Self-avoiding Walks
Extra Credit

Jeremy Morris

November 17, 2005

The code I have written will find self-avoiding walks using the Metropolis algorithm. It does this for the general case, in other words, for any choice of d (dimension of the lattice) and n (path length).

In order to check to see if the code is working, I have run two sets of tests. The results of which can be seen in Figures (1) and (2).

The first question is how big should the value N be in order to assure convergence. N is the number of times we carry out the product $\nu = \nu P$. I have chosen the values $N_i = 10, 10^2, 10^4, 10^6$ and kept the values $n = 3, d = 2, \beta = \infty$. The results can be seen in Figure (1) shows the results. It turns out that we get exactly what is expected for very low values of N. For example, when $N = 100$ we already get the uniform distribution on the self-avoiding walks. This may be due to the choice for n, d, β. Unfortunately, it takes a lot of time to compute the stationary distribution for larger values of n and d. At a later time I may try to increase these values to see what happens.

The next question is how does the value of β affect these calulations. I have run some experiments setting $\beta = 1, 10^4, 10^{16}, 10^{32}$. It is interesting that there is only a difference when $\beta = 1$. For all other values of β we have the uniform distribution. Again, this may have something to do with the choice of n and d which remained the same for these experiments.

The next step would be to run some similar experiments when n and d are altered to see how N and β affect the results. This could happen during a long weekend or something when we have more computing time at our disposal.

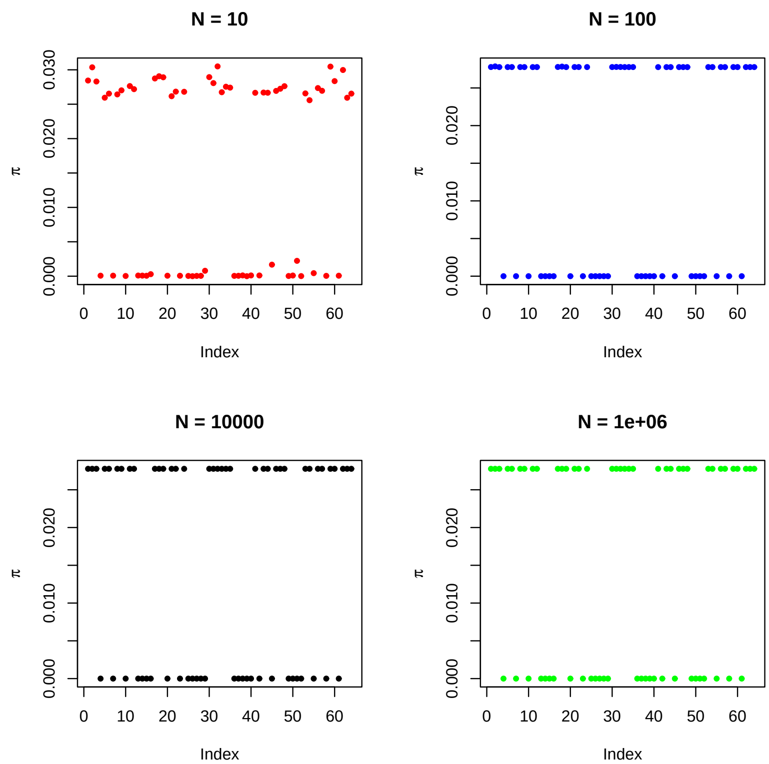

Figure 1: Plots Varying N

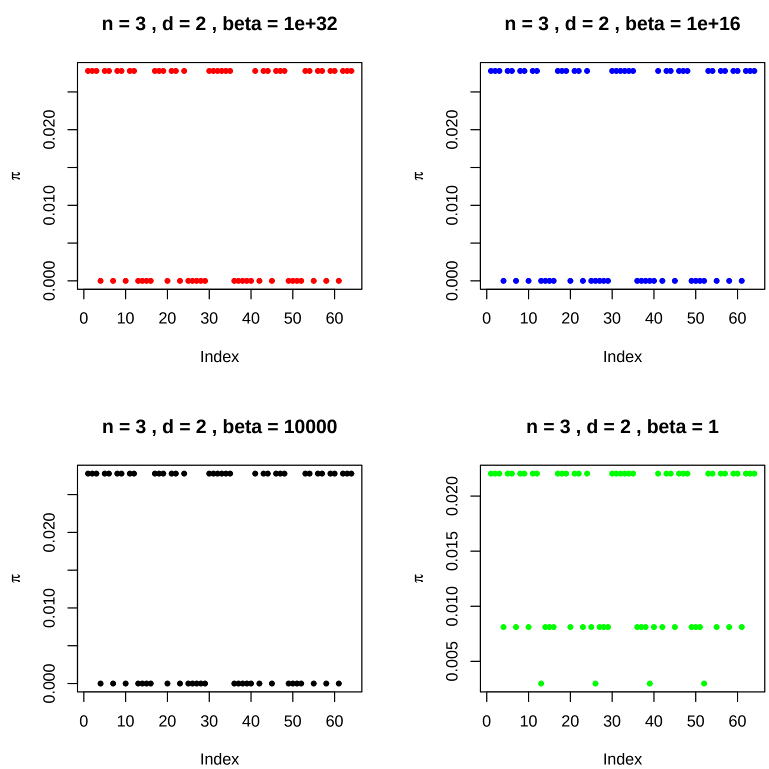

Figure 2: Plots Varying β