Monte Carlo Hand-in Assignment 2

Eliot Montesino Petrén, el6183mo-s, 990618-9130, LU Faculty of Engineering F19 , eliot.mp99@gmail.com,
Kasper Nordenram, ka0884no-s, 990212-7910, LU Faculty of Engineering F19

I. Introduction

This assignment focuses on the exploration of self-avoiding walks in multi-dimensional lattices through the application of Monte Carlo methods, particularly sequential Monte Carlo (SMC) techniques. It's part of an academic exploration within the field of Mathematical Statistics aimed at understanding complex stochastic processes. The objective is to estimate the number of possible self-avoiding paths and other related parameters, such as the connective constant, which are crucial for theoretical and applied physics, especially in the study of polymer chains. This exploration requires a blend of theoretical knowledge and practical application, providing insights into the use of stochastic processes in solving real-world problems.

Part1: Self-avoiding walks in $\mathbb{Z}^d$

Definition from task:

A self-avoiding walk (SAW) is a sequence of moves on a lattice that does not visit the same point more than once. In $\mathbb{Z}^d$, the set $S_n \subseteq \mathbb{Z}^d$ of possible such walks of length n is formally given by

$$\begin{aligned} S_n(d) = \{ & x_{0:n} \in \mathbb{Z}^d(n+1) : x_0 = 0, \\ & \|x_k - x_{k-1}\| = 1, x_\ell \neq x_k, \forall 0 \le \ell < k \le n \} \end{aligned}$$

To compute the number $c_n(d) = |S_n(d)|$ of possible such walks is, when n is large, considered to be a very challenging problem in enumerative combinatorics. The aim of this home assignment is to solve this problem using sequential Monte Carlo (SMC) methods.

1: Why is it that for all $n \geq 1$ and $b \geq 1$ the following is true?

$$c_{n+m}(d) \le c_n(d) c_m(d) \tag{1}$$

Intuitively eq 1 makes sense because when you concatenate a walk of length n with a walk of length m, some of the resulting walks will not be self-avoiding since the second walk could intersect with the path taken by the first walk. Formally, we can write that

$$S_{n+m}(d) \subset S_n(d) \times S_m(d) \tag{2}$$

where the subset relation is strict for almost all n, m, and d. The relation holds because all $n+m$-long walks have a decomposition into an n-long and an m-long walk (note that the order matters, so there are not two decompositions). We also see that in many cases, like mentioned, there are elements of the set to the right that are not part of the set to the left, if the composition of the walks intersects itself.

2: We know from **1** that the sequence $(c_n(d))_{n \geq 1}$, representing the number of self-avoiding walks of length n in a d-dimensional lattice, satisfies the condition:

$$c_{n+m}(d) \le c_n(d) c_m(d) \tag{3}$$

for any positive integers n and m. To apply Fekete's lemma, we first need to show that $(c_n(d)^{1/n})_{n \geq 1}$ is a subadditive sequence.

Taking the logarithm of this, which preserves the inequality since all terms are larger than or equal to 1, we get

$$\log c_{n+m}(d) \le \log c_n(d) + \log c_m(d) \tag{4}$$

Thus, $\log c_n(d)$ is a subadditive sequence, and Fekete's lemma holds, giving that the following limit exists

$$\lim_{n \to \infty} \frac{1}{n} \log c_n(d) := \log \mu_d. \tag{5}$$

We can rewrite $\log \frac{1}{n} c_n(d)$ to $\log c_n(d)^{\frac{1}{n}}$ resulting in

$$\log \mu_d = \lim_{n \to \infty} \log c_n(d)^{\frac{1}{n}}. \tag{6}$$

Since the logarithm is continuous for large numbers (in fact strictly larger than 0), we can move it outside of the limit, getting

$$\log \mu_d = \log \lim_{n \to \infty} c_n(d)^{\frac{1}{n}}. \tag{7}$$

We now see that the limit $\lim_{n \to \infty} c_n(d)^{\frac{1}{n}}$ exists. Finally, taking the exponent of this, we get

$$\mu_d = \lim_{n \to \infty} c_n(d)^{\frac{1}{n}}. \tag{8}$$

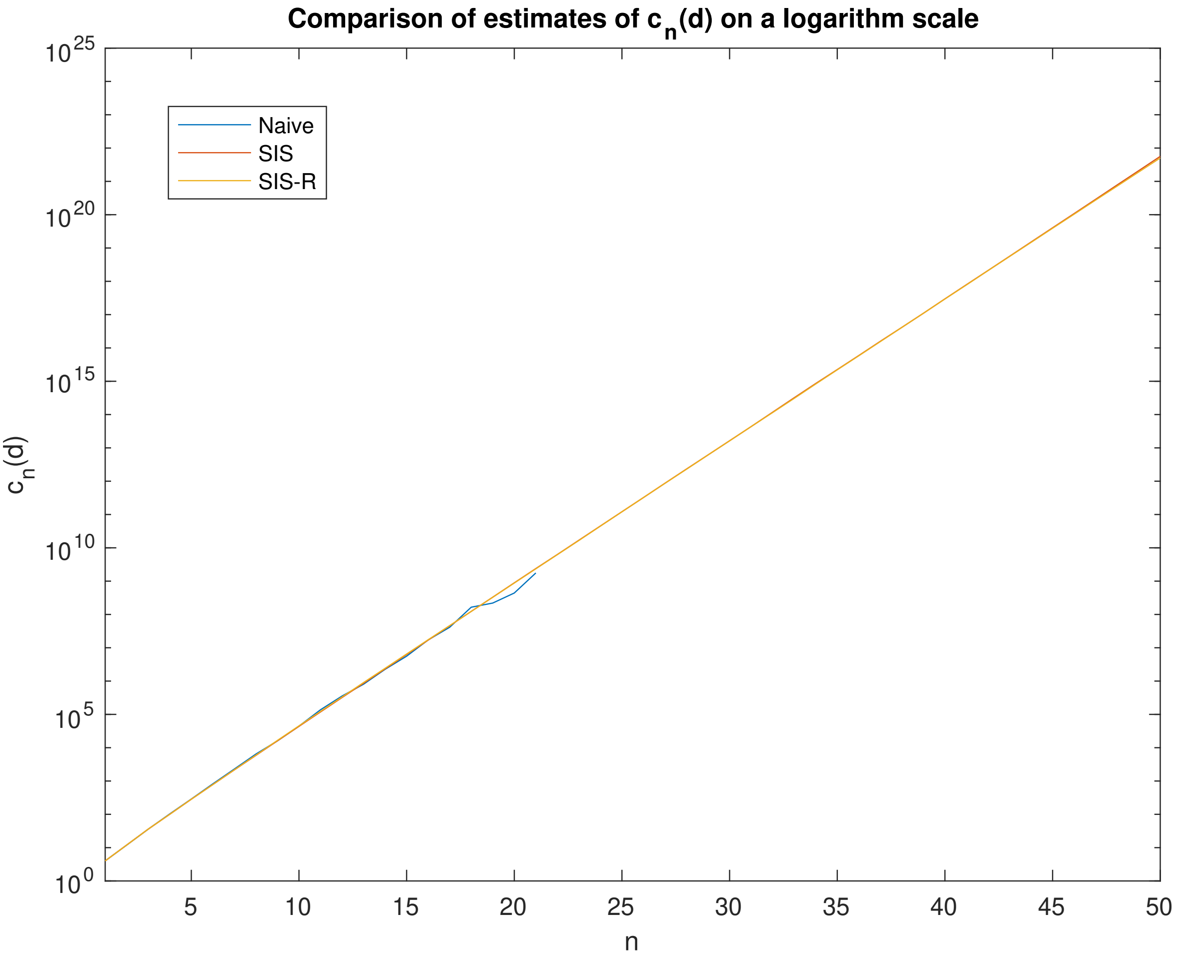

Fig. 1. Estimates of the number of self-avoiding walks. Naive refers to the solution of **3**, SIS to **4**, and SIS-R to **5**.

TABLE I
Estimates in **3**. N = 2500

n	$c_n(2)$
1	4.0
2	11.92
3	35.89
4	103.01
5	287.95
6	$0.0822 \cdot 10^4$
7	$0.2294 \cdot 10^4$
8	$0.6449 \cdot 10^4$
9	$1.6043 \cdot 10^4$
10	$4.4460 \cdot 10^4$
11	$0.0014 \cdot 10^8$
12	$0.0036 \cdot 10^8$
13	$0.0081 \cdot 10^8$
14	$0.0225 \cdot 10^8$
15	$0.0558 \cdot 10^8$
16	$0.1718 \cdot 10^8$
17	$0.4123 \cdot 10^8$
18	$1.6493 \cdot 10^8$
19	$2.1990 \cdot 10^8$
20	$4.3980 \cdot 10^8$

TABLE II
Estimates in **4**. N = 2500

n	$c_n(2)$
1	4.0
2	12.0
3	36.0
4	100.34
5	283.74
6	$0.0782 \cdot 10^4$
7	$0.2158 \cdot 10^4$
8	$0.5869 \cdot 10^4$
9	$1.6060 \cdot 10^4$
10	$4.3696 \cdot 10^4$
11	$0.0012 \cdot 10^8$
12	$0.0032 \cdot 10^8$
13	$0.0086 \cdot 10^8$
14	$0.0232 \cdot 10^8$
15	$0.0626 \cdot 10^8$
16	$0.1697 \cdot 10^8$
17	$0.4575 \cdot 10^8$
18	$1.2357 \cdot 10^8$
19	$3.3014 \cdot 10^8$
20	$8.8998 \cdot 10^8$
30	$1.6389 \cdot 10^{13}$
40	$2.9746 \cdot 10^{17}$
50	$5.6417 \cdot 10^{21}$

3: In Appendix III-A, self-normalized importance sampling is summarized. In this case, the instrumental distribution g_n is such that $g_n = \prod_{k=1}^{n} g_k(x_k|x_{k-1})$ and $g_k(x_k|x_{k-1})$ just a discrete distribution with weight $\frac{1}{4}$ on each of the 4 neighbors. Each sample is then simply a random walk, which might (very likely) have density 0 in the target distribution f_n. In the formulation derived in Appendix III-A, we see that q is the set of all possible walks, which we can call $N_{\text{possible walks}}$. The size of this set is 4^n, since each step has 4 choices. We thus get

$$c_n(2) \simeq 4^n \frac{\sum_{k=1}^{N} \mathbf{1}_{S_n(2)}(X_k)}{N}, \tag{9}$$

so we simulate a lot of random walks (the samples X_k), count which are self-avoiding, and then scale this by the total number of random walks. This approach is computationally inefficient since we will keep simulating from walks that already have density zero. We implement it by simulating a lot of random walks of length n, and then go through them one length at a time. The results with 2500 particles are shown in Figure 1. The instability seen at high n arises because the number of self-avoiding walks decreases as n increases, and when this number gets too small, there is a loss of "resolution". When the line finally disappears, this is because no self-avoiding walks remain. This is a major problem with this method - the fact that the instrumental distribution has much larger support than the target distribution for large n, resulting in the need for very many samples. Some numerical values are shown in Table I.

4: We now use self-avoiding walks as instrumental distribution, so new points are sampled from the free neighbors, and when no free neighbors exist, we just keep repeating the last point. This solution is more statistically and computationally efficient than the one in **3**, because we have a better match between the instrumental and target distributions, and also save resources by not simulating random numbers when a walk is already not self-avoiding and has 0 weight. We now, however, face the problem of not being able to analytically quantify the set q. What we can do, however, is do the estimation sequentially, as described in Appendix III-A. The results with 2500 particles are shown in Figure 1. The results look good, appearing to grow linearly for all investigated n. Some numerical values are shown in Table II.

5: This solution is very similar to the one in **4**, but we resample the points based on their weight. This is statistically more efficient for larger n, as we keep more high-weight points which thus contain a lot of information. In **4**, we can get the problem that a lot of walks get stuck and thus get weight 0, resulting in fewer and fewer points with which to calculate the average. The results with 2500 particles are shown in Figure 1. The results are only slightly different from those in **4**, with the difference growing slightly for larger n. Some numerical values are shown in Table III.

TABLE III
ESTIMATES IN **5**. N = 2500

n	$c_n(2)$
1	4.0
2	12.0
3	36.0
4	100.14
5	283.55
6	$0.0781 \cdot 10^4$
7	$0.2176 \cdot 10^4$
8	$0.5984 \cdot 10^4$
9	$1.6558 \cdot 10^4$
10	$4.5243 \cdot 10^4$
11	$0.0012 \cdot 10^8$
12	$0.0033 \cdot 10^8$
13	$0.0090 \cdot 10^8$
14	$0.0242 \cdot 10^8$
15	$0.0650 \cdot 10^8$
16	$0.1742 \cdot 10^8$
17	$0.4623 \cdot 10^8$
18	$1.2309 \cdot 10^8$
19	$3.2821 \cdot 10^8$
20	$8.7723 \cdot 10^8$
30	$1.6273 \cdot 10^{13}$
40	$2.9527 \cdot 10^{17}$
50	$5.0421 \cdot 10^{21}$

6: We solved this by fitting the function

$$c_n(2) = A_2 \mu_2^n n^{\gamma_2 - 1}, \tag{10}$$

to the found $c_n(2)$ in task 5 (numerically, and in the least squares sense). We used the 21th to the 50th coefficients, and 2500 particles. The smaller coefficients have been excluded to minimize bias, as the model is asymptotic in n. The fitting was done in the logarithm-10 domain, to make the error sizes more comparable (and the fitting thus more stable), so

$$\log_{10} c_n(2) = \log_{10} A_2 + n \log_{10} \mu_2 + (\gamma_2 - 1) \log_{10} n. \tag{11}$$

The process was repeated 25 times, resulting in the parameter histograms seen in Figure 2. Clearly, there is some variance in the estimates. This is likely due to variance in the estimates, due to the randomness of Monte-Carlo methods. The most sensitive parameter is μ_2, as the model is exponential in it, which explains why it has the smallest variance. The parameters can, to some degree, compensate for each other, which also explains the relatively high variance. Some variance may also be due to numerical and fitting issues. There also seems to be some remaining bias compared to the analytic γ_2-value of 43/32, but this is not so surprising, given the still quite small n-values.

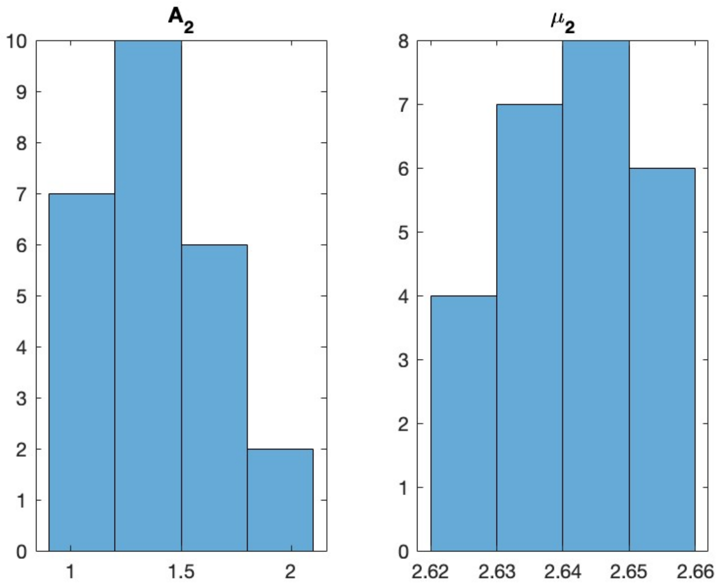

Fig. 2. Parameter estimate histograms from **6**.

7: We are to show why the following bound is true.

$$d \leq \mu_d \leq 2d-1 \tag{12}$$

The bound can be shown in two ways either transforming μ_d up to $c_n(d)$ and argue why the resulting transformed upper and lower bound are true or it can be shown the other way around by directly arguing for an upper and a lower bound for the size of the set of all possible SAW:s ($c_n(d)$) and then transforming it down to μ_d, and we will go with the latter since we already know the bounds following a "proof" inspired by a paper by Gordon Slade [1]. The transformation between $c_n(d)$ and μ_d is made with was shown in task 2 utilizing the equality with the limit, please see eq 8. Let us begin by defining the lower bound.

Let $c_n(d)$ be the number of all possible SAW:s for a given length of the walk n and for a given dimension d. A lower bound for this number is defined in eq 13.

$$d^n \leq c_n(d) \tag{13}$$

The lower bound d^n combinatorically represents the permutation of all possible walks of length n that is made purely in either a given positive or negative direction of d.

Why does the lower bound hold true? By being allowed to only walk in a positive or negative direction, the walk can impossibly cross itself thus fulfilling a valid SAW, but the permutation also clearly does not represent all possible SAW:s thus fulfilling $d^n \leq c_n(d)$.

The result is quite intuitive to understand. When the walk is only allowed to walk in either a positive or negative direction it is restricted from making a "rotation" and a crossing requires that the walk's forward direction can be rotated four times in a plane (in two dimensions), in the given space $\mathbb{Z}^d$, $90° * 4 = 360°$ to form a *closed loop*, and this is equivalent to being allowed to walk in both d and $-d$ at least once, for a given dimension. Three rotations $3 * 90° = 180°$ in a plane corresponds to a walk that is made at least 1 step in both d and $-d$ in one direction. The only other way for a walk to be an invalid SAW is of course the trivial invalid SAW of taking a step in the opposite direction of the latest step made.

An upper bound is defined in eq 14.

$$c_n(d) \leq (2d)(2d-1)^{n-1} \tag{14}$$

The upper bound combinatorically represents taking the first step in any of the possible directions available of a total of $2d$ choices and then taking the rest of the steps of the walk of $n-1$ steps in any of the directions available that is not in the opposite direction of the last step taken of a total of $2d-1$ choices, finally forming a total of $2d(2d-1)^{n-1}$ permutations.

Why does the upper bound hold true? The restriction of not being able to walk in the opposite direction of the last step taken discounts all the trivial invalid SAW:s of taking a step back to the point it just was at. Following the argumentation from the lower bound, the permutation does not however restrict the walk to not form *closed loops* (that are invalid SAW:s) but it the permutation does represent all possible SAW:s of length n as well, thus the permutation must be able to be equal or larger than the size of the set of all possible SAW:s $c_n(d) \leq (2d)(2d-1)^{n-1}$.

There are cases where the upper bound permutation is always the same size as $c_n(d)$. When $1 \leq n \leq 3$ the walk is not able to form a closed loop and for these n always $c_n(d) = (2d)(2d-1)^{n-1}$. Let us also inspect the special case when $d = 1$. In thus special case then the relation is always equal as well $c_n(1) = (2 * 1)(2 * 1 - 1)^{n-1} = 2$. The space $\mathbb{Z}^1$ is then a line and the upper bound represents choosing a positive or negative direction and then just walking in a a straight line. This is equal to all possible SAW:s no matter the length of the walk $1^n = c_n(1)$ as forming a closed loop is impossible and the only way of making an invalid SAW is taking a step in the opposite direction of the one last taken.

We have now obtained eq 15.

$$d^n \leq c_n(d) \leq (2d)(2d-1)^{n-1} \tag{15}$$

Let apply the operation and the limit in eq 8 on eq 15 and we obtain eq 16.

$$\lim_{n\to\infty} d^{n/n} \leq \mu_d \leq \lim_{n\to\infty} (2d)^{1/n}(2d-1)^{(n-1)/n} \tag{16}$$

$$d \leq \mu_d \leq \lim_{n\to\infty} \frac{2d^{1/n}}{(2d-1)^{1/n}}(2d-1) \tag{17}$$

$$d \leq \mu_d \leq 2d-1 \tag{18}$$

Where in eq 16 the limit in eq 8 is applied to $c_n(d)$ giving the middle bound, and where in eq 17 the lower bound is given since the limit is irrelevant for when the exponent $d^{n/n} = d$ and between eq 17 and eq 18 the limit is applied to the fraction giving $\lim_{n\to\infty} \frac{2d^{1/n}}{(2d-1)^{1/n}} = 1$ since the nominator and denominator grow asymptotically equally. This marks the end of this task and the general bound in eq 18 is shown.

8: We are tasked to show that

$$A_d \geq 1 \text{ for } d \geq 5. \tag{19}$$

We can utilize the given conjecture in eq 20 (that also holds for $d \geq 5$) and simplify it since we already assume that $d \geq 5$. This produces eq 21.

$$c_n(d) \sim \begin{cases} A_d\mu_d^n n^{\gamma_d - 1} & d = 1, 2, 3,\ d \geq 5 \\ A_d\mu_d^n (\log(n))^{1/4} & d = 4 \end{cases} \quad \text{as } n \to \infty, \tag{20}$$

$$c_n(d) \sim A_d\mu_d^n n^{\gamma_d - 1}, d \geq 5,\ \text{as } n \to \infty \tag{21}$$

And this is further simplified as included in the conjecture it is proven that $\gamma_d = 1$ for $d \geq 5$ and thus $n^{\gamma_d - 1} = n^0 = 1$. We obtain eq 22

$$c_n(d) \sim A_d\mu_d^n, d \geq 5,\ \text{as } n \to \infty \tag{22}$$

This can be inserted to the previously shown inequality from eq 1 that is repeated in eq 23 asserting a new total length of the SAW as $n + m$ where these lengths are also assumed in the same equation.

$$c_{n+m}(d) \leq c_n(d)c_m(d) \quad \text{for all } n \geq 1, m \geq 1. \tag{23}$$

Put eq 22 in eq 23 and obtain eq 24 (observe that eq 23 is valid for all $n \geq 1, m \geq 1$ and $(n+m) \to \infty$ is valid for this)

$$\begin{aligned} A_d\mu_d^{n+m} \leq A_d\mu_d^n A_d\mu_d^m \quad &= A_d^2\mu_d^{n+m}, \\ d \geq 5, n \geq 1, m \geq 1 \text{ and } (n+m) \to \infty \end{aligned} \tag{24}$$

We can simplify the inequality in eq 24 to eq 25 that is invariant of the lengths of walk n and m.

$$A_d \leq A_d^2, d \geq 5 \tag{25}$$

Eq 25 at first hand gives that $|A_d| \geq 1$ but we also have to realize that A_d must be positive, finally yielding $A_d \geq 1$ for $d \geq 5$. A_d must be positive as negative values of it would correspond to the possibility of negative values of $c_n(d)$ in the conjecture in eq 21, which is not allowed as you cannot have a negative number of possible SAW:s.

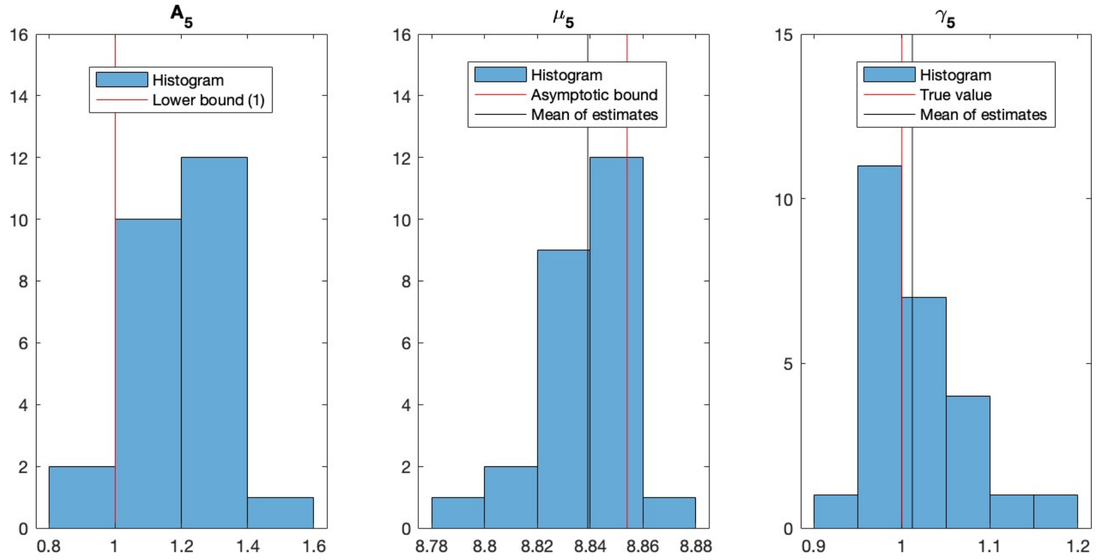

Fig. 3. Parameter estimate histograms from **9**.

9: The method from **6** has been repeated here for $d = 5$. The results have been plotted with some of the bounds from **7** and **8** as references. The μ_5-lower and upper bounds of $d = 5$ and $2d - 1 = 9$ have been excluded as they are so far outside the found range of values. We can see that some A_5-values are below the lower bound, but given the high variance in the estimate, this is not so worrying. We see some remaining bias in the γ_5-estimate but, once again, this is not so surprising since it is an asymptotic result and we have only gone up to $n = 50$.

PART2: FILTER ESTIMATION OF NOISY POPULATION MEASUREMENTS

10

We have the model

$$X_{k+1} = R_{k+1} X_k (1 - X_k) \tag{26a}$$

$$X_0 \in U(C, D) \tag{26b}$$

$$R_{k+1} \in U(A, B), \mathrm{iid} \tag{26c}$$

$$Y_k | X_k = x \in U(Gx, Hx) \tag{26d}$$

where $k = 0, 1, 2, \ldots$, and the scalar parameters are $A = 0.8$, $B = 3.8$, $C = 0.6$, $D = 0.99$, $G = 0.8$ and $H = 1.25$. The aim is to estimate the filter expectation, $\mathbb{E}\{X_k | Y_{0:k}\}$ up to $k = n$. The employed method is SISR (sequential importance sampling with resampling), however, we can sample directly from the target distribution, so the importance and the target distribution coincide.

a: The results for one run are plotted in Figure 4. Clearly, the estimates are all very close to each other. The average root mean square errors for the estimates over 100 runs are 0.10263, 0.10256, and 0.10251, respectively. As expected, the error decreases slightly with increasing N. Depending on the application, $N = 500$ could be seen as the most reasonable choice, as it requires less computational power than the other choices, with very small performance improvement. However, if the small increase in performance is critical and computational resources are abundant, it might be worth the extra effort to go up to 1000 or 10000 points.

b: By sorting the simulated particles, we can approximate the 95% confidence intervals for all X_k, as seen in Figure 5. These plots have been produced from the same simulation as the one in Figure 4.

Clearly, the intervals are all very similar, and the number of points outside the intervals are in all cases slightly lower than expected given a 95% confidence interval, so they are all likely close to optimal. As in **a**, $N = 500$ can be considered the most reasonable choice as it saves, computational power with just a small loss of performance, but this would, again, depend on the application.

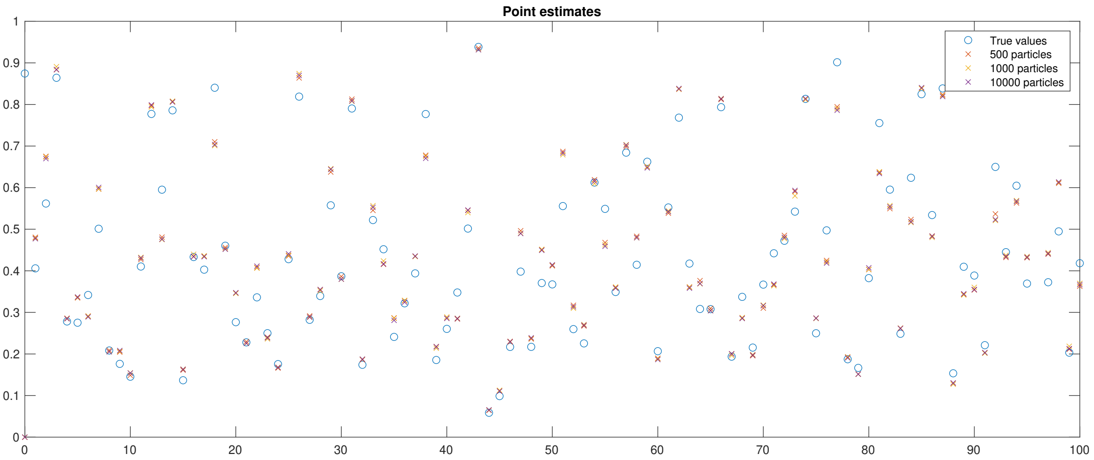

Fig. 4. Point estimates in noisy population measurements.

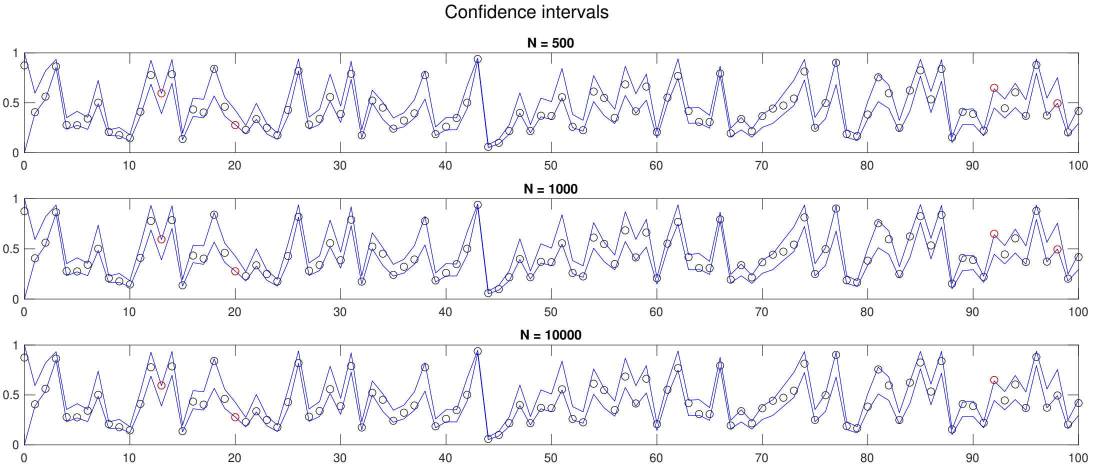

Fig. 5. Estimated 95% confidence intervals in noisy population measurements. Red dots mark points outside the confidence interval.

TABLE IV

95% Confidence Interval in noisy population measurements. Every 10 th value starting from the first entry. N = 10000.

Entry number (index)	**1**	**11**	**21**	**31**	**41**	**51**	**61**	**71**	**81**	**91**	**101**
Upper Bound	1.0000	0.1818	0.4257	0.4672	0.3521	0.5091	0.2151	0.3889	0.5091	0.4370	0.4490
Lower Bound	0	0.1202	0.2793	0.3060	0.2303	0.3325	0.1442	0.2543	0.3392	0.2864	0.2938
Difference	1.0000	0.0616	0.1464	0.1612	0.1218	0.1766	0.0709	0.1346	0.1699	0.1506	0.1552

II. References

[1] Gordon Slade. *BOUNDS ON THE SELF-AVOIDING-WALK CONNECTIVE CONSTANT*. URL: https://personal.math.ubc.ca/~slade/kahane.pdf.

[2] Magnus Wiktorsson. *Lecture 3, slide 23*. 2024.

[3] Magnus Wiktorsson. *Lecture 6, slide 20*. 2024.

[4] Magnus Wiktorsson. *Lecture 7, slide 19*. 2024.

III. Appendix

A. Self-normalised importance sampling (with resampling)

When using self-normalized importance sampling, an unknown normalizing constant of the distribution f for the random variable X can be approximated by

$$c = \mathbf{E}_g\{\omega(X)\} \simeq \frac{1}{N}\sum_{i=1}^{N}\omega(X_i), \tag{27}$$

where g is an instrumental distribution, from which the X_i:s are drawn. Clearly, the variance of this estimate is smaller, the fewer points where ω is zero are included, but no points where ω is nonzero can be excluded. In the context of self-avoiding random walks, we see that [2]

$$c_n(d) \simeq \frac{1}{N}\sum_{i=1}^{N}\frac{\mathbf{1}_{S_n(d)}(X_i)}{\frac{\mathbf{1}_Q(X_i)}{q}} = q\frac{\sum_{i=1}^{N}\mathbf{1}_{S_n(d)}(X_i)}{N}, \tag{28}$$

where Q is the set of allowed walks in the intrumental distribution, and q is the size of this set. As we can see, this only helps if we can analytically calculate the size of the instrumental set of walks, and we also see that every time $\mathbf{1}_Q(X_i)$ is zero, $\mathbf{1}_{S_n(d)}(X_i)$ has too also be zero, or we get an undefined result.

If we cannot determine the size of q analytically, we may be successful using a sequential implementation. In this case, we know that $q_0 = c_0 = 1$. We can then proceed by generating samples X_i^{r+1} from the distribution $g_{r+1}(x_i^{r+1}|x_i^r)$, and get $X_i^{0:r+1}$ as $(X_i^{0:r}, X_i^{r+1})$. We then get the weights [3]

$$\omega_i^{r+1} = \left[\frac{z_{r+1}(X_{0:r+1})}{z_r(X_{0:r})g_{r+1}(x_i^{r+1}|x_i^r)}\right]\cdot\omega_i^r, \tag{29}$$

where $z_n(x_{0:n}) = \mathbf{1}_{S_n(d)}(X_i)$, and $g_{r+1}(x_i^{r+1}|x_i^r)$ is simply (the number of free neighbors)$^{-1}$, or any number if the number of free neighbors is zero since $z_{r+1}(X_{0:r+1})$ will be 0. We can remove $z_r(X_{0:r})$ from this expression, because if it is zero, also $z_{r+1}(X_{0:r})$ will be zero, and the old weight will be 0 - resulting in a new weight of 0. So we have the expression

$$\omega_i^{r+1} = \frac{z_{r+1}(X_{0:r+1})}{g_{r+1}(x_i^{r+1}|x_i^r)}\cdot\omega_i^r, \tag{30}$$

Each particle (walk) thus gets its own weight. Referring back to Equation 27, we thus see that we can form the estimate of the normalizing constant as the mean of the weights.

If we want to improve the efficiency of this estimate, we can resample from the particles. Using multinomial selection [4], we get

$$\tilde{X}_i^{0:n} = X_j^{0:n} \text{ with probability } \frac{\omega_j^n}{\sum_{l=1}^{N}\omega_l^n}. \tag{31}$$

The weights after resampling are usually all set to 1, however, since we are after the average weight here, this means that we lose all information. We need to set them to the same value, and here it is fitting to use the last average weight, i.e. $c_n(d)$. We can also note that particles with weight zero are never chosen, so we get

$$\omega_i^{r+1} = \frac{z_{r+1}(X_{0:r+1})}{g_{r+1}(x_i^{r+1}|x_i^r)}\cdot c_n(d). \tag{32}$$

To simplify things, we can sequentially calculate the modifiers $\frac{1}{N}\sum_{i=1}^{N}\frac{z_{r+1}(X_{0:r+1})}{g_{r+1}(x_i^{r+1}|x_i^r)}$, and, knowing that $c_0(d) = 1$, we can cumulatively multiply the modifiers to get the estimates. This is less computationally heavy, and keeps the values between in the order of magnitude of 1. The weights without the $c_n(d)$ can also be used for resampling.