

Physics 750: Exercise 4

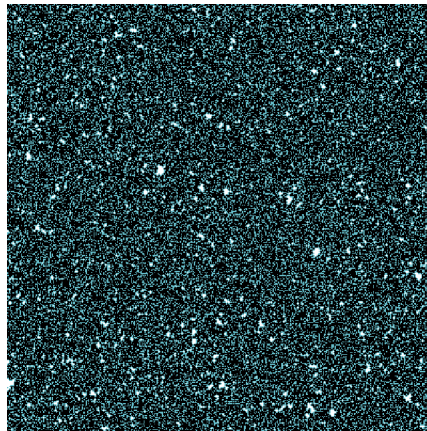
Tuesday, September 5, 2017

1. Use the curl command to download from the class website everything you'll need for this exercise.

```
$ WEBPATH=http://www.phy.olemiss.edu/~kbeach/  
$ curl $WEBPATH/courses/fall2017/phys750/src/exercise4.tgz -O  
$ tar xzf exercise4.tgz  
$ cd exercise4
```

2. The files `perc_cluster.cpp` and `perc_cluster2.cpp` implement the kinetic growth model we discussed in lecture. The first is a naive implementation, and the second is much more efficient. Look over the code and try to understand why.
3. Run the program a few times and see if you can manually estimate the critical value of the growth probability that coincides with the percolation threshold.

```
$ make perc_cluster2  
g++ -o perc_cluster2 perc_cluster2.cpp -O2 -Wall -ansi -pedantic ...  
$ ./perc_cluster2  
Usage: perc_cluster (0 < prob < 1)  
$ ./perc_cluster2 0.5  
[ctrl-c]  
$ ./perc_cluster2 0.7  
[ctrl-c]
```



Shown above are kinetic growth model final configurations for $p = 0.5$ (left) and $p = 0.7$ (right) on a 400×400 lattice with periodic boundary configurations.

4. Write a new program that will compute the critical probability to a few decimal places using binary search.
5. Another program is included that implements the Deposition Diffusion and Aggregation (DDA) algorithm. Explore its behaviour over the range of input arguments.

```
$ make diffusion  
g++ -o diffusion diffusion.cpp -O2 -Wall -ansi -pedantic ...  
$ ./diffusion  
diffusion (0.0 < initial population probability < 1.0) (0.0 < deposition rate)  
$ ./diffusion 0.0 2.5
```

6. Develop a stopping condition for the algorithm.
7. Compute the effective dimension—possibly fractal—of the resulting configuration.
8. Check out how slowly the program runs when the linear dimension of the lattice is increased:

```
const size_t L = 100; // change the value on line 26 to 200 or 400, say
```

The problem is that the implementation is poor. Modify the code to get a comparable efficiency gain as seen in `perc_cluster2.cpp` with respect to `perc_cluster.cpp`.

9. Using the `bitmap.hpp` header from Exercise 1, create a new version of the program that dumps its final configuration to `output.png` rather than to the screen via OpenGL. You'll have to modify the `makefile` accordingly.