mathematics

MDPI

Article

Layered Growth of 3D Snowflake Subject to Membrane Effect and More than One Nucleation Center by Means of Cellular Automata

César Renán Acosta *,†, Irma Martín † and Gabriela Rivadeneyra †

Applied Physics Department, Faculty of Engineering, Autonomous University of Yucatan, Mérida 97310, Yucatán, Mexico; irma.martin@correo.uady.mx (I.M.); g.rivadeneyra@correo.uady.mx (G.R.)

* Correspondence: crenan@correo.uady.mx

† All authors contributed equally to this work.

Abstract: In this work, it is taken into account that in nature, due to pressure and temperature, water drops in general are either spherical or ellipsoidal. Thus, starting from a more general structure, a 3D elliptical surface (oblate spheroid) is constructed, which, by means of parameters, can be turned into a spherical shape. Hexagons are built on a rectangular horizontal plane, then this plane is passed through an elliptical surface at height h, which is determined by a parameter θ. As a result of the cutting of these surfaces, a curve and a plane are obtained, both horizontal ellipsoidal; if these hexagons are within the perimeter of the horizontal ellipse obtained as a function of θ, they are marked with an N, and if they are outside the perimeter, they are marked with an E. Several frozen nucleation centers are established, either in the same layer or in different planes, marking them with an F and their first eight neighbors with a B. The calculations based on a modified snowflake model are carried out tile by tile and layer by layer, governed by the thermodynamic factors α, β, and γ, leading to results that depend on the position of the nucleator, which can be symmetrical or asymmetrical for a snowflake with more than one nucleation center and an external surface formed by water vapor that functions as a membrane.

Keywords: cellular automaton; snowflake; hexagonal lattice; discrete Laplacian

MSC: 37M05

check for updates

Received: 21 December 2024
Revised: 16 January 2025
Accepted: 20 January 2025
Published: 28 January 2025

Citation: Acosta, C.R.; Martín, I.; Rivadeneyra, G. Layered Growth of 3D Snowflake Subject to Membrane Effect and More than One Nucleation Center by Means of Cellular Automata. *Mathematics* **2025**, *13*, 433. https://doi.org/10.3390/math13030433

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In general, the use of cellular automata to model snowflake growth is based on the Reiter model [1] or on variants of it [2–4], which have been successful in obtaining 2D models. The underlying idea to bring these same growth dynamics to a 3D structure is that the layers formed by the water hexagons in their different phases are interconnected, which is given by considering that the hexagonal tiles have six first neighbors in their same plane, while there are two additional first neighbors that connect the automaton, one with the immediate upper layer and another with the adjacent lower layer, as seen on Figure 1. Thus, the thermodynamics that reproduce the model for the snowflake propagates in all the layers [5–7].

In the Reiter model, there are four states for a tile: F (frozen cell), B (boundary cell), N (non-receptive cell) and E (edge cell). In the case of growth for 3D models, there are the edge cells at the perimeter of the layer, but also the two boundary layers at the bottom and top of the volumetric configuration—all its cells are edge cells. These four states are

controlled by an iteration function, which involves the application of the diffusion function, calculated through a graph associated with a central tile with its eight neighboring cells.

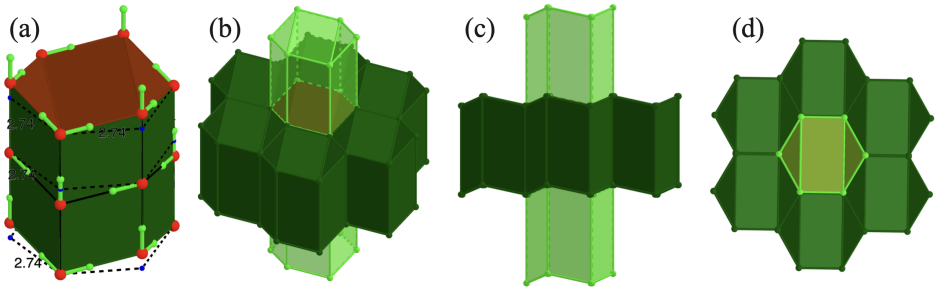


Figure 1. (**a**) Toothed hexagonal polyhedron (the automaton) formed by water when it crystallizes, where the red atoms represent oxygen and the green atoms represent hydrogen. (**b**–**d**) Horizontal and vertical lateral views of the first neighbors when eight of these polyhedra join together.

In terms of computational efficiency, there are two ways to apply these models: one is where the element that triggers the process is the frozen cell and the first two or three neighbors of the frozen tile are checked. In terms of efficiency, it is very fast, with visually acceptable 2D results, but in 3D, they generate a central layer that grows very quickly, being also symmetrical with respect to this central layer. In a second way of applying the model (this second method is the one we use in this work), where the process activator is the surrounding medium through pressure and temperature (i.e., thermodynamics), each stage of the process starts with the bottom layer, checking it tile by tile, performing the averages and applying the formulation in both modes. Once the first layer is finished, we move on to the second and to the third . . . ; what can be seen is that in the first stage of the process, both methods are the same, but from the second and subsequent stages onwards, the averages begin to differ, not only for the first two or three neighbours, but for a number of neighbours that grow at each stage, even when the tile marked as (boundary cell) remains in a non-frozen state [8–11].

In order to establish a limit to the growth of the snowflake, a surface is designed in space in the form of an oblate spheroid (flattened ellipse), which contains the drop of water in its three different phases (vapour, liquid, and solid) which we will call "the membrane", which, with the procedure of cellular automata, is led to form the snowflake. This ellipsoidal appearance is acquired due to the temperature and pressure conditions of the medium, and it could also acquire a spherical configuration or any other form. The function of the membrane is to limit the planes of hexagons and separate the types of cells (E or N).

Thus, in this work, the Reiter model, which is specifically designed for 2D snowflakes (six first neighbors in a plane), is modified by adding two first neighbors, one at the top of the automaton and another at the bottom; this results in adjacent surfaces communicating through thermodynamic parameters, giving rise to the generation of a 3D object. Likewise, other nucleation centers are randomly added (in the original model there is only one nucleation center), which generates a greater variety of shapes [12–14].

In addition to the modifications to the model, an oblate spheroid surface is designed with the aim of approximating the behavior of the automaton to that of a snowflake in nature.

2. The Elliptical Membrane

To define the oblate spheroid (or just spheroid) surface in space with parameters θ and φ in the first octant of the coordinate system, we will do so through the major semi-axis a

and minor semi-axis b, which determine the size and shape of the drop. It is noted that the sweep of θ is from 0° to 180° and that of φ is from 0° to 360°.

$$g(\theta, \varphi) = \left(a \sin\theta \cos\varphi + a, b \sin\theta \sin\varphi + b, \frac{b}{a} \cos\theta + \frac{b}{a} \right) \tag{1}$$

Based on these semi-axes, the number of rows and columns of tiles that conforms to a horizontal flat surface is calculated, which is passed at different heights determined by a parameter θ_1 through the spheroid surface.

$$\begin{cases} \text{número de columnas} & n = \frac{b}{\sigma \sin\left(60 \frac{\pi}{180}\right)} \\ \text{número de filas} & m = \frac{2a}{\sigma} \end{cases} \tag{2}$$

where σ is the side of the hexagon that forms a tile. With the same function, a plane with parameters r, φ is defined, which is the result of cutting the spheroid surface (Figure 2) with the rectangular tile plane.

$$h(r, \varphi) = \left(r \sin\theta_1 \cos\varphi + a, \frac{br}{a} \sin\theta_1 \sin\varphi + b, \frac{b}{a} \cos\theta_1 + \frac{b}{a} \right) \tag{3}$$

It is noted that the sweep of r is from 0 to a and that of φ is from 0° to 360°. This plane is located at a height defined by θ_1, which is established according to the heights of the planes of the tiles; that is, we place planes starting from 30° to 150° every 30°—this angular difference is established based on the height between the tiled planes.

The horizontal and rectangular planes with the tiles are passed through the spheroid surface; so, where they intersect, they form an elliptical perimeter, which is used to determine if a tile is inside or outside that perimeter. The tiles that are inside the perimeter can acquire the different states N, B, or F, so those that are outside can only be in state E.

The function of the horizontal perimeter ellipses (Figure 2b) formed when the tile planes and the spheroidal surface (Figure 2a) are cut is specified in Equation (4); the resulting ellipsoidal horizontal planes are shown in Figure 2c, which contain the cells that are to form the 3D snowflake layer by layer.

$$\psi(\varphi) = \left(\frac{ab \sin\theta_1 \cos\varphi}{\sqrt{a^2 \sin^2\varphi + b^2 \cos^2\varphi}} + a, \frac{ab \sin\theta_1 \sin\varphi}{\sqrt{a^2 \sin^2\varphi + b^2 \cos^2\varphi}} + b, \frac{b}{a} \cos\theta_1 + \frac{b}{a} \right) \tag{4}$$

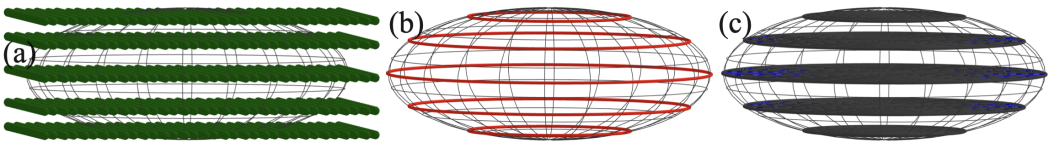


Figure 2. (**a**) Spheroidal surface in space and horizontal planes placed at every 30° where they intersect, forming (**b**) ellipsoidal perimeter and (**c**) plane (also ellipsoidal) with tiles inside.

Equation (4) is used for an elliptic curve since it describes an ellipse centered at (h, k), where $h = a$ and $k = b$ and which is at a height of $z = b/a \cos\theta_1 + b/a$. Therefore, the function of the membrane is to define the tiles, with their possible states to reach, that remain within the relevant perimeter, in addition to limiting the growth.

3. Snowflake Model

The modeling of the growth of a snow crystal from an ice seed, which is a speck of dust, or pollen, or ash that groups water molecules that freeze, is given by the diffusion transport of vapor within the system and is carried out through the diffusion Equation (5) [1–5].

$$\frac{\partial \varphi}{\partial t} = D\nabla^2 \varphi \tag{5}$$

where D is the diffusion constant. Considering the hexagonal structure at the base that water forms when it crystallizes as a graph, which is formed by a vertex joined to its first eight neighbors, the approximation of the weighted discrete Laplacian, which is represented as $\nabla^2 \varphi \approx \Delta_\gamma \varphi$ at a position v and at a time t, with equal weights γ_{uv} on each edge, affecting a function that is denoted as $\varphi(t, v)$, is given by the following [8–11]:

$$(\Delta_\gamma \varphi)(t, v) = \sum_{u:d(u,v)=1} \gamma_{uv}[\varphi(t, v) - \varphi(t, u)] \tag{6}$$

where $d(u, v)$ in Equation (2) is the distance of the graph between vertices u and v. Therefore, this sum is applied to the nearest neighbors u of the vertex v. The approximation for a hexagonal lattice at the base and two additional neighbors just above and below the cell under review is

$$\varphi(t+1,\ v) \approx \varphi(t, v) + \frac{\alpha}{12}\left[-8\varphi(t, v) + \sum_{N \in nn(v)=1}^{N} \varphi(t, N)\right] \tag{7}$$

where it is taken into account that $D\gamma_{uv} = \alpha/12$ in Equation (3)—that is, the center of the graph contributes with a weight of $2/3$ and the nearest neighbors with $1/12$ when $\alpha = 1$. The schematic description of the beginning of a tessellation process is shown in Figure 3.

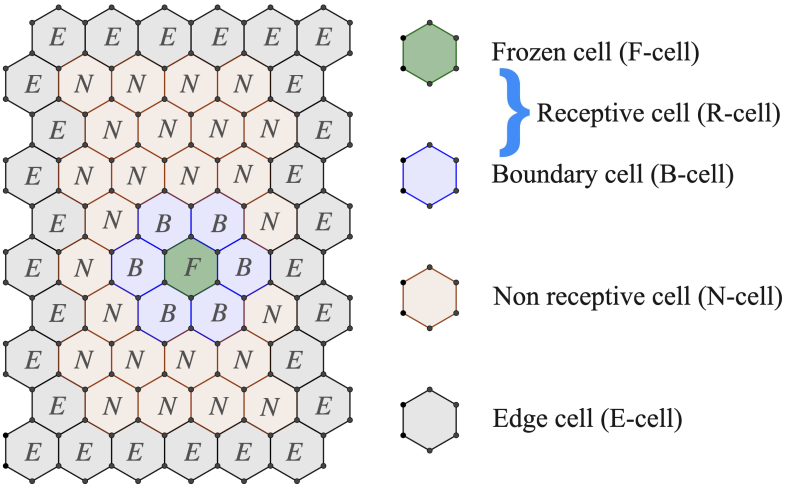


Figure 3. The cellular automata are shown in their different states deployed around a nucleator. The frozen tiles are marked with *F*, the boundary cells with *B*, the non-receptive cells with *N*, and the edge cells with *E*.

All cells are described by a continuous state function $s_t(z)$, which is taken as an indirect measure of the amount of water stored in each cell z [6,7], whose values are between 0 and 1 for unfrozen tiles, which represents the frozen state; this state $s_t(z)$ updates its value based on the thermodynamic factors α, β, and γ. In their initial state, one or more cells are defined as $s_0(z) = 1$, which can be in defined positions in a single layer or in different planes or any other randomly determined location. All the remaining tiles take the value of β, so if $s_t(z) \geqslant 1$, it is a frozen cell (F), but if one of its first neighbors is frozen, then it is a boundary cell (B). It is also established that if $s_t(z) < 1$ and none of its first neighbors has the condition of $s_t(z) \geqslant 1$ then it is a non-receptive cell (N). Finally, if the tile is at one end of the defined surface, then it is an edge cell (E).

We then have that the procedure for marking the tiles begins with the complete lower and upper layers, labelled as edge cells (E). The "interior" layers initially have two types of tiles: those that are on the perimeter or outside of it (in this case, each layer is identified as edge cells (E)) and the remaining ones within the perimeter are non-receptive cells (N). We locate a nucleator that is a frozen cell, which replaces the state of that cell, which now has one N, at a point in the central layer or in any other layer, whether defined or randomly chosen, with its first eight neighbors marked with a B, replacing the N that it had previously.

The initial conditions in this three-dimensional model [1,2] are as follows:

$$s_0(z) = \begin{cases} 1 & \text{si} \quad z = \mathcal{O} \\ \beta & \text{si} \quad z \neq \mathcal{O} \end{cases} \tag{8}$$

where $\mathcal{O}$ refers to the position of the cells with the characteristic of being frozen—that is, they are F cells. Additionally, the following functions are defined in a z cell: a measurement of the amount of water stored in the cell that participates in diffusion $utu_t(z)$, which evaluates the freezing state of a non-receptive or edge cell (N or E)—edge cells participate in diffusion but have a constant state $u_t(z) = \beta$; there is also a measurement of the amount of water stored in the cell, which does not participate in diffusion $v_t(z)$, which evaluates the freezing state of a receptive cell (B or F). Therefore, the state function is as follows:

$$s_t(z) = u_t(z) + v_t(z) \tag{9}$$

where if z is a receptive cell (B or F cells), we have $u_t(z) := 0 \rightarrow v_t(z) := s_t(z)$. Now, if z is a non-receptive cell (N cells), we have $v_t(z) := 0 \rightarrow u_t(z) := s_t(z)$. It is worth mentioning that no punctual operations are performed on the edge tiles (E), but they do contribute to the diffusion of water vapor when operations are performed on their neighboring tiles, whether they are receptive (B or F) or non-receptive (N).

In this model [1], the state of a cell evolves as a function of the states of its neighboring tiles and it has the following possibilities:

For any receptive cell (B or F), its posterior state $v_{t+1}(z)$ is updated via the following:

$$v_{t+1}(z) := v_t(z) + \gamma + \frac{2\alpha}{3}\left[\overline{u_t(z)}\right] \tag{10}$$

In Figure 3 and Equation (10), it can be seen that the boundary cells (B and $0 < v_t(z) < 1$) have as their first neighbors some non-receptive cells N with $u_t(z) \neq 0$—thus, Expression (10) applied to a boundary cell B that has $u_t(z) = 0$ only has the average of the first eight neighbors, but this average only takes values for the non-receptive cells N, implying that the boundary cells also have diffusion (Equation (6)). The frozen cells F are surrounded by boundary cells B, so the average is zero $\overline{u_t(z)} = 0$; therefore, frozen cells (F) are not affected by diffusion.

For any non-receptive cell (N), its posterior state $u_{t+1}(z)$ is updated through the following:

$$u_{t+1}(z) := u_t(z) + \frac{2\alpha}{3}\left[\overline{u_t(z)} - u_t(z)\right] \tag{11}$$

We have used subscripts in Equations (10) and (11) to denote the updated functions given the varying state of a cell before and after a step is completed, and written $\overline{u_t(z)}$ for the average of $u_t(z)$ for the eight nearest neighbors of cell z, which only takes values for N-type cells (Equations (10) and (11)). The cells at the edge (E cells) of the model setup are referred to as edge or marginal cells, in which $u_{t+1}(z) := \beta$ is set. Therefore, in the diffusion process, water vapor is added to the system through these marginal cells.

Combining the two intermediate states that have been updated gives the revamped state of the cell (Equation (8)) at the end of a stage [1,2]:

$$s_{t+1}(z) = u_{t+1}(z) + v_{t+1}(z) \tag{12}$$

By varying the parameters α, β, and γ in this model, the geometric shapes of the snow crystal on each surface develop and this contributes, through the effect of diffusion with its neighboring cells, to the evolution under the same state functions of the upper and lower surfaces, generating the 3D patterns of the snowflakes.

4. Structuring the DataFrame

To construct a tessellated surface where each tile is a regular hexagon and the central point of each hexagon represents the entire tile, alternating lines guided by two base vectors generating hexagons must be constructed (Equation (13)), for which the end points of these base vectors are represented as follows:

$$A = (a,\ 0)\ ;\ B = \left(a\cos\left(\frac{60\pi}{180}\right),\ a\sin\left(\frac{60\pi}{180}\right)\right) \tag{13}$$

where a is the side of the hexagon. To build a frame n rows high and m columns wide given by (Equation (2)), the building must be carried out line by line with the following:

lpc = Sequence(Sequence(2kB + qA, q, −k, m − k), k, 0, n)

lsc = Sequence(Sequence((2k + 1)B + qA, q, −k, m − k), k, 0, n)

Each sequence is defined by the function to be evaluated and the parameter that controls the function, where the discrete evaluation of the function starts and ends—that is, we assume that to go from $-k$ to $m - k$, we carry out the process in steps of 1 (likewise for k to go from 0 to n). This generates a frame in which the central points have the coordinates obtained with the sequences lpc and lsc; the two sequences have been grouped to have a continuous count, whose representation with their indexes is shown in Figure 4.

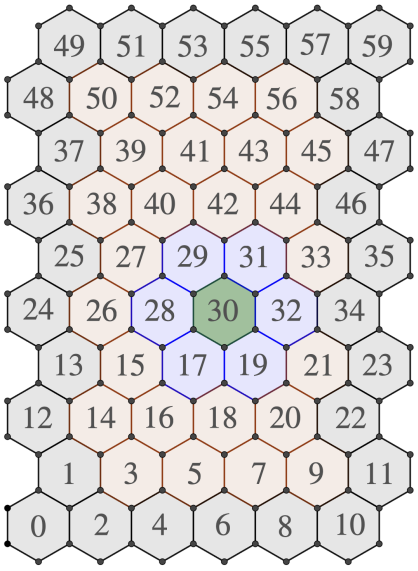


Figure 4. Indexes of the points generated in the two sequences lpc and lsc, which have been grouped to have a continuous count. For the array shown, $n = 4$ and $m = 5$.

The indices of those points can therefore be used to locate the first six neighbors of each hexagon, complementing them with the points that are in the layers just above and below the cell being operated on. Thus, if we repeat this same count in all the layers, they all have

the same first neighbors, only changing the vertical coordinate of each point. A description of this database is shown in Figure 5.

index numbers	x Coordinate	y Coordinate	Six first neighbors	z Coordinate	Cell type	$s_t(z)$	$u_t(z)$	$v_t(z)$		
40602	posx00	posy00	cellvec00	posz03	tcell03	scell03	utp03	vtp03	posz04	tcell04
20293	48.25	43.73	[20291, 20292, 20294, 20295, 20694, 20696]	2.61	N	0.50	0.50	0.00	3.92	N
20294	48.50	43.30	[20292, 20293, 20295, 20296, 19891, 19893]	2.61	N	0.50	0.50	0.00	3.92	N
20295	48.75	43.73	[20293, 20294, 20296, 20297, 20696, 20698]	2.61	N	0.50	0.50	0.00	3.92	N
20296	49.00	43.30	[20294, 20295, 20297, 20298, 19893, 19895]	2.61	N	0.50	0.50	0.00	3.92	N
20297	49.25	43.73	[20295, 20296, 20298, 20299, 20698, 20700]	2.61	N	0.50	0.50	0.00	3.92	N
20298	49.50	43.30	[20296, 20297, 20299, 20300, 19895, 19897]	2.61	N	0.50	0.50	0.00	3.92	N
20299	49.75	43.73	[20297, 20298, 20300, 20301, 20700, 20702]	2.61	B	0.50	0.00	0.50	3.92	N
20300	50.00	43.30	[20298, 20299, 20301, 20302, 19897, 19899]	2.61	B	0.50	0.00	0.50	3.92	N
20301	50.25	43.73	[20299, 20300, 20302, 20303, 20702, 20704]	2.61	F	1.00	0.00	1.00	3.92	B
20302	50.50	43.30	[20300, 20301, 20303, 20304, 19899, 19901]	2.61	B	0.50	0.00	0.50	3.92	N
20303	50.75	43.73	[20301, 20302, 20304, 20305, 20704, 20706]	2.61	B	0.50	0.00	0.50	3.92	N
20304	51.00	43.30	[20302, 20303, 20305, 20306, 19901, 19903]	2.61	N	0.50	0.50	0.00	3.92	N
20305	51.25	43.73	[20303, 20304, 20306, 20307, 20706, 20708]	2.61	N	0.50	0.50	0.00	3.92	N
20306	51.50	43.30	[20304, 20305, 20307, 20308, 19903, 19905]	2.61	N	0.50	0.50	0.00	3.92	N
20307	51.75	43.73	[20305, 20306, 20308, 20309, 20708, 20710]	2.61	N	0.50	0.50	0.00	3.92	N
20308	52.00	43.30	[20306, 20307, 20309, 20310, 19905, 19907]	2.61	N	0.50	0.50	0.00	3.92	N

Figure 5. Characteristic description of the database used for the implementation of the snowflake. In the *cellvec*00 column, some sets formed with the first neighbor indexes are shown. The dataframe continues with the columns "pos-n" (height at which the layer is located), "tcell-n", "scell-n", "utp-n" and "vtp-n", which are added for each layer with the values obtained and actualized at every stage by the application of the model.

Each column of the database has the same number of elements. Next, by adding another layer to the structure, the following columns are added: a column for height "pos-n", another column for the cell type "tcell-n", another to measure the total freezing state $s_t(z)$ "scell-n" of a receptive (B or F) or non-receptive (N) cell, one more $u_t(z)$ column for "utp-n", and a final one for $v_t(z)$ "vtp-n".

An important element is the central tile, since it can be used to locate an initial frozen core or as a reference for the location of these cores; so, to find this central tile, the number of rows (m) and columns (n) given by (Equation (2)) is taken, where n is the element used to define where the central cell is. Thus, if n is even, the location is calculated with the following:

$$\text{Central cell of the layer} = (m+1) * (n+1) \tag{14}$$

While if the number of columns is odd, we have the following:

$$\text{Central cell of the layer} = (m+1) * n \tag{15}$$

With all these concepts already defined, we have an initial dataframe that contains the information of the complete snowflake, in this case, with a spatially defined ellipsoidal shape. This dataframe is a dynamic object in which the states of every tile that is part of the snowflake are modified at each stage; nevertheless, in terms of computing efficiency, it is not "profitable" to interact directly with this dataframe. Thus, one strategy that was used to calculate the state changes in a layer is to download the information from a surface to one-dimensional arrays, operate with them, and at the end of the calculations with that layer, return them already modified to the dataframe, after which one can download the data from the next layer and recalculate. Thus, for all the layers in a stage, this process is repeated for all the surfaces for each stage.

5. Assigning Values to Parameters

An ellipsoidal frame containing the snowflake was built with seven horizontal planes that cut the spheroid surface and each plane with 40,602 cells (201 rows and 202 columns)

and the parameters α, β, and γ were varied. The snowflake is a structure that develops in stages. Each stage is a space of time that begins with a change in the lowest layer (layer 1) that is not an edge (the first edge is layer 0) and ends with the penultimate upper layer, since the last one is again an edge, this marks the end of a stage and continues with the next one, starting again with layer 1. In general, the number of stages that were handled for each snowflake was around 1000 stages, directly depending on the values of α, β, and γ. If too many stages are performed, the structure becomes saturated and the object obtained takes the shape of the edge layers. In the cases shown, this object is an elliptical spatial surface, but it could be a sphere or a cylinder or the shape established for these edges and boundary layers.

With the same considerations of the Reiter model [1] regarding the states of the tiles (F, B, N, and E), though modified for its application in space, while taking into account a membrane that limits the size of the planes in space and considering several nucleation centers (shown in the previous figures), it is observed that the shapes that the structure acquires and the number of stages it takes before saturating depend on the parameters α, β, and γ, α and γ being those that establish the thermodynamics inside the snowflake through the Laplacian (expressed as a graph). The parameter β is the one that has the temperature and pressure information from outside the snowflake and that carries it through the tiles at the edges and the two boundary layers.

6. Conclusions

In this work, a modification was made to the Reiter model in which eight first neighbors are considered, and a procedure was added to consider a spatial surface as a membrane that contains the horizontal planes, which group the hexagonal tiles. These cells, which are the cellular automata, are made up of water molecules in three states, solid (F), liquid (B), and gaseous (N); the starting automata (F) are the nucleators (which can be one or several), which, when interacting with the surrounding medium, have an effect on a finite space whose limit generates what we call the membrane and all the tiles that are outside this limit are automata in a gaseous state (E). However, due to their distance from the nucleator, they do not participate in the change of state, which does occur between cells B and N through the thermodynamic parameters α and γ. Nevertheless, these E cells are in charge of transmitting the effect of the surrounding medium by means of parameter β, which remains fixed throughout all calculation process, though β could be modified, reflecting a change in the medium external to the snowflake.

Thus, the thermodynamics of the process are carried out considering a tile and its first eight neighbors, six forming hexagons and located in the plane, plus two others, one in the immediately superior plane and another in the adjacent inferior plane. This entire process, which involves the application of the model with Equations (8)–(11), is carried out using a continuous numbering of the indexes corresponding to each tile. However, there are other ways of indexing, such as first considering the even cells and then the odd cells or using matrix arrangements. These ways of treating the indexes affect the ease or difficulty of finding the first and second neighbors. With the indexing mode used, these first neighbors were treated as a subset of six indexes associated with each tile, complementing the calculations with the tiles immediately above and below the position of the particular cell with which we are operating.

The results shown in Figures 6–15 are based on the design of a data structure called the dataframe, which is dynamically modified layer by layer, which implies that the layer information is downloaded to arrays; through the indexes of these arrays, which are the same as those of the dataframe, each tile is individually operated on by applying the model.

This strategy in the interaction with the dataframe makes the entire operation more efficient, since it goes from hours of calculation to obtain a snowflake to minutes.

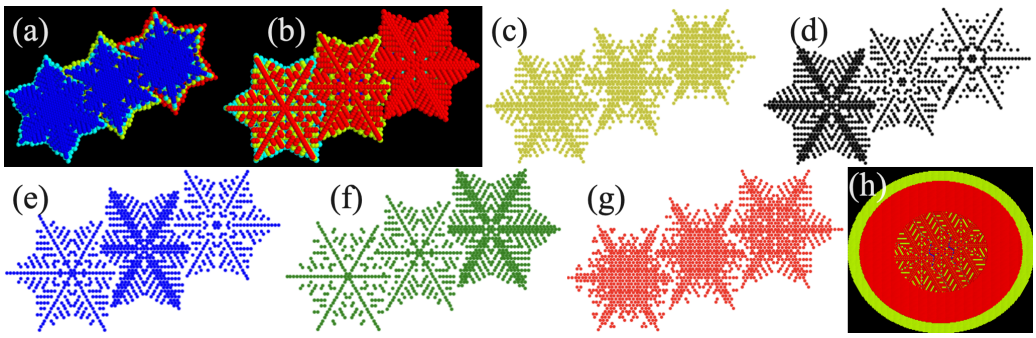


Figure 6. Snowflake with 40 stages: $\alpha = 0.2$, $\beta = 0.9$, $\gamma = 0.0$, with a view of layers 1 and 4 together (marked (**a**,**b**)), where the nucleation centres are placed with one in the middle layer (layer 3), another in the upper layer (layer 4) that is 1650 tiles away from the centre to the right, and a third nucleation centre in the lower layer (layer 2), also 1650 tiles away from the centre to the left. The individual layers (**c**–**g**) are also shown. After 300 stages, all layers were already saturated and took the shape of a container; in the overall view, layer 4 is shown in red (**h**).

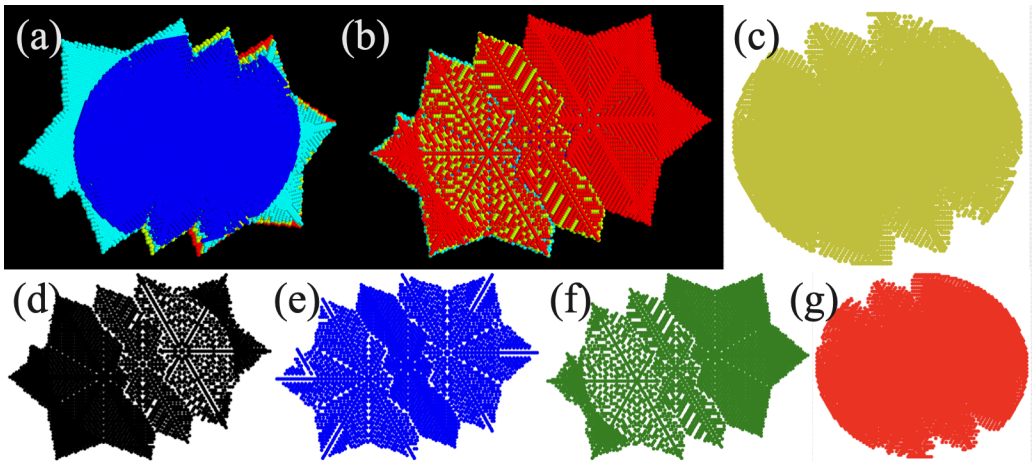


Figure 7. Snowflake with 100 stages: $\alpha = 0.2$, $\beta = 0.9$, $\gamma = 0.0$, with a view of layers 1 and 4 together (marked (**a**,**b**)), where the nucleation centers were placed leaving one in the central layer (layer 3), another in the upper layer (layer 4) that is 1650 tiles away from the center, and a third nucleation center in the lower layer (layer 2), also 1650 tiles away from the center. The individual layers (**c**–**g**) are also shown.

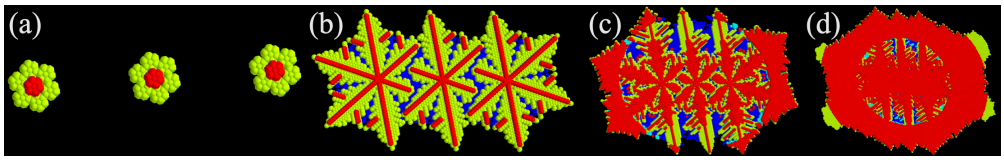


Figure 8. Snowflake with 100 stages (**a**) with parameters $\alpha = 0.5$, $\beta = 0.3$, $\gamma = 0.001$. The nucleation centers were all placed in the central layer (layer 3), one in the center of that layer and the other two 1650 tiles away from the center—one to the right and the other to the left. After 500 stages (**b**), the snowflake already shows the beginnings of saturation at 1000 stages (**c**) and after 1500 stages (**d**), all layers are already saturated and have taken the shape of a container.

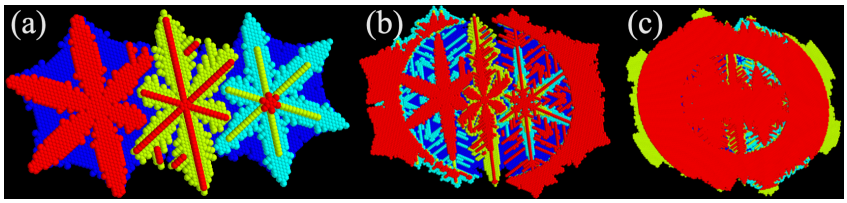


Figure 9. Snowflake with 500 stages (**a**) with parameters $\alpha = 0.5$, $\beta = 0.3$, $\gamma = 0.001$, then 1000 (**b**) and finally, 1500 (**c**) stages. The nucleation centres were placed one in the central layer (layer 3), another in the upper layer (layer 4) but 1650 tiles away from the centre, and finally, a third nucleation centre in the lower layer (layer 2), also 1650 tiles away from the centre.

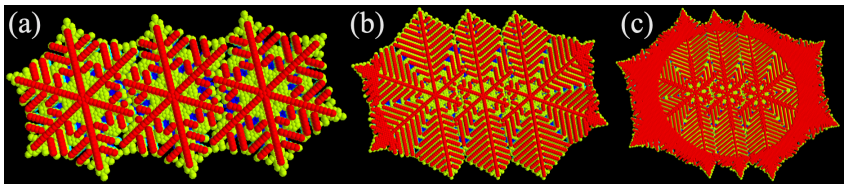


Figure 10. Snowflake with 100 stages (**a**) with parameters $\alpha = 0.5$, $\beta = 0.4$, $\gamma = 0.001$, then 200 (**b**) and, finally, 300 stages (**c**). The nucleation centers were all placed in the central layer (layer 3)—one in the center of that layer and the other two 1650 tiles away from the center, one to the right and the other to the left.

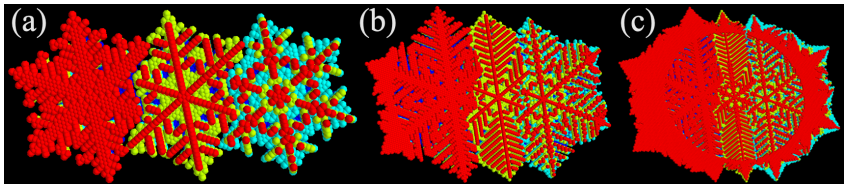


Figure 11. Snowflake with 100 stages (**a**) with parameters $\alpha = 0.5$, $\beta = 0.6$, $\gamma = 0.001$, then 200 (**b**) and 300 stages (**c**)—with this number of stages, it was already saturated. The nucleation centers were placed one in the central layer (layer 3), another in the upper layer (layer 4) that was 1650 tiles away from the center, and a third nucleation center in the lower layer (layer 2), also 1650 tiles away from the center.

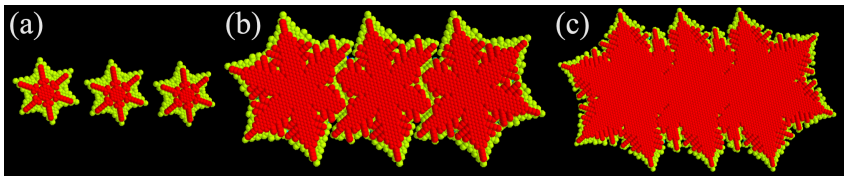


Figure 12. Snowflake with 100 stages (**a**) with parameters $\alpha = 0.35$, $\beta = 0.4$, $\gamma = 0.01$, then 200 (**b**) and 300 stages (**c**). The nucleation centers were all placed in the central layer (layer 3), one in the center of that layer and the other two 1650 tiles away from the center—one to the right and the other to the left.

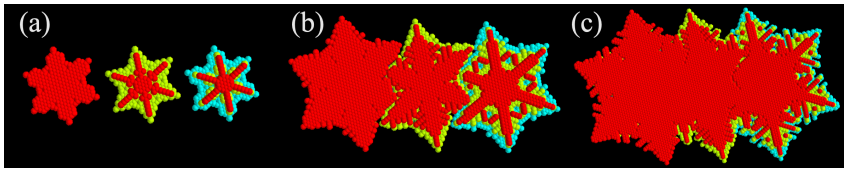


Figure 13. Snowflake with 100 stages (**a**) and parameters values $\alpha = 0.5$, $\beta = 0.4$, $\gamma = 0.01$, then 200 (**b**) and 300 srafes (**c**). The nucleation centers were placed one in the central layer (layer 3), another in the upper layer (layer 4) 1650 tiles away from the center, and a third nucleation center in the lower layer (layer 2), also 1650 tiles away from the center.

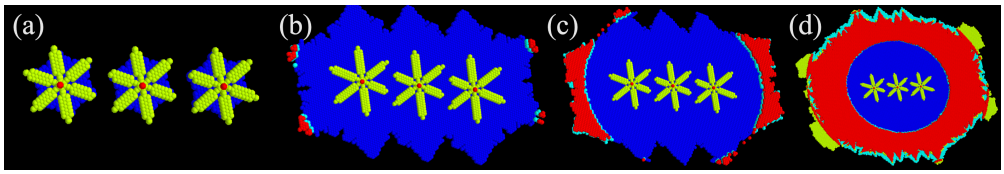


Figure 14. Snowflake with 500 stages (**a**) with parameters $\alpha = 0.35$, $\beta = 0.2$, $\gamma = 0.0$, then 1500 (**b**), 2500 (**c**), and 400 stages (**d**). The nucleation centers were all placed in the central layer (layer 3): one in the center of that layer and the other two 1650 tiles away from the center—one to the right and the other to the left.

The richness of the shape of snowflakes is completely deterministic, since for a set of values α, β, and γ, the same structure is obtained for all the layers of that snowflake, with any change in the parameters leading to different figures, which implies a high dependence on the initial conditions—in other words, displaying the typical behavior of chaos.

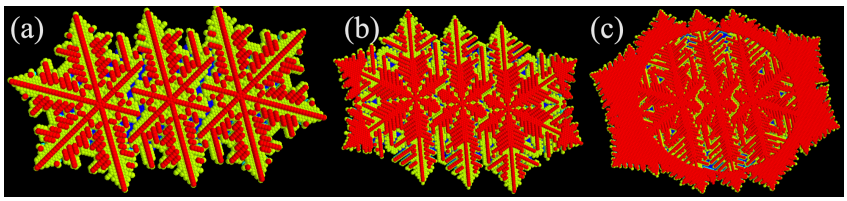


Figure 15. Snowflake with 500 stages (**a**) with parameters $\alpha = 0.35$, $\beta = 0.4$, $\gamma = 0.001$, then 700 (**b**) and 1000 stages (**c**). The nucleation centers were all placed in the central layer (layer 3): one in the center of that layer and the other two 1650 tiles away from the center—one to the right and the other to the left.

The colors used to represent each layer of the snowflake are only to emphasize the difference between the layers. Since planes one and five are in contact with the medium, they saturate more quickly than the inner planes. In general, a perspective was taken from the positive part of the z-axis downwards, showing the surfaces from one to four by color.

Author Contributions: Conceptualization, methodology, and investigation, C.R.A.; validation, visualization, I.M.; formal analysis, G.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article; further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Reiter, C.A. A local cellular model for snow crystal growth. *Chaos Solitons Fractals* **2005**, *23*, 1111–1119. [CrossRef]
2. Li, J.; Schaposnik, L.P. Interface control and snow crystal growth. *Phys. Rev. E* **2016**, *93*, 023302 . [CrossRef] [PubMed]
3. Coxe A.M.; Reiter C.A. Fuzzy hexagonal automata and snowflakes. *Comput. Graph.* **2003**, *27*, 447–454. [CrossRef]
4. Lazzari, P.; Seriani, N. Two-dimensional cellular automata—Deterministic models of growth. *Chaos Solitons Fractals* **2024**, *185*, 114997. [CrossRef]
5. Kondo, S.; Watanabe, M.; Miyazawa, S. Studies of Turing pattern formation in zebrafish skin. *Philos. Trans. A* **2021**, *379*, 20200274. [CrossRef] [PubMed]
6. Libbrecht, K.G. The physics of snow crystals. *Rep. Prog. Phys.* **2005**, *68*, 855–895. [CrossRef]
7. Gravner, J.; Griffeath, D. Modeling snow-crystal growth: A three-dimensional mesoscopic approach. *Phys. Rev. E* **2009**, *79*, 011601. [CrossRef] [PubMed]
8. Shultz, M.J.; Bisson, P.J.; Brumberg, A. Best Face Fordward: Crystal-Face Competition at the Ice-Water Interface. *J. Phy. Chem. B* **2014**, *118*, 7972–7980. [CrossRef] [PubMed]
9. Zakrzewski, W.J. Laplacians on Lattices. *J. Nonlinear Math. Phys.* **2005**, *12*, 530–538. [CrossRef]

10. Kumar, A. Isotropic finite-differences. *J. Comput. Phys.* **2004**, *201*, 109–118. [CrossRef]
11. Moghari, S.; Ghorani, M. A symbiosis between cellular automata and dynamic weighted multigraph with application on virus spread modelling. *Chaos Solitons Fractals* **2022**, *155*, 111660. [CrossRef]
12. Wang, F.; Zhang, Y.; Jin, D.; Jiang, Z.; Liu, Y.; Knoll, A.; Jiang, H.; Ying, Y.; Zhou, M. Magnetic Soft Microrobot Design for Cell Grasping and Transportation. *Cyborg Bionic Syst.* **2024**, *5*, 0109. [CrossRef] [PubMed]
13. Song, Y.; Huang, G.; Yin, J.; Wang, D. Three-dimensional reconstruction of bubble geometry from single-perspective images based on ray tracing algorithm. *Meas. Sci. Technol.* **2024**, *36*, 016010. [CrossRef]
14. Zhu, J.; Pan, S.; Chai, H.; Zhao, P.; Feng, Y.; Cheng, Z.; Zhang, S.; Wang, W. Microfluidic Impedance Cytometry Enabled One-Step Sample Preparation for Efficient Single-Cell Mass Spectrometry. *Small* **2024**, *20*, 2310700. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.