Abstract

This report corresponds to the evaluation of the results obtained in the first project of Simulation Methods, as well as, the solutions of some theoretical questions asked in the explanatory document of the project. In order to solve the proposed questions, we have prepared a script in C, which is also attached in the delivery. We shall divide the report in two parts. The first one, will be destined to the explanation of the implemented code and, in the second one, we will answer the four questions asked in the statement.

1 Code implementation

As stated in the abstract, we have prepared a script in C in order to make our delivery. In this section, we will explain how we have implemented the code, focusing on the use of the library *pgplot* to make interactive graphics in C.

To run our program, we should have installed the *pgplot* library, specifically, *cpgplot*, which allows compiling a script in C using that library. Once we have installed the package, we can execute the script using the following commands (at least for macOS users):

```
gcc -o Henon.exe Henon.c -I/opt/local/include -L/opt/local/lib -lcpgplot -lpgplot -lm -lX11
```

```
./Henon.exe
```

Once this is done, a black window with a square $[0,1]\times[0,1]$ will appear. If we click in any point of the square, it will appear the dynamics of the conservative Henon map at that point. More specifically, 1000 iterations of the Henon map in the clicked point is shown. We can click 20 times, once achieved, the window will not accept more points and we will be able to reproduce an image similar to Figure 1. We should return to the terminal to continue the program.

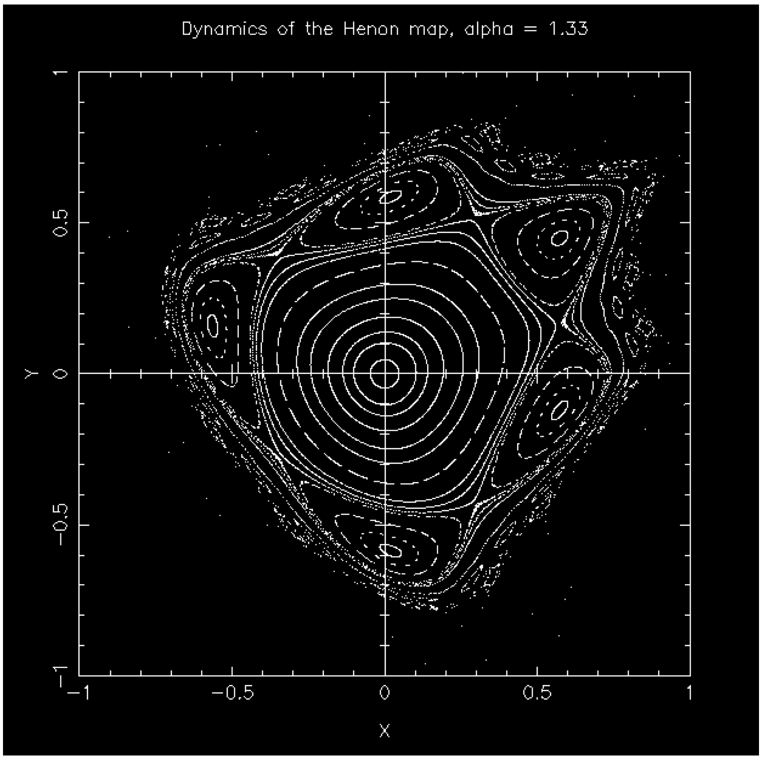

Figure 1: Dynamics of the Henon map with α=1.33

We will have to add *return* to the terminal to continue with the script (at least for macOS users). At that moment, the program computes a Newton's method algorithm to the function $F(x,y) := f^5(x,y) - (x,y)$ where f is the conservative Henon map. With this, since Newton's method is a root-finding algorithm, we will be able to find a zero of F, this is equivalent to find a fixed point by f^5, i.e., a periodic orbit of period 5[1]. Nevertheless, let us recall that Newton's method will only find one point, the fixed point for f^5 closest to the given initial condition, hence, to find the complete orbit, we will have to choose carefully the initial conditions. Looking at the picture, we have prepared ten initial conditions, stored in two vectors, *initial_cond_hyper* and *initial_cond_elipt*, to find both orbits, the hyperbolic and the elliptic respectively. The result is displayed on the screen.

Now that we have understood how the program works, let's go into details. Let us explain more about the *pgplot* library. To make the interactive graphic in C, I have implemented the function *dynamics_plot*, that opens the black window in the screen using the function *cpgopen*. Then, to set up the plot and include some labels, we have used the functions *cpgenv* (in which we set the range of the displayed square) and *cpglab* (in which we add the title and the x, y-labels). Finally, we start a loop in which we pick up the mouse click coordinate using the function *cpgcurs* and later on, we compute iterations of the Henon map in that point, plotting it with the function *cpgpt*.[2]

Let us explain the Newton's method part. I have prepared several functions in order to implement this, let us explain them. Remember that our goal now, is to compute a zero of F, where $F(x, y) = f^5(x, y) - (x, y)$ and f is the Henon map. A function with name F is implemented to compute the image of a given point (x, y) by F. Let us recall the Newton's method schema:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - [J_F(\mathbf{x}^{(k)})]^{-1} F(\mathbf{x}^{(k)}), \qquad \mathbf{x} = (x, y) \in \mathbb{R}^2.$$

Therefore, we need to compute the Jacobian matrix of F. To determine it, we could use the following theorem:

Theorem 1.1. *Consider differentiable functions $f : \mathbb{R}^m \to \mathbb{R}^k$ and $g : \mathbb{R}^n \to \mathbb{R}^m$, and a point β in $\mathbb{R}^n$. Let $J_f(\beta)$ denote the Jacobian matrix of f at β. Then,*

$$J_{f\circ g}(\beta) = J_f(g(\beta))J_g(\beta)$$

Proof. Drectly from the chain rule. □

Using this, we can compute easily the Jacobian matrix of F at a point $\beta \in \mathbb{R}^2$:

$$J_F(\beta) = \left(\prod_{k=1}^{4} J_f(f^{5-k})\right) J_f(\beta) - I_2$$

where I_2 is the identity matrix of order two. Notice that this is simply a matrix multiplication! To implement this in C, we have prepared two functions called *Jacobian* and *Jacobian_5*, that computes the Jacobian matrix of f and F respectively at a given point. Notice that the Jacobian matrix of f at a point (x, y) is the following:

$$J_f(x, y) = \begin{pmatrix} cos\alpha + 2xsin\alpha & -sin\alpha \\ sin\alpha - 2xcos\alpha & cos\alpha \end{pmatrix} \tag{1.1}$$

Observe that this is what *Jacobian* function returns. With the help of this function, we can compute the iterates of f and perform the desire matrix multiplication to obtain the Jacobian

[1]Notice that we have two periodic orbit of period 5, a hyperbolic and an elliptic.

[2]Detailed information of every function used here can be found in the pgplot documentation website or in the following link: https://d1qmdf3vop2l07.cloudfront.net/pamoc.cloudvent.net/compressed/e3da9587fdfa2166e3fc574d70c4f490.pdf

of F(this is what has been implemented in the *Jacobian_5* function).

Now, we are able to perform Newton's method. We perform a maximum of 50 Newton's iterations until $||\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}||_2 < 10^{-15}$ or $||F(\mathbf{x}^{(k+1)})||_2 < 10^{-15}$. Notice that to compute the inverse of the Jacobian matrix, we have prepared a function called *inv*, that computes the inverse of a 2×2 matrix.

Finally, we have also studied the eigenvalues related to the elliptic and the hyperbolic periodic orbits to study its stability. We have computed the eigenvalues of the Jacobian matrix of f^5 evaluated in the periodic points. In order to implement this, our program computes the eigenvalues of a 2×2 matrix, using its trace and its determinant. Remember that, for a given 2×2 matrix $A = (a_{i,j})_{1 \leqslant i,j \leqslant 2}$,

$$\begin{aligned} p_A(t) = det(A - tI_2) &= (a_{1,1} - t)(a_{2,2} - t) - a_{1,2}a_{2,1} \\ &= t^2 - (a_{1,1} + a_{2,2})t + a_{1,1}a_{2,2} - a_{1,2}a_{2,1} \\ &= t^2 - tr(A)t + det(A) \end{aligned}$$

Therefore, to find the eigenvalues, we only have to compute the solutions of a second degree polynomial. Firstly, we have computed the discriminant of the equation, $\Delta = tr(A)^2 - 4det(A)$. If $\Delta \geq 0$, a function called *solver_real* computes the real solutions of the equation, otherwise, a function called *solver_complex*, computes the complex solutions of the equation using the *complex.h* library.

2 Proposed exercises

Exercise 1. Show that this is an area-preserving map.

Proof. To see that the conservative Henon map is an area-preserving map, it is enough to see that $|\det(J_f(x,y))| = 1$ for every $(x,y) \in \mathbb{R}^2$, where f is the conservative Henon map. Notice that, using the expression of the Jacobian given in (1.1), it is easy to see that $\det(J_f(x,y)) = cos^2\alpha + sin^2\alpha = 1$. □

Exercise 2. Compute the elliptic periodic orbit of period five mentioned before. Give all five points.

Proof. Using the delivered script in C, we can obtain the elliptic periodic orbit of period five. The results are shown in the following table:

Initial guess	Orbit
(0,-0.600)	(0.002,-0.586)
(0.600,-0.200)	(0.575,-0.120)
(0.600,0.480)	(0.575,0.450)
(0,0.600)	(0.002,0.587)
(-0.600,0.200)	(-0.565,0.159)

Table 1: Elliptic periodic orbit of period five and the initial conditions used to find them

□

Exercise 3. Compute the hyperbolic periodic orbit of period five mentioned before. Give all five points.

Proof. As before, using the delivered script in C, we can obtain the hyperbolic periodic orbit of period five. The results are shown in the following table:

Initial guess	Orbit
(0.35,-0.50)	(0.30,-0.43)
(0.55,0.20)	(0.58,0.17)
(0.35,0.48)	(0.30,0.52)
(-0.38,0.35)	(-0.35,0.39)
(-0.35,-0.30)	(-0.35,-0.27)

Table 2: Hyperbolic periodic orbit of period five and the initial conditions used to find them

□

Exercise 4. Compute the eigenvalues related to the elliptic and the hyperbolic periodic orbit. Do they depend on the starting point of the orbit? What can you tell about their product?

The eigenvalues are computed in the delivered script in C. Notice that the eigenvalues does not depend on the starting point of the orbit. For each point in the elliptic periodic orbit, we have $\lambda_{1,2} \approx 0.87 \pm 0.49i$, and, for each point in the hyperbolic periodic orbit, we have $\lambda_1 \approx 1.53$ and $\lambda_2 \approx 0.65$.

Notice that $\lambda_1\lambda_2 = 1$ and it is not a coincidence. Evidently, since $\det(J_f(x,y)) = 1$ for each $(x,y) \in \mathbb{R}^2$, then, $\det(J_{f^5}(x,y)) = 1$. Moreover, since the determinant of a matrix is equal to the product of its eigenvalues, we obtain that $\lambda_1\lambda_2 = 1$.