

Numerics of the Branching Wiener Sausage

S Amarteifio,...

(Dated: November 1, 2016)

Abstract

The C code can be used to run the Branching Wiener Sausage process for different systems sizes, number of dimensions and boundary conditions. Details of the process and specifics about simulation are described.

1 Overview

The BWS process is a branching and diffusion process. Any particle either hops (diffuses) with probability h , branches with probability σ or becomes extinct with probability ϵ . The critical branching rate is given; we fix $\sigma = \epsilon$. In the current implementation parameters are chosen such that $h + \sigma + \epsilon = 1$ and h is chosen to be high e.g. 0.95. Each of these sub processes is a *Poisson* process. As the system evolves, an exponential waiting time for an event is simulated as $-\ln(1-U)/N$ where $U \sim U(0,1)$ and N is the size of the population. Time is stored as a `long double`. The process is *bosonic*. When branching occurs, offspring are added *locally* i.e. to the same site as the parent.

Typically we choose system sizes of powers of 2 (specifically $2^n - 1$) starting from $2^4 - 1$. We choose an upper bound that accounts for memory usage for higher dimensions. For example, two dimensions can easily exceed 2^{10} while three and five dimensions may struggle depending on the implementation and the hardware.

1.1 Lattice & Boundary Conditions

The D dimensional lattice is mapped to a one dimensional data structure using a suitable mapping $f : \mathcal{R}^D \rightarrow \mathcal{R}$, specifically we use

$$f(x_0, x_1, \dots, x_d) = \sum_{d=0}^{D-1} x_d L^d \quad (1)$$

With this mapping, a particle that hops 1 site in any direction in \mathcal{R}^D jumps $\pm L^d$ in R^1 for a given component axis¹.

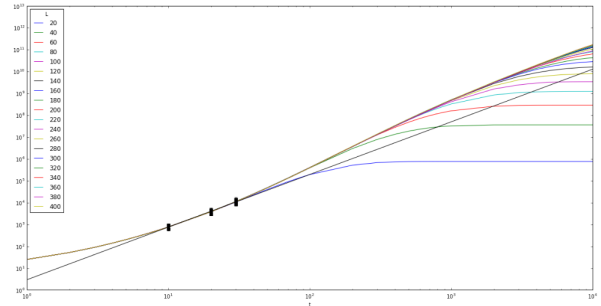
The boundary conditions are enforced by creating a function that maps each site to a *partition* of the space. This function, g , maps a site i to a partition in the context of some axis/dimension d . Simply, $g(i, d) = \text{floor}(i/L^{d+1})$. It is illegal for a particle to

hop between partitions². When a jump is computed along an axis such that $i' = i + L^d$, we can check the condition $g(i, d) \stackrel{?}{=} g(i', d)$.

A bitmask is used to close boundaries on given axes. For example setting bits 0,1 closes the first component axis at both ends to create a toroidal axis (as opposed to closed/reflecting boundaries).

1.2 Exponents for 2D

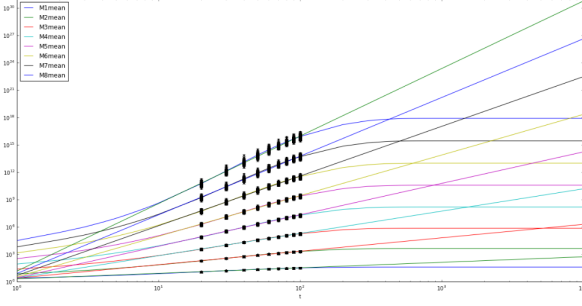
Considering various boundary conditions for various system sizes, we study behaviour in a region dominated by system size or extinction. There is a certain upper regime where *distinct sites visited* is dominated by extinction. For small systems, the boundary conditions dominate. Looking at the first moment for a number of system sizes, one can see that there is a region where, independent of system size, the observable grows with t (power law).



The first eight moments are shown for a system of size 100

¹Conventionally we assign even numbers in 2D to positive(forward) moves and odd numbers to negative (back) moves

²TODO: Performance hint to predetermine boundary sites with lattice flags to avoid checking partitions each time.



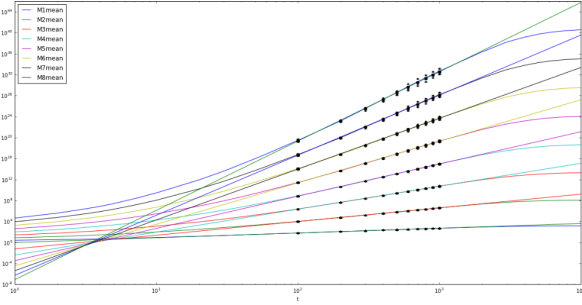
Amp	Index	Moment n=	Δ Index
1.00	0.91	1.00	nan
0.06	2.61	2.00	1.70
0.00	4.37	3.00	1.76
0.00	6.14	4.00	1.77
0.00	7.92	5.00	1.77
0.00	9.69	6.00	1.77
0.00	11.46	7.00	1.77
0.00	13.22	8.00	1.76

Amp	Index	Moment n=	Δ Index
2.82	0.58	1.00	nan
1.04	1.77	2.00	1.18
0.39	3.02	3.00	1.25
0.19	4.27	4.00	1.24
0.11	5.49	5.00	1.23
0.09	6.70	6.00	1.21
0.07	7.89	7.00	1.19
0.07	9.07	8.00	1.18

For 2D (open boundaries) the table shows the exponents.

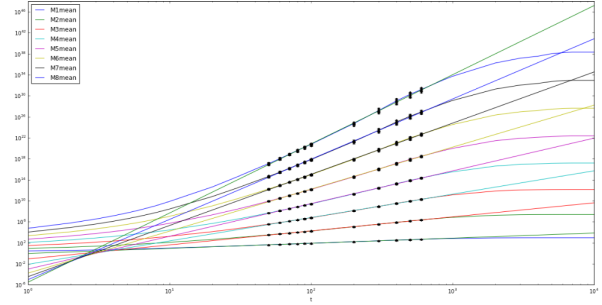
1.3 Exponents for 3D

The first eight moments are shown for a system of size 100



1.4 Exponents for 5D

The first eight moments are shown for a system of size 100



Amp	Index	Moment n=	Δ Index
0.94	0.97	1.00	nan
0.09	2.67	2.00	1.70
0.01	4.45	3.00	1.78
0.00	6.23	4.00	1.78
0.00	8.00	5.00	1.77
0.00	9.75	6.00	1.75
0.00	11.47	7.00	1.72
0.00	13.17	8.00	1.70