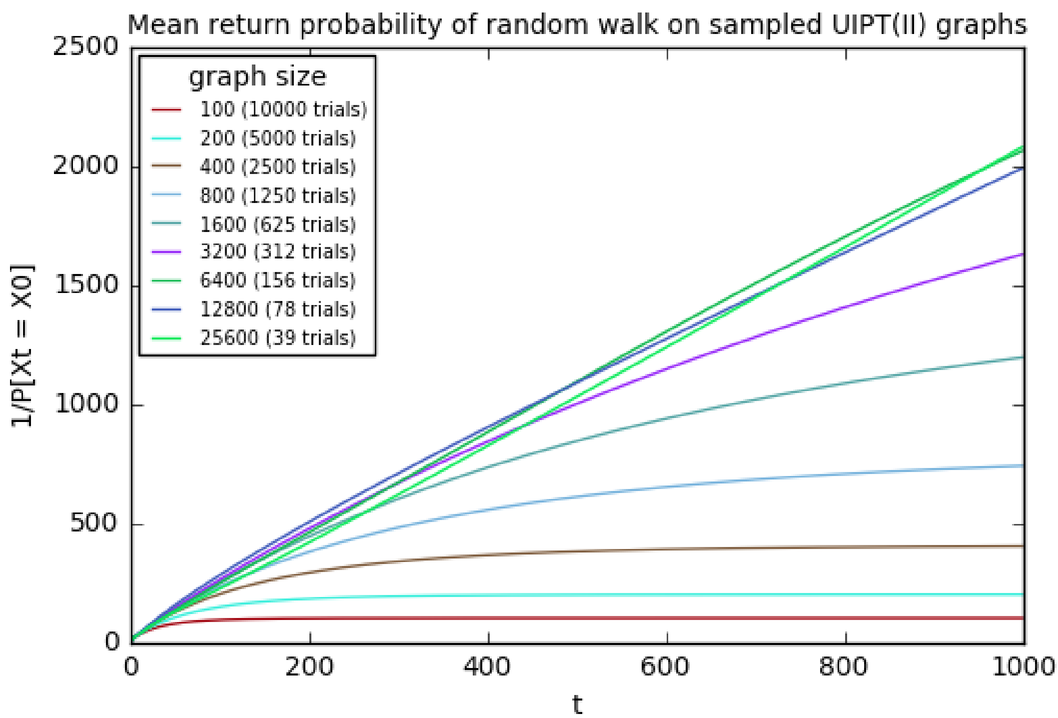
(Note everything here is for type II triangulations, which allow multiple edges.)

1 Return probability of random walk

Picking a random starting vertex x_0 on a uniform planar triangulation of n vertices, and treating $\{x_1 \dots x_t\}$ as the vertices visited on the random walk, look at $\mathbb{P}[x_0 = x_t]$. This is provably at least $1/t \log t$ (thanks), although it seems to go more like $1/t$.

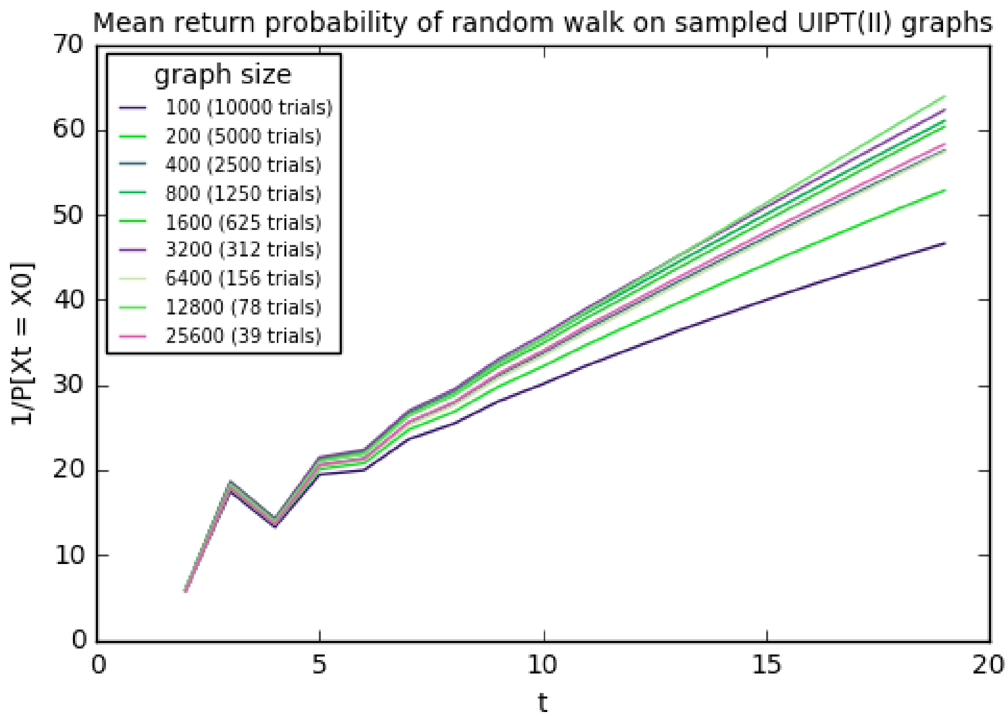
I generated large numbers of uniform planar triangulations of various sizes, and found the exact distribution of the random walk (for that particular triangulation and starting point) by doing sparse matrix-vector multiplications on the initial distribution (the relevant unit basis vector). Fancy exponentiation doesn't seem to be helpful here, because for this particular task, we seem to care about when the number of steps taken is much less than the total number of nodes, and sparsity goes away pretty quickly. I ran this for 1000 steps (in my current implementation, running any of these random walks for more than 100 steps takes longer than generating the triangulation in the first place). Data is on github.

Note that this is the inverse of the mean return probability:



For t high enough, because these graphs are (unfortunately) not infinite, the return probability will level off to $1/n$, which is clearly visible for small n.

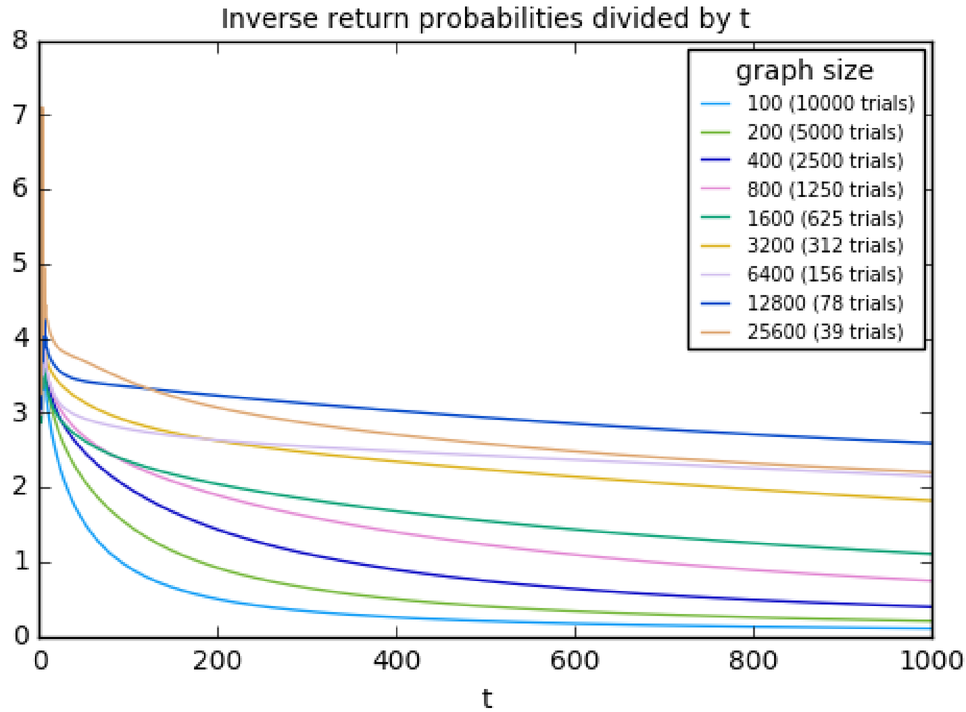
It's not clear to me how much this effect is ruining the results: it seems like we should be relatively fine to pretend the graph is infinite if $t < r^{\frac{1}{4}}$ (just because of the ball growth rate), but can we do it for longer? Here's the same plot, zoomed in t $t = 20$. If the return probabilities for the random walk are noticeably higher for small n, then that's a sign that we need to use bigger graphs.



The jaggedness at the beginning is expected – there's zero chance that you're back where you started after 1 step, and (comparatively) small chance after three steps, five steps, etc, although it evens out.

I don't know if $t = 20$ is high enough to say anything at all about asymptotics – we have this weird tradeoff between wanting to look at long walks because we care about the asymptotics, but also hating long walks because they require large graphs in order to be relevant (plus the stochastic nature of the graph generation to begin with). Not sure what the middle ground is, other than computing for longer.

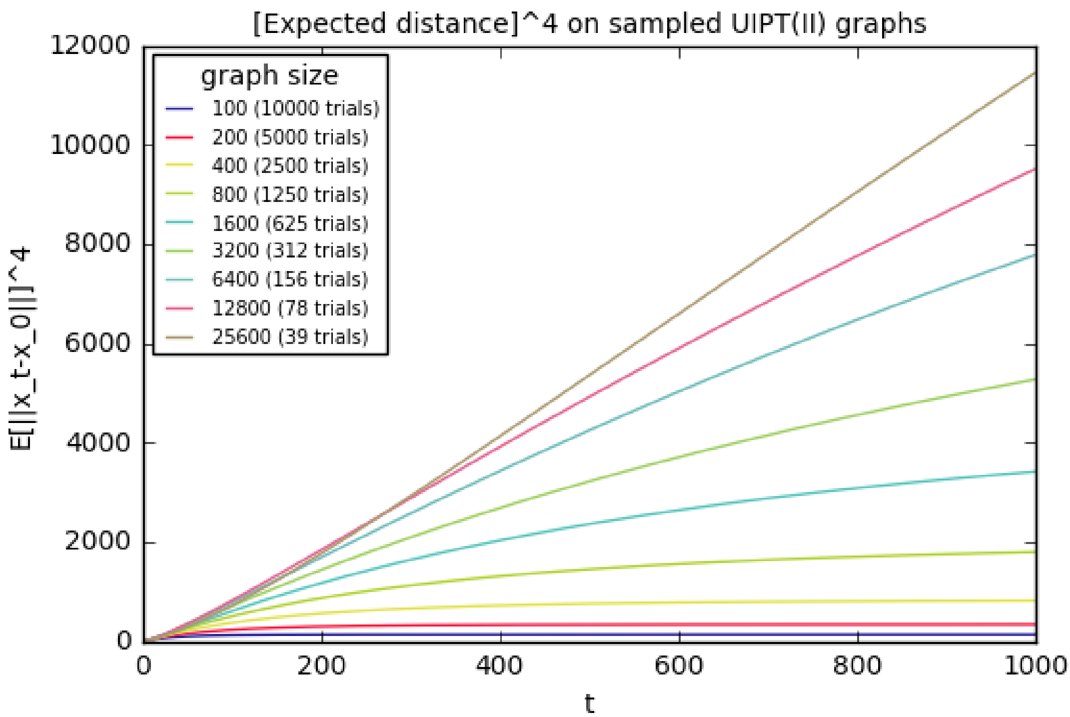
That said, it looks pretty linear – here's the original plot, divided by t.



If this went with $1/t$, we'd expect this to be constant (up until the point where the finiteness is a problem). If it went with $\frac{1}{t\log(t)}$, then we'd expect this graph to be about $\log(t)$. But it's actually sublinear! Hard to say whether that's a quirk of having small graphs, or something more meaningful.

2 Distance

Here's the average distance from the initial vertex x_0, according to the same process as return probabilities. Note that the distance is raised to the 4th power in the graph. As before, data is on github.



We have the same problem where on finite graphs, eventually there aren't many vertices that are more than a small distance away from the start. However, it looks pretty clearly to be $t^{\frac{1}{4}}$: putting a larger or smaller exponent makes it not look linear anymore.