Prof. Dr. A.K. Hartmann
WiSe 2023/24, 5.04.4521
Blatt 7, 26.02.2024

Exercises Computational Physics

7 Spanning probability for percolation

The aim is to detect the percolation transition not by alanyzing the size of the largest component. Instead, it shall be checked whether the largest component *spans* the lattice. This means that the largest component occupies in every x-column of the system at least one lattice site. [1]

1. Download the programm `percolation1.c` from StudIP. Also you need `percol.h`, `percol.c`, `stacks.c` and `stacks.h`. You may compile by using
   `cc -o percol percolation1.c percol.c stacks.c -Wall -g`

2. Extend the program by a function which checks whether a cluster spans the system:

```
/****************** percol_spanning() ******************/
/** Determines whether cluster 'cluster_ID' occupies   **/
/** for each x ordinate at least one site.             **/
/** For each site, in 'cluster[i]' the ID of the       **/
/** cluster (starting at 0) is stored                  **/
/** PARAMETERS: (*)= return-paramter                   **/
/**           N: number of sites                       **/
/**     cluster: ID of clusters sites are contained in **/
/**  cluster_ID: ID of the cluster which is tested     **/
/**    x_length: length of system in x direction       **/
/** RETURNS:                                           **/
/**     1 if all x values are occpied (<->spanning)    **/
/*********************************************************/
int percol_spanning(int N, int *cluster,
    int cluster_ID, int x_length)
```

   The function is supposed to work in all dimensions! (6 P)

3. Remove the commentary around the call to this function in `main()`.

4. Test your function thoroughly with the debugger `gdb`. Explain your tests. (1 P)

5. Perform simulations for different lattice sizes L (at least 1000 realizations of the disorder). You should consider enough different values of p close to the expected transition at $p_c \approx 0.59$. Start with small sizes like $L = 12$ and go as far as you can go, at least $L = 24, 48$ should work. (1 P)

[1] This is a different criterion than "wrapping".

6. Plot your estimator of the spanning probability, i.e., the fraction of systems where the largest cluster (2. output column) spans, as a function of p (1. column).

   The percolation transition happens (about) where the curves intersect. (1 P)

7. Perform a data collapse by rescaling the p-axis with $(p - p_c)L^{1/\nu}$ such that the different curves collapse, for suitable value of ν onto a single curve.

   Hint: the system size L is printed in the 3rd output column. Use suitable values for `pc`, `nu` in `gnuplot` with the command `plot "datei_L1" u (($1-pc)*$3**(1/nu)):2 w l` for rescaling on single data file, similar for several files.

   Which value of ν do you obtain for the best collapse? (1 P)