

1 Introduction

A simple random walk in both two and three dimensions was implemented in continuous space (i.e. not on a grid). Ensembles of walks of length up to $N = 8192$ positions were generated non-self avoiding and self-avoiding and the effect of ensemble size M and length N on the ensemble average end-to-end square distance and its error observed.

2 Algorithm Description

The algorithm is implemented in continuous space using polar coordinates. For the case of self avoiding random walks, special consideration had to be given to detect and abort a walk which has gotten stuck inside its own geometry. The main random walk algorithm's structure is as follows:

- Start random walk at origin $(0, 0, 0)$; disregard z-axis here and below for 2D case
- Generate random starting polar coordinate angles ϕ, θ
- While random walk has less than N sites:
 - Randomly modify ϕ (2D); ϕ, θ (3D)
 - Calculate next site position using ϕ, r (2D); ϕ, θ, r (3D)
 - If self avoiding:
 - * Check new position doesn't overlap with previous sites
 - If overlap: Increase attempt count & terminate walk if count == limit, else go back to modify ϕ, θ for next candidate position
 - Else if no overlap: Set attempt count to zero; add new position to chain
 - Else if not self avoiding:
 - * Add new position to chain
- return squared end-to-end distance origin \leftrightarrow final position N

3 Results

The program was implemented as described above and submitted with this report.

3.1 Task 1

The error Δ was found to behave as $\Delta \propto \frac{\langle R^2 \rangle}{\sqrt{M}}$ for $M \rightarrow \infty$ as expected and as shown in figure 1. Selecting an ensemble size of $M = 10^4$ thus yields an error of approximately 1%.

Also, as figure 2 shows, the ensemble average squared end-to-end distance behaves as expected with $\langle R^2(N) \rangle \propto N$ for $N \rightarrow \infty$.

3.2 Task 2

Now for a self avoiding random walk in three dimensions, the error behaves as in task 1 as $\Delta \propto \frac{\langle R^2 \rangle}{\sqrt{M}}$ for $M \rightarrow \infty$ as shown in figure 3.

blublub

As figure 4 shows, the ensemble average squared end-to-end distance no longer behaves as in the non-self avoiding 2D where we had $\langle R^2(N) \rangle \propto N$ for $N \rightarrow \infty$, but seemingly as a power law with an exponent of slightly above one. This is reflected by the slope of the straight line in figure 4.

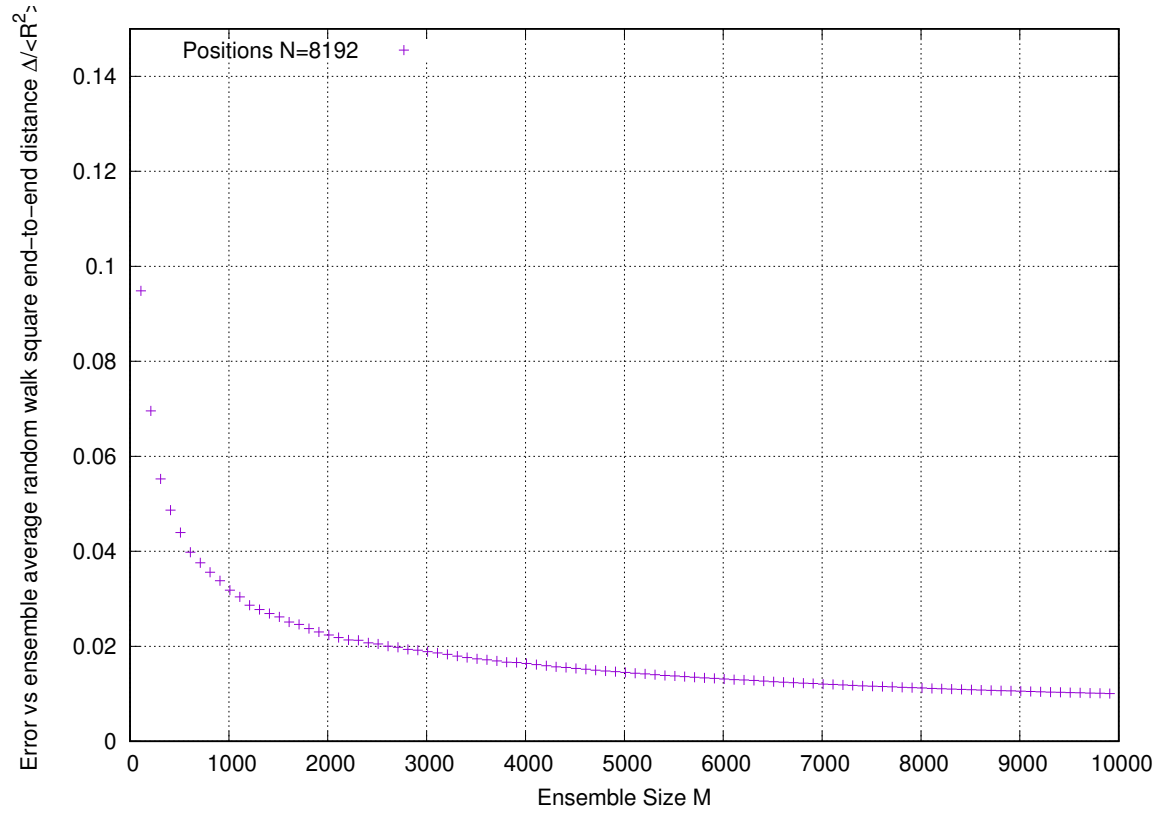


Figure 1: Relative error $\Delta/\langle R^2 \rangle$ vs ensemble size M for non-self avoiding 2D continuous random walk.

4 Discussion

The results were all in line with expectations. To me, the biggest challenge was to write the algorithm such that trapped walks get thrown away when really stuck but not too early as otherwise the chance to obtain complete walks with larger N drops rapidly.

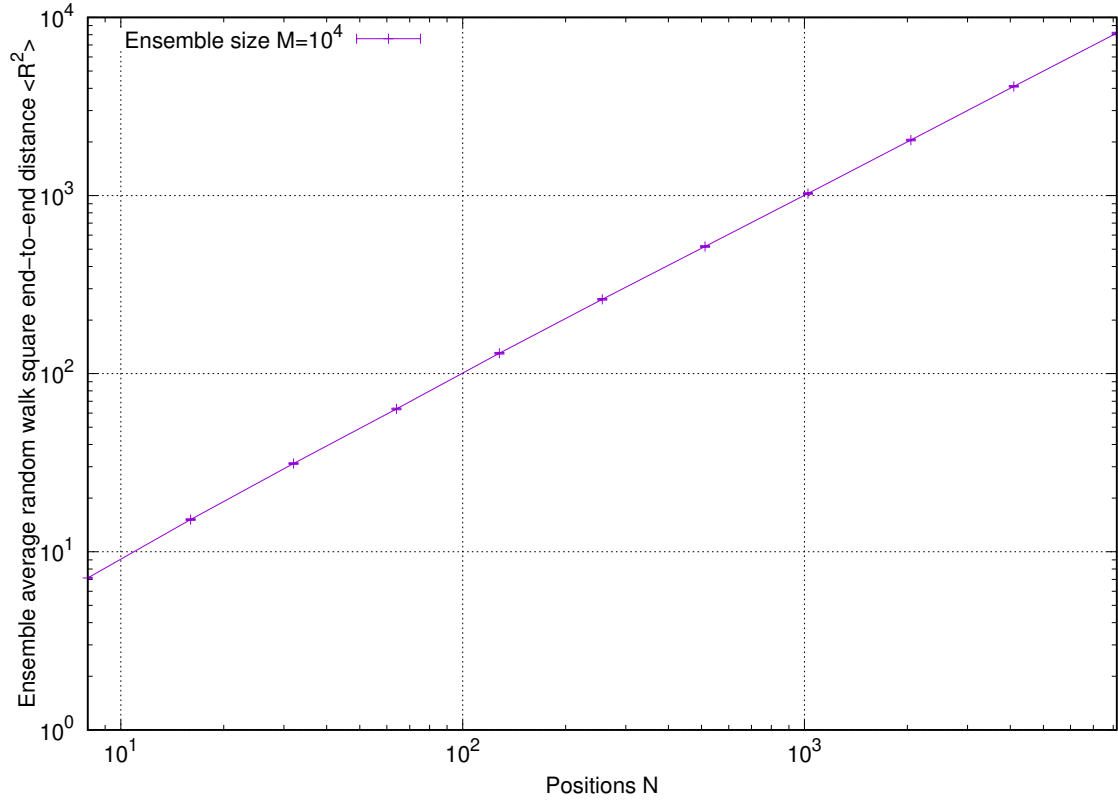


Figure 2: Ensemble average squared end-to-end distance $\langle R^2(N) \rangle$ vs random walk length N for non-self avoiding 2D continuous random walk.

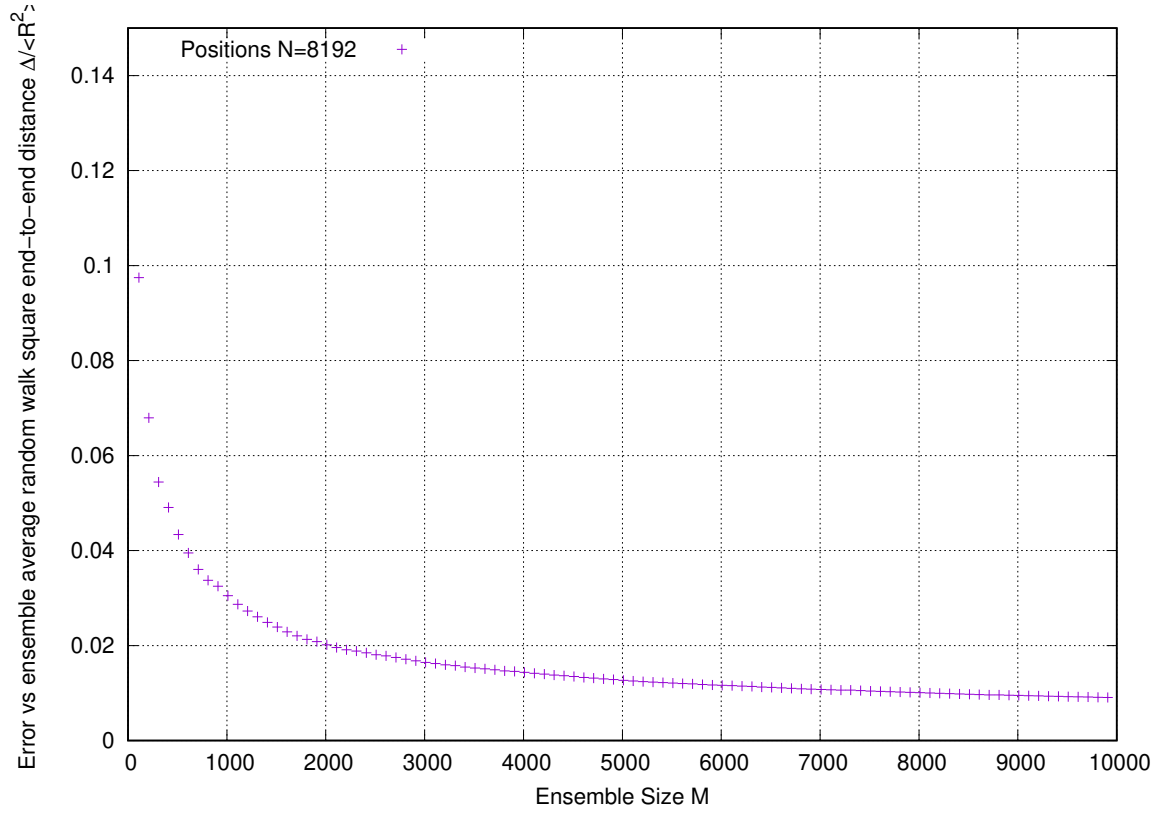


Figure 3: Relative error $\Delta / \langle R^2 \rangle$ vs ensemble size M for self avoiding 3D continuous random walk.

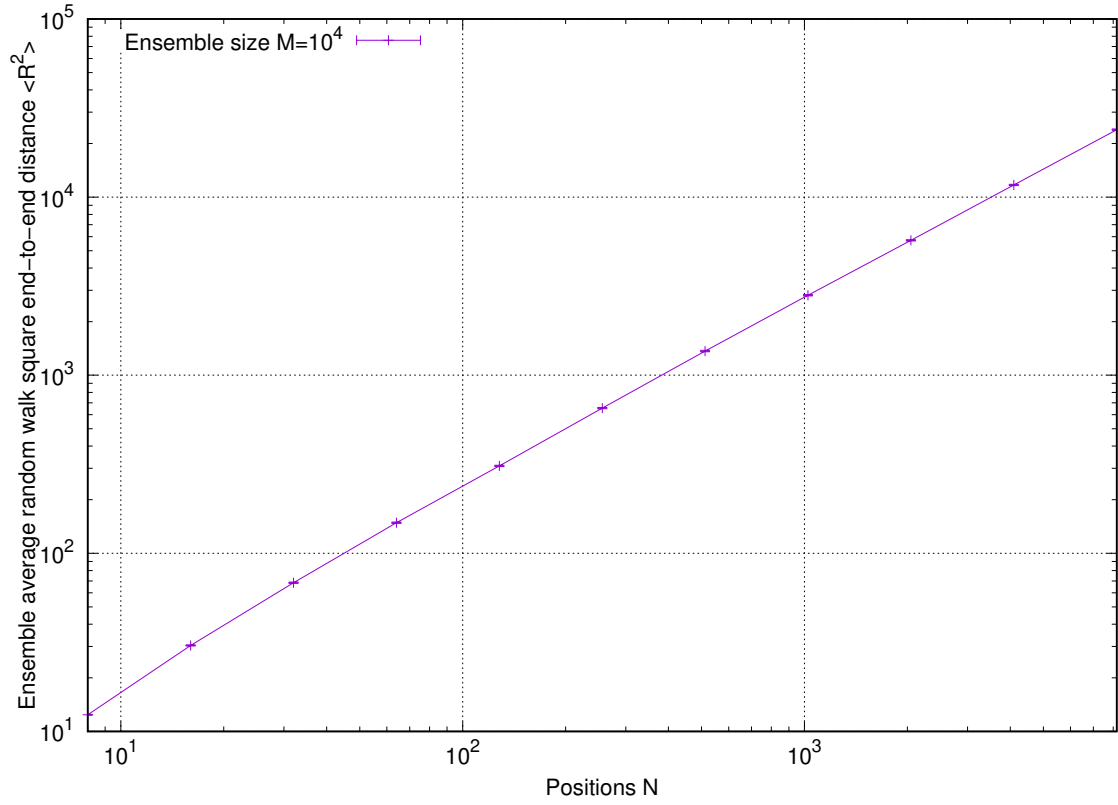


Figure 4: Ensemble average squared end-to-end distance $\langle R^2(N) \rangle$ vs random walk length N for self avoiding 3D continuous random walk.