

## Übungen zur Computerorientierten Physik

### 7 Fraktale Oberfläche bei Perkolation

Ziel ist es, die fraktale Dimension der Oberfläche des größten Clusters am Perkolationsübergang zu bestimmen. Die Oberfläche  $\delta S$  beinhaltet die alle "Flächenelemente" (oder Linien bei  $d = 2$  Dimensionen) zwischen einen Gitterplatz der zum größten Cluster gehört und einem leeren Gitterplatz. Die fraktale Dimension  $d_f$  ist dann definiert durch das Größenwachstum der gemittelten Oberfläche als Funktion der Gittergröße  $L$  gemäß  $\delta S \sim L^{d_f}$

1. Laden Sie das Programm `percolation1.c` vom StudIP. Weiter benötigen Sie `percol.h`, `percol.c`, `stacks.c` und `stacks.h`. Sie können es übersetzen mit  
`cc -o percol percolation1.c percol.c stacks.c -Wall -g`
2. Erweitern Sie Ihr Programm um eine Funktion zur Bestimmung der Oberfläche des größten Clusters. Dazu müssen Sie über alle Gitterplätze iterieren. Für jeden Gitterplatz, der zu dem größten Cluster gehört, wird über alle Nachbar-Gitterplatz iteriert. Für jeden Nachbarn `nn`, der leer (`cluster[nn]==-1`) ist, wird die Oberfläche des Clusters um einen erhöht.

```
/****** percol_surface() *****/
/** Calculates surface of the largest cluster          **/
/** 'id_largest'                                       **/
/** PARAMETERS: (*)= return-paramter                 **/
/**      num_n: number of neighbours                  **/
/**      N: number of sites                           **/
/**      next: gives neighbours (0..N)x(0..num_n-1)    **/
/**      0 not used here. Use NEXT() to access        **/
/**      num_cl: number of clusters                    **/
/**      cluster: id of clusters sites are contained in **/
/**      (-1 if not occupied)                         **/
/** id_largest: id of largest cluster                  **/
/** RETURNS:                                          **/
/**      size of surface of largest cluster           **/
/******
int percol_surface(int num_n, int N, int *next, int num_cl, int *cluster,
                  int id_largest)
```

Diese Funktion sollte für alle Dimensionen funktionieren.

(6 Punkte)

3. Entfernen Sie die Kommentarzeichen beim Aufruf der Funktion in `main()`. Achtung: Der Aufruf von `percol_largest()` muss weiterhin erfolgen, auch wenn der direkt zurückgegebene Wert nicht benötigt wird, wohl aber der ebenfalls bestimmte `id_largest`.
4. Testen Sie Ihre Funktion ausführlich mit dem Debugger `gdb`. Erläutern Sie wie Sie getestet haben (1 Punkt)
5. Führen Sie Simulationen am kritischen Punkt  $p_c = 0.59$  für verschiedene Gittergrößen  $L$  durch (mind. 100 Realisierungen). Fangen Sie mit kleinen Größen an und sehen Sie, wie weit Sie kommen, mindestens  $L = 12, 16, 24, \dots 256$  sollte gehen. (1 Punkt)
6. Plotten Sie den Schätzwert für die Oberfläche (2. Ausgabespalte), als Funktion von  $L$  (3. Ausgabespalte). Fitten Sie ein Potenzgesetz  $\delta S \sim L^{d_f}$ . Welchen Wert für  $d_f$  finden Sie? (2 Punkte)