	51.2158%	540468	8.4230%	174500	2.7195%
16	17245332	8666956	50.2568%	1385652	8.0349%	441812	2.5619%
17	46466676	22861604	49.2000%	3528868	7.5944%	1112284	2.3937%
18	124658732	60301764	48.3735%	9080964	7.2847%	2831496	2.2714%

Table 2: Generating function coefficients for all SAWs with different single periodic boundary conditions.

The coefficients for the normal case without periodicity are confirmed to be true, as can be seen in other works e.g. first [1]. I am very confident that the

coefficients for the periodic case should also be correct even though there is no reference data for them. The same coefficients are obtained by doing the enumeration by hand, which can be done up to a length of 3 with relative ease.

It is known, that c_N should be scaling with respect to (2.3). As one can see in figure 4, this holds true even for the periodic walks in the different categories.

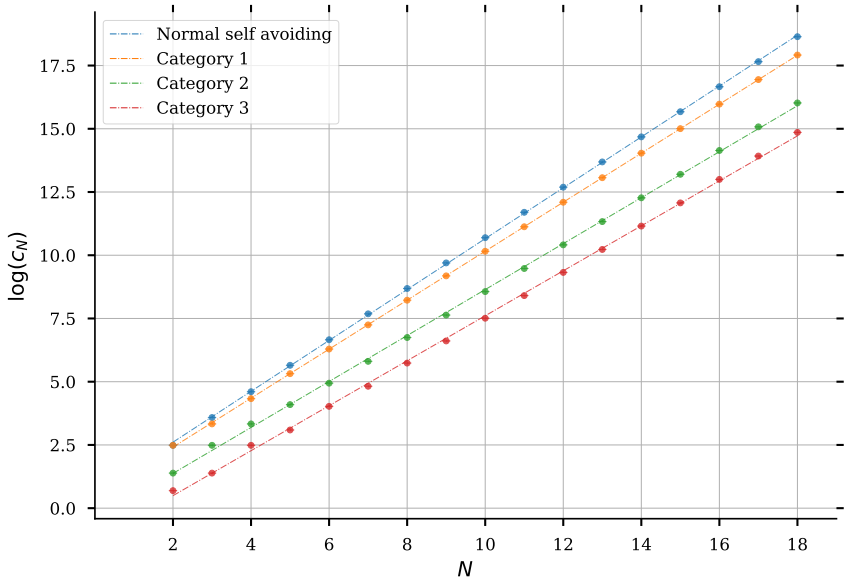


Figure 4: Fit for the coefficients c_N versus the length N. Linear regression was performed to get an estimate for the connective constant. The fitting parameters can be seen in table 4.

It is also interesting to look at which percentage f_N of the normal walks are periodic. In figure 5 one can see the behaviour of this percentage. It seems, that the percentage is going to approach a bound as the length N gets bigger.

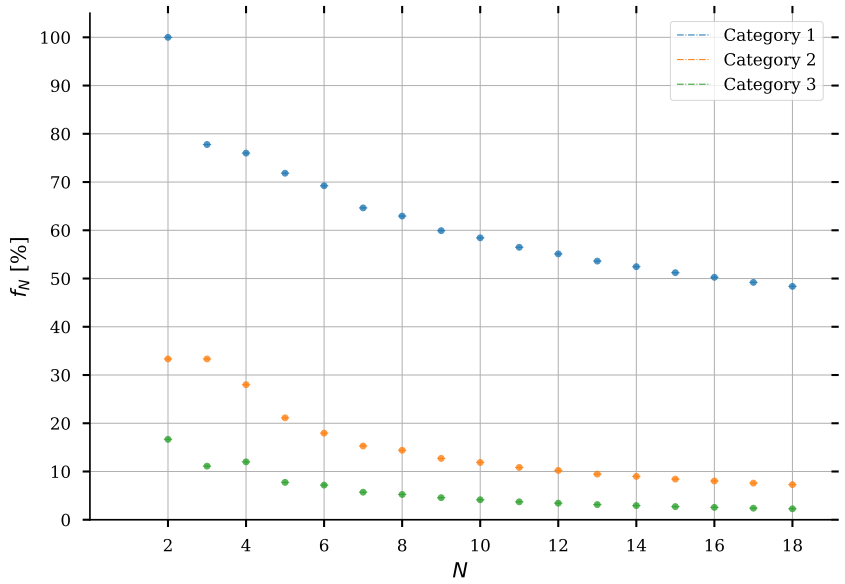


Figure 5: Percentage of periodic walks f_N with respect to normal self-avoiding walks versus the length of the walks N.

4.2 Connective constant

Using the number of possible walks c_N, we obtained as seen before and the corresponding length of the walks N, one can easily calculate an upper bound for the connective constant μ. This can be seen in (2.4) and (2.5), whereby the results are noted down in table 3.

	None	Category 1	Category 2	Category 3
μ_{16}	2,7689	2,6448	2,3405	2,2713
μ_{17}	2,7642	2,6443	2,3529	2,2860
μ_{18}	2,7593	2,6439	2,3653	2,3005

Table 3: Upper bound for the connective constant μ obtained from exact enumeration for all single periodic boundary conditions.

It does seem, that the upper bound does not converge for periodic walks in the category two and three i.e. the connective constant increases with the

walk length N. But that should change for larger N, which is confirmed later in the pivot algorithm section.

As expected, the more the walks are restricted with periodic boundaries the less valid walks c_N are obtained. In conclusion, one could also expect a smaller connective constant μ. This seems to even hold true for these small lengths N.

On the other hand, it is possible to use (2.3) and linear regression to get a rough estimate for the connective constant μ. One takes the logarithm on (2.3), which yields

$$\log c_N = \log A + N \log \mu + (\gamma - 1) \log N \; . \tag{4.1}$$

For small N the last part $(\gamma - 1) \log N \approx 0$ since it is known, that $\gamma = \frac{43}{32}$ in the square lattice case [7, p. 7]. This linear regression can be seen above in figure 4 with the results for the amplitude A and the connective constant μ noted down in table 4.

	None	Category 1	Category 2	Category 3
A	1,8180±0,0262	1,6091±0,0129	0,6422±0,0513	0,2770±0,0602
μ	2,7339±0,0024	2,6340±0,0012	2,4806±0,0046	2,4328±0,0054

Table 4: Connective constant μ and amplitude A obtained from linear regression as seen in figure 4 with there respective errors σ.

The result seems quite promising and are not that far away from the expected results of $\mu_{\text{sq}} = 2,63815853031(3)$ [5].

4.3 Critical exponents

To obtain the critical exponents ν one has to calculate the mean squared end to end distance $\langle |\omega(N)|^2 \rangle$ or mean squared gyration radius $\langle |S(N)|^2 \rangle$ as defined in equation (2.6) and respectively (2.10). These two properties were recorded for every enumerated length $N < 19$.

Afterwards, one uses weighted least square regression, for which the theoretical background can be seen in section 3.3, to obtain estimates for the critical exponent ν and the amplitudes E and F.

In other words, we fit the function

$$Y = AN^{2\nu}(1 + BN^{-\Delta}) \tag{4.2}$$

respectively for the end to end distance $Y = \langle |\omega(N)|^2 \rangle$ and the gyration radius $Y = \langle |S(N)|^2 \rangle$.

The resulting fits can be seen in figure 6 for the means squared end to end distance and in figure 7 for the mean squared gyration radius. The fits looks nearly identical with the only exception being the different amplitude A.

One could also do a linear fit by applying the logarithm, this is done later for the results obtained via the pivot algorithm. These two different fitting methods should yield the same fitting parameters though.

The estimates for the critical exponents ν and amplitudes A, that were obtained, can be found with their respective errors in table 5. Again sorted for the different categories defined above.

Since our length is still pretty small, one adds the dominant correction to scaling term $BN^{-\Delta}$ as seen above in section 2.5. I considered Δ as constant with the value $\Delta = 3/2$ as compelling evidence can be found to confirm this [8].

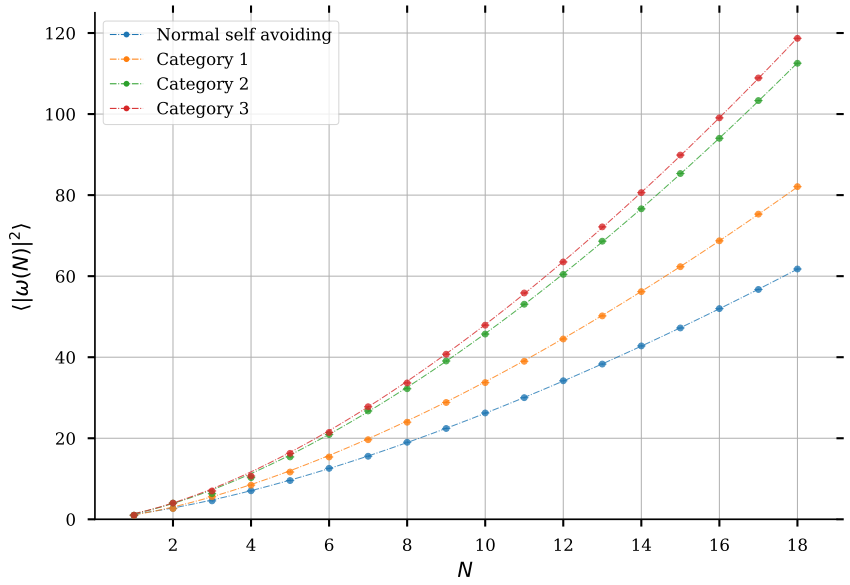


Figure 6: Fit for obtaining the critical exponent ν via mean squared end to end distance.

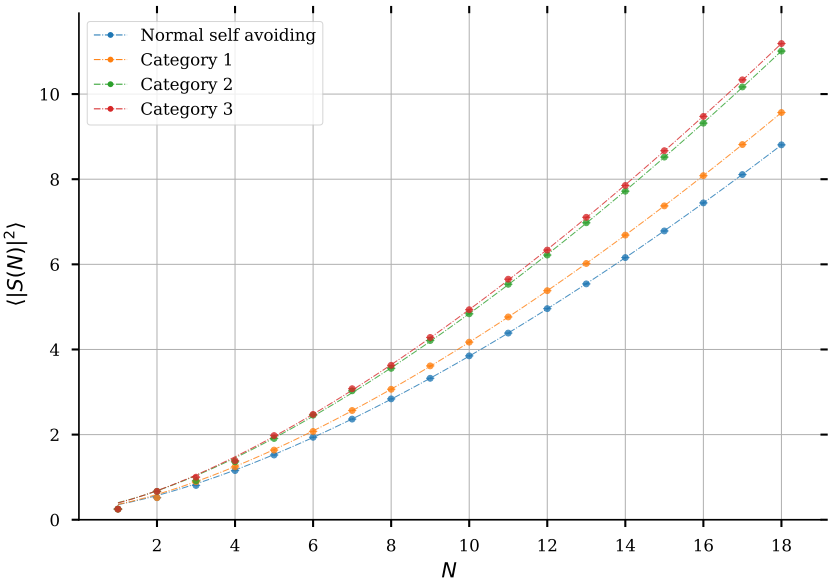


Figure 7: Fit for obtaining the critical exponent ν via mean squared gyration radius.

	none	K_1	K_2	K_3
A_{ee}	0,8645±0,0222	1,0704±0,0099	1,3523±0,0159	1,3843±0,0199
ν_{ee}	0,7373±0,0039	0,7506±0,0017	0,7650±0,0019	0,7702±0,0026
A_{gy}	0,1239±0,0018	0,1369±0,0012	0,1683±0,0058	0,1766±0,0057
ν_{gy}	0,7334±0,0022	0,7310±0,0013	0,7202±0,0053	0,7149±0,0049

Table 5: Fitting parameters for the critical exponent fits, as can be seen in figure 6 and figure 7.

Even though the results seem too small in comparison to the anticipated result $\nu = 3/4$ [14], they are by far in the margin of error for these small lengths N. The values for ν considering the different periodic walks do not diverge much, which could already be a small evidence for the independence of the critical exponent to the periodic boundary.

Overall, it seems that the more one restricts the walks by periodic boundary conditions, the bigger the end to end distance gets. Therefore, the amplitude increases from one category to the next. The same applies to the gyration radius and is quite interesting.

5 Pivot algorithm

As before in the enumeration case, the simulation was written in C++ and analysed with python mainly using the scipy library [13]. In extend the program was parallelized/multi-threaded, which has to be considered in the analysis.

5.1 Preliminary tests

In order to test the slightly modified pivot algorithm, 10^7 SAWs of lengths $N = 15$ and $N = 18$ were generated on the square lattice. I compared the mean squared end to end distance $\langle |w_N|^2 \rangle$ and mean squared gyration radius $\langle |S_N|^2 \rangle$ with the known exact values from the enumeration. All this was done for all categories defined beforehand.

Operation	none	K_1	K_2	K_3
$\langle \|w_{15}\|^2 \rangle$	47,1793 (47,2177)	62,3112 (62,3329)	85,3263 (85,3286)	89,9511 (89,8814)
$\langle \|S_{15}\|^2 \rangle$	6,7514 (6,7843)	7,35052 (7,3734)	8,5106 (8,5192)	8,6739 (8,6699)
$\langle \|w_{18}\|^2 \rangle$	61,7420 (61,7664)	82,0710 (82,0911)	112,4713 (112,5375)	118,5662 (118,68507)
$\langle \|S_{18}\|^2 \rangle$	8,76263 (8,80733)	9,53353 (9,5666)	10,9855 (11,0093)	11,1677 (11,1869)

Table 6: Preliminary tests for the slightly modified pivot algorithm.

The known exact values are shown in parentheses and were obtained via exact enumeration, which can be seen above. The test yields quite promising results i.e, they agree with the enumeration within statistical error of about $\pm 0,5\%$.

5.2 Critical exponent

On the two dimensional square lattice walks of length N ranging from 20 to 1000 were generated. For every length, 10^6 walks were evaluated. In other words, 10^6 sweeps were recorded after considering autocorrelation.

Autocorrelation, for which an example can be seen in figure 8, was determined and according to it only every fourth sweep recorded. This was done for every running thread in the simulation, since they were initialized independent from each other. In hindsight, it would have been enough to only consider it once, since for a lag greater than 4 the correlation coefficient is smaller than $0,5$. This holds true for every thread.

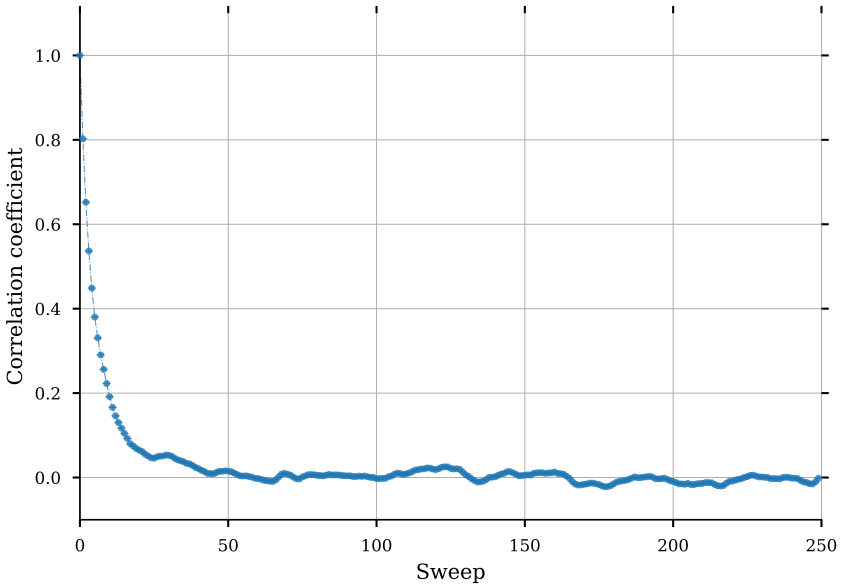


Figure 8: Autocorrelation plot for the end to end distance with length $N = 300$ on a single single thread

I also discarded the first ∼500 sweeps depending on the thread, since it looked like the equilibration time was achieved after that period i.e. the properties only fluctuate around the normal distribution after these first sweeps and there seems to be no drifting behaviour afterwards. An example for this can be seen in figure 9. Where the deviation from the mean is plotted against the time series. Also the standard deviation of the mean is illustrated in the background. If one looks at the first 500 sweeps one can see this small drifting behaviour towards this standard deviation of the mean. This was done because other than noted before, the initial state of the simulation was not random but a rod.

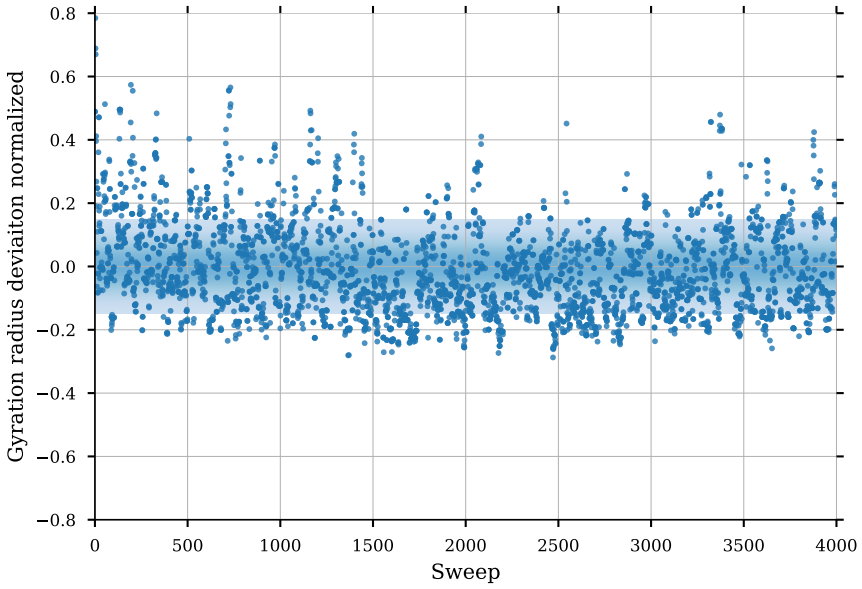


Figure 9: Mean deviation of the gyration radius versus the first 4000 sweeps. The length of the simulated walk was $N = 150$ and if looked at closely, it is possible to see the drifting behaviour of the first ∼500 sweeps.

Now to obtain the critical exponent ν via regression, one has to calculate the mean squared end to end distance $\langle |w_N|^2 \rangle$ or the mean squared gyration radius $\langle |S_N|^2 \rangle$. This was done while the simulation was running via the moving average.

Afterwards, considering every length N, the values for $\langle |w_N|^2 \rangle$ and $\langle |S_N|^2 \rangle$ were plotted against their corresponding walk length. Weighted least square regression was performed to obtain the critical exponent. There are two different types of regression, that were considered, i.e. weighted non-linear least squares and weighted linear least squares. The fitting function for the non-linear least square regression can be seen in equation (2.7). For the linear regression one modifies this function by applying the logarithm such that

$$log(Y) = log(DN^{2\nu}) = 2\nu log(N) + log(D) \ , \tag{5.1}$$

which can be seen at greater extend in section 3.3.

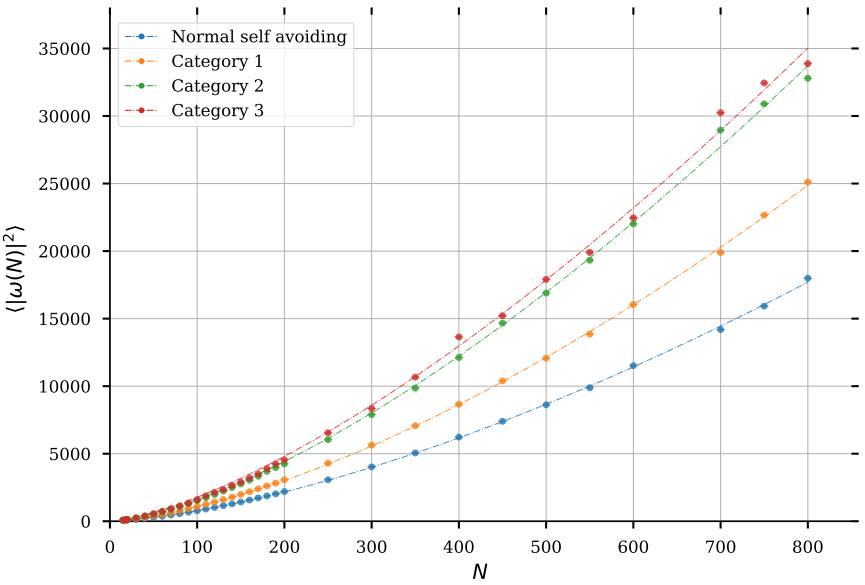
Whereby, the slope of the linear fit is 2ν and the interception is $log(D)$. For

both methods no correction to scaling term was used, since the length N is quite big already. But as before in the enumeration case one has to use weights considering the inverse variance.

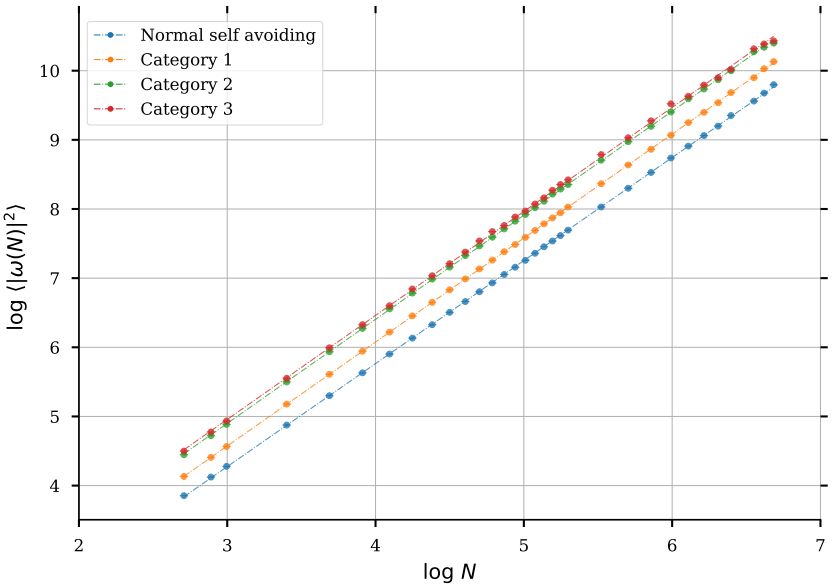
Both regression models were done for the mean end to end distance and the mean gyration radius. For the end to end distance an example using both methods can be seen in figure 10. Considering the gyration radius one repeats the regression explained above. This is not shown in any figure because it looks exactly the same as the end to end distance and yields nothing new.

The resulting estimates for the fit parameters are noted down in the table 7 and table 8. For every linear regression the goodness of the fit is godlike (≈ 0.9999), in contrast to the least square regression, where the goodness of fit is a bit smaller but still nearly perfect (≈ 0.98). Hence, the fit parameters for the linear regression should be a bit better.

It seems, that the critical exponent ν does not change too much from one category to another (about 1% deviation). As in the enumeration case this could prove that the critical exponent ν is indipendent from the choosen boundary condition. This behaviour was expected, since the critical exponent is considered to be universal for the dimension of the system.



(a) Non-linear fit for the mean squared end to end distance versus the length using weighted least squares. Errorbars are plotted but too small to see.



(b) Linear fit for the mean squared end to end distance versus the length using weighted least squares.

Figure 10: Fitting of critical exponent ν and amplitude D via mean squared end to end distance.

On the other hand, the amplitudes get bigger the more restrictions are applied on the walks i.e. the different categories. Furthermore, collapsed walks are less likely to be periodic, hence bigger amplitudes.

One could interpret the results as confirmation, that the critical exponent ν is independent from periodicity. This looks quite promising, but the critical exponents are still varying a bit. After considering even longer walks, one should be able to give a more accurate prediction about this independence.

	Fit model	None	K_1	K_2	K_3
D	non-linear	0,7439±0,0375	1,0589±0,0393	1,6908±0,0841	2,6726±0,4246
	linear	0,8302±0,0012	1,0705±0,0007	1,4596±0,0015	1,5348±0,0039
ν	non-linear	0,7532±0,0040	0,7518±0,0029	0,7406±0,0039	0,7577±0,0125
	linear	0,7439±0,0012	0,7511±0,0007	0,7532±0,0015	0,7540±0,0039

Table 7: Estimated parameters for the critical exponent ν and the amplitude D obtained from fitting the mean end to end distance versus the length N.

	Fit model	None	K_1	K_2	K_3
F	non-linear	0,1076±0,0033	0,1243±0,0020	0,1561±0,0084	0,2031±0,0307
	linear	0,1184±0,0018	0,1290±0,0018	0,1514±0,0023	0,1558±0,0043
ν	non-linear	0,7502±0,0024	0,7452±0,0013	0,7363±0,0042	0,7140±0,0119
	linear	0,7418±0,0018	0,7415±0,0018	0,7382±0,0023	0,7362±0,0043

Table 8: Estimated parameters for the critical exponent ν and the amplitude F obtained from fitting the mean gyration radius versus the length N.

5.3 Connective constant

To get an estimate of the connective constant from a Monte Carlo simulation, one can use an approach first done by Nathan Clisby [15].

In short words, we create two walks ω_1, ω_2 with length N. We add them together with one ending at $(0, 0)$ and the other one starting at $(0, 1)$. This is in the following expressed as $\omega_1 \diamond \omega_2$.

It is checked if the combined walk of length $2N + 1$ is still valid i.e. we calculate our indicator function which is defined as

$$B(\omega_1, \omega_2) = \begin{cases} 0 \text{ if } \omega_1 \diamond \omega_2 \text{ not self-avoiding} \\ 1 \text{ if } \omega_1 \diamond \omega_2 \text{ self-avoiding} \end{cases} \tag{5.2}$$

and update the mean value of $B(\omega_1, \omega_2)$ which is defined as

$$\tilde{B}_N = \frac{1}{c_N^2} \sum_{|w_1|=N, |w_2|=N} B(\omega_1, \omega_2) \tag{5.3}$$

$$= \frac{C_{2N+1}}{4C_N^2}. \tag{5.4}$$

The longest walks, I enumerated on the two dimensional hypercubic lattice, have a length of 18. We can get an estimate for higher order number of steps c_N with our mean indicator B_N recursively using the result from the

enumeration:

$$\begin{aligned} c_{37} &= \tilde{B}_{18} c_{18}^2 \\ c_{75} &= \tilde{B}_{37} c_{37}^2 = \tilde{B}_{37} \tilde{B}_{18}^2 c_{18}^4 \\ c_{151} &= \tilde{B}_{75} c_{75}^2 = \tilde{B}_{75} \tilde{B}_{37}^2 \tilde{B}_{18}^4 c_{18}^8 \\ &\vdots \end{aligned} \tag{5.5}$$

With this recursion and the definition of the connective constant

$$\begin{aligned} \mu_N &\equiv c_N^{1/N} \\ \log \mu_N &= \frac{1}{N} \log c_N \end{aligned} \tag{5.6}$$

one can get a good estimate of the connective constant while only using Monte Carlo methods. All of this can be seen at greater length in [15], it is a quite pleasant read. I would recommend to read it for in depth details about this method.

I did this not only for self-avoiding walks, but also for the periodic self-avoiding walks. The algorithm is done exactly the same with the only exception being that the walks are also checked for PBCs and accordingly added to different lists. This can be done parallel for the different categories defined above.

In the following table 9 the results (obtained via the recursive algorithm) for the connective constant and for each of my categories defined above are noted down.

	None	Category 1	Category 2	Category 3
μ_{18}	2,75(9)	2,64(3)	2,36(5)	2,30(0)
μ_{37}	2,74(2)	2,56(8)	2,18(0)	1,97(8)
μ_{75}	2,69(8)	2,49(0)	2,04(1)	1,82(2)
μ_{151}	2,67(4)	2,44(6)	1,97(1)	1,74(3)
μ_{303}	2,65(8)	2,42(0)	1,92(8)	1,69(6)
μ_{607}	2,64(9)	2,40(5)	1,90(5)	?

Table 9: Estimate of the connective constant μ obtained via Monte Carlo simulation.

I could not get enough valid periodic saw configurations for the third category at a length $N = 607$. The reason for this is just computing time. The

effective length of these walks, if one considers periodicity, is

$$N_{eff} = 3N (= 1821)$$

and the acceptance rate for the third category and this length is less than $0,01\%$. As such one has to obtain quite a few walks to get a usable estimate.

In comparison to the earlier results from the enumeration, which did not seem to converge, these results clearly approach a bound. For the normal self-avoiding case it is believed that $\mu = 2,63815$ as was mentioned before and can be seen in [16].

Using the recursion (5.5) one also gets an rough estimate for the number of possible walks c_N, which are noted down in table 10.

	None	Category 1	Category 2	Category 3
c_{18}	$1{,}24(6) \cdot 10^{8}$	$6{,}03(0) \cdot 10^{7}$	$9{,}08(1) \cdot 10^{6}$	$2{,}83(1) \cdot 10^{6}$
c_{37}	$1{,}63(5) \cdot 10^{16}$	$1{,}44(8) \cdot 10^{15}$	$3{,}33(4) \cdot 10^{12}$	$9{,}22(6) \cdot 10^{10}$
c_{75}	$2{,}16(2) \cdot 10^{32}$	$5{,}23(4) \cdot 10^{29}$	$1{,}75(0) \cdot 10^{23}$	$3{,}53(7) \cdot 10^{19}$
c_{151}	$3{,}17(0) \cdot 10^{64}$	$4{,}76(6) \cdot 10^{058}$	$3{,}18(4) \cdot 10^{44}$	$2{,}90(0) \cdot 10^{36}$
c_{303}	$4{,}76(4) \cdot 10^{128}$	$2{,}04(4) \cdot 10^{116}$	$2{,}75(5) \cdot 10^{86}$	$3{,}36(2) \cdot 10^{69}$
c_{607}	$7{,}40(8) \cdot 10^{256}$	$2{,}20(1) \cdot 10^{231}$	$9{,}71(1) \cdot 10^{169}$	$2{,}71(0) \cdot 10^{135}$

Table 10: Rough estimate for the number of possible walks c_N obtained via Monte Carlo simulation.

It is also possible to get a lower bound or estimate for even higher order coefficients using the generating function and the obtained connective constant. By comparison of coefficients in (2.9) and (2.8), one sees that

$$c_N \geq \mu^N \ . \tag{5.7}$$

Using this method and μ_{607}, one obtains roughly the same coefficients in comparison to the recursion. This can be seen in table 11.

	Method	None	Category 1	Category 2	Category 3
c_{607}	Recursive	$7{,}40(8) \cdot 10^{256}$	$2{,}20(1) \cdot 10^{231}$	$9{,}71(1) \cdot 10^{169}$	$2{,}71(0) \cdot 10^{135}$
c_{607}	$c_N \geq \mu^N$	$7{,}42(0) \cdot 10^{256}$	$2{,}17(1) \cdot 10^{231}$	$9{,}53(3) \cdot 10^{169}$	$1{,}89(2) \cdot 10^{135}$

Table 11: Comparison between two different methods for obtaining the number of walks c_N.

Even another approach to obtain the connective constant μ would be to look at the plot of the different indicator functions $\tilde{B}_N$ versus the length N, where one could use the exponential decay

$$\tilde{B}_N = ae^{-bN} + c \tag{5.8}$$

as fitting function for the relation. This was done in figure 11, whereby the fitting parameters can be seen in table 12.

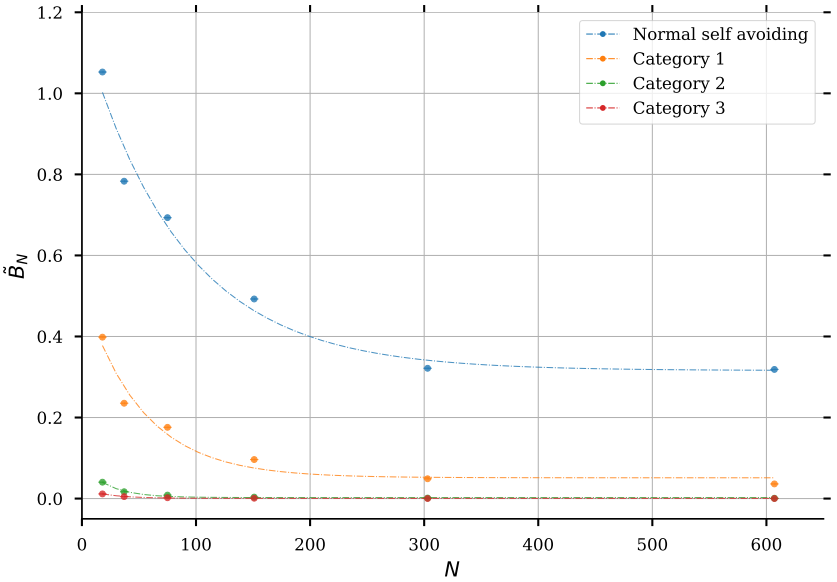


Figure 11: Indicator function $\tilde{B}_N$ versus the length N, whereby an exponential decay was fitted and the obtained fitting parameters can be seen in table 12.

	None	Category 1	Category 2	Category 3
a	0,8452±0,0911	0,4644±0,0699	0,0831±0,0146	0,0238±0,0033
b	0,0116±0,0031	0,0196±0,0056	0,0442±0,0085	0,0435±0,0067
c	0,3159±0,0498	0,0513±0,0216	0,0025±0,0014	0,0005±0,0003

Table 12: Estimate of the connective constant μ obtained via Monte Carlo simulation.

The fitting parameters, as seen in table 12, have quite big errors, e.g. in the third category the error and value for c are nearly identical. This value is

also very close to zero, which could be an indicator for convergences towards zero. Furthermore, if the probability $\tilde{B}_N$ is zero, no walks get accepted and that method could not be valid for the periodic case.

Afterwards, it is possible to use the function with their respective fitting parameters to calculate an estimate for the connective constant μ. The results for this approach can be seen in comparison to the recursion in table 13.

		None	Category 1	Category 2	Category 3
μ	Exponential fit	2,63(9)	2,38(6)	1,91(2)	1,65(2)
μ	Recursion	2,64(9)	2,40(5)	1,90(5)	1,69(6)

Table 13: Estimate of the connective constant μ obtained via fitting of the indicator function $\tilde{B}_N$.

6 Conclusion

Periodic walks are a fun small extension to normal self-avoiding walks. The extension gives insight on multiple global phenomena such as the critical exponent ν and the connective constant μ.

The critical exponent ν was considered to be lattice depended and as such should still remain the same as one introduces periodic boundary conditions. This periodic independence was already suspected after looking at the enumeration data. Furthermore, by using the pivot algorithm even more evidence was found, that this indeed holds true.

Looking at the amplitudes A of the critical exponent fits $Y = AN^{2\nu}$, the amplitudes get bigger the more we restrict the walks with periodic boundary conditions. Therefore, the end to end distance is bigger the more periodic boundary conditions we propose. This also holds true for the gyration radius and could be explained by simply considering collapsed walks, which are overall less likely to be periodic.

By taking a lot of inspiration from Nathan Clisby [15] method, the connective constant μ was determined for every proposed category of periodic self-avoiding walks. It seems, that the introduced acceptance rate $\tilde{B}_N$ does converge towards zero in the third category. Therefore, this could imply that this method can not be used for larger walks with a lot of periodic restrictions.

Using this acceptance rate, the coefficients c_N were calculated. As expected, these coefficients scale according to $c_N \propto A\mu^N N^{\gamma-1}$, this still holds true for any periodic boundary condition proposed.

The connective constants get smaller as one restricts the self-avoiding walks with more periodic boundary conditions, as was done by introducing three different categories. This behaviour was expected since less periodic walks exist and furthermore the number of walks c_N and the connective constant are directly proportional.

7 Future prospects

There is a lot of work done for the normal self-avoiding walk, what can not be said for the periodic case. It would be quite interesting to see some of the same approaches for the periodic case. Especially considering other categories than chosen in this bachelor's thesis.

It would be especially interesting to see the simulations done in higher dimensions and/or on other lattice types. Honeycomb lattice could be quite informative, since the exact connective constant is known for non-interacting walks on it and maybe it is possible to get some mathematically rigorous results in the periodic case [17].

Implementation of a faster and more efficient pivot algorithm would enable the simulation of way longer walks [4]. Furthermore, the error of the resulting properties could be scaled down. Maybe even the independence of the critical exponent with regards to periodicity could be shown with greater certainty.

One could also look at phase transitions of this model by implementing a self-attractive case, which could also be quite interesting to see.

Overall, there are a lot of different quite fascinating cases, which can be considered in the future. All of these would be nice additions to the as of now very narrow contemplations of periodic self-avoiding walks.

References

[1] A R Conway, I G Enting, and A J Guttmann. "Algebraic techniques for enumerating self-avoiding walks on the square lattice". In: *Journal of Physics A: Mathematical and General* 26.7 (Apr. 1993), pp. 1519–1534. DOI: 10.1088/0305-4470/26/7/012. URL: http://dx.doi.org/10.1088/0305-4470/26/7/012.

[2] Nathan Clisby and Iwan Jensen. "A new transfer-matrix algorithm for exact enumerations: self-avoiding polygons on the square lattice". In: *Journal of Physics A: Mathematical and Theoretical* 45.11 (Feb. 2012), p. 115202. DOI: 10.1088/1751-8113/45/11/115202. URL: https://doi.org/10.1088%2F1751-8113%2F45%2F11%2F115202.

[3] N. Madras and A. D. Sokal. "The Pivot Algorithm: A Highly Efficient Monte Carlo Method for the Self-Avoiding Walk". In: *J. Stat. Phys.* 50 (1988), pp. 109–186.

[4] Nathan Clisby. "Efficient Implementation of the Pivot Algorithm for Self-avoiding Walks". In: *Journal of Statistical Physics* 140.2 (May 2010), pp. 349–392. ISSN: 1572-9613. DOI: 10.1007/s10955-010-9994-8. URL: http://dx.doi.org/10.1007/s10955-010-9994-8.

[5] Iwan Jensen. "A parallel algorithm for the enumeration of self-avoiding polygons on the square lattice". In: *Journal of Physics A: Mathematical and General* 36.21 (May 2003), pp. 5731–5745. DOI: 10.1088/0305-4470/36/21/304. URL: https://doi.org/10.1088%2F0305-4470%2F36%2F21%2F304.

[6] I G Enting. "Generating functions for enumerating self-avoiding rings on the square lattice". In: *Journal of Physics A: Mathematical and General* 13.12 (Dec. 1980), pp. 3713–3722. DOI: 10.1088/0305-4470/13/12/021. URL: https://doi.org/10.1088%2F0305-4470%2F13%2F12%2F021.

[7] N. Madras and G. Slade. *The Self-Avoiding Walk*. Probability and Its Applications. Birkhäuser Boston, 2013. ISBN: 9781461241324. URL: https://books.google.de/books?id=JsoFCAAAQBAJ.

[8] A. R. Conway and A. J. Guttmann. "Square Lattice Self-Avoiding Walks and Corrections to Scaling". In: *Phys. Rev. Lett.* 77 (26 Dec. 1996), pp. 5284–5287. DOI: 10.1103/PhysRevLett.77.5284. URL: https://link.aps.org/doi/10.1103/PhysRevLett.77.5284.

[9] I G Enting. "Generating functions for enumerating self-avoiding rings on the square lattice". In: *Journal of Physics A: Mathematical and General* 13.12 (Dec. 1980), pp. 3713–3722. DOI: 10.1088/0305-4470/13/12/021. URL: https://iopscience.iop.org/article/10.1088/0305-4470/13/12/021.

[10] A. Kloczkowski and R. L. Jernigan. “Transfer matrix method for enumeration and generation of compact self-avoiding walks. I. Square lattices”. In: *The Journal of Chemical Physics* 109.12 (1998), pp. 5134–5146. DOI: 10.1063/1.477128. URL: https://doi.org/10.1063/1.477128.

[11] M.E.J. Newman and G.T. Barkema. *Monte Carlo Methods in Statistical Physics*. Clarendon Press, 1999. ISBN: 9780198517979. URL: https://books.google.de/books?id=J5aLdDN4uFwC.

[12] Makoto Matsumoto and Takuji Nishimura. “Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator”. In: *ACM Trans. Model. Comput. Simul.* 8.1 (Jan. 1998), pp. 3–30. ISSN: 1049-3301. DOI: 10.1145/272991.272995. URL: https://doi.org/10.1145/272991.272995.

[13] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* (2020). DOI: https://doi.org/10.1038/s41592-019-0686-2.

[14] Le Guillou, J.C. and Zinn-Justin, J. “Accurate critical exponents from the ansion”. In: *J. Physique Lett.* 46.4 (1985), pp. 137–141. DOI: 10.1051/jphyslet:01985004604013700. URL: https://doi.org/10.1051/jphyslet:01985004604013700.

[15] Nathan Clisby. “Calculation of the connective constant for self-avoiding walks via the pivot algorithm”. In: *Journal of Physics A: Mathematical and Theoretical* 46.24 (May 2013), p. 245001. DOI: 10.1088/1751-8113/46/24/245001. URL: http://dx.doi.org/10.1088/1751-8113/46/24/245001.

[16] I Jensen and A J Guttmann. “Self-avoiding walks, neighbour-avoiding walks and trails on semiregular lattices”. In: *Journal of Physics A: Mathematical and General* 31.40 (Oct. 1998), pp. 8137–8145. DOI: 10.1088/0305-4470/31/40/008. URL: http://dx.doi.org/10.1088/0305-4470/31/40/008.

[17] Hugo Duminil-Copin and Stanislav Smirnov. *The connective constant of the honeycomb lattice equals* $\sqrt{2+\sqrt{2}}$. 2010. arXiv: 1007.0575 [math-ph].

Statement of Authorship

I hereby declare that I am the sole author of this bachelor thesis and that I have not used any sources other than those listed in the bibliography and identified as references. I further declare that I have not submitted this thesis at any other institution in order to obtain a degree.

April 7, 2020

Sebastian Mohr