Random Walk and Self-Avoiding Walk analysis

Patrizio Spada

October 8, 2023

Contents

Abstract

In the theory of stochastic processes [1], **Random Walk** (RW) is an example of stochastic process; it is the formalization of the motion of a point (Random Walker, RWr) who is moving without destination. Starting from this, it is also possible to define a similar walk, **Self-Avoiding Walk** (SAW) that is based on a constraint: the point cannot return on a site that has already been occupied. Analysis of polymers structure is an example of notable applications of RW and SAW.

1 Introduction

Fixed a probability measure $\mathbb{P}$ and fixed s, τ, p and q greater than zero with the condition $p + q = 1$, once defined a sequence $\{X_j\}_{j=0\dots}$ of random variables such that

$$X_0 = 0 \quad \mathbb{P}\text{-a.s.} \tag{1}$$

and

$$X_j = \begin{cases} +s & \mathbb{P}(X_j = +s) = p \\ -s & \mathbb{P}(X_j = -s) = q \end{cases} \qquad j = 1\dots \tag{2}$$

the stochastic process $X(t)$ defined by

$$X(t) = \sum_{j=0}^{\infty} X_j \cdot \theta(t - \tau \cdot j) \quad t \geq 0 \tag{3}$$

is said to be a **one-dimensional Random Walk**. With the same parameter the process $\vec{R}(t)$, with $\vec{R}$ vector of a D-dimensional euclidean vector space,

$$\vec{R}_0 = \vec{0} \quad \mathbb{P}\text{-a.s.} \tag{4}$$

and

$$R^{(i)}(t) = \sum_{j=0}^{\infty} X_j^{(i)} \cdot \theta(t - \tau \cdot j) \qquad t \geq 0 \tag{5}$$

$$X_j^{(i)} = \begin{cases} +s & \mathbb{P}(X_j^{(i)} = +s) = p \\ -s & \mathbb{P}(X_j^{(i)} = -s) = q \end{cases} \qquad \text{for a } i \text{ in } \{1 \dots D\}, \qquad \text{for every } j = 1 \dots \tag{6}$$

$$X_j^{(k)} = 0 \quad \mathbb{P}(X_j^{(i)} = 0) = 1 \qquad \text{for every } k \text{ in } \{1 \dots D\} \text{ with } i \neq k \tag{7}$$

A random walk is said to be **symmetric** if $p = q$.

Now is possible to define the **Self-avoiding walk**:

$$X[m] \equiv \sum_{j=0}^{m} X_j \qquad \text{with } m \text{ in } 0 \dots; \tag{8}$$

$X(t)$ is a one-dimensional SAW iff

$$X[m+1] \neq X[n] \qquad \text{for every } n < m+1. \tag{9}$$

The generalization in a D-dimensional euclidean vector space is straightforward.

1.1 RW and SAW implementation

Usually in codes implementations the condition $s = 1$ and $\tau = 1$ are imposed; the typical applications of the RWr and SAWr time evolution include the following conditions:

- maximum value of time coordinate is fixed;
- maximum values of spacial coordinates are fixed.

The change of the variable t determines the change of X and $\vec{R}$ in (3) and (5): this defines the time evolution of the RWr or SAWr: for each unitary increment of t, only one of the D dimensions is involved: the point moves for a **step** towards a direction randomly chosen. This picture, along with the condition of equiprobability of the two opposite directions, defines a symmetrical **D_RW time evolution** o a symmetrical **D_SAW time evolution** (D stands for D-dimensional and symmetrical is understood) according to the absence or the presence of the condition on the preceding occupied points.

Each possible position is called **site** and the set of sites is a **lattice**: the point moves in a D-dimensional lattice (D is fixed in the problem); the position at time t defines a the **Euclidean distance**:

$$r(t) = \sqrt{\sum_{i=1}^{D} X_i^2(t)}. \tag{10}$$

The variance of (10) is

$$V(t) = \left\langle r^2(t) \right\rangle \tag{11}$$

because the term $\langle r \rangle^2$ vanishes as consequence of the symmetry of the problem.

1.2 RW and SAW first passage time and RW race

In theory of stochastic processes, for a D-dimensional RW or SAW, if each run until the distance N is reached, the **first time passage law** is the function that describes how N is in the relation with the amount of time elapsed t:

$$t = fpt(N). \tag{12}$$

One can expect that for the RWr the relation between the first passage time and the reached distance is

$$\text{first passage time} \propto \langle (\text{reached distance})^2 \rangle. \tag{13}$$

Moreover, one can focus on the problem of producing statistic which describes the probability of RWr to beat a time record on a fixed distance race.

1.3 Report aims explication

In light of what has been treated, numerical simulations are carried out with MATLAB software focused on the study of

1. first passage time law for a 2RWr;
2. first passage time law for a 2SAWr;
3. time records history of a 1RWr race.

2 Methods for the fpt law simulations

Some ideas of the codes are taken from the lectures [2].

2.1 2RWr

The function $t = dist2DRW(N)$ receives the number N (distance from origin), and returns the amount of time used by the 2RWr for reaching the distance N. The algorithm of the function is defined as follow: the integer variables x and y of are declared and initialized to zero; these values are to be updated in the process. A *switch case* is used; it is based on the random choice of the oriented direction of motion: 1 is a step to the right, 2, 3, 4 are respectively to left, up and down. Before entering in the *switch case*, one integer number is randomly extracted between 1 and 4; the number is randomly selected through the function *randi* (*randi(4)*, as written in the code). Till now the process is described only for one step; the total process is constructing by using a *while cycle* in which the aforementioned algorithm is included; the *while cycle* condition is

$$\sqrt{x^2(t) + y^2(t)} < N. \tag{14}$$

2.2 2SAWr

In order to construct $t = dist2DSAW(N)$, the analogous function for the 2SAWr, it is necessary to introduce in the algorithm the information of the points that has already been occupied: this means that, for each point, a generic **RW-allowed** step can be **allowed** or **not allowed**. Knowing that N is the distance (input) that, once reached, determines the end of the trail, the $(2*N+2) \times (2*N+2)$ matrix *mem* (memory) is introduced: all its components are set to zero; the algorithm is based on the idea that every RW-allowed step must be checked by *mem*, which is built through the part of trajectory that has already been traced. The starting point is chosen as $(N+1, N+1)$ and $mem(N+1, N+1)$ is set to 1.

A *while cycle* based on (14) is open; two flag variables $flag0$ and $flag1$ are declared and initialized to 1; another *while cycle* is open (with the condition $flag1 == 1$) after the introduction of *randi(4)* (with the same purpose described in 2.1), a *switch case* is open, according to the logic written in section 2.1. $flag1$ variables allows to check if the 2SAWr is in the following configuration: no movement is allowed (in this case the function returns the $flag1 == 0$); is the condition $flag1 == 1$ fulfilled then a movement variable is updated and the $flag0$ is updated with the corresponding value in *mem*. If it is 1 the movement is forbidden and the nested *while cycle* continues; on the contrary the execution continues after the nested *while cycle* (the SAWr has done a step) and the time variable, the time evolution variables and the suitable *mem* value are updated.

3 Methods for the time records race simulation

The analysis the statistic of records of 1RWr has to focus on the logic of two functions:

- first function, $[ttv, trv, l] = record\_rw\_1D(d)$;
- second function $[avt, avr, lm] = record\_rw\_1D\_mc(d, copies, lim)$.

3.1 First function

1. ttv trial time (between two subsequent records) vector
2. trv rw time (between two subsequent records) vector
3. l number or record

The algorithm of $[ttv, trv, l] = record\_rw\_1D(d)$ is exposed with the following few lines.

1. A 1RWr is run until the RWr reaches the distance of d (input).
2. The amount of time elapsed in the trial is calculated through the function $t = trw\_1D(d)$; if it is different from d then the record can be beaten; otherwise it cannot and the program is ended (the subsequent while cycle cannot be entered).
3. A sequence of random walks is run until the RW with the minimum time value has achieved; at the end of each random walk one of the two following conditions takes place: the current random walk beats/does not beat the current record; the variables update follows this outcome.

3.2 Second function

The second function receives the distance, the number of repetitions to execute and the maximum value (lim) of amount of time in order for the RWr to be taken into account; the function return the two averages vector and the length of this vector.

4 Results and conclusions

4.1 Fpt law simulations

The Monte Carlo simulation has been out carried with 80 repetitions ("copies", as written in the code) of 2RW and 2SAW stopped when a distance is reached; the maximum value of the distance contemplated is 20 ("maxdis" in the code). Hence the supposition (13) appear to be confirmed. Moreover, with a comparison between the two plots (figure (1) and (2)) it is possible to state the 2SAW achieve, on average, the same distance in less amount of time.

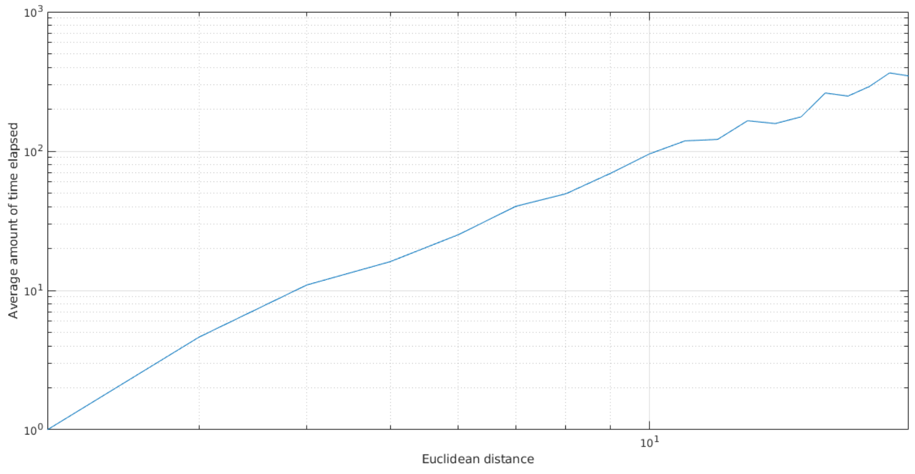

Figure 1: Fpt law simulation results in bi-logarithmic scale for a 2RWr

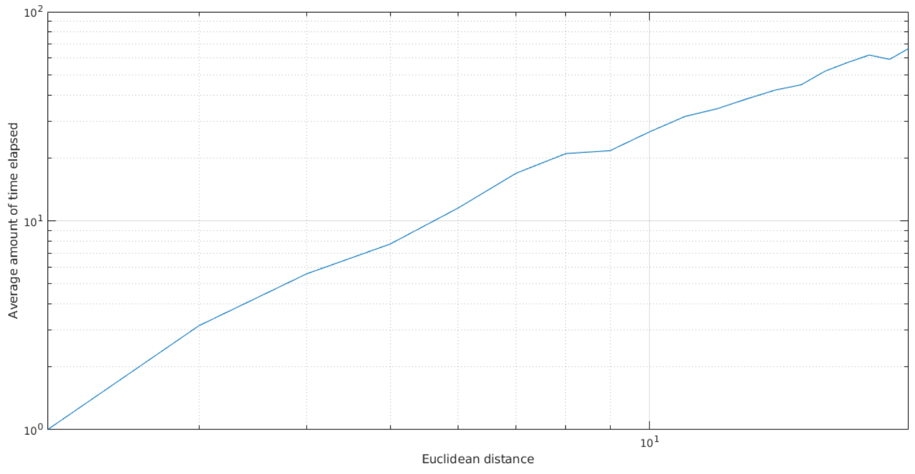

Figure 2: Fpt law simulation results in bi-logarithmic scale for a 2SAWr

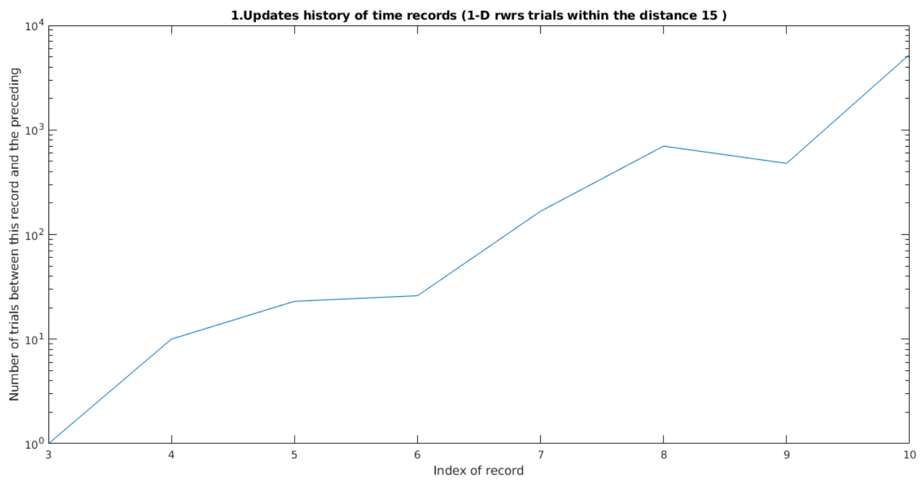

Figure 3

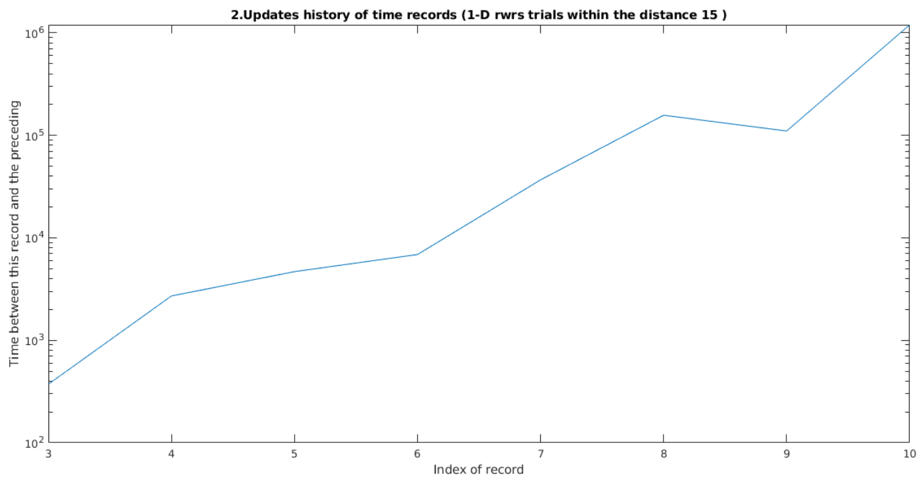

Figure 4

4.2 Time records race simulation

The Monte Carlo simulation has been out carried with 60 repetitions ("copies", as written in the code) of 1RW; value of the distance contemplated is 20 ("d" in the code). As expected, the trend of both curves is increasing.

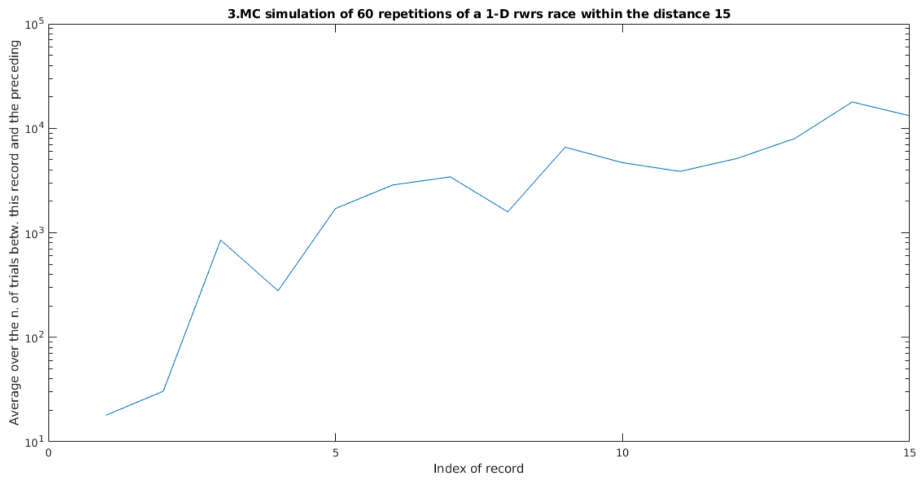

Figure 5

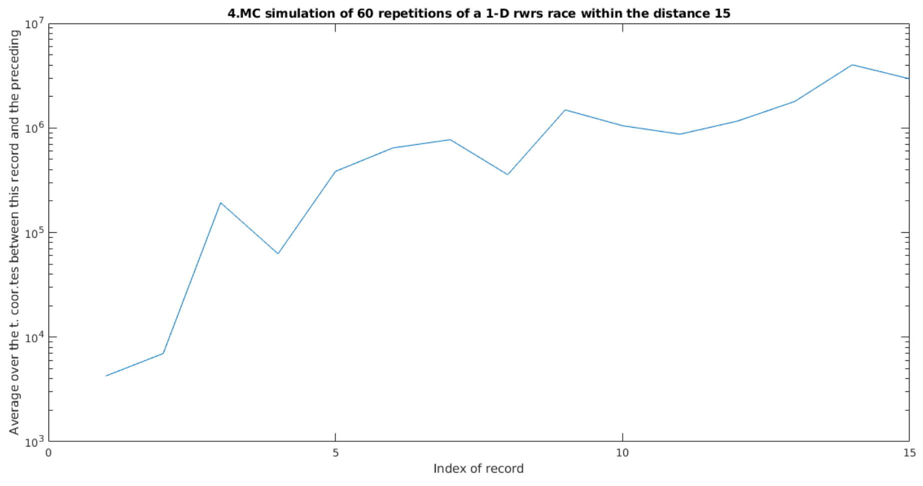

Figure 6

References

[1] Nicola Cufaro Petroni, *Lectures on Probability and Stochastic Processes* (academic year 2017/18).

[2] *Sebastiano Stramaglia lectures* (academic year 2019-20).