Can Equations Be Exciting?

Neli Dimitrova

Institute of Mathematics and Informatics
Bulgarian Academy of Sciences

1 Introduction

Nowadays computer generated fractal patterns can be seen everywhere, from squiggly designs on computer art posters to illustrations in serious scientific journals. Interest continues to grow among scientists and, rather surprisingly, artists and designers. This paper provides visual demonstrations of complicated and beautiful structures that can arise as solutions of even simple equations.

Fractals are geometric objects which are usually the result of an iterative or recursive construction or an algorithm, i. e. they are not just static images, but created by a dynamical process. Think about the beautiful forms that we can see in nature – plants as a result of their dynamic growth; mountains as a result of past tectonic activity as well as erosion processes... Fractals cannot be described by algebraic formulae like e. g. some plane figures in the Euclidean geometry.

It is not difficult to imagine that if a system is described by complicated mathematical equations, then its solution might be complicated and unpredictable. What has come as a surprise to most scientists is that even simple systems, described by simple equations, can have "strange" solutions. The most famous example is the equation, used to model a single species time evolution and known as the *logistic map*. More about the solutions of the logistic map can be found in [1].

Here we shall consider systems of two discrete nonlinear equations, known also as recurrence (or difference) equations or as iterated maps. This kind of systems might arise, for example, from an ecological predator-prey model. Such two classical Lotka-Volterra models are presented in Section 2 and used to demonstrate the idea on bounded and stable solutions (called trajectories) of such a system. Examples of general quadratic, cubic and other nonlinear iterated maps are presented in the next Sections 3, 4 and 5. As it can be seen from the figures there, the solutions totally differ from the ones presented in Section 2; the trajectories (bounded but unstable) describe pieces in the plane that genuinely can be compared to pieces of art! If you want to learn more about the properties of such kind of solutions of iterated maps, you can read Section 6. The visualization in the paper is carried out in the computer algebra system *Maple 13*. Simple Maple commands producing some of the images are given in the Appendix.

2 Discrete predator-prey models

Consider a two-species predator-prey discrete model in which one species preys on another. Examples in the nature include sharks and fish, lynx and snow-shoe hares, ladybirds and aphids, wolves and rabbits. A very simple model, known as the Lotka-Volterra model [2], is the following:

$$\begin{aligned} x_{n+1} &= x_n(1+p_1-p_2x_n-p_3y_n) \\ y_{n+1} &= y_n(1-q_1+q_2x_n), \quad n=0,1,2,\dots \end{aligned} \tag{1}$$

Here p_1, p_2, p_3, q_1 and q_2 are nonnegative constants; x_n and y_n represent the number of prey and predator populations respectively at time n. The terms appearing in the right-hand sides of the equations, have a biological meaning as follows:

• $(1+p_1)x_n - p_2x_n^2$ represents the logistic growth of the population of prey in the absence of predator;

• $p_3x_ny_n$ and $q_2x_ny_n$ represent species interaction: the population of prey suffers and predators gain from the interaction;

• $(1-q_1)y_n$ represents the extinction of predators in the absence of preys.

There are three particular types of outcome that are often observed in the real world. In the first case, there is coexistence, in which the two species live in harmony. In nature, this is the most likely outcome. In the second case, one of the species becomes extinct, and in the third case both species go to extinction.

Having some values at the initial time $n = 0$, say (x_0, y_0), we can consecutively compute by means of (1) an infinite sequence of points in the (x, y)-plane

$$(x_1, y_1),\ (x_2, y_2), \ldots, (x_n, y_n),\ (x_{n+1},\ y_{n+1}), \ldots \tag{2}$$

This sequence describes the evolution of the populations as time increases and is called *a trajectory of* (x_0, y_0); (x_0, y_0) is called *initial point* or initial condition. Obviously, the values of the sequence members depend on the choice of (x_0, y_0) as well as on the values of the constants p_1, p_2, p_3, q_1 and q_2. The main question is: given some initial point (x_0, y_0) what can we say about the behavior of the trajectory (2) for sufficiently large n? Figure 1(a) presents three trajectories within

$$p_1 = 0.05,\ \ p_2 = 0.0001,\ \ p_3 = 0.001,\ \ q_1 = 0.03,\ \ q_2 = 0.0002 \tag{3}$$

for three different initial conditions $(x_0, y_0) = (20, 5)$, $(x_0, y_0) = (100, 10)$ and $(x_0, y_0) = (50, 40)$, denoted by solid boxes. As you can see, when n increases, the three trajectories approach one point in the plane and remain close to it. Such a point is called a *stable steady state* or *attractor*. You can find it by replacing in (1) $x_{n+1} = x_n = x$, $y_{n+1} = y_n = y$ and then solve the obtained nonlinear system for x and y. You will obtain three different solutions $(x, y) = (0, 0)$, $(x, y) = (500, 0)$ and $(x, y) = (150, 35)$. The third point $(150, 35)$ is the attractor, shown in Figure 1(a). The other two steady states are called *unstable*, because the trajectories of initial points, even slightly different from these steady states, move away from them as time n increases.

Now set $p_2 = 0$ in the model (1); in this way you change the growth rate law in the prey population. The system becomes

$$\begin{aligned} x_{n+1} &= x_n(1 + p_1 - p_3 y_n) \\ y_{n+1} &= y_n(1 - q_1 + q_2 x_n), \quad n = 0, 1, 2, \ldots \end{aligned} \tag{4}$$

Figure 1(b) presents one trajectory with the same coefficient values for p_1, p_3, q_1 and q_2 from (3) and with initial condition $(x_0, y_0) = (20, 5)$, denoted by a solid box. Now we see a totally different picture: the two populations oscillate, building a *stable cycle*. How can the system (4) be interpreted in terms of species behavior? If the ratio of predators to prey is relatively high, then the population of predators drops. When the ratio of predators to prey drops, then the population of prey increases. If there is sufficient quantity of prey, the predator number starts to increase. The resulting cyclic behavior is repeated over and over.

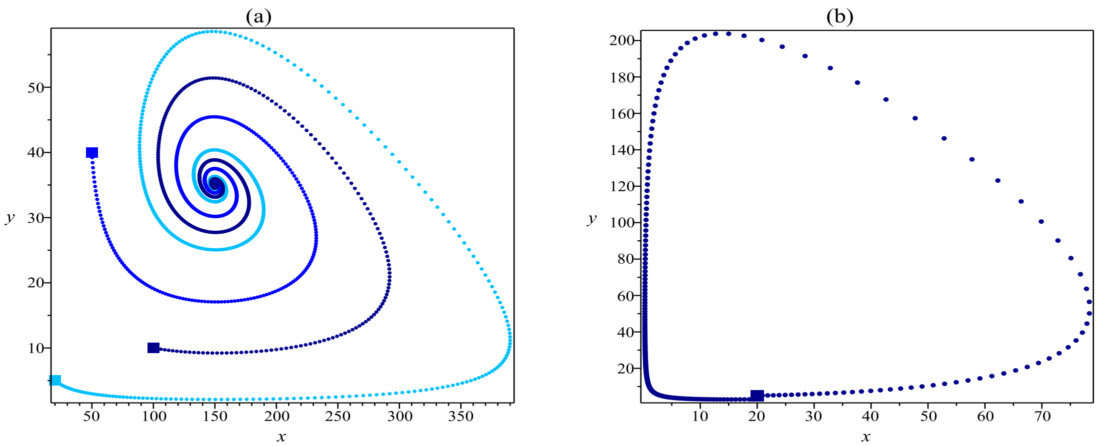

Figure 1: Trajectories: *(a)* of model (1); *(b)* of model (4)

The trajectories in the above two examples have a common feature: they all are *bounded*, because there exist quadrangles in the (x,y)-plane (see Figure 1) which enclose all points (x_n,y_n), $n=1,2,\dots$ for any initial condition (x_0,y_0) inside these quadrangles. In the two systems there is either a stable steady state, that attracts all trajectories, or the trajectory builds a stable cycle. In both cases we say that the trajectories are stable.

The equations in (1) and (4) are called *discrete iterated systems* or *iterated maps* because the next values of the x- and y-quantities are predicted by the previous values. In the next sections we shall consider more general examples of iterated maps, whose trajectories are bounded but *unstable*: such a trajectory will never move off to infinity, but will also never settle down to a point or a cycle. Initial conditions are drown to a special type of attractors, called a *strange* or *chaotic* attractor, which is not a point or even a finite set of points but rather a complicated geometrical object, called *fractal*.

3 Iterated quadratic maps

The iterated maps (1) and (4) have terms of the form x_n^2 and $x_n y_n$ as their highest order term, thus they are maps of order 2 or quadratic maps. The general form of a quadratic map is

$$\begin{aligned} x_{n+1} &= a_1+a_2x_n+a_3x_n^2+a_4x_ny_n+a_5y_n+a_6y_n^2 \\ y_{n+1} &= b_1+b_2x_n+b_3x_n^2+b_4x_ny_n+b_5y_n+b_6y_n^2, \quad n=0,1,2,\dots \end{aligned} \tag{5}$$

For example, we have in the map (1)

$a_1=0,\ a_2=1+p_1,\ a_3=-p_2,\ a_4=-p_3,\ a_5=a_6=0;\ \ b_1=b_2=b_3=0,\ b_4=q_2,\ b_5=1-q_1,\ b_6=0.$

Denote by $a=(a_1,a_2,a_3,a_4,a_5,a_6)$ and by $b=(b_1,b_2,b_3,b_4,b_5,b_6)$ the vectors of the coefficients in the quadratic iteration scheme (5). Table 1 contains six numerical examples for a and b [4]. The initial point in all examples is chosen to be $(x_0,y_0)=(0,0)$.

Table 1: Examples of quadratic maps with chaotic attractors

Example	Coefficient vectors a and b	Figure
1	$a=(-1.2,\ -0.6,\ -0.5,\ 0.1,\ -0.7,\ 0.2)$ $b=(-0.9,\ 0.9,\ 0.1,\ -0.3,\ -1,\ 0.3)$	2 (left)
2	$a=(-1.1,\ -1,\ 0.4,\ -1.2,\ -0.7,\ 0)$ $b=(-0.7,\ 0.9,\ 0.3,\ 1.1,\ -0.2,\ 0.4)$	2 (right)
3	$a=(-0.9,\ 0.6,\ 1.2,\ 0.8,\ -0.8,\ -1)$ $b=(-0.4,\ 0.1,\ -0.6,\ 0.4,\ 0.1,\ 0.9)$	3 (left)
4	$a=(-0.3,\ 0.7,\ 0.7,\ 0.6,\ 0,\ -1.1)$ $b=(0.2,\ -0.6,\ -0.1,\ -0.1,\ 0.4,\ -0.7)$	3 (right)
5	$a=(0.2,\ 0.8,\ -0.6,\ -0.7,\ -0.3,\ -0.2)$ $b=(-0.9,\ -0.5,\ 0.6,\ -1.2,\ -0.3,\ 0.8)$	4 (left)
6	$a=(0.7,\ 0.6,\ -0.4,\ -0.1,\ 0.8,\ 0.1)$ $b=(-0.9,\ 0.4,\ 0.6,\ -0.4,\ -0.7,\ -1.2)$	4 (right)

Now enjoy the fascinating shapes of the chaotic attractors of these maps in Figures 2 to 4.

The computations and graphic visualizations are carried out in the computer algebra system *Maple 13*. The *Maple* commands producing the left image on Figure 2 are given in the Appendix. Note the value of the constant `iterations`, which is equal to 35000! This value corresponds to the number n of the points in (2), which are used to produce the image. Some of the examples in Table 1 require even more iterations – about 50000! Can you imagine that you would be able to reproduce the picture by hand-made calculations?... The solution images could not be seen in their full glory without computers and computer technologies and this is one of the greatest gifts of the 21-st century. It is not in vain that some fractals were regarded as exceptional objects, as counter examples, as "mathematical monsters" during 19-th century.

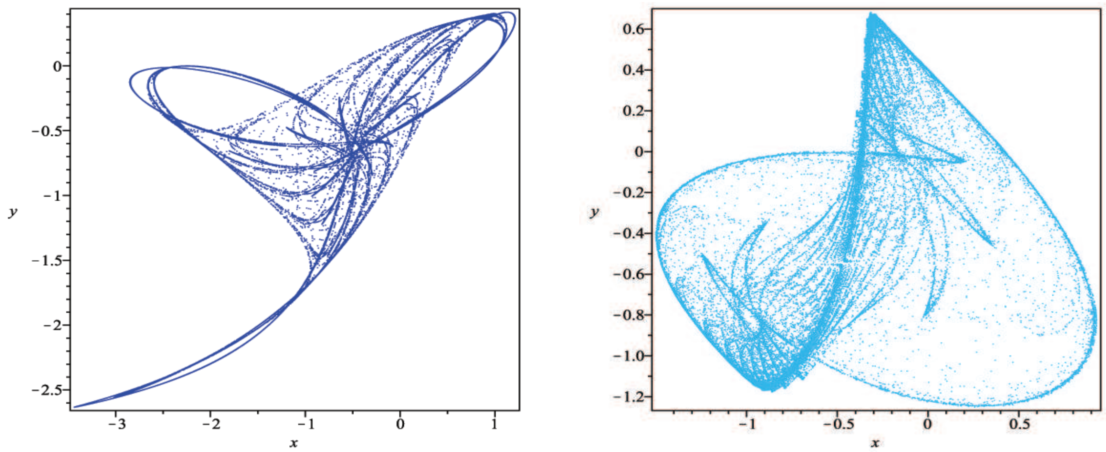

Figure 2: Chaotic attractors of quadratic maps: Examples 1 and 2

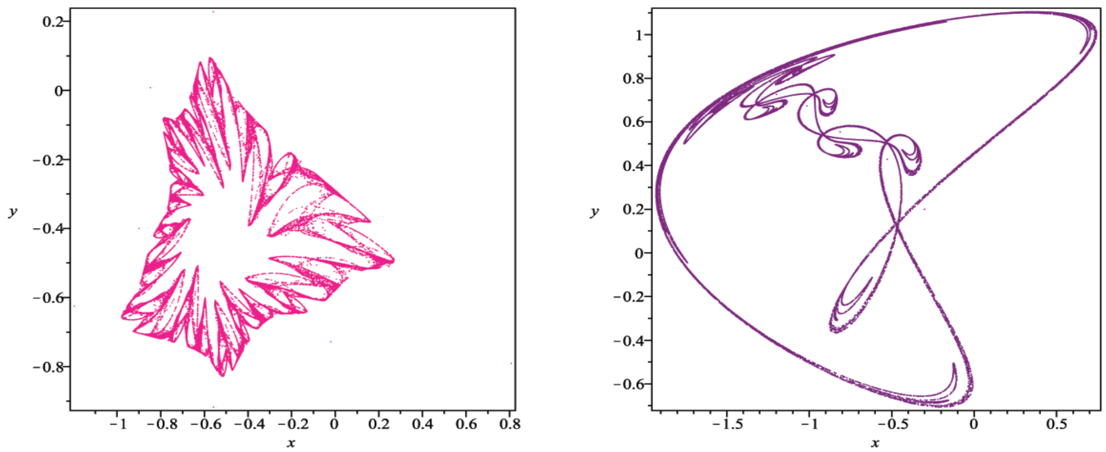

Figure 3: Chaotic attractors of quadratic maps: Examples 3 and 4

Task 1. Visualize the chaotic attractors of the well known Hénon map [2], proposed by the French astronomer Michel Hénon in 1976:

$$\begin{aligned} x_{n+1} &= 1 + cx_n^2 + y_n \\ y_{n+1} &= dx_n, \quad n = 0, 1, 2, \ldots \end{aligned}$$

with initial condition $(x_0, y_0) = (0,0)$ and coefficients: (i) $c = -1.2$, $d = 0.4$; (ii) $c = -1.4$, $d = 0.3$.

Task 2. Visualize the chaotic attractors of the following quadratic maps, starting with initial condition $(x_0, y_0) = (0,0)$ and coefficient vectors

$$\begin{aligned} (i) \quad a &= (0.8, -0.8, -1.1, -0.3, -0.1, -1), & b &= (-0.9, -0.4, 0.6, -0.4, -0.4, 0.4); \\ (ii) \quad a &= (1.3, 0.3, 0, 0.6, -0.6, -1), & b &= (0.1, -0.7, 0.5, -0.8, 0.1, -0.6). \end{aligned}$$

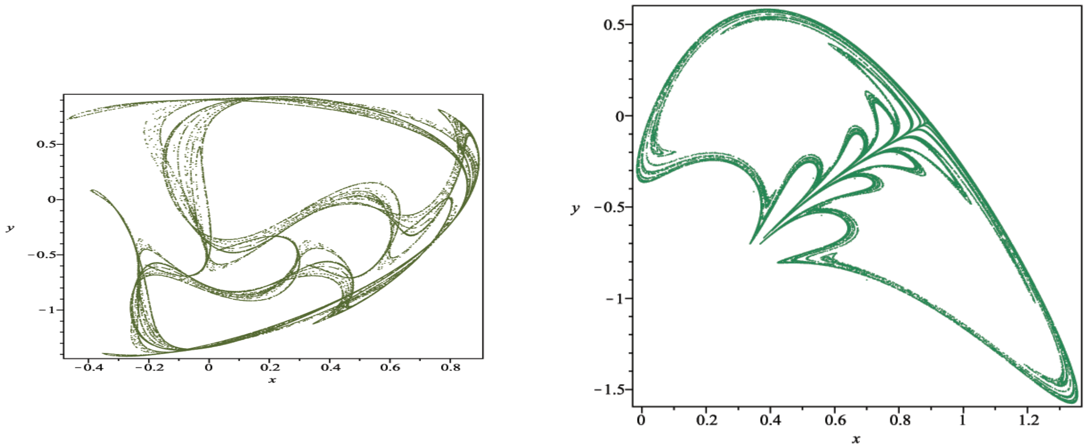

Figure 4: Chaotic attractors of quadratic maps: Examples 5 and 6

4 Iterated cubic maps

If we introduce in the quadratic map (5) terms of order 3 like x_n^3, $x_n^2 y_n$, $x_n y_n^2$, y_n^3, we obtain a cubic map. The general form of the cubic map is

$$\begin{aligned} x_{n+1} &= a_1 + a_2 x_n + a_3 x_n^2 + a_4 x_n^3 + a_5 x_n^2 y_n + a_6 x_n y_n + a_7 x_n y_n^2 + a_8 y_n + a_9 y_n^2 + a_{10} y_n^3 \\ y_{n+1} &= b_1 + b_2 x_n + b_3 x_n^2 + b_4 x_n^3 + b_5 x_n^2 y_n + b_6 x_n y_n + b_7 x_n y_n^2 + b_8 y_n + b_9 y_n^2 + b_{10} y_n^3 \\ & n = 0, 1, 2, \ldots \end{aligned} \tag{6}$$

Denote by $a = (a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10})$ and by $b = (b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10})$ the vectors of coefficients in (6). Table 2 contains four numerical examples [4] of cubic maps with chaotic attractors. The images of the chaotic attractors are shown in Figures 5 and 6. Would you imagine that so simple equations can produce so beautiful solutions?

Table 2: Examples of cubic maps with chaotic attractors

Example	Coefficient vectors a and b	Figure
7	$a = (-0.1,\ -0.6,\ 0.5,\ 0.2,\ -0.2,\ -0.3,\ -0.7,\ -0.8,\ -0.1,\ -0.9)$ $b = (-0.6,\ -0.2,\ 1.1,\ 0.6,\ 0.8,\ -0.8,\ -0.8,\ 1,\ 1.2,\ -0.8)$	5 (left)
8	$a = (-0.4,\ 0.6,\ 0,\ -0.5,\ 0.4,\ -1,\ -0.5,\ 0.3,\ -0.9,\ -0.7)$ $b = (-0.2,\ -0.7,\ -1.1,\ -0.2,\ -0.8,\ -1.2,\ -0.1,\ -0.4,\ -0.7,\ -0.9)$	5 (right)
9	$a = (0,\ -0.6,\ -0.6,\ 0.1,\ -0.9,\ 0.3,\ -0.5,\ 1,\ 0.2,\ 0.1)$ $b = (-0.2,\ -0.7,\ 0.4,\ 0.8,\ -0.4,\ -0.4,\ -0.5,\ -1.1,\ 0.9,\ 0.3)$	6 (left)
10	$a = (0.2,\ 0.9,\ -0.7,\ -0.2,\ 1,\ -0.2,\ -0.8,\ -0.4,\ -1.1,\ 0.3)$ $b = (-0.6,\ 0.1,\ 1.2,\ 0.3,\ 0.9,\ -0.2,\ 1,\ -1,\ 1.2,\ 0.8)$	6 (right)

The *Maple* commands producing the image on Figure 5 (left) are given in the Appendix. In Examples 7 to 10, the initial conditions are chosen to be $(x_0, y_0) = (0, 0)$.

It is interesting to note, that for most of the strange attractors the initial point does not matter, i. e. all initial conditions (x_0, y_0) result in the same image (check it!). In other words, any initial point will generate the same set of points, although they will be generated in a different order.

Task 3. Develop the image of the chaotic attractor for a cubic map with coefficient vectors

$$\begin{aligned} a &= (-0.3,\ 1.2,\ -1,\ -1.1,\ 0,\ 0.1,\ -0.7,\ 0.1,\ 1.2,\ 0.2) \\ b &= (-0.8,\ 0.3,\ 1.2,\ 0.8,\ -0.6,\ -0.5,\ -0.5,\ -0.8,\ 0.6,\ 0.8). \end{aligned}$$

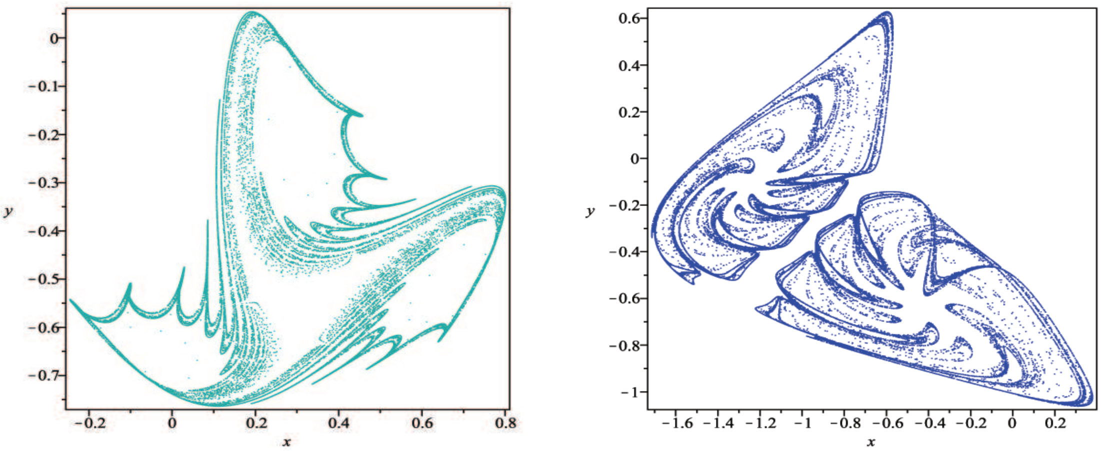

Figure 5: Chaotic attractors of cubic maps: Examples 7 and 8

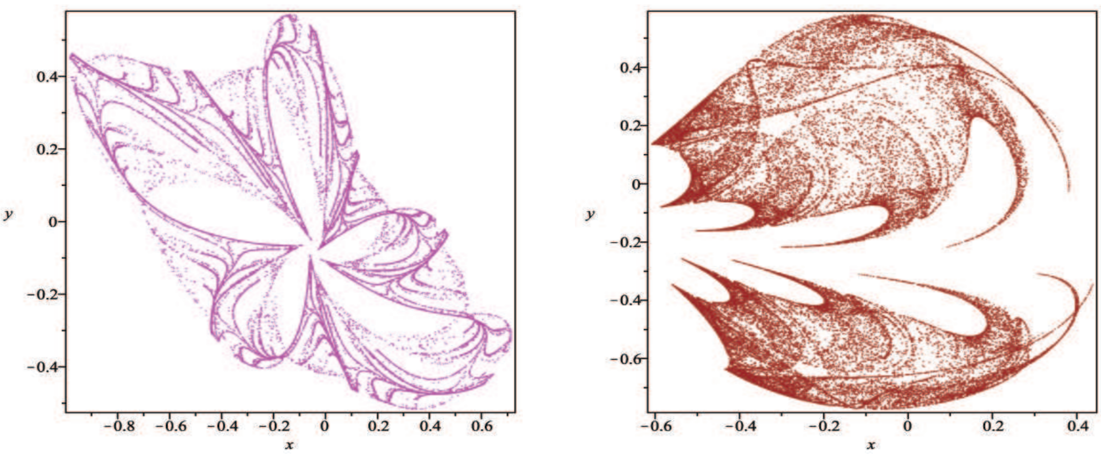

Figure 6: Chaotic attractors of cubic maps: Examples 9 and 10

Task 4. Write down the general form of a quartic iterated map, i. e. a map with the highest order term being 4. Modify the *Maple* commands to visualize the chaotic attractor of the quartic map with coefficient vectors

$$
\begin{aligned}
a &= (-0.4, -0.9, -0.4, -0.9, 0.6, -1.1, 0.7, 0.3, 0.1, -0.9, -1.1, 0.6, -0.6, 0.2, -0.2) \\
b &= (0.2, -0.2, -0.6, -1.2, -0.2, 0, -1, -1, -0.9, 0.1, 1.1, -0.7, -0.5, 1, 0.4).
\end{aligned}
$$

5 Other fascinating iterated maps

In the previous two sections we considered polynomial iterated maps, i. e. the right-hand sides of (5) and (6) were polynomials of two variables of second and third degree respectively. Instead of polynomials we can consider iterated maps, involving other nonlinear functions such as trigonometric functions, absolute value, square root, signum function etc. – all functions that are usually supported by any software environment. Below we present two such examples.

The King's Dream. It is a simple, yet beautiful fractal [7]. The equations to produce it are:

$$
\begin{aligned}
x_{n+1} &= \sin(b y_n) + c \sin(b x_n) \\
y_{n+1} &= \sin(a x_n) + d \sin(a y_n), \quad n = 0, 1, 2, \ldots
\end{aligned}
\tag{7}
$$

with initial condition $(x_0, y_0) = (0.1, 0.1)$ and $a = -0.966918$, $b = 2.879879$, $c = 0.765145$, $d = 0.7447228$. The fractal is shown in Figure 7 (left).

If we slightly change the values of b, c and d, taking $b := b + 0.1$, $c := c + 0.01$, $d := d - 0.2$, then we obtain a new fractal, presented in Figure 7 (right).

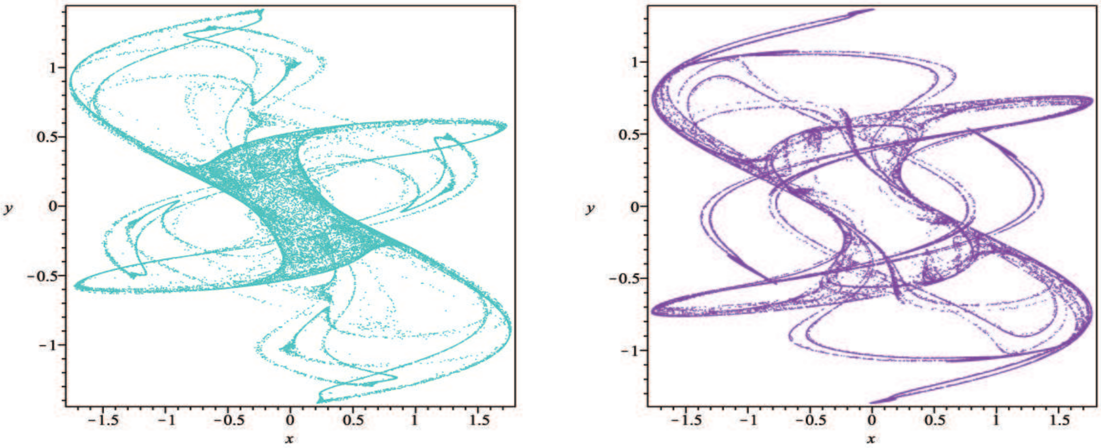

Figure 7: The King's Dream (left) and its modification (right)

Task 5. Construct another fractal using (7) with coefficient values $a = -0.967$, $b = 2.89$, $c = 0.769$, $d = 0.785$ and initial condition $(x_0, y_0) = (0.1, 0.1)$.

Barry Martin fractal. It is produced by the following discrete iterated system

$$\begin{aligned} x_{n+1} &= y_n - sign\,(x_n)\sqrt{|bx_n - c|} \\ y_{n+1} &= a - x_n, \quad n = 0, 1, 2, \dots; \quad x_0 = y_0 = 0.1 \end{aligned}$$

Three special functions are used in the first equation: the square root, the absolute value function, and the signum function *sign*. The *sign* function returns a value of 1 if the x_n-value is positive, and a value of -1 if the x_n-value is negative. The constants a, b and c may take any values. The fractals, presented in Figure 8 are computed for $a = 1$, $b = 2$, $c = 3$ (left plot) and for $a = 0.4$, $b = 1$, $c = 0.1$ (right plot).

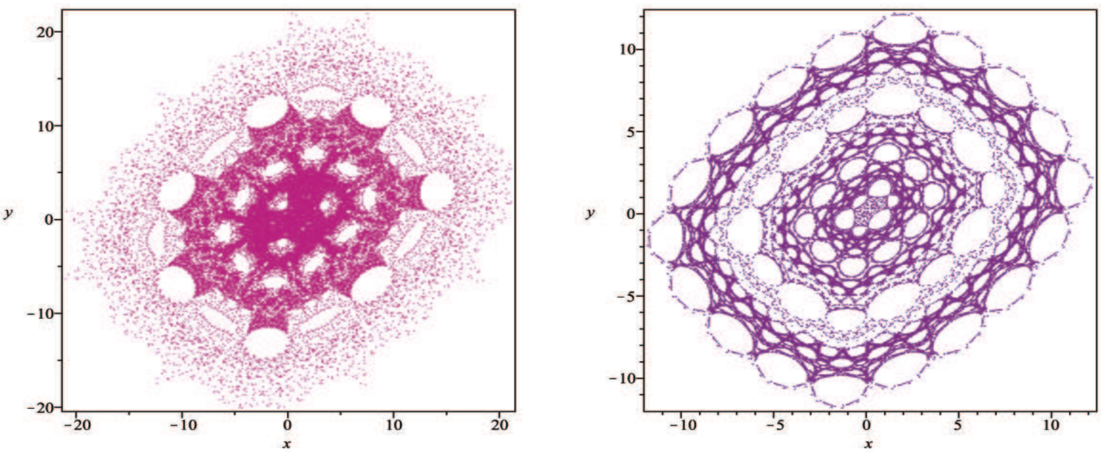

Figure 8: Barry Martin strange attractors

Task 6. Construct other Barry Martin fractals with initial conditions $x_0 = y_0 = 0.1$ and coefficient values

(*i*) $a = 1$, $b = -1$, $c = 2$;

(*ii*) $a = 1$, $b = -1$, $c = -1$;

(*iii*) $a = 0.2$, $b = -0.1$, $c = 0.2$.

6 If you want to learn more...

Classical fractals such as the Koch curve, the Sierpinski triangle or hexagon (see [3], [5], [6], [8] for details) are objects that display self-similarity under magnification and can be constructed using a simple motif (an image repeated on ever-reduced scales). Many objects in nature display this self-similarity at different scales; for example, cauliflowers, ferns, trees, and even blood vessel networks in our own bodies have some fractal structure. Fractals appeared also in art, for example in the paintings of the Dutch artist *Maurits C. Escher* [9], before they were widely appreciated by mathematicians and scientists. Fractals are being applied in many branches of science, for example in computer graphics and image compression (take a closer look at the images on the Web).

The fractals shown on the figures in the previous sections do not display exactly self-similarity; they have only regions that are self-similar. Are there such objects in the real world?

The main ingredient in the definition of a fractal is its *dimension*. Isolated points have dimension zero, line segments have dimension one, surfaces have dimension two, and solids have dimension three. This is their "usual" or so called *topological dimension*. A fractal has a dimension that exceeds its topological dimension. In most cases, fractals possess non-integer dimension, called *fractal dimensions*. The fractal dimension gives finer information about the roughness or complexity of the set. There are however fractals with integer dimension; in this case the fractal dimension must exceed the topological dimension. For example the Sierpinski tetrahedron (a tetrahedral analogue of the triangle) has fractal dimension two, but topological dimension one [3].

The uniqueness of the strange attractors is due to the fact that one does not know exactly where on the attractor the system's trajectory will be. Two points on the attractor that are near each other at one time will be arbitrarily far apart at later times. The only restriction is that the state of the system remains on the attractor. Moreover, the motion of the system never repeats – there are no cycles. The motion on these strange attractors is what is called *chaotic behavior of the system*.

Imagine a set of initial points filling a small square region in the (x,y)-plane. After one iteration the points will move to new positions in the plane, occupying for example an "elongated" region like a parallelogram. The square has contracted in one direction and expanded in the other. With each iteration the parallelogram gets longer and narrower. The orientation of the parallelogram also changes with each iteration. The quantitative interpretation of these effects are given by the so called *Lyapunov exponents*. The name comes from the late 19th-century Russian mathematician Aleksandr M. Lyapunov. An iterated map described by two equations, like the quadratic or cubic maps, possesses two Lyapunov exponents – a positive one corresponding to the direction of expansion, and a negative one corresponding to the direction of contraction. Typical for chaos is that at least one of the Lyapunov exponents should be positive. The method for computing the Lyapunov exponents is slightly complicated and will not be discussed here. The *Maple* commands for computing the Lyapunov exponents in Examples 1 and 7 are given in the Appendix. Table 3 contains the Lyapunov exponents L_1 and L_2 of the quadratic and the cubic maps from Examples 1 to 10.

There is a close relation between the fractal dimension and the Lyapunov exponents. Assume that L_1, L_2 are known and $L_1 > 0$, $L_2 < 0$; as a rule they should satisfy the relation $|L_1| < |L_2|$. Then the fractal dimension FD can be computed via the formula

$$FD = 1 - \frac{L_1}{L_2}. \tag{8}$$

The value FD is called Lyapunov dimension and also Kaplan-Yorke dimension after the names of the mathematicians who proposed the rule (8). The interested reader can consult the books [2], [3] and [4] for more details on this topic. The fractal dimensions of the chaotic attractors from Examples 1 to 10 are given in Table 3, column FD.

Let us finish this article by Barnsley's words [10]: "*Fractal geometry will make you see everything differently... You risk the loss of your childhood vision of clouds, forests, galaxes, leaves, feathers, rocks,*

Table 3: Lyapunov exponents L_1, L_2 and fractal dimensions FD of the maps in Tables 1 and 2

Example	L_1	L_2	FD	Figure
1	0.18	−0.45	1.4	2 (left)
2	0.093	−0.099	1.94	2 (right)
3	0.029	−0.032	1.91	3 (left)
4	0.16	−0.32	1.5	3 (right)
5	0.18	−0.37	1.48	4 (left)
6	0.13	−0.24	1.54	4 (right)
7	0.11	−0.19	1.58	5 (left)
8	0.095	−0.16	1.59	5 (right)
9	0.049	−0.11	1.45	6 (left)
10	0.043	−0.058	1.74	6 (right)

mountains, torrents of water, carpets, bricks, and much else besides. Never again will your interpretation of these things be quite the same."

Appendix: Maple commands

The Appendix contains *Maple* commands for computing and visualizing the chaotic attractor as well as its Lyapunov exponents and fractal dimension for Example 1 from Table 1 and Example 7 from Table 2. The interested reader can produce the images of the other examples by simply replacing the vector components of *a* and *b* with the corresponding numerical values from the tables.

The *Maple* procedures are kept as simple as possible. The more experienced programmer can translate them in her/his favorite programming environment.

• Computing and visualizing the chaotic attractor of the quadratic map in Example 1

```
 > restart:
 > with(plots):
 > iterations:=35000:
 > a:=array(1..6,[-1.2, -0.6, -0.5, 0.1, -0.7, 0.2]);
   b:=array(1..6,[-0.9, 0.9, 0.1, -0.3, -1, 0.3]);
 > x:=array(0..iterations):
   y:=array(0..iterations):
 > x[0]:=0: y[0]:=0: #initial condition
 > for i from 0 to iterations-1 do
     x[i+1]:=a[1]+a[2]*x[i]+a[3]*(x[i])^2+a[4]*x[i]*y[i]
            +a[5]*y[i]+a[6]*(y[i])^2:
     y[i+1]:=b[1]+b[2]*x[i]+b[3]*(x[i])^2+b[4]*x[i]*y[i]
            +b[5]*y[i]+b[6]*(y[i])^2
   end do:
>  points:=[[x[n],y[n]]$n=1..iterations]:
>  pointplot(points,style=point,symbol=solidcircle,symbolsize=4,
   color=blue,axes=boxed,labels=['x','y']);
```

• Computing the Lyapunov exponents and the fractal dimension of the quadratic map in Example 1

```
> itermax:=500:
```

```
> a:=array(1..6,[-1.2, -0.6, -0.5, 0.1, -0.7, 0.2]);
  b:=array(1..6,[-0.9, 0.9, 0.1, -0.3, -1, 0.3]);
> x:=0: y:=0:
> vector1:=<1,0>: vector2:=<0,1>:
> for i from 1 to itermax do
    x1:=a[1]+a[2]*x+a[3]*x^2+a[4]*x*y+a[5]*y+a[6]*y^2:
    y1:=b[1]+b[2]*x+b[3]*x^2+b[4]*x*y+b[5]*y+b[6]*y^2:
    x:=x1:
    y:=y1:
    J:=Matrix([[a[2]+2*a[3]*x+a[4]*y,a[4]*x+a[5]+2*a[6]*y],
               [b[2]+2*b[3]*x+b[4]*y,b[4]*x+b[5]+2*b[6]*y]]):
    vector1:=J.vector1:
    vector2:=J.vector2:
    dotprod1:=vector1.vector1:
    dotprod2:=vector1.vector2:
    vector2:=vector2 - (dotprod2/dotprod1)*vector1:
    length_vector1:=sqrt(dotprod1):
    area:=abs(vector1[1]*vector2[2] - vector1[2]*vector2[1]):
    L1:=evalf(log(length_vector1)/i):
    L2:=evalf(log(area)/i-L1)
  end do:
> print('L1'=L1, 'L2'=L2); #Lyapunov exponents, L2<0<L1, |L1|<|L2|
> FD:=1 - L1/L2;  #the fractal dimension
```

• Computing and visualizing the chaotic attractor of the cubic map in Example 7

```
> restart:
> with(plots):
> iterations:=35000:
> a:=array(1..10,[-0.1,-0.6,0.5,0.2,-0.2,-0.3,-0.7,-0.8,-0.1,-0.9]);
  b:=array(1..10,[-0.6,-0.2,1.1,0.6,0.8,-0.8,-0.8,1,1.2,-0.8]);
> x:=array(0..iterations): y:=array(0..iterations):
> x[0]:=0: y[0]:=0: #initial conditions
> for i from 0 to iterations-1 do
    x[i+1]:=a[1]+a[2]*x[i]+a[3]*(x[i])^2+a[4]*(x[i])^3
           +a[5]*(x[i])^2*y[i]+a[6]*x[i]*y[i]+a[7]*x[i]*(y[i])^2
           +a[8]*y[i]+a[9]*(y[i])^2 + a[10]*(y[i])^3:
    y[i+1]:=b[1]+b[2]*x[i]+b[3]*(x[i])^2+b[4]*(x[i])^3
           +b[5]*(x[i])^2*y[i]+b[6]*x[i]*y[i]+b[7]*x[i]*(y[i])^2
           +b[8]*y[i]+b[9]*(y[i])^2 + b[10]*(y[i])^3:
  end do:
> points:=[[x[n],y[n]]$n=1..iterations]:
> pointplot(points,style=point,symbol=solidcircle,symbolsize=4,
            color="LightSeaGreen",axes=boxed,labels=['x','y']);
```

• The Lyapunov exponents and the fractal dimension of the cubic map in Example 7

```
> Digits:=30:
> x:='x': y:='y':
> itermax:=500:
> a:=array(1..10,[-0.1,-0.6,0.5,0.2,-0.2,-0.3,-0.7,-0.8,-0.1,-0.9]);
  b:=array(1..10,[-0.6,-0.2,1.1,0.6,0.8,-0.8,-0.8,1,1.2,-0.8]);
```

```
> x:=0: y:=0:
> vector1:=<1,0>: vector2:=<0,1>:
> for i from 1 to itermax do
    x1:=a[1]+a[2]*x+a[3]*x^2+a[4]*x^3+a[5]*x^2*y
        +a[6]*x*y+a[7]*x*y^2+a[8]*y+a[9]*y^2 + a[10]*y^3:
    y1:=b[1]+b[2]*x+b[3]*x^2+b[4]*x^3+b[5]*x^2*y
        +b[6]*x*y+b[7]*x*y^2+b[8]*y+b[9]*y^2 + b[10]*y^3:
    x:=x1: y:=y1:
    J:=Matrix([[a[2]+2*a[3]*x+3*a[4]*x^2+2*a[5]*x*y+a[6]*y+a[7]*y^2,
          a[5]*x^2+a[6]*x+2*a[7]*x*y+a[8]+2*a[9]*y+3*a[10]*y^2],
          [b[2]+2*b[3]*x+3*b[4]*x^2+2*b[5]*x*y+b[6]*y+b[7]*y^2,
          b[5]*x^2+b[6]*x+2*b[7]*x*y+b[8]+2*b[9]*y+3*b[10]*y^2]]):
    vector1:=J.vector1:
    vector2:=J.vector2:
    dotprod1:=vector1.vector1:
    dotprod2:=vector1.vector2:
    vector2:=vector2 - (dotprod2/dotprod1)*vector1:
    length_vector1:=sqrt(dotprod1):
    area:=abs(vector1[1]*vector2[2] - vector1[2]*vector2[1]):
    L1:=evalf(log(length_vector1)/i):
    L2:=evalf(log(area)/i-L1)
  end do:
> print('L1'=L1, 'L2'=L2); #Lyapunov exponents, L2<0<L1, |L1|<|L2|
> FD:=1 - L1/L2;  #the fractal dimension
```

Recommended reading

[1] Dimitrova N. *Order and Chaos in a Model of Population Biology*, In: J. Andersen et al, $MATH_2EARTH$: Bringing Mathematics to Earth. 141876-LLP-1-2008-1-AT-COMENIUS-CMP, Publ. PRVOKRUH, Prague, Czech Republic, 49–54 (in English), 148–154 (in Bulgarian), 2010.

[2] Lynch S. *Dynamical Systems with Applications Using Maple*^{TM}, Birkhäuser, Boston, 2010.

[3] Peitgen H.-O., Jürgens H., Saupe D. *Fractals for the Classroom. Part One: Introduction to Fractals and Chaos*, Springer, New York, 1992.

[4] Sprott J. C. *Strange Attractors: Creating Patterns in Chaos*, 2000.

[5] Ulovec A., Hohenwarter, H.: *Fractals – broken with no need to repair*, in this volume.

[6] `http://math.bu.edu/DYSYS/`

[7] Pickover C. *Chaos in Wonderland*, St. Martin's Press, 1994.

[8] Sendova, E. *Introducing a Little Chaos to Break the Tradition*, Mathematics and Education in Mathematics, Proc. 31st Spring Conf. UBM, 35–47, 2002 (in Bulgarian).

[9] `http://www.mcescher.com/`

[10] Barnsley, M. F. *Fractals Everywhere*, Academic Press, 1988.