# Numerical experiments.

**Abstract**

This is an addition to an article "Percolation of three fluids on a honeycomb lattice". The notions here are taken from this article.

In order to verify Conjectures 2-3 about percolation from the center, a computer program was written.

Each side of the hexagon $P$ is taken to be parallel to some side of a lattice cell. The program considers $k$ random colorings of the polygon $M_n$. Depending on input $p = 1, 2$, or 3, it computes the number $T_p$ of colorings with percolation of all the fluids $1, \dots, p$. (For example, if $p = 2$, then the program computes the number of colorings with percolation of both fluids 1 and 2.)

| Input | Output |
|---|---|
| $n$ | $T_p$ |
| $k$ | |
| $p$ | |

Output results for some $n$, $k$, and $p$ are shown in Table 1. Additionally, the value $P = (T_3/k)/(T_1/k)^3 \approx \frac{P(A_{1,n} \cap A_{2,n} \cap A_{3,n})}{P(A_{1,n})P(A_{2,n})P(A_{3,n})}$ is computed. Presence of values $P < 1$ does not disprove Conjecture 2, because it can be caused by statistical deviations. For $n = 500, k = 1000000, p = 1$ the program runs approximately 2 hours.

Table 1: Output results

| n | k | $T_1$ | $T_2$ | $T_3$ | $P$ | $P-1$ |
|---|---|---|---|---|---|---|
| 3 | 10 000 000 | 9 844 112 | 9 690 354 | 9 538 785 | 1.000 | <0.001 |
| 4 | 10 000 000 | 9 843 390 | 9 691 475 | 9 537 528 | 1.000 | <0.001 |
| 5 | 10 000 000 | 9 655 567 | 9 327 500 | 9 009 425 | 1.001 | 0.001 |
| 6 | 10 000 000 | 9 575 758 | 9 166 785 | 8 779 321 | 1.000 | <0.001 |
| 7 | 10 000 000 | 9 478 577 | 8 985 300 | 8 522 311 | 1.001 | 0.001 |
| 8 | 10 000 000 | 9 368 765 | 8 778 158 | 8 229 285 | 1.001 | 0.001 |
| 9 | 10 000 000 | 9 285 683 | 8 630 914 | 8 016 759 | 1.001 | 0.001 |
| 10 | 10 000 000 | 9 211 830 | 8 475 622 | 7 813 657 | 1.000 | <0.001 |
| 15 | 10 000 000 | 8 898 951 | 7 893 907 | 7 030 482 | 0.998 | -0.002 |
| 20 | 10 000 000 | 8 657 067 | 7 466 236 | 6 462 095 | 0.996 | -0.004 |
| 25 | 10 000 000 | 8 461 617 | 7 166 402 | 6 016 836 | 0.993 | -0.007 |
| 50 | 1 000 000 | 787 626 | 621 883 | 490 967 | 1.005 | 0.005 |
| 100 | 1 000 000 | 733 994 | 539 606 | 395 491 | 1.000 | <0.001 |
| 150 | 1 000 000 | 707 948 | 500 289 | 354 320 | 0.999 | -0.001 |
| 200 | 1 000 000 | 681 800 | 466 133 | 318 058 | 1.004 | 0.004 |
| 300 | 1 000 000 | 660 490 | 429 462 | 283 512 | 0.984 | -0.016 |
| 400 | 1 000 000 | 633 621 | 402 712 | 259 337 | 1.019 | 0.019 |
| 500 | 1 000 000 | 621 187 | 386 527 | 242 821 | 1.013 | 0.013 |
| 1500 | 1 000 000 | 555 244 | 299 740 | 166 587 | 0.973 | -0.027 |

Let us describe the algorithm checking whether fluid 1 percolates from the center $O$ to the boundary in a given coloring. Since the color of the center is irrelevant, we may assume that the center has color 0. (See Table 2 for examples.)

Fix a coloring. At the beginning, a beetle sits at the center. The beetle is allowed to move to an adjacent cell with color 0 or 1. We are going to find out if the beetle can reach the boundary.

**Step 1.** If the beetle is already in a boundary cell, then go to END1. Otherwise go to Step 2.

**Step 2.** If the color of the cell below the beetle is 0 or 1, then move the beetle downwards, and go to Step 1. If the color of the cell below the beetle is 2 or 3, then go to Step 3.

**Step 3.** (Building of a wall; see Table 2)

Construct a finite sequence (called a *wall* in what follows) of distinct adjacent sides of cells (called *wall segments*) as follows. The first term of the sequence is the bottom side of the cell where the beetle is located, and the second term of the sequence is adjacent to the left endpoint of the first wall segment. All the cells from one side of the wall have color 0 or 1, and all the cells from the other side of the wall have color 2 or 3. And finally, either the last wall segment lies between two boundary cells or all the wall segments surround some collection of cells of $M_n$. In the former case go to END1 and in the latter case go to Step 4.

(It is easy to see that the wall is uniquely defined by the conditions above.)

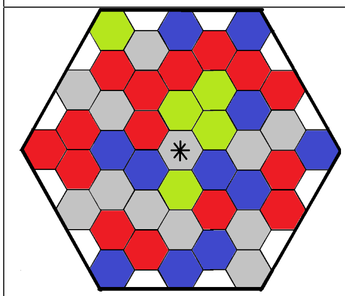
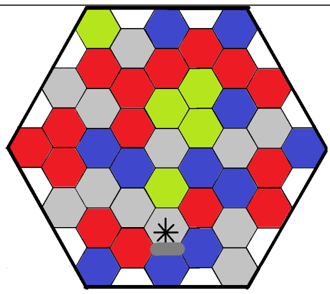
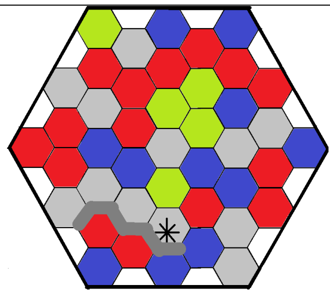
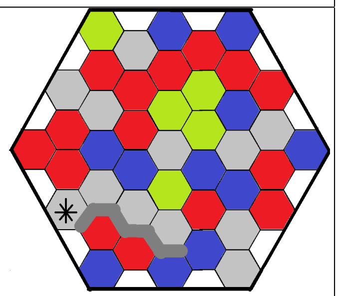
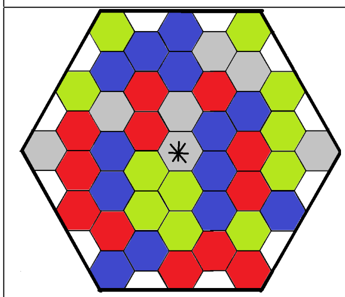
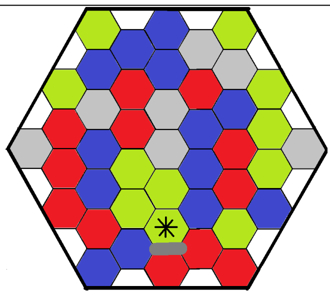
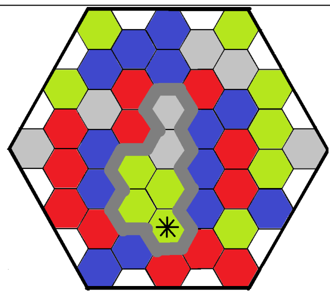
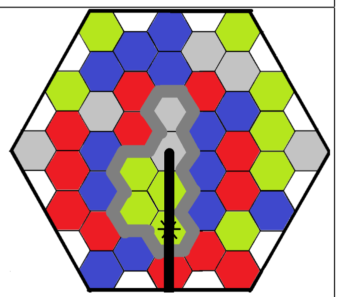
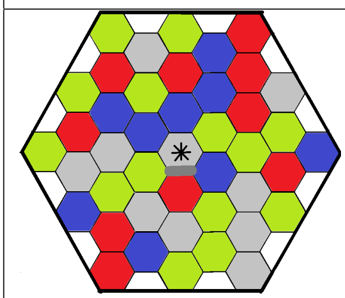
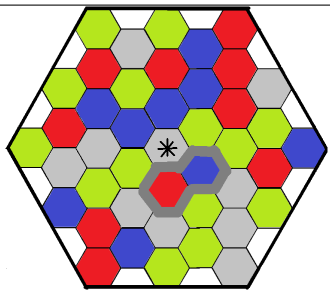
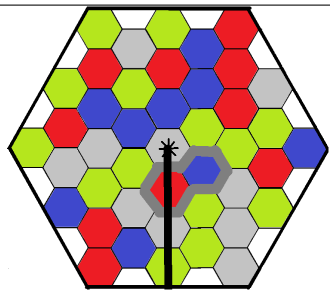
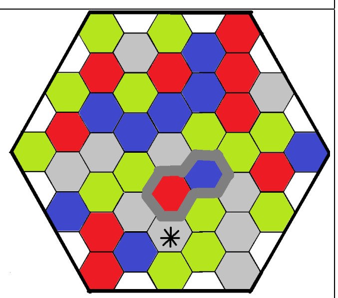
**Step 4.** Draw the ray from the center of $M_n$ to the bottom. If the ray intersects with wall segments (which were constructed in Step 3) an odd number of times, then go to END2. Otherwise, move the beetle to the cell right below the lowest wall segment intersecting the ray, and go to Step 1.

**END1.** There is a percolation from the center to the boundary.

**END2.** There is no percolation from the center to the boundary.

The following table shows how the algorithm works. In all three examples $n = 7$.

Table 2: Percolation-checking algorithm

<table>
<tr><td colspan="4">
</td></tr>
<tr><td colspan="4">Example 1</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td>At the beginning, the beetle is sits at the center.</td><td>The beetle moves downwards until it hits a blue cell. We put a wall segment right below the beetle.</td><td>We build the wall until we put a wall segment between two boundary cells.</td><td>The beetle can move along the wall to reach the boundary.</td></tr>
<tr><td colspan="4">Example 2</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td>At the beginning, the beetle sits at the center.</td><td>The beetle moves downwards until it hits a red cell. We put a wall segment right below the beetle.</td><td>We build the wall until it encloses some area of $M_7$.</td><td>We draw the ray from the center to the bottom. The ray intersects a unique wall segment. Hence the beetle cannot reach the boundary.</td></tr>
<tr><td colspan="4">Example 3</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td>At the beginning, the beetle sits at the center. It cannot move downwards, thus we put a wall segment right below the beetle.</td><td>We build a wall until it surrounds some area of $M_7$.</td><td>We draw the ray from the center to the bottom. The ray intersects two wall segments.</td><td>The beetle gets around the wall. Afterwards, it moves down and reaches the boundary.</td></tr>
</table>