Modelling and Simulation
Practical Assignment 2: Percolation

Rick van Veen (s1883933)* Laura Baakman (s1869140)*

October 15, 2015

1. Introduction

Kenzel et al. [2] give the following interpretation of the percolation model; it describes the geometry of the randomly generated pores in a porous material through which only certain particles can percolate if the pores form continuous paths. We model this material using a finite lattice, although different lattices are possible, we consider only a square lattice.

The exact percolation model is described in section 2, this section also presents an implementation of the model in pseudo code. In section 3 we discuss the experiments we have performed with the model and their results.

Algorithm 1: `percolation`$(mask, N, p)$

input : N size
p probability
mask $r \times c$ binary matrix.
output: *grid* $(2N+1) \times (2N+1)$ matrix

```
center := (N + 1, N + 1)
push(queue, center)
grid := initGrid(N, N)
while not isEmpty(queue) do
    site = pop(queue)
    sites = grow(grid, site, mask, p)
    if onBorder(site) then
        break
    push(queue, sites)
```

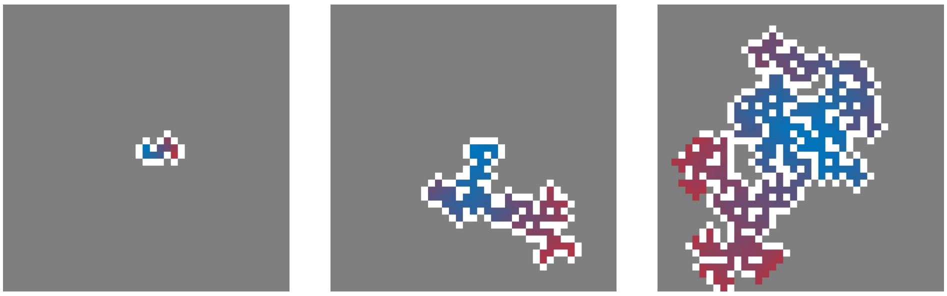
2. Method

Algorithm 1 presents our iterative growth process, the method `percolation` expects three arguments `N`, `probability` and `mask`. Given the size parameter `N`, the grid used for the percolation is $(2N+1) \times (2N+1)$, since this causes the grid to have an uneven number of rows and columns. The grids center is now always clearly defined as $(N+1, N+1)$. The parameter $p \in [0, 1]$ is the probability that a given site in the cluster becomes occupied. The `mask` is a binary matrix with r rows and c columns that determines the used connectivity. Until section 3.4 we only consider four-connected clusters for which the mask presented in figure 7a is used.

Initially the only site we have to consider is the center site, which is consequently the only site in the queue at the first iteration.

Each iteration we pop the next `site` from the queue. We grow this point, using the function `grow`. This method considers all neighbors that are connected to `site` determined by the discussed `mask`. For each of these neighbors we generate the value z, which is randomly sampled from an uniform distribution with the

*These authors contributed equally to this work.



(a) *$N = 20$ and $p = 0.3$* **(b)** *$N = 20$ and $p = 0.5$* **(c)** *$N = 20$ and $p = 0.6$*

Figure 1: *Examples of (a) a small finite cluster, (b) a larger finite cluster and (c) a percolating cluster using four-connected neighbours. The colours of the elements in the cluster indicate when that point was added to the cluster, the 'colder' the color the earlier in the percolation it was added to the cluster. White cells are empty and gray cells are undetermined.*

range $[0, 1]$. If $z \leq p$ we mark the neighbor site as occupied, otherwise it is marked empty. The method `grow` returns the newly occupied neighboring sites, which are then added to the queue, so they can be processed by the `grow` process in a later iteration.

The growth of the clusters stops when the queue is empty and it thus cannot grow anymore or if it has reached one of the borders of the grid. In the first case the cluster is finite, which means that all neighboring sites of the cluster, according to the connectivity defined by the `mask`, are marked as empty. In algorithm 1 we test for this condition via the guard of the loop; if the queue is empty there are no more neighbors to consider, consequently the cluster must be finite.

A percolating cluster is a cluster that has reached the border of the grid, i.e. if there is a occupied site with row or column number 1 or $2N + 1$. We test for this condition with the method `onBorder`. It should be noted that we only check if a site is on the border of the grid after we have already grown the site.

Figure 1 presents three clusters grown with the algorithm. We can clearly see that the finite clusters are completely surround by a white border, which indicates that these sites are empty and processed. The illustration of the percolation cluster, figure 1c, shows that although there are still sites that can grow indicate by the lack of white neighbors, there is one site on the border near the bottom left corner of the image that has stopped the percolation.

3. Experiments

This section presents our exploration of the parameter space of the percolation model. In section 3.1 we discuss the influence of the probability parameter p. In this section we often use the size of the cluster to compare the influence of the different parameters. We have defined the cluster size as the number of points that are occupied.

Section 3.2 explores the effect of the size of the system on the generated clusters. In section 3.3 we attempt to determine the fractal dimension of the finite clusters as a function of p. Finally, section 3.4 presents a short analysis of the impact of the used connectivity.

3.1. Probability

One important property of clusters is their size, and how that size depends on the parameter p. Kenzel et al. [2] describe this relation as follows: for small values of p we get a large number of small clusters. As p

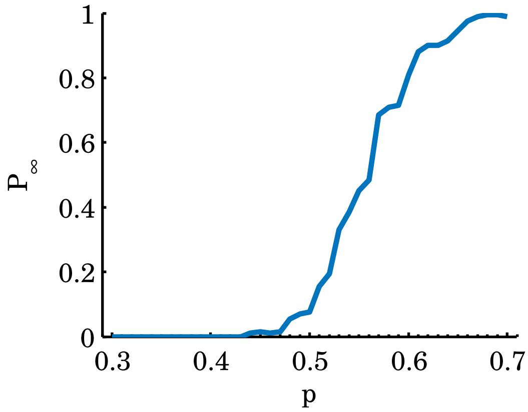


Figure 2: *Ratio of percolating clusters, P_∞, as a function of p. Ratios are calculated over $r_{max} = 200$ runs on a 41×41 grid.*

increases we find positive correlation between p and the average cluster size until p reaches some threshold value p_c. For $p > p_c$ we get either a small finite cluster or a percolating cluster. As $p > p_c$ increases the probability of ending with a finite cluster decreases, until we always get the percolating cluster for $p = 1$. Note that although in theory this cluster should cover the full grid, this is not necessarily the case in our model, since it stops growing as soon as one border site is occupied.

To find an indication of the value of p_c with our model we have let it generate a cluster on a 41×41 grid for $p = 0.31, 0.32, \ldots, 0.7$. For each value of p we grow $r_{max} = 200$ clusters.

Figure 3 presents the mean and standard deviation of the size of the finite clusters as a function of p. In this figure we observe the effect of p on the mean cluster size described by Kenzel et al. Furthermore, based on these data one would estimate p_c to be approximately 0.55.

Kenzel et al. also predicted that the number of percolating clusters relative to the number of finite clusters would grow for $p > p_c$ until $p = 1$, where the only possibility would be a percolating cluster. To observe this effect figure 2 shows p_∞, which is the ratio of the number of percolating clusters to the number of finite clusters. This graph is based on the same data as figure 3. Based on this graph we would say that $p_c \approx 0.4$. This number is lower than the value for p_c based on figure 3. This is probably caused by the relatively small grid sizes, which causes us to classify some clusters as percolating, that are actually finite.

Stauffer [4] has found p_c to be approximately 5.928e^{-1} for a square lattice. As stated earlier our lower estimation of p_c is quite likely caused by our small grid.

3.2. System Size

In section 3.1 we postulated that our relatively small grid influenced the found value of p_c. This section qualitatively discusses the relation between the size of the grid and the cluster. We repeat the experiment discussed in section 3.1 but varied $N = 2, 6, \ldots, 60$ instead of p. Since we are mostly interested in the size of finite clusters we choose $p = 0.5 < p_c$. Instead of the size of grid we now measure Q the ratio of the size of finite clusters to the number of sites in the grid.

Figure 4 shows the mean and standard deviation of Q as a function of N. In this graph we observe that average Q decreases as N increases, i.e. the size of the clusters does not grow as hard as the number of sites in the grid.

Figure 5 shows that $P\infty$ reaches zero for $N > 40$, which indicates that for this value of N there are no more percolating clusters. This fits with the theory discussed in section 3.1, which states that we get only finite clusters for $p < p_c$. We have percolating clusters for $p = 0.5 < p_c$ since are grid is not large enough to hold the finite clusters.

If the lattice size is infinite we are no longer

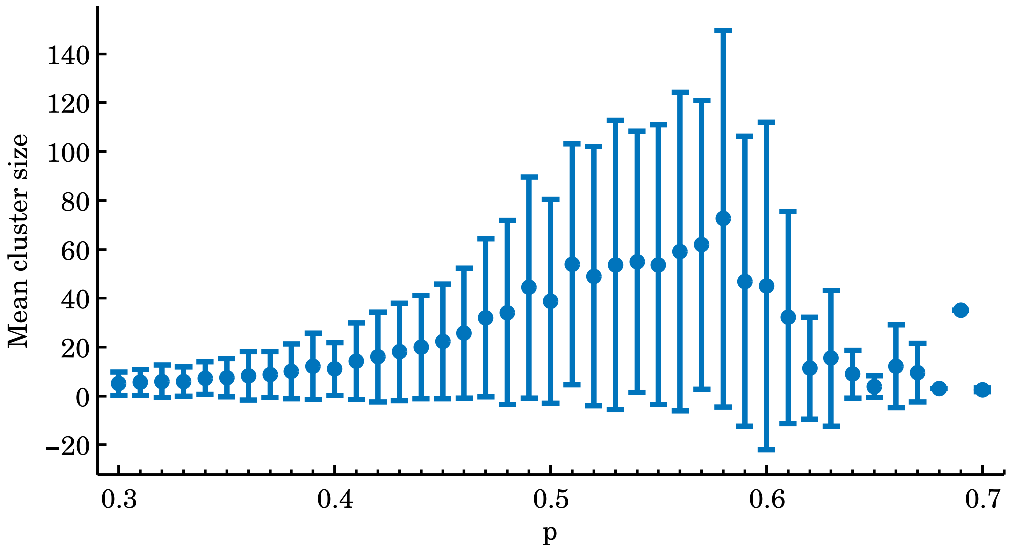


Figure 3: *Mean cluster sizes, represented as points, and standard deviations, indicated by the vertical error bars, as a function of p, with a step size of 0.1. The mean and standard deviation were calculated over 200 runs on a 41×41 grid.*

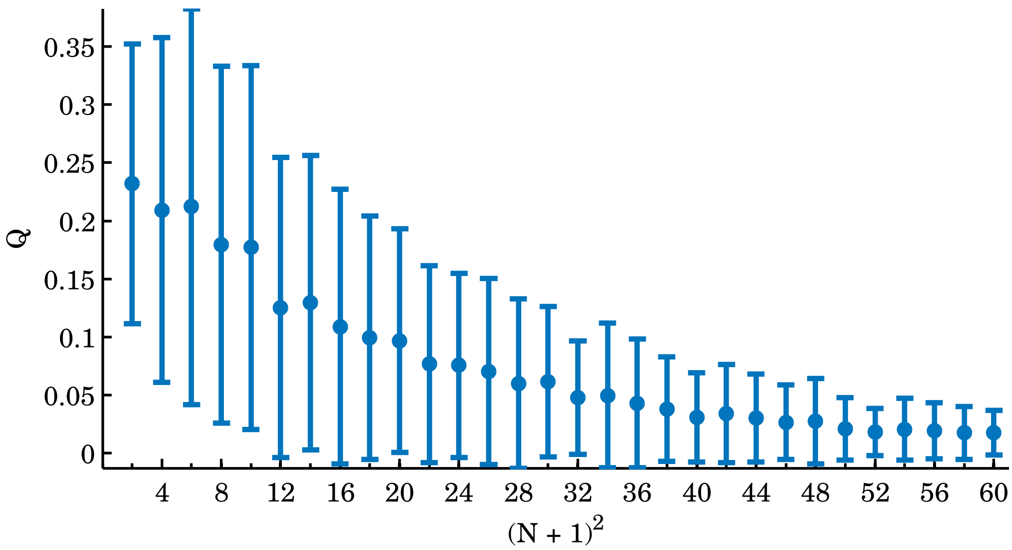


Figure 4: *The mean, represented as points, and standard deviations, indicated by the error bars, of the ratio of the cluster size to number of sites in the grid, i.e. $(N+1)^2$.The mean and standard deviation were calculated over 200 runs with $p = 0.5$.*

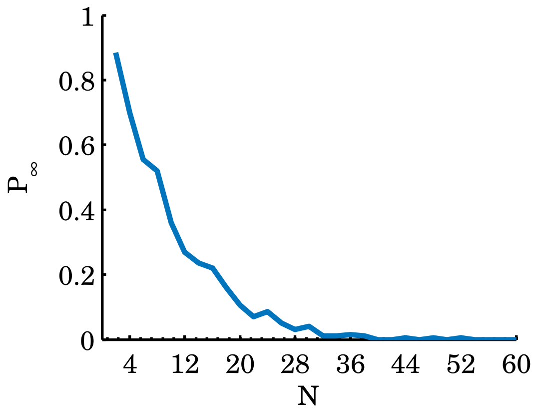


Figure 5: *Ratio of percolating clusters to the number of finite clusters, P_∞, as a function of N. Ratios are calculated over $r_{max} = 200$ runs with $p = 0.5$.*

limited by the size of the lattice, in essence we remove one of our stop conditions. In this situation the theory presented in section 3.1 holds, i.e. as long as $p < p_c$ we only have finite clusters. As $p > p_c$ the number of percolating clusters relative to the number of finite clusters increases until we always get a percolating cluster for $p = 1$.

3.3. Fractal Dimension

Falconer [1] describes the fractal dimension as some number ρ such that

(1) $M_\varepsilon(\rho) \sim c\varepsilon^{-s}$

where c and s are constants and $M_\varepsilon(\rho)$ are measurements at different scales ε for $\varepsilon \to 0$. Falconer then shows that the fractal dimension can be estimated "as minus the gradient of a log-log graph plotted over a suitable range of ε"[1].

One way to get the measurements M_ε is to use box-counting. When box-counting is used the different scales mentioned in Falconer's definition are the sizes of the boxes.

We have used the function `box-count` by Moisy [3] to determine the fractal dimension of Percolation clusters. This method uses box sizes that are power of two. Consequently $\varepsilon = 1, 2, 3, \ldots 2^q$ where q is the smallest integer such that $q \leq (2N + 1)$.

We have used the box-counting algorithm on a cluster generated with $N = 80$, $p = 0.7$, the used cluster is shown in figure 6a. Figure 6b presents the number of boxes as a function of the size of the boxes. The box-counting dimension can be read from figure 6c to be 1.879, which neatly approximates the dimension 1.896 mentioned by Stauffer [4]. The small difference between these numbers can be explained by the relatively small size of our cluster.

3.4. Connectivity

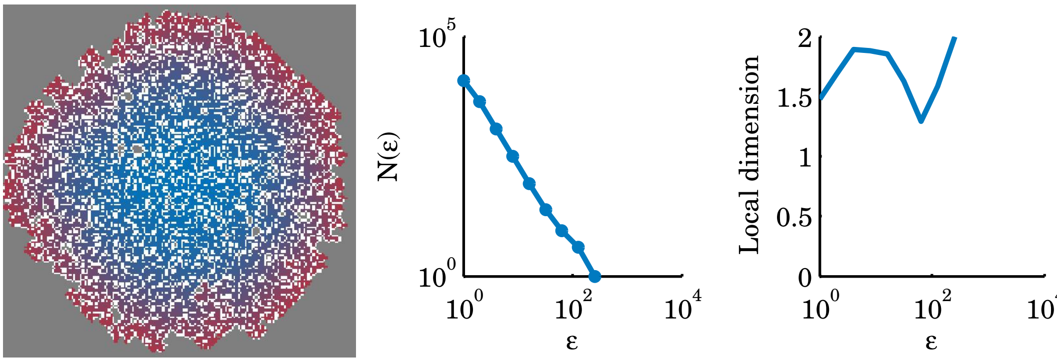
We consider two different connectivities, namely four- and eight-connectivity, shown in figure 7. In this section we discuss the same properties we have discussed for the four-connected percolation model.

The results of performing the same experiment as discussed in section 3.1 with the eight-connectivity mask, are presented in figure 8 and 9. We have changed the range of p to $p = 0.01, 0.02, \ldots, 0.99$, since using the range used for four-connectivity seemed too small.

It should be noted that for some values of p, especially larger values there are no mean cluster sizes, since no finite clusters were found. This indicates that one is much more likely to encounter a percolating cluster with eight-connectivity than with four-connectivity.

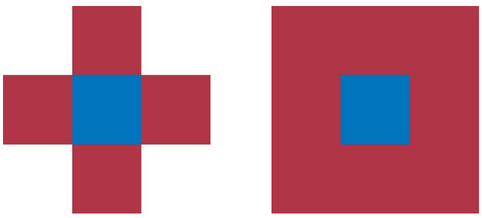
This is confirmed by figure 9, where we find that only for very low values of p P_∞ is zero, and that P_∞ quickly approaches 1.

These findings suggest that p_c is much lower when eight-connectivity is used. Based on this, admittedly small experiment, one would guess p_c to be approximately 0.2 when eight-connectivity is used. More research is needed to find the actual value of p_c when



(a) *The cluster.* **(b)** *The number of boxes as function of the box size.* **(c)** *The gradient of figure 6b.*

Figure 6: *(a) The cluster ($N = 80, p = 0.7$) used to compute the box-counting dimension. (b) The number of boxes used to cover that cluster as a function of the box size. (c) The gradient of the function plotted in (b).*



(a) *4-connectivity* **(b)** *8-connectivity*

Figure 7: *Connectivity masks for (a) four-connectivity and (b) eight-connectivity. The red squares are considered neighbors of the blue center square.*

eight-connectivity is used.

References

[1] Kenneth Falconer. *Fractal geometry: mathematical foundations and applications.* John Wiley & Sons, 2004.

[2] Wolfgang Kenzel et al. *Physics by computer.* Springer-Verlag New York, Inc., 1997.

[3] F. Moisy. *Computing a fractal dimension with Matlab: 1D, 2D and 3D Box-counting.* 2008. URL: `http://www.fast.u-psud.fr/~moisy/ml/boxcount/html/demo.html`.

[4] Dietrich Stauffer. *Introduction to percolation theory.* Taylor & Francis, 1985. Chap. 2, pp. 15–58.

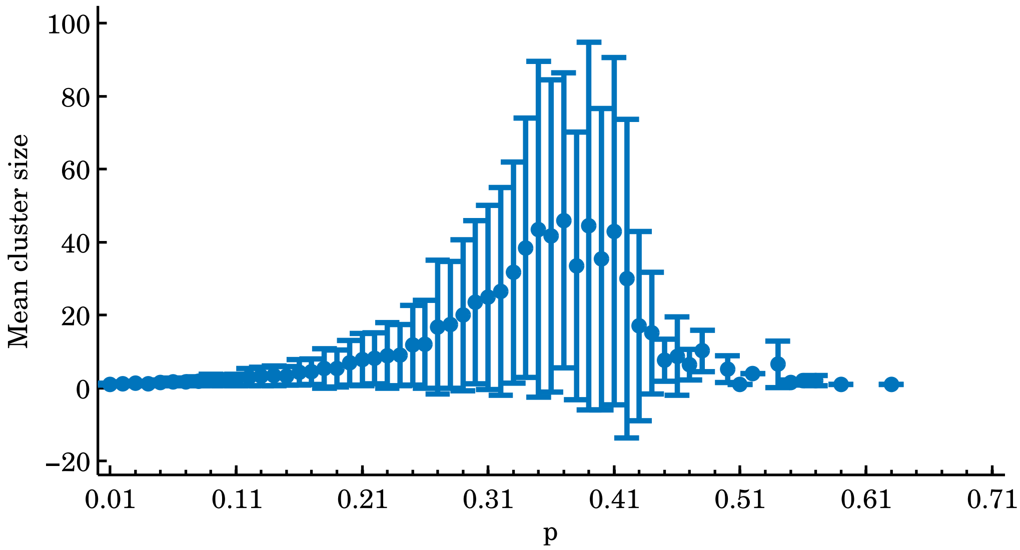


Figure 8: *Mean cluster sizes, represented as points, and standard deviations, indicated by the vertical error bars, as a function of* $p = 0.01, 0.02, \ldots, 0.99$ *when eight-connectivity is used. The mean and standard deviation were calculated over* 200 *runs on a* 41×41 *grid.*

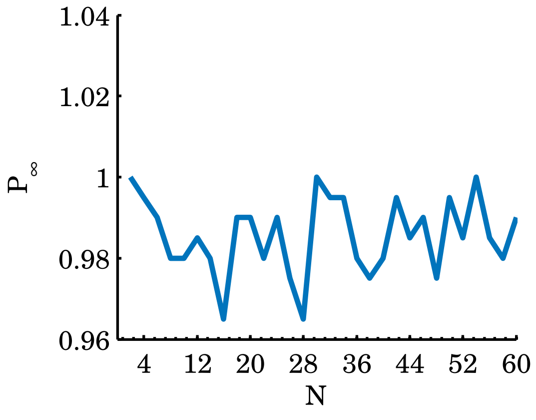


Figure 9: *Ratio of percolating clusters,* P_∞*, as a function of* $p = 0.01, 0.02, \ldots, 0.99$ *when eight-connectivity is used. Ratios are calculated over* $r_{max} = 200$ *runs on a* 41×41 *grid.*