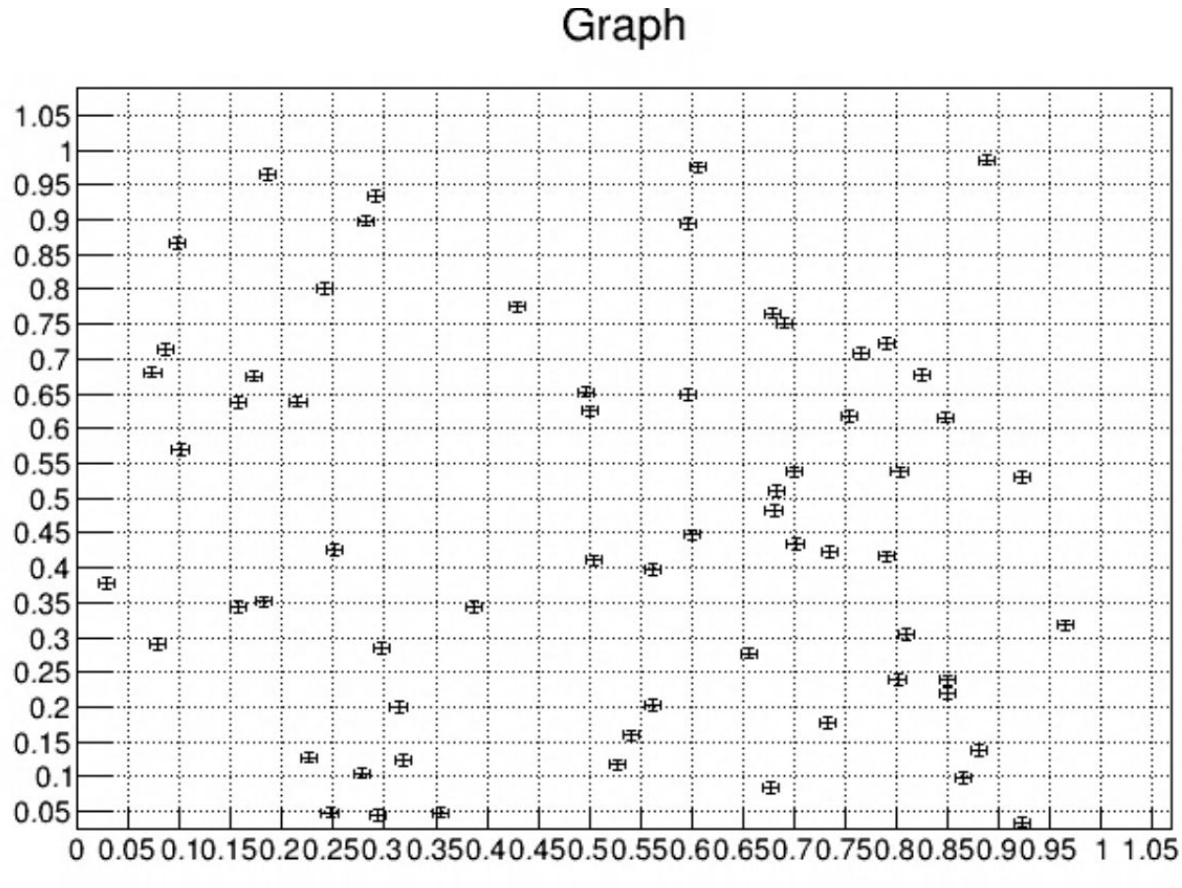
**instructions for running programs are in README.md*

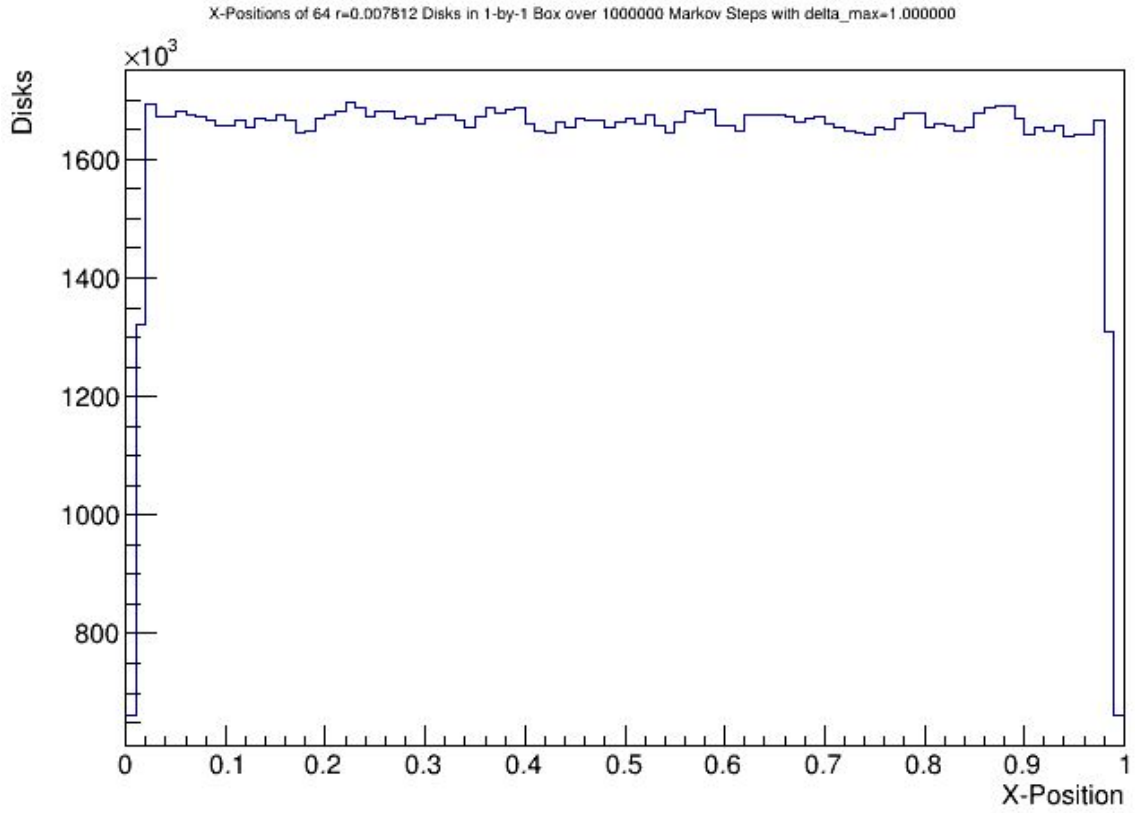
Markov Disks (naive implementation)

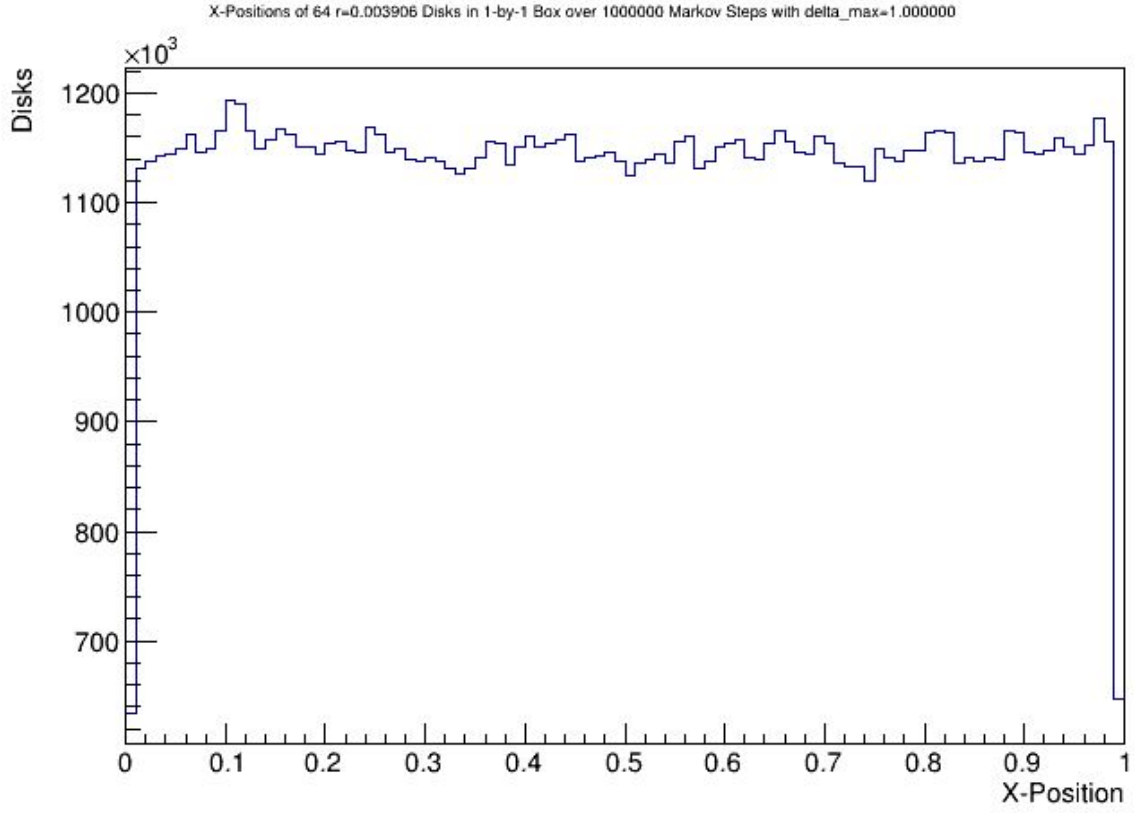
Our simulation ran one-million Markov chain operations on 64 disks of radius 1/128 in a 1 unit square box. Each disk had a delta-movement of up to 1 unit, meaning that each disk could, during its walk, change positions to any valid position in the box (valid meaning within the bounds of the box, and not overlapping with any other disk).

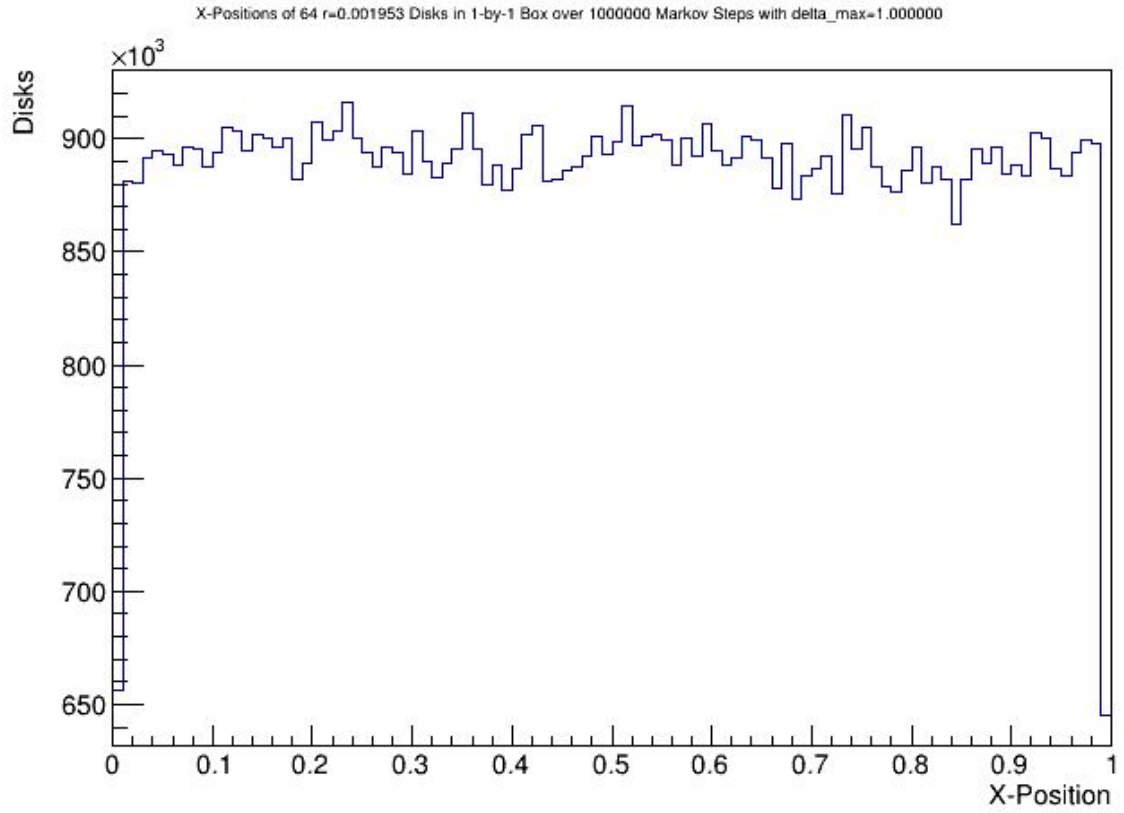
The radius of the disks is always at most half the length of the box divided by the number of disks. Included with our programs are graphs that use error bars to illustrate the rough position and size of each disk in the initial state.

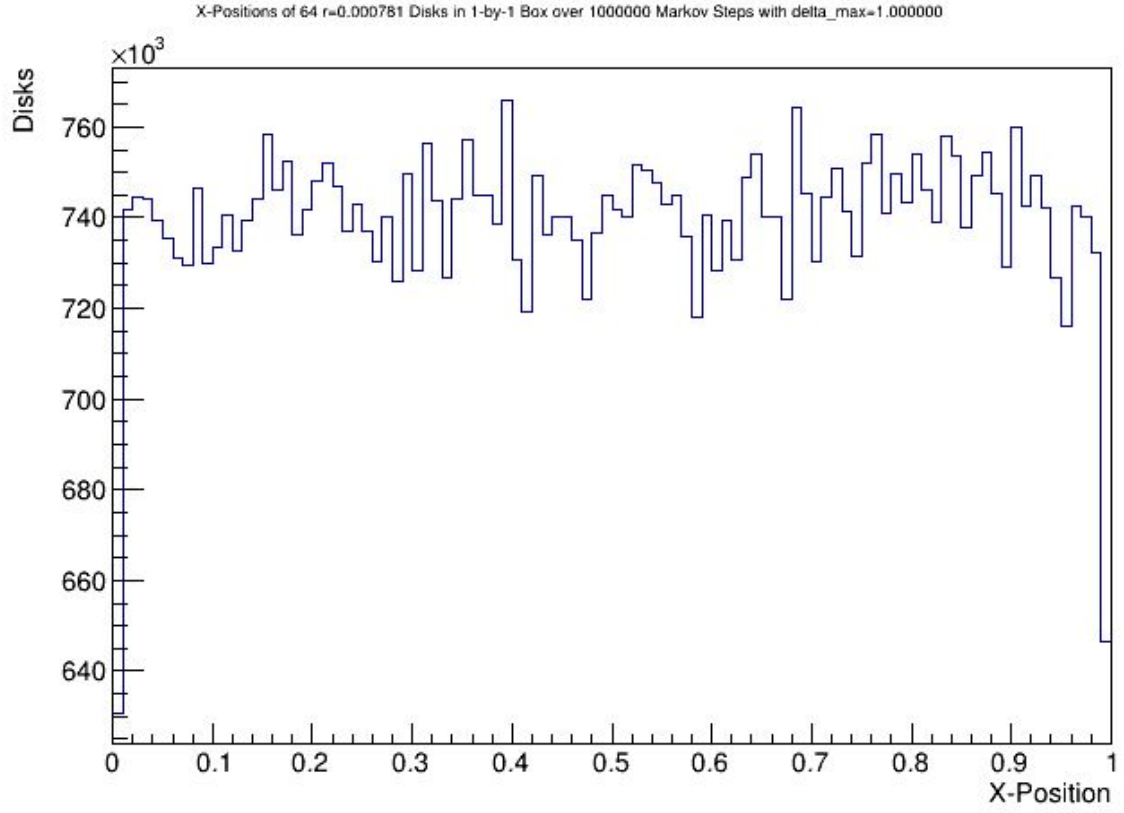


Below are graphs showing the densities of the Markov disk simulation for disks of radius 1*r_max, 0.5*r_max, 0.25*r_max, and 0.1*r_max. Each simulation took about 21 seconds to run.


X-Positions of 64 r=0.007812 Disks in 1-by-1 Box over 1000000 Markov Steps with delta_max=1.000000
Disks
×10^3
1600
1400
1200
1000
800
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
X-Position


X-Positions of 64 r=0.003906 Disks in 1-by-1 Box over 1000000 Markov Steps with delta_max=1.000000
Disks
×10^3
1200
1100
1000
900
800
700
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
X-Position


X-Positions of 64 r=0.001953 Disks in 1-by-1 Box over 1000000 Markov Steps with delta_max=1.000000
Disks
×10^3
900
850
800
750
700
650
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
X-Position

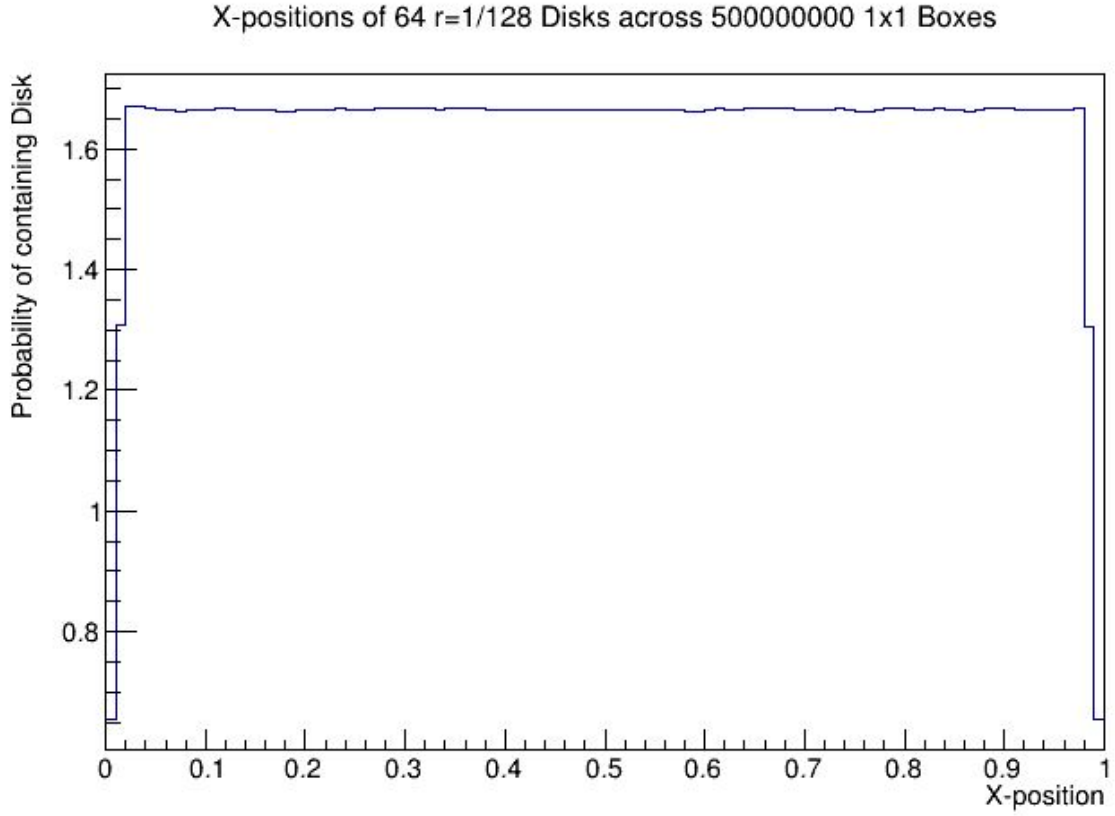

X-Positions of 64 r=0.000781 Disks in 1-by-1 Box over 1000000 Markov Steps with delta_max=1.000000
Disks
×10^3
760
740
720
700
680
660
640
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
X-Position

Comparison to Event-Disks

In comparison to …
<event disks graph>

Comparison to Direct-Disks

In comparison to Direct-Disks, the Markov simulation runs much more efficiently. For the conditions outlined above, the direct-disk simulation finished in about 60 seconds, about double that of the naive Markov's 25 seconds. However, the direct disk simulation produced much smoother results than that of the Markov simulation. Our speculation is that this is because the Markov simulation moves only one disk per step, meaning that under Markov, each iteration is more likely to share similarities and partially matching configurations. Whereas, with the direct disk implementation, each configuration is generated from scratch, which although slower, produces much more varied configurations.



Comparison to Markov Disks with Grid Partitioning

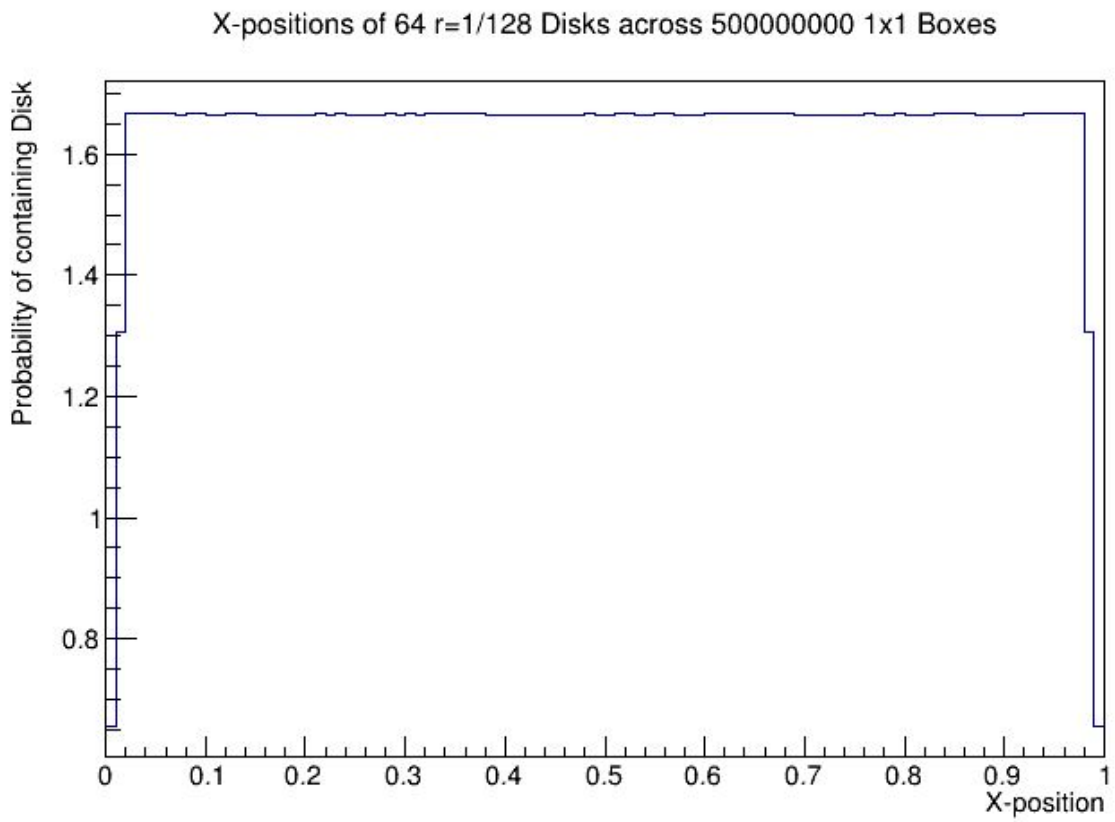
An attempt to speed up the Markov simulation was made by reducing the number of comparisons made between disks when evaluating iterations. This was done by dividing the box

into a grid of smaller squares. The new position of the walked disk would then only be checked against other disks in its cell along with those of the 8 neighboring cells. Our partition divided the box into 100 smaller cells, which across 64 disks, meant that each cell was home to about one disk. Already, we can see large potential gains, cutting the number of comparisons per iteration from 63 down to about 8.

From the tests, we found that this grid-cell addition cut the runtime of the program from 25 seconds down to about 21 seconds. These aren't amazing gains, but keeping in mind that our program only runs for a relatively small one million iterations, the overhead from the bookkeeping may be keeping the grid version from outperforming the naive version, at least for smaller numbers of data points. We ran the programs again but for ten million points, and found that the grid version's 230 seconds about matches the naive version's 235 seconds. This could be due to an implementation error, with improper handling of references which creates unnecessary copying.

Large Batch Job for Inefficient Direct-Disks Algorithm

We ran our least efficient algorithm, the direct disks algorithm, for 500,000,000 steps, which should have no more than 12 hours, since 1,000,000 ran for about one minute. The program terminated earlier than expected, at about 5.5 hours, and produced the graph below. There is not much visual difference between the 1 million and 500 million point version, probably limited by resolution.


X-positions of 64 r=1/128 Disks across 500000000 1x1 Boxes
Probability of containing Disk
1.6
1.4
1.2
1
0.8
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
X-position