ABSTRACT

In this paper we give three examples of expending patterns defined by hyperbolic cellular automata whose growth seems to be very similar to the growth of colonies of bacteria.

KEYWORDS: cellular automata, hyperbolic geometry, tilings, bacteria colonies.

1. INTRODUCTION

A very intriguing phenomenon of diffusion is given by the growth of colonies of bacteria, see [1]. As explained by Professor Ben-Jacob, very surprising structures can be obtained by putting such colonies in very severe conditions, see Figures 1 and 11. This gives a striking power of adaptability of these colonies. These experimental data are comforted by the discovery of bacteria in almost every possible hard conditions as geysers, ocean fathoms, core of the earth and even atomic piles. In the introduction of [1], Professor Ben-Jacob says:

Eons before humans, bacteria inhabited a very different Earth. As the earliest life form they devised ways to counter the spontaneous course of increasing entropy and convert high-entropy, inorganic substances into low entropy, organic molecules...
To change environmental hazards, bacteria resort to a wide range of cooperative strategies...
They collectively glean information from the environment, communicate, distribute tasks, perform distributed information processing and learn from past experience.

In many cases, their growth on plates used by microbiologists to study them constitute figures with a more or less fractal symmetry. Cellular automata were already used to model this, see [2]. Now, fractal symmetry may address to hyperbolic geometry. This is why we tried the other way: let us start from hyperbolic geometry, in fact from an appropriate tiling of the hyperbolic plane and try to simulate the observed growth with hyperbolic cellular automata.

In Section 2, we give the needed information for the reader about what to know about hyperbolic geometry in order to understand how our grid is obtained and to see how cellular automata are implemented in this context. In Section 3, we see how to proceed to the simulations indicated in the abstract. In Section 4, we conclude with ideas about possible continuations.

2. A TRIANGULAR TILING OF THE HYPERBOLIC PLANE

2.1. Hyperbolic Geometry

Here, we shall use one of the most popular models of hyperbolic geometry, Poincaré’s disc. This model is represented by Figure 2.

Inside the open disc represented in the figure we have the points of the hyperbolic plane. Note that by definition, the points on the border of the disc do not belong to the hyperbolic plane. However, these points play an important role...
in this geometry and are called **points at infinity**. Lines are trace of diameters or circles orthogonal to the border of the disc, e.g. the line $m$. In this model, two lines which meet in the open disc are called **secant** and two lines which meet at infinity, *i.e.* at a point at infinity are called **parallel**. In the figure, we can see a line $s$ through the point $A$ which cuts $m$. Now, we can see that two lines pass through $A$ which are parallel to $m$: $p$ and $q$. They touch $m$ in the model at $P$ and $Q$ respectively which are points at infinity. At last, and not the least: the line $n$ also passes through $A$ without cutting $m$, neither inside the disc nor outside it. This line is called **non-secant**.

The tiles of the heptagrid are very big. So, to obtain our grid, we first split each tile of the heptagrid into seven triangles whose vertices are the centre of the tile and the endpoints of its edges. This gives generation 1, see Figure 4. Then, we split the triangles into four triangles by taking the mid-points of the previous triangles: the new triangles are defined by a vertex of the previous triangles and the mid-points of the edges of the previous triangle which meet at the vertex. This defines generation 2, see Figure 5. We can go on the process inductively: the generation $n+1$ is obtained from the generation $n$ as generation 2 is obtained from generation 1. However, in this paper we shall focus on generation 2 only. Let us call it the **second triangular heptagrid, heptatrigrid** for short.

### 2.2. The Grid of our Simulation

From a famous theorem established by Poincaré in the late 19th century, it is known that there are infinitely many tilings in the hyperbolic plane, each one generated by the reflection of a regular convex polygon $P$ in its sides and, recursively, in the reflection of the images in their sides.

In the paper, we shall focus our attention on one of the simplest tilings belonging to this family: the tiling $\{7,3\}$, which we call the **heptagrid**, see Figure 3. Here, the polygons are regular convex heptagons, all vertices being shared by three of them.

### 2.3. Cellular Automata on the Hyperbolic Triangular Grid

Remember that such an automaton consists in a finite automaton $A$ attached to each 2-triangle of the heptatrigrid. A cell of the cellular automaton consists of $A$ and a 2-triangle which is the **support** of the cell. The neighbours of a cell $c$
with $T$ as its support have, as their supports, the 2-triangles which share a side with $T$.

Figure 6 indicates the basic elements of the location of a triangle. A coordinate is defined by four numbers in the format $(\sigma, \nu, \tau, \pi)$: $\sigma$ is the number of the sector in which the triangle lies; $\nu$ is the number of the heptagon of the sector in which the triangle lies; $\tau$ is the number in $[1..7]$ of the generation 1 triangle in which the triangle lies; and in this triangle, $\pi$ is the number of the triangle itself.

The numbering of the generation 1 triangle, we say later 1-triangle, is defined by the number of the side of the heptagon on which the 1-triangle is built, a number in $[1..7]$. For the other heptagons, side 1 is shared by the heptagon of the sector number $i$.

We consider that the father of the root of the tree in each sector is the central cell. For precise explanations on these notions, the reader is referred to [3].

In each 1-triangle, we have four triangles of generation 2, we call them 2-triangles. The four 2-triangles of a 1-triangle are numbered from 0 to 3. First, we number the vertices and the sides of a 1-triangle $T$ from 0 to 2: 2 is the centre of the heptagon, 0 and 1 are the vertices of the side of the heptagon defining $T$; the side 0 is opposite to the vertex 1. Following the counter-clockwise orientation, 0 comes before 1. Now, the number of a vertex of $T$ is the number of the 2-triangle which possesses this vertex. Accordingly, 3 is the number of the 2-triangle whose vertices are the mid-points of the edges of $T$. This numbering can be repeated for any further generation, see [4]. This numbering has interesting properties. The interested reader is referred to [4] for more information.

Now, to implement cellular automata, we have to compute the coordinates of the neighbours of a cell $c$ from the coordinate of $c$ itself.

Let $T$ be the 2-triangle which supports the cell. Number the neighbours of $c$ from 0 to 3, $c$ itself being neighbour 0. For the other numbers, the neighbour $i$ of $c$ is the 2-triangle which shares the side $i-1$ of $T$. Let $(\sigma, \nu, \tau, \pi)$ be the coordinate of $T$, the support of $c$. The coordinates of the neighbours of $T$ are given by Table 1.

![Figure 6. From the Heptagrid to the Heptatrigrid](image)

| neighbour | sector | number | slice | place |
|-----------|--------|--------|-------|-------|
| 0         | $\sigma$ | $\nu$ | $\tau$ | 0     |
| 1         | $\sigma$ | $\nu$ | $\tau$ | 3     |
| 2         | $s(\sigma, \nu, \tau)$ | $v(\tau, \nu)$ | $l(\nu, \tau)$ | 1     |
| 3         | $\sigma$ | $\nu$ | $\tau \oplus 1$ | 1     |
|           | $\sigma$ | $\nu$ | $\tau$ | 0     |
| 2         | $\sigma$ | $\nu$ | $\tau \oplus 1$ | 2     |
| 3         | $\sigma$ | $\nu$ | $\tau$ | 3     |

| neighbour | sector | number | slice | place |
|-----------|--------|--------|-------|-------|
| 0         | $\sigma$ | $\nu$ | $\tau$ | 2     |
| 1         | $\sigma$ | $\nu$ | $\tau \oplus 1$ | 2     |
| 2         | $\sigma$ | $\nu$ | $\tau \oplus 1$ | 2     |
| 3         | $\sigma$ | $\nu$ | $\tau$ | 3     |

As can be seen in the table, each 2-triangle has at least one neighbour which is in the same heptagon and in the same slice of the heptagon. Note that a 2-triangle with place 3 has all its neighbours in the same slice of the same heptagon. A 2-triangle with place 2 has all its neighbours in the same heptagon, but two neighbours are in the slices which are adjacent to its own one. This is indicated by the expressions $\tau \oplus 1$ and $\tau \ominus 1$. As the number of a slice is in $[1..7]$, subtracting 1 from 1 gives 7 and adding 1 to 7 gives 1. For $\tau \in [2..7]$, $\tau \ominus 1 = \tau - 1$, and for $\tau \in [1..6]$, $\tau \oplus 1 = \tau + 1$. For the 2-triangles with place 0 and 1, the computation of the coordinates of their neighbours is more complex. Indeed, in each case, one of the neighbours do not belong to the same heptagon $H$, but to a heptagon $K$ neighbouring $H$. This changes the value of $\nu$ and it may also change the values of $\sigma$ and $\tau$. This is indicated in Table 1 by the expressions $s(\sigma, \nu, \tau)$, $v(\tau, \nu)$ and $l(\nu, \tau)$.

We have no room to explain the computation of these ex-
expressions. The reader can found them in [7], more explanations and computations are to be found in [3, 4].

Clearly, the coordinate of a neighbour $K$ of a heptagon $H$ with coordinate $\nu$ depends on the side $\tau$ shared by $H$ and $K$. Now, the side numbered by $\tau$ in $H$ does not receive the same number in $K$ and we shall say that $K$ is the neighbour $\tau$ of $H$. The correspondence between these numbers gives the value of the function $t(\nu, \tau)$ and, for completeness, we give it in Table 2. Note that the sides of the central cell are all numbered by 1 in its neighbours. For the other cells, the correspondence depends on the status of $H$ and it may also depend on that of $H$. Side 7 is always the side shared by a neighbour which is on the same level of the tree, even when there is a change of tree by the change of sector. If $H$ is black, its side 7 is numbered 2 on the other side. If $H$ is white, the number of its side 7 in the other neighbour $K$ depends on the status of $K$ as indicated in the table.

| black in $H$ | white in $H$ |
|-------------|--------------|
| in $K$      | in $K$       |
| $3^KwK$     | $4^KwK$      |
| 1           | 1            |
| 2           | 2            |
| 3           | 3            |
| 4           | 4            |
| 5           | 5            |
| 6           | 6            |
| 7           | 7            |

Table 2. Correspondence between Side Numbers

On the left-hand side, the numbers in a heptagon $H$. On the right-hand side, the numbers in a heptagon $K$, $H$ and $K$ sharing the considered edge. Note that if $H$ is white, the other number of side 1 may be 4 or 5 when $K$ is white and that it is always 5 when $K$ is black.

To conclude with this section, let us remember that cellular automaton have been implemented in several grids of the hyperbolic plane. The complexity classes of these cellular automata have been investigated leading to very surprising results. Several universal cellular automata also have been implemented in these grids. We refer the reader to [4, 5] for more information and more references.

3. THE SIMULATIONS

The basic idea behind our simulations is the propagation of the tree structure of the heptagrid by a cellular automaton. The result, illustrated by Figure 7 convinced us that we could try to simulate colonies of bacteria. We propose three of them which are examined in Sub-section and which differ by the number of states of the cellular automaton which is used for the simulation.

3.1. First Simulation: the Propagation of the Tree Structure

The tree structure of the heptagrid can be implemented by cellular automata on this grid: this was illustrated in [6] in order to give a toy example of a cellular automaton on this grid.

![Figure 7. Cellular Automata and the Tree Structure](image)

The propagation of the tree structure of the heptagrid with the indication of the grid.

We can do the same here and Figure 7 gives the 36th step of execution of this automaton starting from an initial configuration in which the seven 2-triangles of place 2 of a heptagon are in the same state, in red in the figure: we call this the core-2 configuration. As we can see, the automaton has a non-small number of states: 18 of them. In [6], we had 5 states only. In fact, it is possible to have 4 states in the case of the heptagrid if we do not need to differentiate the two white sons of a white node. We need much more states here as we wish to diffuse the structure of the tree with its two types of rules. For programming reasons, it was easier to program the automaton by implementing the following strategy: when the automaton enters a heptagon, it goes as soon as possible to the 2-triangles with place 2. There, by a counting process, it determines the directions of the sons from the direction of the father which is the direction from which the automaton entered the cell.

The way the automaton is working can be seen as an animation on the slides which are deposited on [8].

It seems to us that the result has a striking similarity with pictures about the growth of colonies of bacteria in highly stressed conditions, see Figure 8.
3.2. The other Simulations

In this sub-section, we successively examine three attempts to simulate the propagation of colonies of bacteria. We shall consider the number of states we use as well as the information that the cells are assumed to know about themselves. We shall try to give the states and these assumptions a kind of biological flavour. We have to keep in mind the specificity of the cellular automaton programming. A cell cannot directly act upon another one. Such an action has to be 2-stepped: if \( c \) wants to act on a neighbour \( n \), \( c \) has to signal this intention by taking a particular state. Seeing this state on \( c \), and possibly seeing an additional information displayed by its other neighbours, \( n \) can interpret the intention and take the desired state. However, we often speak in a direct manner, that \( c \) acts on \( n \) in this or that way.

With two states

The states are white for the medium, black for the colony. The cells want to propagate, but competition is not encour-aged. This can be rather simply formulated as follows:

(a) A black cell remains black.
(b) A white cell becomes black if and only if it has exactly one black neighbour at this time.

Figure 9 illustrates the \( 36^{th} \) time of this situation starting from the core-2 configuration. We can see that the colony invades almost all the space, leaving holes unoccupied. The condition on the change of the white cell to a black one has, as a consequence, that a white cell which has two black neighbours exactly remains white. This is the reason of the pairs of adjacent white cells which are regularly produced in the evolution of the automaton.

With four states: version 1

Now we have four states: \( W, R, Y \) and \( V \) calling them white, red, yellow and vermilion respectively. White represents the medium. Red is almost the initial configuration which is, here again, the core-2 configuration.

The action of the cells is now:

(a) A red, yellow or vermilion cell remains in its colour.
(b) A white cell becomes red, yellow or vermilion if and only if, at this time, it has exactly one neighbour which is red, yellow or vermilion respectively.

Here too, when a white cell becomes non-blank, it keeps the new colour for ever.

The above picture of Figure 10 illustrates the \( 36^{th} \) step of computation starting from the core-2 configuration. The non white cells occupy a heptagon exactly, with the pattern we have in the figure for the red cells. We call this the heptagonal core configuration. We can notice that in this case also, the cells which remain white are the same as those of
the previous automaton. We also notice that, thanks to the core-2 configuration, the red state no more occurs in the computation. After the initial time, the computation outside the heptagonal core involves three states only: white, yellow and vermilion.

Figure 10. New Comparison, 4 States and Bacteria

Above, a diffusion process with 4 states based on local knowledge of the colony. Below, the colony of Figure 1, picture by courtesy of Professor Ben-Jacob.

In view of the bottom picture of Figure 10, it seems reasonable to consider that yellow and vermilion states together represent the colony.

With four states: version 2

Here, we again have four states. But we also assume that the colony has some knowledge of the geometry of the space. This can be viewed as an acquired experience of the space by the colony. We assume that a cell knows its place and whether it is in slice 1 or not. It is easy to see that this is a 2-bit information only. We take the same colours as previously, with white as the state for the space. Here too the computation starts from the core-2 configuration.

This time too, the formulation of the rules is in the same style as previously but it becomes more intricate, as it involves the place and the slice of a 2-triangle.

(a) A red, yellow or vermilion cell remains in its colour.
(b) If a white cell has two white neighbours and if its slice is 1, then it takes the colour of its third neighbour.
(c) If a white cell has two white neighbours and if its slice is not 1 but its place is 2 then if its third neighbour is red, yellow or vermilion, it becomes yellow, vermilion or red respectively.
(d) If a white cell has two white neighbours and if its third neighbour is red then, if its place is 3, 0 or 1, it becomes red, vermilion or yellow respectively.
(e) If a white cell has the states white, yellow and vermilion among its three neighbours, if its slice is 1 and if its place is 3, then it becomes red.

With these rules, the cellular automaton behaves in a somewhat different manner. As can be seen from the upper picture of Figure 11, although the initial configuration is the same as previously, the four states are now involved during the whole computation. Moreover, the colony does not occupy the whole space: the branches which regularly spread out are far away from each other, which avoid any kind of competition. Also, we can see that this time we have a cooperation between the states. The knowledge whether a cell is in a slice 1 or not allows the colony to take advantage of the topology in order to invade the center of a heptagon, according to the scenario contained in the condition c of the rule. Next, the conditions d and e allow to occupy the slices 4 and 5 of the heptagon and those ones only, without knowing the number of the slice. This is obtained by the combination of the conditions c, d and e. Once this is checked for one heptagon around the central cell, this is repeated for all the heptagons which are the 4- and 5-neighbours of a heptagon. This way, we obtain seven binary trees which grow from the heptagonal core.

In Figure 11, we compare the growth of the upper picture the figure with a picture of another bacteria colony. Note that the upper picture of Figure 11 does not show the drawing of the heptagrid, as the above picture of Figure 8 from Figure 7. The computer program which draws the figure writes down a PostScript file from the information obtained by performing the simulation up to the 36th step, starting from the heptagonal core. In this writing, the program simply removes the drawing commands, simply keeping the filling commands which allow to paint closed areas defined for drawing the same figure.
It should be remarked that in all the previous simulations, the computation may be as long as wished within the time and memory limits of a computer. Due to the exponential growth of the number of 2-triangles as we go away from the central heptagon, these limits are rapidly reached and improvements in technology may perhaps allow us by one round of 2-triangles further each time the capacity is multiplied by 3. However, for simulations of actual colonies of bacteria, this is not a problem as their growth is not only finite but also small in the hyperbolic scale.

It seems to us that this hyperbolic simulation gives an interesting approximation of the phenomenon observed in real experiments. The above discussion about the space of computation indicates that it could be interesting to investigate generation 3 of triangles and so, to look at what we obtain for 3-triangles. Most probably, we could get a finer simulation but certainly at the price of a bigger number of states. The interpretation of these states from a biological point of view is of course a question as well as how much of the knowledge of the space could be allowed for 3-triangles where a third parameter within the place is necessary. These are directions for further work on this topic.

4. CONCLUSION

ACKNOWLEDGEMENTS

The author is very much in debt to Professor Ben-Jacob for the pictures he sent him and for the permission of publishing them in this paper. He also much in debt to the organizers of the Special Section CAAA, Dominique Désérable, Rolf Hoffmann and Genaro Juárez Martínez, for helping him in the submission process of this paper.

REFERENCES

[1] E. Ben-Jacob, “Social behavior of bacteria: from physics to complex organization”, European Physical Journal B, 65(3), (2008), 315-322.

[2] G.B. Ermentrout, L. Edelstein-Keshet, “Cellular automata approaches to biological modelling”, Journal of Theoretical Biology, 160(1), (1993), 97-133.

[3] M. Margenstern, CELLULAR AUTOMATA IN HYPERBOLIC SPACES, vol.1, THEORY, Old City Publishing, Philadelphia, (2007), 422p.

[4] M. Margenstern, CELLULAR AUTOMATA IN HYPERBOLIC SPACES, vol.2, IMPLEMENTATION AND COMPUTATIONS”, Old City Publishing, Philadelphia, (2008), 360p.

[5] M. Margenstern, “An algorithmic approach to tilings of hyperbolic spaces: 10 years later”, Lecture Notes in Computer Sciences, 6501, (2010), 32-52.

[6] M. Margenstern, “A universal cellular automaton on the heptagrid of the hyperbolic plane with four states”, Theoretical Computer Science, 412, (2011), 33-56.

[7] M. Margenstern, “Bacteria inspired patterns grown with hyperbolic cellular automata”, arXiv:1012.2771, (2010), 16p.

[8] File on the personal site of M. Margenstern: http://www.lita.univ-metz.fr/~margens/look_test.pdf