3D Modelling Seashells[*]

Francesco De Comité
University of Lille, Sciences and Technology
France
francesco.de-comite@univ-lille1.fr

Abstract

We detail how 3D printing techniques can be used to design and create concrete models of seashells, gathering the works of 19th century biologists on seashell shapes and more recent results of mathematicians and computer scientists about shell decoration patterns. Using only open-source softwares and standard computers, we illustrate the fact that the exploration of seashells variety can be conducted by anyone.

Introduction

Apprentice painters used to learn their art by copying the masters again and again. Programmers and mathematicians like the challenge of imitating existing things. By proving they can produce realistic images of natural objects, they show they have captured part of the logic behind their construction. Moreover, if the way they use to obtain this copy is simple, and can be described with few words (or program lines), there is a chance that this was the proper approach to address the problem. According to the Occam's Razor principle, short explanations have more chance to be the right ones.

Trying to understand Nature has been the goal of a lot of scientists (and not only programmers) through the ages. This comprehension can use either the elaboration of physical laws, the resolution of mathematical equations, the description of chemical reactions or the writing of computer programs. The history of the study of seashells is a good example mixing all these fields of science. Studies of seashell shapes date back to the 18th century, and regain interest with the advent of computers, which made it possible to visualize virtual models. If we consider that the history of seashell shape simulation is closely connected to the development of the graphic capacities of computers, the advent of 3D printers should have encouraged programmers to use these new tools, in order to produce copies of existing shells. However, this does not seem to be the case. The only example I found at this time (May 2017) is David Bachman's artwork *Shell-f Like*, presented at 2015 Bridges Art Exhibit. If I compare Bachman's [3] work to my method, I can notice several differences.

[*]This work was supported by European funds through the program FEDER SCV-IrDIVE and by French National Research Agency (ANR-11-EQPX-0023).

Bachman used post-processing to make his shell more realistic by modelling the aperture, and mapped the photograph of a real seashell pattern for colouring the model. In my method, I don't use post-processing, and the pigmentation pattern is computed, as explained in section *Decorating the Shells*. Bachman also used a professional program (Rhino/Grasshopper). But part of the challenge was to use only open-source tools (Blender). I used Rhino/Grasshopper only in the early stages of the project, in order to verify its feasibility.

The other interesting aspect of seashell generation is the simulation of pigmentation patterns. Their regularity, tinted with a little touch of randomness, is very appealing for mathematicians and computer scientists. Can we define simply, by means of as few equations and parameters as possible, the huge diversity of these patterns? That is to say, can we understand, or at least mimic, the laws that rule the disposition of colours on the external surface of the shell?

The rest of the paper is organized as follows: The next section will describe the method used to generate shell shapes. The following section will deal with pigmentation pattern generation. Next section is about obsolescence in the field of computer science, and the difficulty of preserving knowledge in this field, compared to other scientific domains. I will compare my work to previous ones. I will show the points were I was able to go farther, and also the limits I was not able to go over, together with new problems arising. Pictures of successful 3D prints will demonstrate the validity of the method.

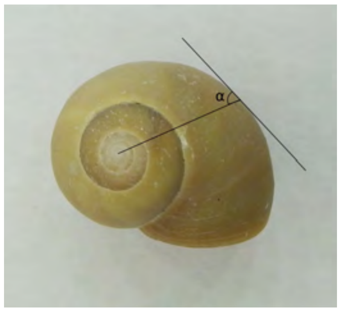
Modelling Seashells

The task of modelling seashells can be broken down into two distinct parts: describing the shell shape, and computing the decoration pattern. At first, one can program these parts separately, and finally assemble them in the same source code.

Seashell Shapes

Mathematically defining seashell shape was done in several steps dating back to the 19th century. First, in 1838, Moseley[7] collected specimens on which he made measurements, deducting the role of the exponential function in the description of seashells. Then, in 1917, D'Arcy Thompson [11] gave a list of three parameters (to say nothing of the generating curve) which can describe almost any seashell shape:

- α, the constant angle of the equiangular spiral.
- β, the angle between the spiral axis and the tangent to the whorls.
- γ, the *angle of retardation*, measuring the overlap between whorls. This parameter is rather difficult to evaluate, and seems to be inversely proportional to the increase rate of the curve.



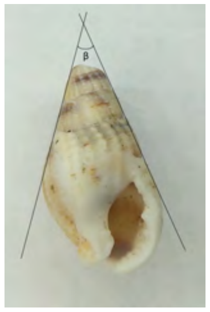


Figure 1: Two parameters out of three used by D'Arcy Thompson.

In 1962, D. Raup [9] proposed a mathematical model defining the seashell as the envelope of a generating curve turning around an axis and translating in the direction of this axis, while becoming progressively larger. This definition was using four parameters :

- The shape of the generative (closed) curve, hereafter called *the aperture*.
- The position of the curve relatively to the axis.
- The rate of increase of the curve.
- The curve's rate of translation along the axis.

Tuning these parameters, Raup was able to describe almost all kind of existing seashells. I am using his definition for designing seashells. More formally, let r be the distance from the center of the aperture to the vertical axis, z the height of the center, and θ the angle of rotation along the spiral. We can express the position of the center as a function of θ.

$$r(\theta) = r_0\, k_1^{\frac{\theta}{2\pi}} \tag{1}$$
$$z(\theta) = z_0\, k_2^{\frac{\theta}{2\pi}} \tag{2}$$

Here r_0 and z_0 are the initial coordinates of the aperture center, while k_1 and k_2 represent the growth factor of r and z, respectively. Hence, in this model, r (z) is multiplied by k_1 (k_2) each time θ makes a complete turn. Typically, $k_1 = k_2$. This is not enough to define the seashell shape: the aperture rotates from an angle θ around its vertical axis, and it has also to be scaled: the higher θ, the higher the scale. In the simple case where the aperture is a circle, and $k_1 = k_2 = k$, the aperture radius satisfies the equation:

$$rad(\theta) = \sqrt{r_0^2 + z_0^2}\,\frac{k-1}{k+1}\,k^{\frac{\theta}{2\pi}} \tag{3}$$

Equation 3 ensures that two images of the aperture for θ and $\theta + 2\pi$ will be tangent circles (It is a direct application of the Pythagorean theorem).

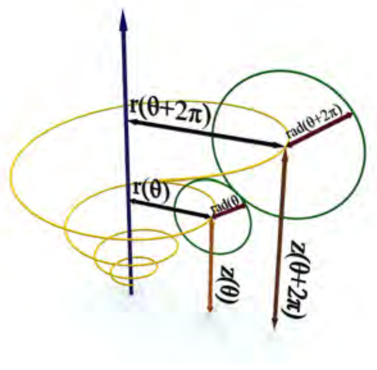


Figure 2: The parameters describing a seashell.

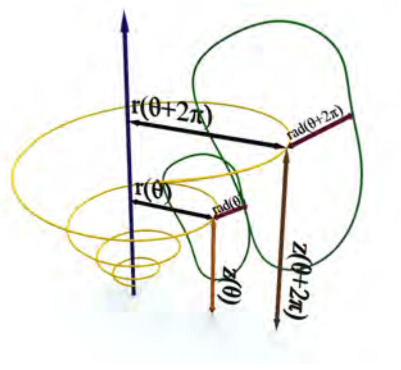


Figure 3: The non-circular aperture case.

When the aperture is not a circle, things become more difficult, and one has to tune the radius growth factor by trial and error. It would be worth considering other heuristics, or interactive ways of defining the aperture and its radius growth rate: this is one of the direction of planned future works.

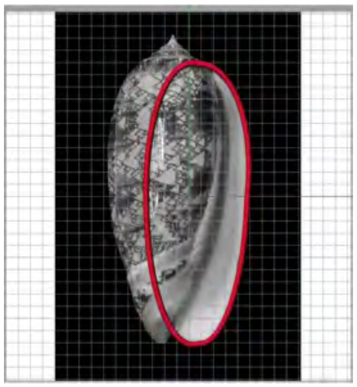


Figure 4: Aperture for Olivia Porphyria.

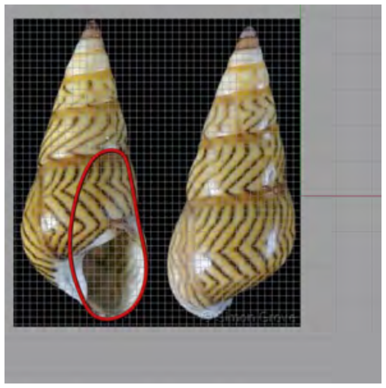


Figure 5: Aperture for Bankivia Fasciata.

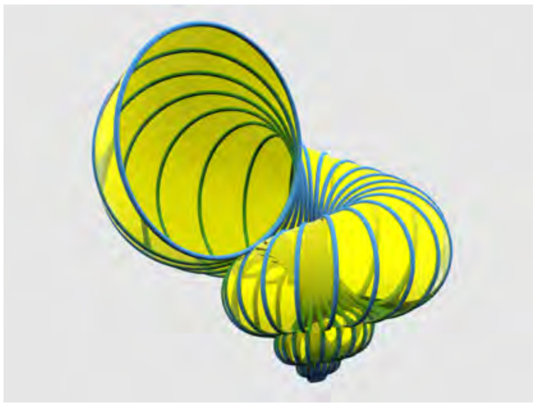


Figure 6: A surface enveloping a set of armatures.

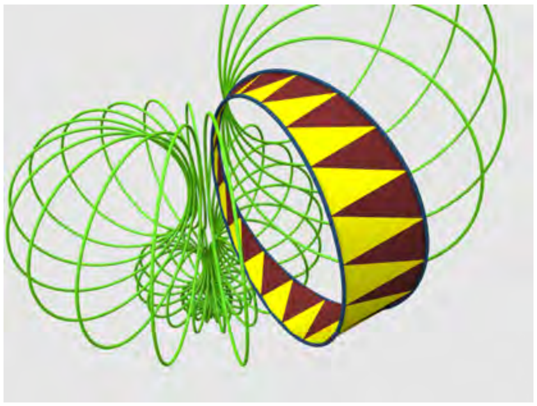


Figure 7: Triangles between armatures.

On the other hand, 3D printing software are tolerant, and they accept a certain amount of overlap in the generated surface. Experiments with a circular aperture reveal that we can represent many (but not all) of the seashell shapes existing in nature. If we want to capture more shell forms, we have to allow for a more generic aperture shape. Figure 2 illustrates the role of each parameter. Figure 3 shows that parameters r and z are still valid in the case where the aperture is not a circle.

The scale of the aperture must be computed in a way similar to equation 3. In order to work with a realistic non-circular seashell aperture, we can upload the image of a real seashell in the modelling software, and create a curve following the shape of the aperture, as illustrated in Figure 4 and Figure 5. Once we have computed a set of scaled, rotated and moved images of the aperture (we call them *armatures*), we need to interpolate a surface between them: see Figure 6. Figures 10 to 12 show some examples of rendered external seashell surfaces. These images illustrate the power of the algorithm. Figure 13 depicts a shell form created by the algorithm that doesn't exist in nature (at least, not to my knowledge). At this point of the study, we could achieve D. Raup's wish of drawing a map *of the total spectrum of possible variation in snail form*. It is tempting to halt here and produce a lot of different and beautiful shapes. But we can go farther: giving some thickness to these surfaces, in order to finally obtain 3D printed models, and thereafter put some decoration on the shells.

3D printers are only able to print volumes. We need to add thickness to the surfaces created at this point. For that, we compute a smaller surface, similar to the external shell shape. This surface will be nested into the external one. If we follow strictly the morphogenesis of real seashell, this internal wall should begin at the

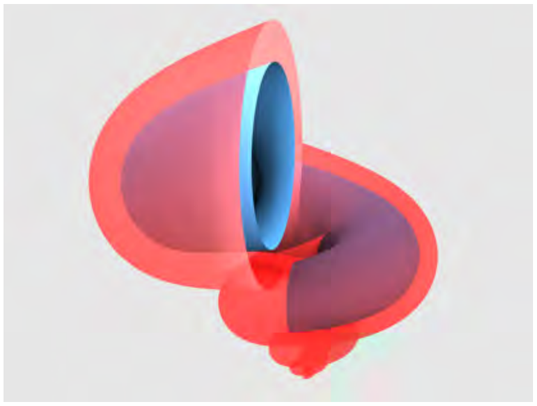


Figure 8: Internal and external walls.

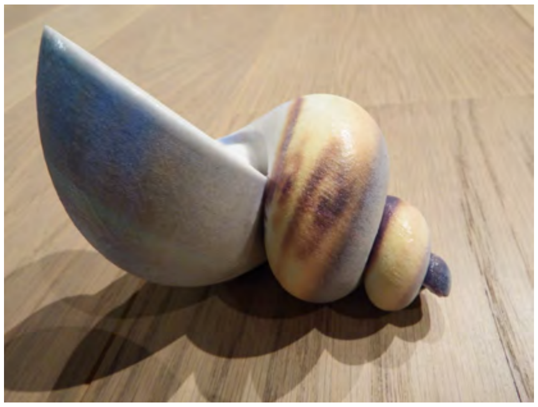


Figure 9: Mapping the Gioconda on a seashell.

same angle θ as the initial surface. This would lead to very thin walls at the origin, and induce weaknesses in the printed model. Experiments show that a short internal wall is sufficient to create the illusion of a shallow seashell: one or two turns are enough. A constraint induced by the 3D printing process is that any part of the object must have a minimal thickness (3mm for the Zcorp Project 460 used for this study): we must verify this minimal thickness while defining the depth and scale of the shell hole. Figure 8 shows the two walls in place. 3D printing compatibility constraints ask for the mesh to be *watertight* in order to be printed: it must be a real solid, with an interior, an exterior and some thickness. At this stage of the construction process, what we have is two tubes, one inside the other, with a total of four open ends. We need to perform three more operations: close the little end of exterior and interior surface and connect together the edges of the large ends of both surfaces.

Decorating the Shells

Objects defined so far are already 3D printable, and the procedure used is very similar to the one David Bachman defined. The difference is that I kept the memory of each elementary triangle of the shell, and I can now define the colour of any point of the shell (relative to the scale of the elementary triangle I defined). Any colouring strategy is applicable there; I experimented by mapping images pixel by pixel, but the procedure gives poor results (see figure 9); what can be the meaning of mapping rectangular images on helical surfaces? Nevertheless the door is kept open for further experiments. While many existing models capture the spiraling shape of seashells very well, my goal was to write a program that would produce realistic models of the surface patterns as well.

For most seashells, patterns are generated once for all as the shell aperture continues its development[1]. The basic hypothesis is that the colour of a point on this aperture depends only on the corresponding point generated at the previous stage, and its neighbours. The first approaches were made in 1969 by Waddington and Cowe [13], and then Wolfram [14] in 1984, using cellular automata. This purely computer-oriented vision of pigmentation pattern growth is not biologically grounded, but is worth exploring. Another approach, not far from Turing's vision of morphogenesis [12] by means of instability of chemical substances, was taken by Meinhardt and Klingler [6]. It uses a model of short range chemical activation and long-range inhibition: two different chemical substances (an inhibitor and an activator) struggle for local predominance. Their local concentration is based upon the former concentration at each point at the preceding stage. Differential equations are ruling the process. Meinhardt and Klingler exhibit different models of equations, and demonstrate

[1]Cypraea patterns are generated by a two-dimensional process [10]

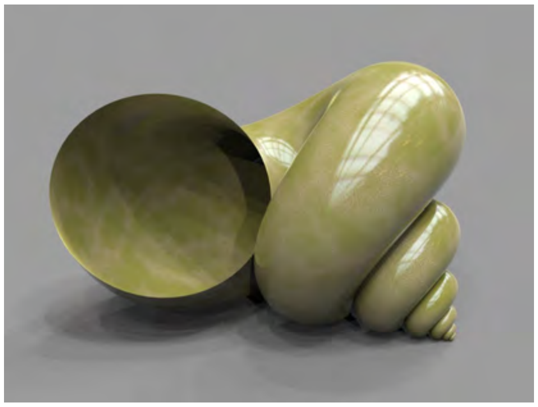


Figure 10: Massive seashell.

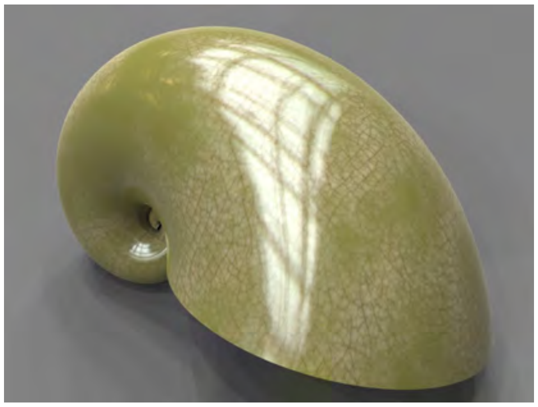


Figure 11: Nautilus-like external shape.

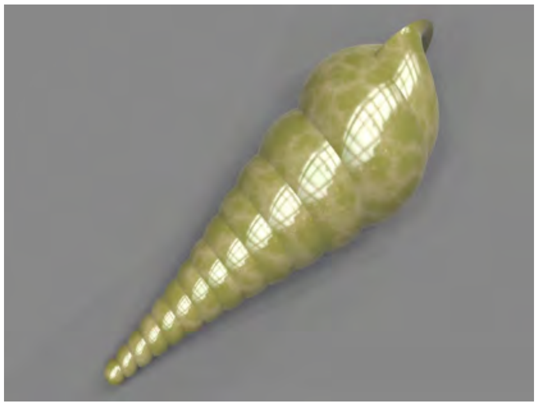


Figure 12: Elongated shell.

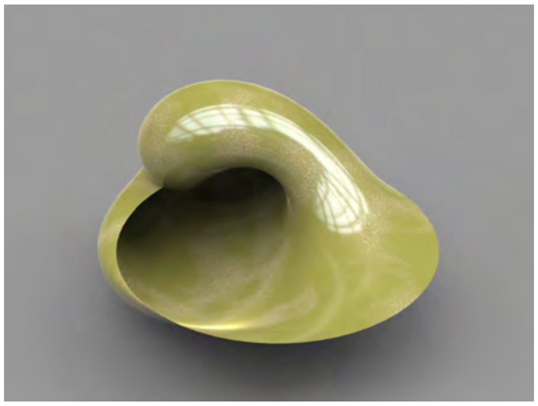


Figure 13: Utopian seashell.

that these equations generate patterns very close to natural shell pigmentation patterns. Since they published not only the equations, but also the computer programs and the values of their parameters, we should be able to reproduce the experiments, some thirty years later, with other software tools, more powerful computers, an above all new manufacturing tools: colour 3D printers.

Cellular automata generally deal with only two colours (otherwise automata become difficult to describe), and are not able to express the complexity of pigmentation. Figure 14 shows two such misses. But here also, I keep the door open for further experiments. On the other hand, the chemical approach was more fruitful. Meinhardt and al. [5] consider that seashell pigmentation results from a struggle between two substances: an activator and an inhibitor, the colour of each point of the shell depending on the respective concentration of each substance. Concentrations of activator and inhibitor at time t and position p depend upon the former concentrations at time $t-1$ around position p. This scheme can be applied in two-dimensional arrays (for zebra stripes, for example). Our case is simpler. Since the shell growth takes place at the aperture, we only have to know the concentration states along a one-dimensional array (the aperture) in order to generate the next iteration of states. The basic set of equations defining the system has the following form :

$$\frac{\partial a}{\partial t} = \rho s \left(\frac{a^2}{1+\kappa a^2} + \rho_0 \right) - \mu a + D_a \frac{\partial^2 a}{\partial x^2} \tag{4}$$

$$\frac{\partial s}{\partial t} = \sigma - \rho s \left(\frac{a^2}{1+\kappa a^2} + \rho_0 \right) - \nu s + D_s \frac{\partial^2 s}{\partial x^2} \tag{5}$$

The active edge of the aperture is identified with the x axis. An *activator*, with the concentration a, diffuses along this axis at rate D_a and decays at the rate μ. In an auto-catalytic process, activator is produced at a rate proportional to a^2 (first term of equation 4). Similarly, a *substrate*, with the concentration s, is produced at rate σ and diffuses at rate D_s. Production of activator decreases the amount of substrate. Starting from this basic equation, Fowler and al. [4] and Meinhardt and al. [5] defined a set of variations in order to be able to depict all the patterns met on seashell surfaces. Tuning the parameters allows one to mimic a lot of situations, if not all. From the previous stage of our seashell construction, we have to find the colours

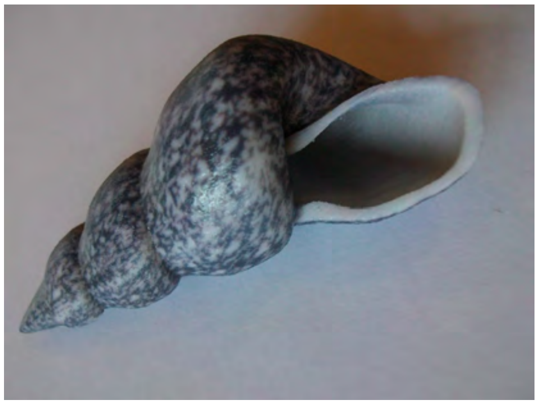
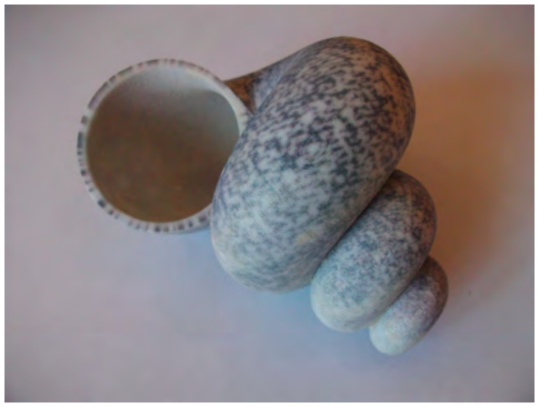



Figure 14: Two examples of synthetic seashells decorated with patterns based on cellular automata.

(i.e., concentrations of substances) associated to the set of triangles we define between each couple of armatures. Using the examples, programs and sets of parameters given in Meinhardt [5], I was able to decorate shells with different realistic patterns. Figures 16 to 18 show some examples actually 3D printed. Figure 26 shows several ready-to-print examples: they will be processed as soon as possible. At the moment the shape and the pigmentation of the shells are not synchronized, but I plan to print realistic shells reproducing accurately real world specimens. Concentrations computed in these programs were translated into different shades of colours (typically 20 to 50), which is much more than the two colours used with cellular automata. This helped to obtain better looking results, since concentrations were more or less continuous, and avoid the pitfall in which I fell with automata. Figure 19 is an interesting example of failure: trying to reproduce the pattern of *Tapes*, a wrong choice of parameters resulted in this quite regular pattern, which corresponds to no real seashell, as far as I know. This illustrates the fact that exploring the universe of pattern possibilities can be fruitful.

Archeology in the Numerical Age

Modelling seashell shapes is based upon works beginning in the 19th century [7] [11], and are expressed in mathematical terms the programmer can translate in his favorite programming language. It is only later on that computer scientists took over and describe seashell genesis directly in terms of computer code [9] [8].The situation is different as far as the modelization of pattern is concerned. If we except Turing's seminal

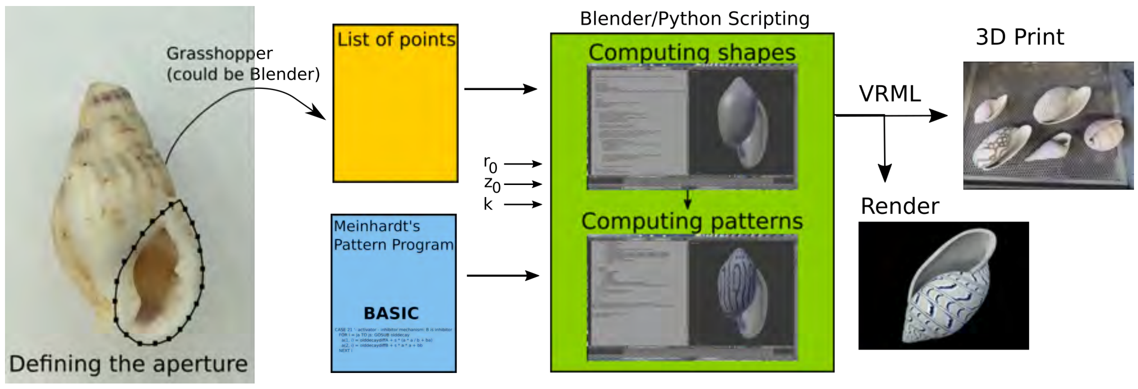


Figure 15: A survey of the complete seashell modelisation process.

work [12], Meinhardt and al. [5] [6] publications are using computer simulations to validate their chemical reactions hypotheses. Their book "The Algorithmic Beauty of Sea Shells" was contained a $3\frac{1}{2}$ floppy disk. It included programs (written in Basic) implementing the algorithms presented in the book. The reader was able to tune the parameters in order to create patterns variations. Doing so, he could have a better understanding of the influence of each parameter. Despite the fact that the book is not so old (last edition year seems to be 2009), both the media support and the programming language have become obsolete. It seemed also difficult to find a used book which still contains the diskette. After some researches, I found a copy belonging to the Pierre Bartoli library in Montpellier [1]. People there were kind enough to send me a copy of the floppy disk content (their first task was to find a working floppy disk drive, and install it on a computer). At this point, I had in hand the source code of the simulations, but I was not able to find a compatible Basic interpreter. The only solution was then to literally read the source code, and translate it. The first step was to translate it in Java, since I used ImageJ [2] to obtain quick previews of the patterns (Figures 20 and 21), and then in Python, in order to integrate the code into the Blender script generating the seashell. Figures 22 and 23 show the corresponding pattern obtained in Blender when Meinhardt's equations are coded in the Python script window. This illustrates the volatility of both programming languages (Basic) and data supports (floppy disks and data reading devices). Testing pattern generation first in Java/ImageJ and then in Python/Blender allows one to verify original source code, and to tune parameters. Quite often, in the original programs, some parameters are more or less implicit (or at least one has to dig into the code to find their meaning and/or value). Since Meinhardt associates each program with a figure in his book, we can verify whether our transcription is faithful to the original code. Moreover, we can play with the parameters and try to obtain more accurate pattern models. As the computation power of computers has dramatically increased since the time when the book was written, experiments that took a long time twenty years ago can be done quite instantly.

Improvements and Limitations

The main improvement of this work is the achievement of real 3D objects similar to existing seashells, by combining shape and decoration. Dan Bachman [3] already produced a 3D printed seashell, differing from our models in several points. On one hand, he modeled a more realistic aperture but on the other hand, the shell decoration was obtained by UV mapping, not by computation. Fowler and al. [4], after having described their method for modelling 2D representations of decorated seashells, listed five open problems, among which I believe to have solved three.

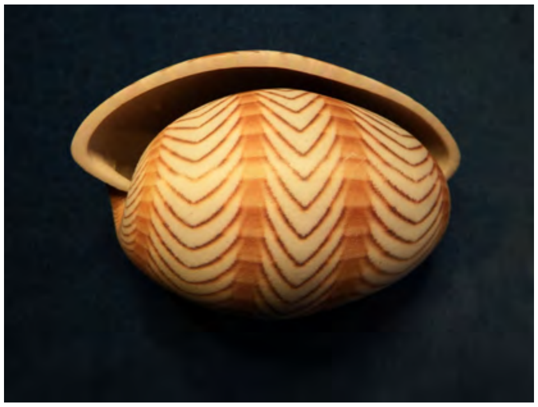


Figure 16: Cyprea Ziczac-like pattern.

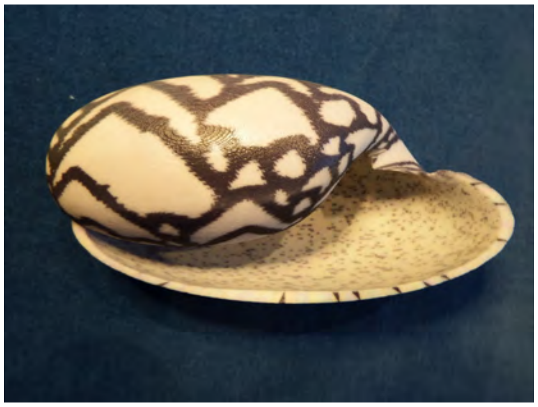


Figure 17: Bankivia Fasciata-like pattern.

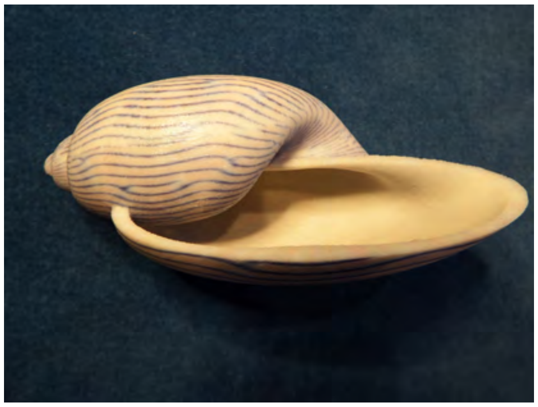


Figure 18: Amorina Undulata-like pattern.

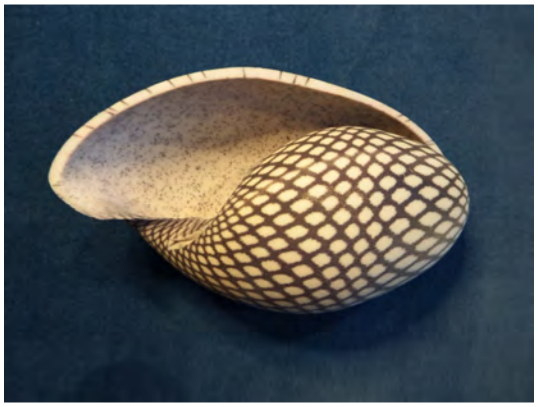


Figure 19: Serendipist pattern.

- *Proper modeling of the shell opening* : I left this problem open, since I focused on the programming aspect of the work. Making the aperture more realistic needs artistic skills.
- *Modeling of spikes* : The method used here does not allow one to take into account geometric irregularities like spikes (figure 24). Nevertheless, limited and quite *regular irregularities* could be handled at low cost (figure 25 illustrates a first experiment in this way).
- *Capturing the thickness of shell walls* : As Fowler and al. wanted to produce 2D images, they did not focus on that point. Since I wanted to make real three dimensional models, I had to solve this problem, by creating an inner thicker spiral wall (a method initiated by Dan Bachman).
- *Alternatives to the integrated model* : Fowler's paper was written in 1992. At this time, handling a mesh with millions of polygons was time and memory consuming, leading the computer to its limits. Nowadays, I can consider meshes of this size with no major problems. 3D printers (I used the Zcorp Projet 460*Plus*) can easily afford such heavy files, and can print several models at the same time.
- *Improved rendering* : Since my goal was not to produce images, but three dimensional objects, the enhancement of the rendering is no more important. Instead of that, I still have to work to make the shells more realistic : size, relative thickness of the walls, texture, weight.

On the other hand, the fact that the models are now three dimensional induces some new shortcomings:

- *Size and thickness*: Models are printed in a material called *Full Color Sandstone*, which is dense,

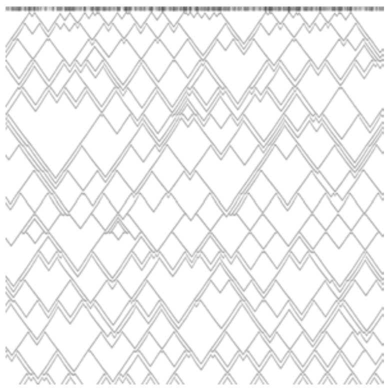


Figure 20: Olivia Porphyria pattern in ImageJ.

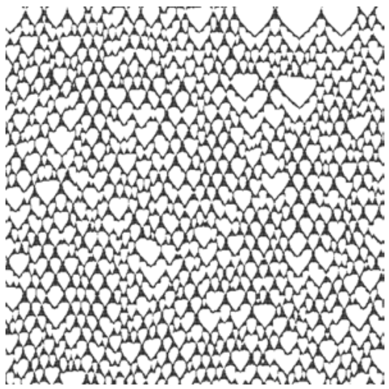


Figure 21: Conus Marmoreus pattern in ImageJ.

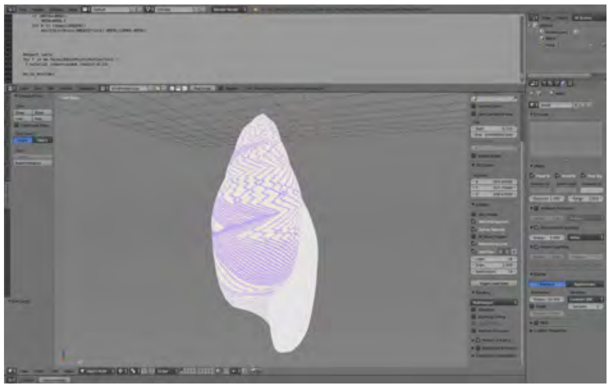


Figure 22: Olivia Porphyria pattern in Blender.

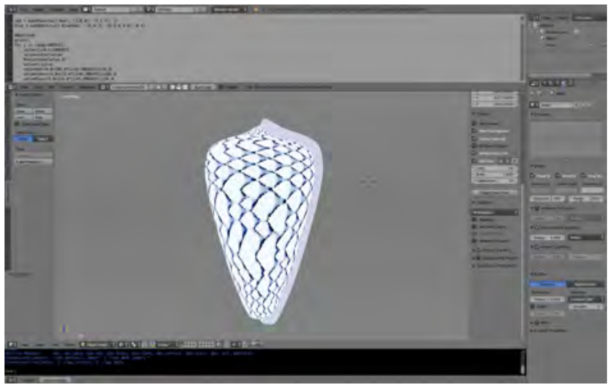


Figure 23: Conus Marmoreus pattern in Blender.

heavy and fragile. Walls must have a minimum thickness of 3mm. This can lead to large and heavy models. Several 3D printing companies tried multicolor plastic, but it seems that this material is not yet available. I am looking forward to seeing the advent of multicolor porcelain 3D printers.

- *Resolution of the aperture*: The algorithm defining the patterns works by discretising the aperture. To obtain realistic patterns, we need a large number of control points (500 is a minimum, the more the better). Multiplied by the number of *armatures*, this leads to a very large number of elementary triangles, at the limit of what standard computers can afford.

Perspectives

I began to work on that project in September 2016. Exploring the universe of possible seashells is still at its beginning, and David Raup's project of *map[ping] the total spectrum of possible variations in snail form* [9] is far to be completed. In Meinhardt's book, there is a picture (figure 10.20 page 184 in the 2009 edition) featuring *A virtual museum of shells*. Based of the present work, this could become *A real museum of virtual shells*. It could be interesting to create chimerical shells, according either to their shape or to their decoration. This could help biologists to understand or explain why we don't meet these specimens in Nature. In the context of exploring hostile environments (on earth or on other planets), one can imagine building shells on demand to shelter little autonomous robots (which could also be *autonomously* 3D printed).

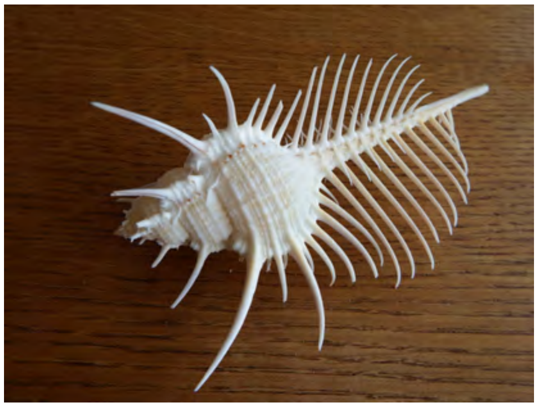


Figure 24: The *Murex* challenge.

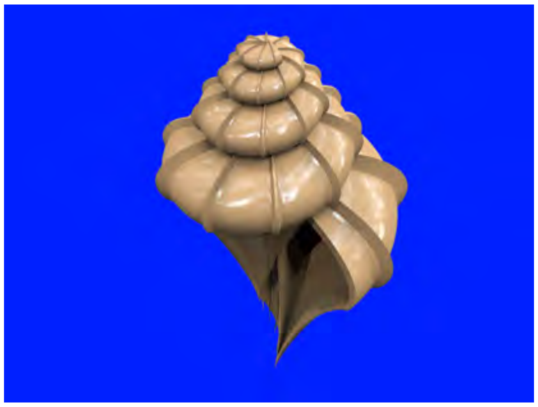


Figure 25: Irregularities on Epitonium shell.

References

[1] Centre de documentation Pierre Bartoli. `http://www1.montpellier.inra.fr/bartoli` (accessed 02/05/2017).

[2] ImageJ website. `https://imagej.nih.gov/ij/` (accessed 02/05/2017).

[3] Modelling seashells (David Bachman blog). `http://mathartblog.com/?p=257` (accessed 07/04/2017).

[4] Deborah R. Fowler, Hans Meinhardt, and Przemyslaw Prusinkiewicz. Modeling seashells. In *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1992, Chicago, IL, USA, July 27-31, 1992*, pages 379–387, 1992.

[5] H. Meinhardt, P. Prusinkiewicz, and D.R. Fowler. *The Algorithmic Beauty of Sea Shells*. The Virtual Laboratory. Springer Berlin Heidelberg, 2003.

[6] Hans Meinhardt and Martin Klingler. *Pattern formation by coupled oscillations: The pigmentation patterns on the shells of molluscs*, pages 184–198. Springer Berlin Heidelberg, Berlin, Heidelberg, 1987.

[7] H. Moseley. On the geometrical forms of turbinated and discoid shells. *Philosophical Transactions of the Royal Society of London*, 128:351–370, 1838.

[8] Clifford A. Pickover. A short recipe for seashell synthesis. *IEEE Computer Graphics and Applications*, 9(6):8–11, 1989.

[9] David M. Raup. Computer as aid in describing form in gastropod shells. *Science*, 138(3537):150–152, 1962.

[10] Enrico Savazzi. The colour patterns of cypraeid gastropods. *Lethaia*, 31(1):15–27, 1998.

[11] D'Arcy Wentworth Thompson. *On Growth and Form*. Cambridge University Press, 1945.

[12] A. M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 237(641):37–72, 1952.

[13] C.H. Waddington and Russell J. Cowe. Computer simulation of a molluscan pigmentation pattern. *Journal of Theoretical Biology*, 25(2):219 – 225, 1969.

[14] S. Wolfram. Cellular automata as models of complexity. *Nature*, 311(5985):419–424, 1985.

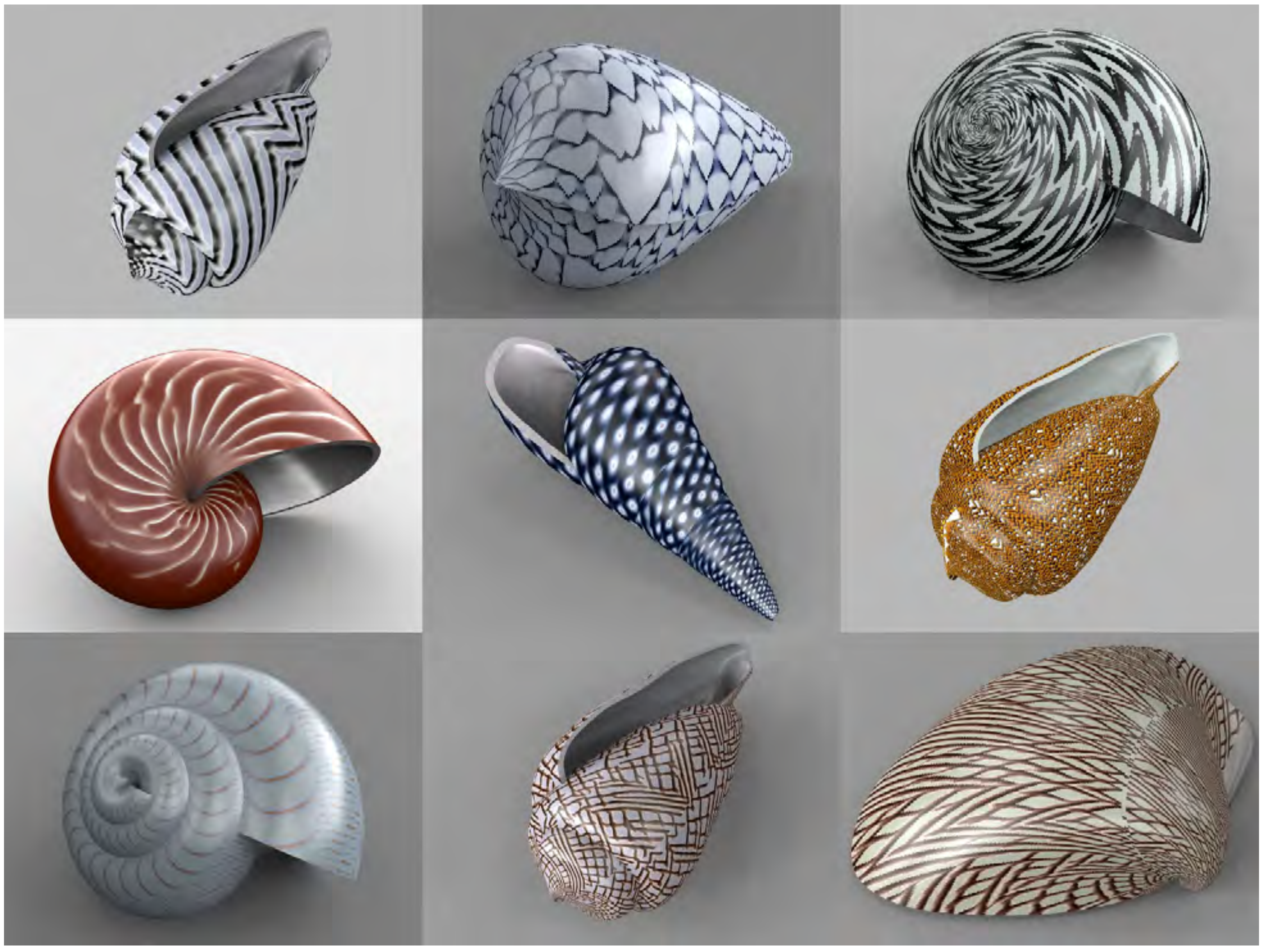


Figure 26: *A bunch of ready to print models*: Persicula Persicula, Conus Marmoreus, Neritodryas Dubia, Nautilus Pompilius, Bankivia Fasciata, Olivia Porphyria, Natica Euzona, Conus Abbas, Tapes Literatus.