

# Chaotic dynamics with Maxima

*Antonio Morante and José A. Vallejo*

e-mail: amorante,jvallejo@fciencias.uaslp.mx

Faculty of Sciences

State University of San Luis Potosí

78290 San Luis Potosí (SLP)

México

## Abstract

*We present an introduction to the study of chaos in discrete and continuous dynamical systems using the CAS Maxima. These notes are intended to cover the standard topics and techniques: discrete and continuous logistic equation to model growth population, staircase plots, bifurcation diagrams and chaos transition, nonlinear continuous dynamics (Lorentz system and Duffing oscillator), Lyapunov exponents, Poincaré sections, fractal dimension and strange attractors. The distinctive feature here is the use of free software with just one ingredient: the CAS Maxima. It is cross-platform and have extensive on-line documentation.*

## 1 Introduction

The use of a computer is a natural way to explore those notions related to dynamical systems; here, we present the basics of chaotic dynamics using Maxima. This is a free, cross-platform, general purpose computer algebra system [Max 11], capable also of doing numerical computations, that we have used in short introductory summer courses (about 10 hours). The participants were a mixture of mathematics, physics and engineering students, and occasionally high-school teachers. The following notes are the outcomes of such courses, and given the target audience, we have made every effort to keep them at an elementary level, so standard calculus and linear algebra should suffice to read them.

We believe that the use of Maxima has the following advantages:

- (1) Only one program is required (even the use of a graphical front-end is not mandatory).
- (2) Students can experiment with well known dynamical models via the built-in functions of Maxima, and modify them to suit their needs, as the whole sources of Maxima are available.
- (3) The learning curve is smooth, because Maxima syntax is quite intuitive. In fact, this syntax is similar to the one used on the blackboard, so there is little risk that the student “get things done by the computer, without understanding what is going on”, as one must know the mathematical meaning of a command even in order to write its arguments.

- (4) Students can write their own Maxima programs in a short time (in our experience, after a couple of two-hours sessions).
- (5) Since the software is free and cross-platform, students and teachers can use the same program at school and at home with no cost. There is no need for renewing licenses or a particular operating system.

Of course, it is impossible to include all topics one would wish, so we have excluded those theoretical results which are hard to implement using the computer, because they involve irrational numbers. Explicitly, we only make some brief remarks about results such as the density of periodic orbits or sensitivity to initial conditions and leave aside topological transitivity.

In writing this paper, we have used Maxima version 5.28.0 and its graphical interface wxMaxima version 12.09.0. The version of Gnuplot was 4.6 patchlevel 0, everything running on a generic desktop PC with Slackware Linux 14.0. The commands used are available as a Maxima session in the personal page of one of the authors: <http://galia.fc.uaslp.mx/~jvallejo/Software.html>.

## 2 The logistic equation: continuous case

The simplest model of population growth is given by:

$$\frac{dP(t)}{dt} = k \cdot P(t) \tag{1}$$

where  $P(t)$  is the population at time  $t$  and  $k$  is a constant, positive for an increasing population and negative for a decreasing one. However, in the long run this is not a good model: it disregards limiting factors such as propagations of diseases or lacks of food supply. A simple modification in (1) which takes into account these factors can be obtained replacing  $k$  with a parameter  $K = K(P)$ , such that it decreases when  $P$  increases. A possible model for this, is  $K = a - b \cdot P$ ,  $a \gg b > 0$ , so (1) takes the form

$$\frac{dP(t)}{dt} = (a - b \cdot P(t)) \cdot P(t). \tag{2}$$

If  $P(t) \simeq 0$ , equation (2) becomes equation (1), with  $k \simeq a$ , but for increasing values of  $P(t)$ ,  $b \cdot P(t)$  approach  $a$ , and therefore  $\frac{dP(t)}{dt} \simeq 0$ , slowing down the population's growth. Making the change of variables

$$P = \frac{ax}{b}, \quad \frac{dP(t)}{dt} = \frac{a}{b} \frac{dx}{dt} \tag{3}$$

equation (2) becomes (with  $a = k$ )

$$\frac{dx}{dt} = k \cdot x \cdot (1 - x), \tag{4}$$

which is called the logistic equation (it was introduced by P. F. Verhulst in 1838, a discrete version was popularized by R. May in the seventies, see [May 76]).

We will use Maxima to solve equation (4) (note the use of the apostrophe to define an equation. The command for solving first *or* second-order ODEs is `ode2`):

```
(%i1) 'diff(x,t)=k*x*(1-x);
```

```
(%o1)  $\frac{d}{dt} x = k (1 - x) x$ 
```

```
(%i2) ode2(%,x,t);
```

```
(%o2)  $\frac{\log(x) - \log(x - 1)}{k} = t + \%c$ 
```

Now, we set an initial condition for this *first-order* (hence the 1 in `ic1`) equation  $x(t = 0) = x_0$ :

```
(%i3) ic1(%,t=0,x=x[0]),logcontract;
```

```
(%o3)  $\frac{\log\left(\frac{x}{x-1}\right)}{k} = \frac{kt + \log\left(\frac{x_0}{x_0-1}\right)}{k}$ 
```

and solving the last equation we get

```
(%i4) solve(%,x);
```

```
(%o4)  $\left[ x = \frac{x_0 e^{kt}}{x_0 e^{kt} - x_0 + 1} \right]$ 
```

Thus, we have the explicit solution (note that the outcome (`%o4`) is a *list* –a collection of elements enclosed between brackets– and we want to define our solution as the right-hand side of its first element):

```
(%i5) define(x(t),rhs(first(%o4)))$
```

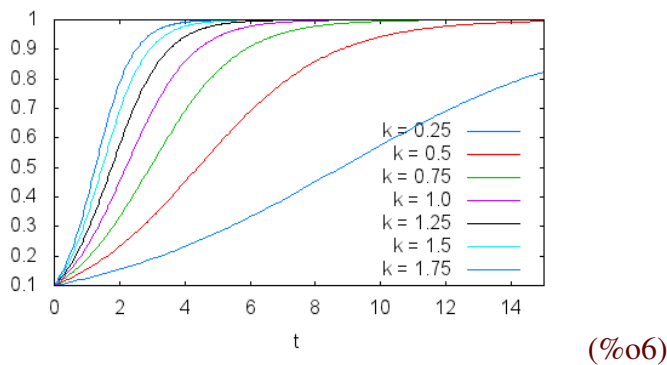
Let us note that normalization (3) implies that for all initial conditions  $0 < x_0$ , the solutions  $x(t) \rightarrow 1$  when  $t \rightarrow \infty$ . In fact, even if we had not known the solutions explicitly, a simple continuity argument would show that all solutions bounded by the steady states  $x = 0$  and  $x = 1$ , tend monotonically to  $x = 1$ <sup>1</sup>. Also, in the unbounded region  $x > 1$  all solutions tend monotonically to the steady state  $x = 1$ . For  $x < 0$  there is no solution (populations are non negative). In general, in a one dimensional continuous system, solutions either converge monotonically to a steady state or diverge.

The parameter  $k$  affects the slope of the solutions, as we can see by graphing them together for different values of  $k$  (in this example, varying from  $k = 0.25$  to  $k = 1.75$  with step 0.25).

```
(%i6) wxplot2d(makelist(subst([k=d*0.25,x[0]=0.1],x(t)),d,1,7),
[t,0,15],cons(legend,makelist(k=d*0.25,d,1,7)),
[gnuplot_preamble,"set key right bottom"]);
```

---

<sup>1</sup>For a system  $\dot{x} = f(x)$ , in the region between two consecutive steady states  $x(t) = x^*$  with  $f(x^*) = 0$ ,  $x_1^* < x_2^*$ , the mapping  $f$  (and therefore the derivative  $\dot{x}$ ) has constant sign, so a solution  $x(t)$  with initial condition  $x_0$  such that  $x_1^* < x_0 < x_2^*$ , is strictly monotonic. This solution is bounded, so  $\lim_{t \rightarrow \infty} x(t) = x_2^*$  if  $x(t)$  is increasing, and  $\lim_{t \rightarrow \infty} x(t) = x_1^*$  if  $x(t)$  is decreasing. Finally, because of the uniqueness theorem for the Cauchy problem,  $x(t)$  can not reach the values  $x_1^*, x_2^*$ . A similar reasoning can be used for unbounded regions.



(%t6)

(%o6)

### 3 The logistic equation: discrete case

The discrete version of equation (4) is the difference equation (with  $r > 0$ )

$$x_{n+1} = r \cdot x_n \cdot (1 - x_n). \quad (5)$$

In general, equation (5) can't be solved exactly<sup>2</sup>. Let us define the so called *logistic map* (in fact, a family of *quadratic maps*),  $F(r, x) = r \cdot x \cdot (1 - x)$ . From (5) we see that, for a given  $r$ , the fixed points of  $F(r, x)$  are the only constant solutions  $x_n = c$  of (5). These *steady states* can be easily calculated:

```
(%i7) solve(r*c*(1-c)=c, c);
```

```
(%o7) [c = (r - 1)/r, c = 0]
```

From our experience with the continuous case, we are tempted to say that any solution of the logistic map is either a sequence bounded by the constant solutions  $x_n = 0$  and  $x_n = (r - 1)/r$  (asymptotically approaching one of these), or a divergent one. But this is not true: The behavior of discrete systems is quite different from that of their continuous counterparts. To be sure, we make a graphical analysis of the solutions. First, we define the *evolution operator* of the system:

```
(%i8) F(r, x) := r*x*(1-x) $
```

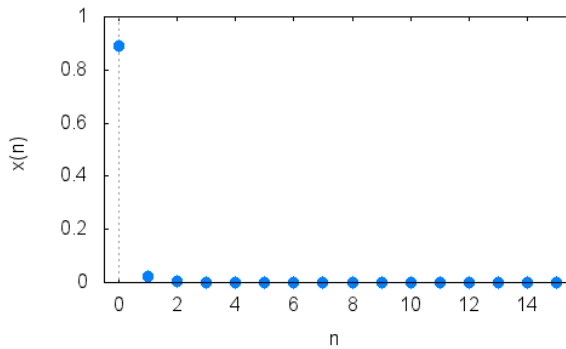
Next, we use the `evolution` command (included in the `dynamics` package) to calculate, for a fixed  $r$ , the  $n + 1$  points  $x_{i+1} = F(r, x_i)$  from  $i = 0$  to  $i = n$ , where the initial value  $x_0$  is given (here we use a pseudo-random initial condition between 0 and 1).

This set of points is a segment of the *orbit* of  $x_0$ . We start with  $r = 0.25$  making 15 iterations:

```
(%i9) load(dynamics) $
```

<sup>2</sup>This sentence deserves a clarification. Actually, "closed" solutions do exist, although defined through implicit functional formulas which are 'hardly useful for computational purposes', see [Bru 98] and references therein. However, there are well-known explicit solutions for particular values of  $k$ , see [http://en.wikipedia.org/wiki/Logistic\\_map#Solution\\_in\\_some\\_cases](http://en.wikipedia.org/wiki/Logistic_map#Solution_in_some_cases).

```
(%i10) set_random_state(make_random_state (true))$
(%i11) x[0]:random(1.0);
(%o11) .8886589561406515
(%i12) evolution(F(0.25, x), x[0], 15, [y, 0, 1]);
```



(%t12)

(%o12)

In this case, the population dies. Indeed, this unlucky fate is independent of the initial condition  $x_0$ , whenever  $0 < r \leq 1$  and  $x_0 \in ]\frac{r-1}{r}, \frac{1}{r}[$  (let us note that this last condition is always fulfilled if  $x_0 \in ]0, 1[$ ). For  $0 < r \leq 1$ , this parameter controls the rate at which population dies, as the reader can check by using other values for  $r$ , such as  $r = 0.123456789$ .

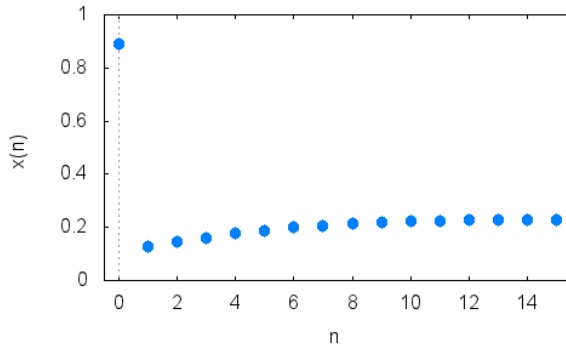
A few experiments will lead us to the following rules for the behavior of the solutions:

- If  $0 < r \leq 1$  and  $x_0 \in ]\frac{r-1}{r}, \frac{1}{r}[$ , the population eventually dies, independently of the initial condition.
- If  $1 < r \leq 2$  and  $x_0 \in ]0, 1[$ , the population quickly approaches the value  $(r - 1)/r$ , independently of the initial condition.
- If  $2 < r \leq 3$  and  $x_0 \in ]0, 1[$ , the population tends again to the value  $(r - 1)/r$ , but in an oscillating way (maybe after a short transient). The rate of convergence is linear, except for  $r = 3$ , where the rate of convergence is quite slow, in fact it is sub-linear.
- If  $3 < r < 1 + \sqrt{6}$  (with  $1 + \sqrt{6} \simeq 3.45$ ), the population oscillates between two values, almost independently of the initial condition. These two values, which depend on  $r$ , are said to have *primary period two*.
- If  $3.45 < r < 3.54$  (approximately), for almost all initial condition the population oscillates between four periodic points.
- If  $r$  increases over 3.54, for almost all initial condition the population oscillates between 8 periodic points, then 16, 32, etc. The size of the intervals formed by the values of the parameter producing oscillations of a given length, becomes small, and the quotient of the size for two consecutive period-doubling intervals approaches the so called *Feigenbaum constant*  $F = 4.669\dots$  This behavior is called a *period-doubling cascade*.

Let us note that when the parameter  $r$  increases, the dynamics differs from that observed in the continuous case.

Let us show some examples. For our initial pseudo-random  $x_0$  and  $r = 1.3$ :

```
(%i13) evolution(F(1.3, x), x[0], 15, [y, 0, 1]);
```

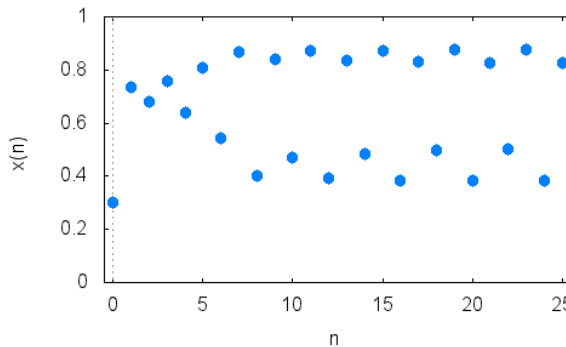


(%t13)

(%o13)

We note that  $(1.3 - 1)/1.3 \simeq 0.23$ . For  $r = 3.5$  and the initial condition  $x_0 = 0.3$ , we observe the transient and four periodic points:

```
(%i14) evolution(F(3.5, x), 0.3, 25, [y, 0, 1]);
```



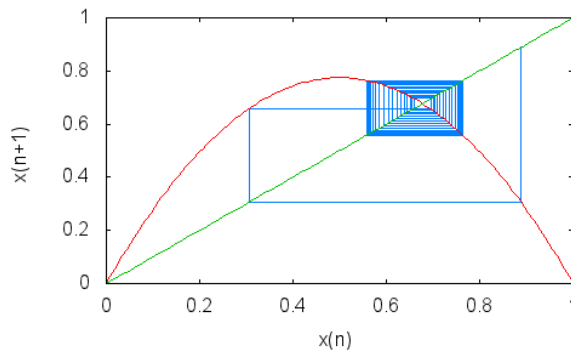
(%t14)

(%o14)

## 4 Staircase diagrams

There is another graphical method to study the phenomenon of the emergence of “attracting” periodic points that we have seen in the last example, based on the particular form of the evolution equation:  $x_{n+1} = f(x_n)$ . For the logistic map ( $f(x) = r \cdot x \cdot (1 - x)$ ), if the coordinates of the points in the plane represent two consecutive values in the orbit of  $x_0$ , the point  $(x, y) = (x_n, x_{n+1})$  can be obtained graphically as the intersection of the vertical line  $x = x_n$  and the graph of the function  $y = f(x)$ . Once  $x_{n+1}$  is known, in order to get  $x_{n+2}$  we just have to let  $x_{n+1}$  play the rôle of  $x_n$  in the previous step. So, we take the intersection of the horizontal line  $y = x_{n+1}$  with the diagonal  $y = x$ . Now  $x_{n+2} = f(x_{n+1})$ , and by iterating the process a predetermined number of times, we get a segment of the orbit of  $x_0$ . Maxima implements this construction with the `staircase` command:

```
(%i15) staircase(F(3.1,x),x[0],250,[x,0,1],[y,0,1]);
```

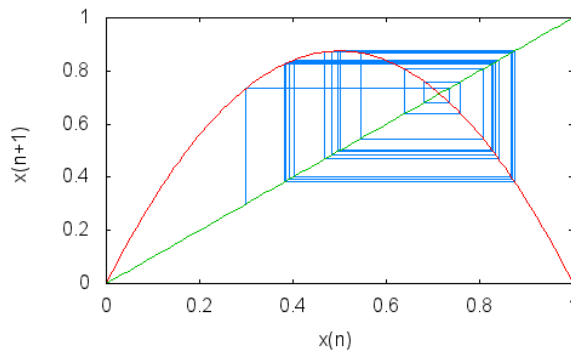


(%t15)

(%o15)

Here we see the orbit oscillating between the periodic points ( $x \simeq 0.56$ ) and ( $x \simeq 0.76$ ). We can also reproduce the behavior of (%o14), where there is a period-four orbit (we make 500 iterations to get a distinguishable path). Let us note that each “rectangle” intersecting the diagonal defines two periodic points:

```
(%i16) staircase(F(3.5,x),x[0],500,[x,0,1],[y,0,1]);
```



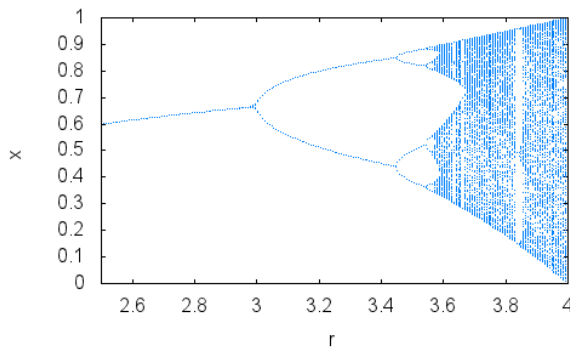
(%t16)

(%o16)

## 5 Bifurcations and chaos

We have seen that the behavior of the orbits for the logistic map depends on the parameter  $r$ . In fact, from  $r = 3$  onwards, the number of periodic points increases in a kind of “period-doubling cascade” (see (%o17) below). It is possible to study the change in the structure of the periodic orbits using the so called *bifurcation diagrams*. In these, we represent the values of the parameter  $r$  on the horizontal axis. For each of these values we mark the values of the corresponding “attracting” periodic points (the use of attracting points is a technical issue, that we will not address here). Maxima implements these plots in its built-in `orbits` command. In the example below, we restrict the plot to the iterations between  $n = 150$  and  $n = 200$ , for a given initial condition  $x_0$ . The range of the values of  $r$  is  $[2.5, 4]$ :

```
(%i17) orbits(F(r,x), x[0], 150, 200, [r, 2.5, 4], [style, dots]);
```

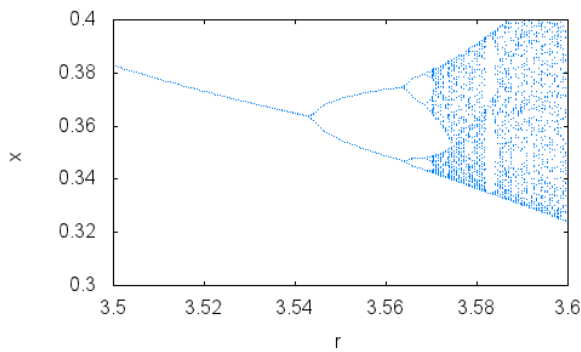


(%t17)

(%o17)

The diagram so obtained displays a special property: it is *self-similar*. This means that after a change of scale, the resulting picture has the same structure that the original one. For example, if we magnify (%o17) centering the zoom on the second bifurcation of the lower branch, we get:

```
(%i18) orbits(F(r,x),x[0],150,200,[r,3.5,3.6],[x,0.3,0.4],
[style,dots]);
```



(%t18)

(%o18)

A measure of the self-similarity of a set is its *fractal dimension* which, unlike the dimension of a vector space, can be a non integer number. In Section 10 we show an example of a set having non integer fractal dimension.

The period-doubling cascade and the appearance of sets with fractal dimension, are usually signs of chaotic behavior. A dynamical system with whose evolution is described by a continuous function  $f : I \subset \mathbb{R} \rightarrow I$  is said to be *chaotic*<sup>3</sup> if:

- (a) It has *sensitivity to initial conditions*: a small variation in the initial condition  $x_0$  of an orbit can produce, in the long run, another orbit far away from the original one.
- (b) *Periodic points are “dense”* in the phase space  $I$  of the system: in general, a set  $S$  is dense in the set  $I \subset \mathbb{R}$  if for any point  $p \in I$  and any  $\varepsilon > 0$ , the interval  $]p - \varepsilon, p + \varepsilon[$  intersects  $S$ .
- (c) It has the *mixing property* (we also say that the system is *topologically transitive*): for any two intervals  $J, K \subset I$  there exist points of  $J$  whose orbits eventually enter  $K$ , that is, there exists an

<sup>3</sup>For simplicity we consider only one-dimensional systems, but the case of multidimensional systems is analogous. For further details on the chaos definition see [Ban 92], [Dev 89], [VB 94]



$n > 1$  such that

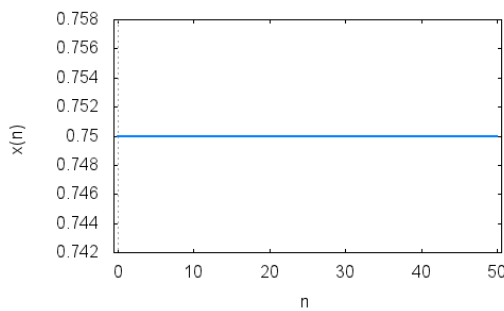
$$f_n(J) \cap K \neq \emptyset,$$

with  $f_n = f \circ \dots \circ f$ . This property is satisfied if and only if the system has an orbit  $\{f_n(x_0) : n \in \mathbb{N} \cup \{0\}\}$  that is dense in  $I$ .<sup>4</sup>

We illustrate each one of these conditions using the logistic map in the examples below.

- (a) Sensitivity to initial conditions. For  $r = 4$ , we consider the initial conditions  $x_0 = \frac{r-1}{r} = \frac{3}{4}$  and  $x_0 + \varepsilon$ , comparing the respective orbits after 50 iterations.

```
(%i19) evolution(F(4,x),3/4,50,[style,[lines,2]]);
```



(%t19)

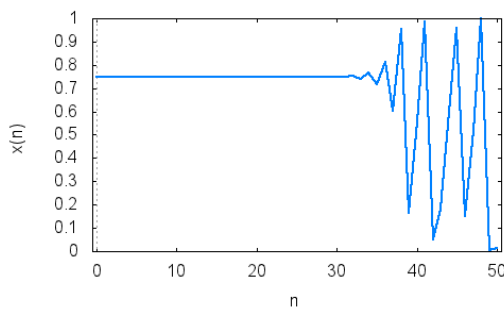
(%o19)

Choosing  $\varepsilon$  small enough (pseudo-random) we see how drastically these orbits diverge (we will see how to measure this divergence in section 7):

```
(%i20) eps:random(0.000000000001);
```

(%o20) 9.87415729237455 10<sup>-13</sup>

```
(%i21) evolution(F(4,x),3/4+eps,50,[style,[lines,2]]);
```



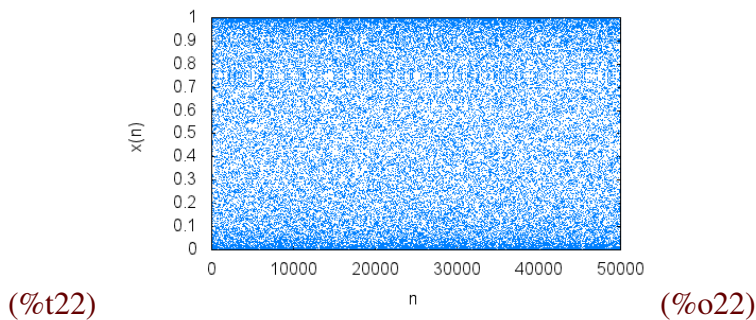
(%t21)

(%o21)

<sup>4</sup>The equivalence is as follows: if the orbit of  $x_0$  is dense in  $I$ , there exists an infinity of points  $x_n = f_n(x_0)$  in any pair of intervals  $J, K$ , so  $f$  is transitive. Conversely, if  $f$  is transitive, for a given interval  $J$  let  $\{A_j\}_{j \in \mathbb{N}}$  be the basis of open intervals with rational end points that are contained in  $J$ ; from the transitivity of  $f$  the family of open sets  $B_j = \cup_{p \in \mathbb{N}} f_{-p}(A_n)$  is dense in  $J$  (since, if  $U \subset J$ , there exists a number  $p$  such that  $f_p(U) \cap A_n \neq \emptyset$ , that is,  $U \cap f_p(A_n) = U \cap B_n \neq \emptyset$ ). From the Baire theorem,  $\cap_{j \in \mathbb{N}} B_j$  is dense in  $J$  too. But this intersection consists of points with dense orbits, since if  $x \in \cap_{j \in \mathbb{N}} B_j$ , for each  $n \in \mathbb{N}$  there is a  $p_n \in \mathbb{N}$  such that  $f_{p_n} \in A_n$ . The sequence  $\{f_{p_n}\}_{n \in \mathbb{N}}$  has points in each  $A_n$  and therefore is dense.

- (b) The set of periodic points is dense in the interval  $I = [0, 1]$ . This is clearly seen in (%o17), where the right side of the unit square is “filled” with periodic points.
- (c) There exists a dense orbit. Again, we can exemplify this behavior with  $k = 4$  and the initial condition  $x_0 = 0.2$  after 50,000 iterations. Its orbit fills the phase space:

```
(%i22) evolution(F(4,x),0.2,50000,[style,dots]);
```



## 6 The Lorenz attractor

As mentioned, the monotonicity of the solutions restricts the presence of chaos in one-dimensional continuous systems. However, for higher order equations (at least of second order with a time dependent non-homogeneous term, as in the Duffing equation shown later) or in systems of first order equations (at least three equations, even in the case of an autonomous system, but necessarily non linear<sup>5</sup>), it is possible to have chaotic behavior. As an example, we consider the Lorenz system, the first system in which the properties of chaotic dynamics were explicitly seen, in 1963 [Lor 63]:

$$\begin{cases} \dot{x} = 10y - 10x \\ \dot{y} = -xz + 28x - y \\ \dot{z} = xy - 8z/3 \end{cases}$$

To numerically study the orbits, we use the 4th-order Runge-Kutta method. The following commands compute the orbit with initial condition  $(-8, 8, 27)$  after 50 time units (with a step of 0.01):

```
(%i23) numer:false$
```

```
(%i24) latractor: [10*y-10*x, -x*z+28*x-y, x*y-8*z/3]$
```

```
(%i25) linitial: [-8, 8, 27]$
```

<sup>5</sup>We have already seen that one dimensional continuous systems can not display chaotic behavior (nor even periodic behavior). For autonomous systems in the plane (or second order differential equations with time-independent coefficients), the uniqueness of solutions together with the Poincaré-Bendixson theorem rule out the existence of chaos. The problem of setting the “minimal conditions” to ensure the existence of chaos is still an open question. An interesting report on this topic is [SL 00].

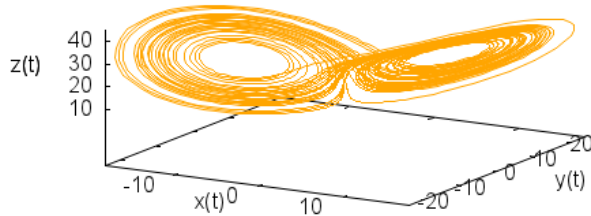
```
(%i26) lsolution:rk(latractor,[x,y,z],linitial,[t,0,50,0.01])$
```

```
(%i27) lpoints: map(lambda([x], rest(x)), lsolution)$
```

```
(%i28) load(draw)$
```

The following command needs that the package `draw` be previously loaded (do `load(draw)` otherwise). Note that the output is actually a separate Gnuplot window, so (by pressing the left button of the mouse and dragging around) the reader can rotate the whole picture to better appreciate its details.

```
(%i29) draw3d(point_type=none,points_joined=true,color=orange,
             xlabel="x(t)",ylabel="y(t)",zlabel="z(t)",
             xtics=10,ytics=10,ztics=10,points(lpoints));
```



(%t29)

(%o29)

The phrase “order within chaos” is frequently found. It means that in some systems, even though the orbits can appear unpredictable and very complex, there is some regularity. For example, in the Lorenz system, the sensitivity to initial conditions prevents us from making accurate predictions, but we can assert that in the long run the orbit will be restricted to a *bounded* region of the space called an *attractor*<sup>6</sup>. This seems clear in figure (%o29), but we can prove it analytically with the aid of Maxima. Let us consider a general Lorenz system, where  $\sigma$ ,  $\rho$  and  $\beta$  are parameters:

$$\begin{cases} \dot{x} = \sigma(y - x) \\ \dot{y} = x(\rho - z) - y, \\ \dot{z} = xy - \beta z \end{cases} \quad (6)$$

```
(%i30) remfunction(x)$
```

```
(%i31) gradef(x(t),%sigma*(y(t)-x(t)))$
```

```
(%i32) gradef(y(t),x(t)*(%rho -z(t))-y(t))$
```

```
(%i33) gradef(z(t),x(t)*y(t)-%beta *z(t))$
```

<sup>6</sup>Attractors are classified as *strange* and *no strange* if they have fractal or integer dimension, respectively. In section 10 we will see how to compute the fractal dimension of an attractor.

We consider the set of those points belonging to an orbit that, for a given instant  $t$ , lie on the sphere with center  $(0, 0, \sigma + \rho)$  and radius  $R(\sigma + \rho)$ , with  $R$  a constant to be determined:

$$\text{\%i34 } x(t)^2 + y(t)^2 + (z(t) - (\sigma + \rho))^2 = R^2 (\sigma + \rho)^2;$$

$$\text{\%o34 } (z(t) - \sigma - \rho)^2 + y(t)^2 + x(t)^2 = (\sigma + \rho)^2 R^2$$

Differentiating equation (o34) (using the chain rule) and substituting (6), we get:

$$\text{\%i35 } \text{facsum}(\text{diff}(\%, t), x(t), y(t), z(t));$$

$$\text{\%o35 } -2\beta z(t)^2 + 2\beta(\sigma + \rho)z(t) - 2y(t)^2 - 2\sigma x(t)^2 = 0$$

This looks like an ellipsoid, and we can prove it is one by “completing the squares”:

$$\text{\%i36 } \text{factor}(\text{solve}((\text{sqrt}(2*\beta)*z(t)-u)^2-u^2=2*\beta*z(t)^2-2*\beta*(\sigma+\rho)*z(t), u));$$

$$\text{\%o36 } [u = \frac{\sqrt{\beta}(\sigma + \rho)}{\sqrt{2}}]$$

So, the points  $(x(t), y(t), z(t))$  should satisfy

$$-2\sigma x(t)^2 - 2y(t)^2 - 2\beta \left( z(t) - \frac{\sigma + \rho}{2} \right)^2 + \beta \frac{(\sigma + \rho)^2}{2} = 0,$$

which is the equation of an ellipsoid with center  $(0, 0, (\sigma + \rho)/2)$ . The semi-axis along directions OX, OY and OZ are, respectively:

$$\text{\%i37 } \text{solve}(1/a = \text{sqrt}(2*\sigma/(\beta*(\sigma + \rho)^2/2)), a);$$

$$\text{\%o37 } [a = \frac{|\sigma + \rho|}{2\sqrt{\frac{\sigma}{\beta}}}]$$

$$\text{\%i38 } \text{solve}(1/b = \text{sqrt}(2/(\beta*(\sigma + \rho)^2/2)), b);$$

$$\text{\%o38 } [b = \frac{|\sigma + \rho|}{2\sqrt{\frac{1}{\beta}}}]$$

$$\text{\%i39 } \text{solve}(1/c = \text{sqrt}(2*\beta/(\beta*(\sigma + \rho)^2/2)), c);$$

$$\text{\%o39 } [c = \frac{|\sigma + \rho|}{2}]$$

These lengths depend only on the parameters  $\sigma$ ,  $\rho$  and  $\beta$ . Thus, taking  $R$  large enough (explicitly  $R > \frac{\sigma + \rho}{2} \max\{2, \sqrt{1 + \beta}, \sqrt{1 + \beta/\sigma}\}$ ), we obtain a sphere containing the initial conditions (here  $(x(0), y(0), z(0)) = (0, 0, 0)$  for simplicity) and such that when they evolve in time, all orbits remain inside this sphere.

As already pointed, the presence of attractors having fractal structure (strange attractors) is a signal for chaotic dynamics, but it is neither a necessary nor a sufficient condition. There are attractors derived from chaotic dynamics that are not strange (for example, in the logistic equation with  $k = 4$  the attractor is the whole interval  $[0, 1]$ ) and there are non-chaotic systems displaying attractors with fractal dimension (see [GOPY 84]).

## 7 Lyapunov exponents

We have seen that a feature of chaotic dynamical systems is the sensitivity to initial conditions. Lyapunov exponents are introduced to qualitatively measure this property. The idea is quite simple: By choosing a fixed orbit, we compare it with other orbits having close initial conditions, and then measure the distance between them with a factor of the form  $\exp(\lambda t)$ . Here  $\lambda$  is the Lyapunov exponent. If it is positive, the orbits diverge asymptotically and there is sensitivity to initial conditions (greater when  $\lambda$  is large). If the exponent  $\lambda$  is negative, the orbits must asymptotically approximate each other and there is no sensitivity to initial conditions. A value of  $\lambda = 0$  indicates that the orbit considered is a stable fixed point.

Let us consider, for simplicity, a one-dimensional dynamical system

$$x_{n+1} = f(x_n)$$

with  $f : \mathbb{R} \rightarrow \mathbb{R}$  derivable except for a finite number of points (this is the case, for example, of the logistic map). We consider the orbits starting at the points  $x_0$  and  $x_0 + \varepsilon$ . After  $N$  iterations the points on the orbits will be, respectively,  $f_N(x_0)$  and  $f_N(x_0 + \varepsilon)$ , where  $f_N = f \circ \dots \circ f$ . So, after  $N$  iterations the distance between the orbits is  $|f_N(x_0 + \varepsilon) - f_N(x_0)|$ , which can be written in the form

$$|f_N(x_0 + \varepsilon) - f_N(x_0)| = \varepsilon e^{N\lambda(x_0)}$$

for a suitable  $\lambda(x_0) \in \mathbb{R}$  that is called the *Lyapunov exponent* at the point  $x_0$ . Is in this way that the Lyapunov exponent gives a measure of how the initial separation  $\varepsilon$  is amplified when the orbits evolve.

Solving the last equation for  $\lambda(x_0)$ :

```
(%i40) assume(%epsilon>0,N>0)$
```

```
(%i41) remvalue(x[0])$
```

```
(%i42) solve(abs(f[N](x[0]+%epsilon)-f[N](x[0]))=
           %epsilon*exp(N*%lambda),%lambda);
```

```
(%o42) [ $\lambda = \frac{\log\left(\frac{|f_N(\varepsilon+x_0)-f_N(x_0)|}{\varepsilon}\right)}{N}$ ]
```

and taking limits with  $\varepsilon \rightarrow 0$  and  $N \rightarrow \infty$  (taking into account the continuity of log and the definition of the derivative), we have:

$$\lambda(x_0) = \lim_{N \rightarrow \infty} \frac{1}{N} \log \left| \frac{df_N}{dx}(x_0) \right|.$$

Just to show the symbolic capabilities of Maxima, we will use it to evaluate the derivative in the last expression. First, we declare the derivative of the function  $f$  by giving it a name (the usual  $f'$ ):

```
(%i43) gradef(f(x), f'(x))$
```

Then, we set the values  $x_1 = f(x_0)$  and  $f'(x_0) = \frac{df}{dx}(x_0)$ :

```
(%i44) atvalue(f(y), y=x[0], x[1])$
```

```
(%i45) atvalue('diff(f(y), y), y=x[0], f'(x[0]))$
```

Now, we compute the derivative of  $f_2 = f \circ f$  at  $x_0$  (Maxima knows the chain rule):

```
(%i46) at(diff(f(f(x)), x), x=x[0]);
```

```
(%o46) f'(x_0) f'(x_1)
```

It is then easy to prove (by induction) that

$$\frac{df_N}{dx}(x_0) = \prod_{i=0}^{N-1} f'(x_i),$$

so

$$\lambda(x_0) = \lim_{N \rightarrow \infty} \frac{1}{N} \log \left| \prod_{i=0}^{N-1} f'(x_i) \right| = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=0}^{N-1} \log |f'(x_i)|. \quad (7)$$

Using equation (7) we can compute the Lyapunov exponents numerically, approximating the limit by a finite sum for  $N$  big enough. As an example, consider the logistic map with parameter  $r = 3$ . In (%i8) we defined the function  $F(r, x)$ , and now we define its derivative:

```
(%i48) define(dF(r, x), diff(F(r, x), x))$
```

Then we set the parameter value:

```
(%i49) r:3$
```

Here we set the number of iterations:

```
(%i50) maxiter:50000$
```

and the initial condition<sup>7</sup>:

```
(%i51) x[1]:random(1.0);
```

```
(%o51) .4823905248516196
```

Now, we construct the orbit from the initial condition:

---

<sup>7</sup>We denote it by  $x_1$  instead of  $x_0$  because we are going to make a list with the iterations, with the initial condition as the first element and, for Maxima, lists are enumerated starting with the index 1.

```
(%i52) for j:2 thru maxiter do (x[j]:F(r,x[j-1]))$
```

Finally, we estimate the corresponding Lyapunov exponent:

```
(%i53) L:0$
```

```
(%i54) for j:1 thru maxiter do L:L+log(abs(dF(r,x[j])))$
```

```
(%i55) '%lambda=L/maxiter;
```

```
(%o55)  $\lambda = -3.145501884323275 \cdot 10^{-4}$ 
```

For this case,  $r = 3$ , we get  $\lambda \simeq 0$ . The reader can experimentally check that for the logistic map  $\lambda(x_0) = \lambda$ , that is, the Lyapunov exponent does not depend on the initial condition  $x_0 \in ]0, 1[$ , it only depends on the parameter  $r$ . For values of  $r$  lesser than (approximately) 3.569945, we have that  $\lambda \leq 0$ , but if  $r > 3.569945$ , the behavior of  $\lambda$  as function of  $r$  becomes quite complicated.

## 8 The Duffing oscillator

We have only considered systems described by first-order equations. However, most of the physical systems of interest are described by second order equations. In this section we present an example of these, the Duffing oscillator.

Let us consider a general Newtonian system, described by a second-order differential equation:

$$\ddot{x}(t) = F(x(t), \dot{x}(t))$$

If  $F = F(x) = -dV/dx$ , we say that the system is conservative and the function  $V = V(x)$  is called the *potential*. For example, the system

$$\ddot{x}(t) = -\frac{1}{4}x^3(t) + x(t)$$

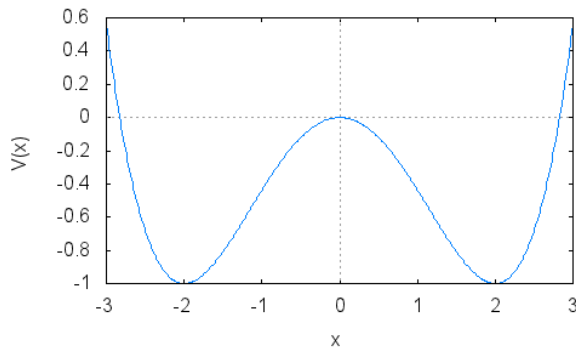
is conservative and its potential is given by

```
(%i56) 'V(x)=integrate(x^3/4-x,x);
```

```
(%o56)  $V(x) = \frac{x^4}{16} - \frac{x^2}{2}$ 
```

The graph of this function is:

```
(%i57) wxplot2d(integrate(x^3/4-x,x), [x,-3,3], [ylabel, "V(x)"]);
```



(%t57)

(%o57)

We observe an unstable equilibrium at  $x = 0$  and two stable equilibria at  $x = -2$  and  $x = 2$ . Now let us modify the system by including a velocity-dependent *damping term*:

$$\ddot{x}(t) = -\frac{1}{4}x^3(t) + x(t) - \frac{1}{10}\dot{x}(t)$$

The system is no longer conservative, since the damping term dissipates energy in the form of heat. As a result, the oscillations made by the system decrease in amplitude until there is no motion. We illustrate this by computing the trajectory with initial conditions  $x(0) = 3$ ,  $\dot{x}(0) = 10$ :

```
(%i58) duff: [v, -v/10+x-x^3/4] $
```

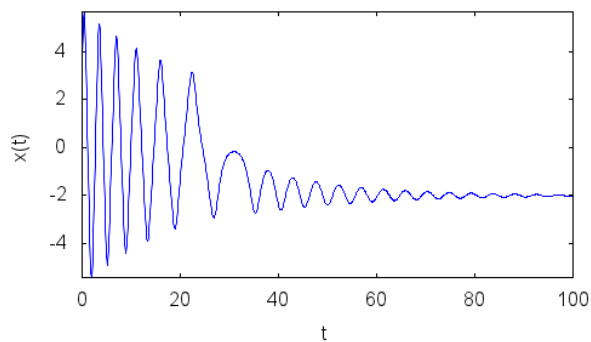
```
(%i59) icduff: [3, 10] $
```

```
(%i60) solduff: rk(duff, [x, v], icduff, [t, 0, 100, 0.1]) $
```

```
(%i61) curveduff: map(lambda ([x], rest (x, -1)), solduff) $
```

```
(%i62) pointsduff: map(lambda ([x], rest (x)), solduff) $
```

```
(%i63) wxdraw2d(point_type=none, points_joined=true,
  xlabel="t", ylabel="x(t)", points(curveduff));
```



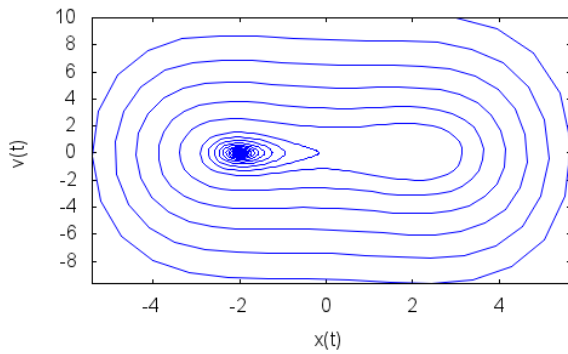
(%t63)

(%o63)

We can also represent the dynamics of the system in the plane  $(x, v = dx/dt)$  (*the phase plane*):

```
(%i64) wxdraw2d(point_type=none, points_joined=true,
  xlabel="x(t)", ylabel="v(t)", points(pointsduff));
```





(%t64)

(%o64)

Note that this result coincides with (%o63), where it is clear that the system evolves to the stable equilibrium  $x = -2$ , oscillating around this value with decreasing amplitude until it stops.

The *Duffing oscillator* is obtained including in the previous system and oscillating force (the *forcing term*)  $\sin(\omega t)$ :

$$\ddot{x}(t) = -\frac{1}{4}x^3(t) + x(t) - \frac{1}{10}\dot{x}(t) + \sin(\omega t)$$

The effect of the forcing term is to restore the energy dissipated by the damping term. If the frequency of the forcing term is the appropriated one, the system will oscillate in a stable way, but if there is no synchronization with respect to the non-forced system, a chaotic behavior will appear. For example, for  $\omega = 1$  starting from the unstable equilibrium ( $x = 0, v = 0$ ), after a short transient the system will evolve to a stable oscillating regime (a *limit cycle*):

```
(%i65) duffing:[v,-v/10+x-x^3/4+sin(t)]$
```

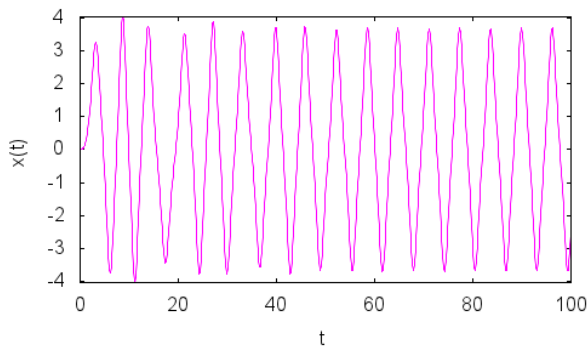
```
(%i66) icduffing:[0,0]$
```

```
(%i67) sduffing:rk(duffing,[x,v],icduffing,[t,0,100,0.1])$
```

```
(%i68) cduffing:map(lambda([x],rest(x,-1)),sduffing)$
```

```
(%i69) pduffing:map(lambda([x],rest(x)),sduffing)$
```

```
(%i70) wxdraw2d(point_type=None,points_joined=True,color=magenta,
xlabel="t",ylabel="x(t)",points(cduffing));
```

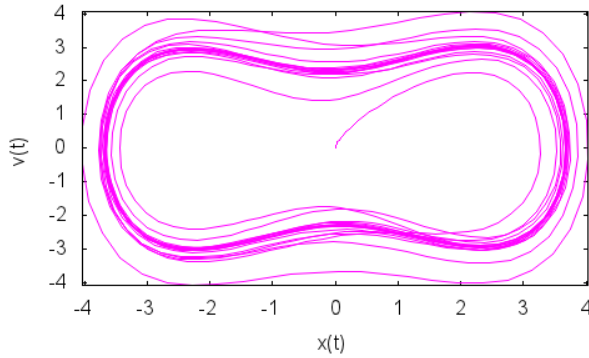


(%t70)

(%o70)

The corresponding phase plane diagram shows that the system tends to a limit cycle, oscillating around the two stable equilibria, instead of approaching just one of them, as happens for the non-forced case:

```
(%i71) wxdraw2d(point_type=None, points_joined=True, color=magenta,
               xlabel="x(t)", ylabel="v(t)", points(pduffing));
```



(%t71)

(%o71)

The behaviors considered do not exhaust the possibilities for the dynamics of the Duffing oscillator. In the next section we will see how to analyze the chaotic case.

## 9 Poincaré sections

In general, for a second order differential equation of the form

$$\ddot{x}(t) = F(t, x, \dot{x})$$

or, equivalently (introducing the variables  $u(t) = x(t)$ ,  $v(t) = \dot{x}(t)$ ) for the first order system

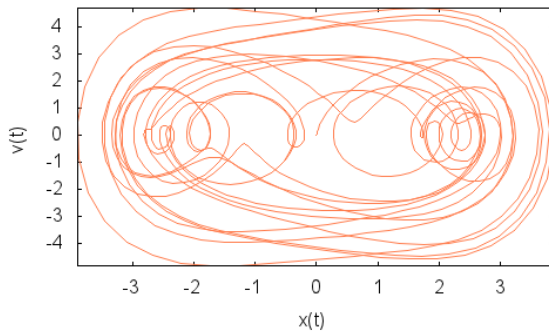
$$\begin{cases} \frac{du}{dt} = v \\ \frac{dv}{dt} = F(t, u, v) \end{cases},$$

the trajectories in the phase plane can intersect themselves and the resulting diagram becomes quite complicated. Let us note that this is not a contradiction with the theorem on the uniqueness of solutions, because here we have a non autonomous differential equation and the phase plane is obtained projecting from the phase space  $(t, x, v) \mapsto (x, v)$ . The theorem applies, when it is the case, in the variables  $(t, x, v)$ . For example, for the Duffing oscillator with forcing term  $2.5 \sin(2t)$ , we get the following phase diagram:

```
(%i72) duffing1:[v, -v/10+x-x^3/4+2.5*sin(2*t)]$
```

```
(%i73) icduffing1:[0,0]$
```

```
(%i74) sduffing1:rk(duffing1, [x,v], icduffing1, [t,0,100,0.1])$
(%i75) pduffing1:map(lambda([x], rest(x)), sduffing1)$
(%i76) wxdraw2d(point_type=none, points_joined=true, color=coral,
    xlabel="x(t)", ylabel="v(t)", points(pduffing1));
```



(%t76)

(%o76)

Poincaré introduced a very useful technique to study the dynamics in these situations. It consists in reckoning a trajectory in the phase space  $(x, v, t)$  by using hyperplanes equally spaced in time as “check points”, and then projecting the resulting points on the plane  $(x, v)$ :

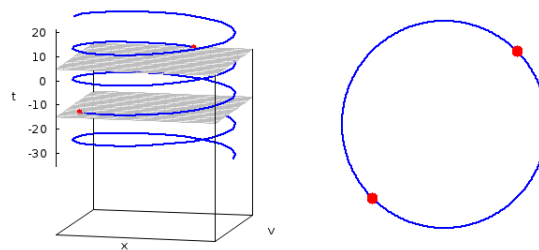


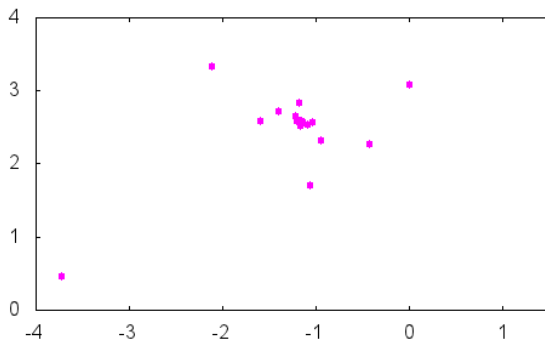
Figure 1: Poincaré sections

Of course, if the time interval between these hyperplanes (called the *Poincaré stroboscopic sections*) is  $T$  and the system is periodic with period  $T$ , we will see just a point on the phase plane. In the figure we see a quasi-periodic system: on the phase plane the trajectories are circumferences, but several points can appear in the sections projection, showing that the system oscillates between several equilibria with a frequency different from the time interval between the Poincaré sections.

When the dynamics is chaotic, the diagram obtained by projecting the Poincaré sections can be really complicated, usually displaying fractal structure.

Let us compute the Poincaré sections for the case of the Duffing oscillator. First, we consider the case of a period  $T = \frac{2\pi}{\omega}$  corresponding to the frequency  $\omega = 1$  (that is, the forcing term is  $\sin(t)$ ). We register the points in the orbit at time intervals of length  $T = \frac{2\pi}{\omega} = 2\pi$ :

```
(%i77) miter:25$
(%i78) %tau:bfloat(%pi)$
(%i79) sduffing2:rk(duffing,[x,v],icduffing,
    [t,0,miter*2*%tau,%tau/30])$
(%i80) pduffing2:create_list(sduffing2[i],i,makelist(i*60,i,1,miter))$
(%i81) poinduffing2:map(lambda([x],rest(x)),
    makelist(pduffing2[i],i,1,miter))$
(%i82) wxdraw2d(point_type=filled_circle,color=magenta,xtics=1,
    ytics=1,xrange=[-4,1.5],yrange=[0,4],points(poinduffing2));
```

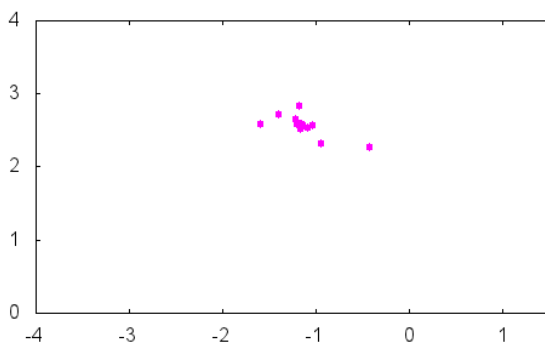


(%t82)

(%o82)

With the exception of some sparse points representing the transient, we see that all the points tend to a cluster (oscillations occur close to a limit cycle). To show that the most distant points corresponds to the transient, we eliminate, for example, the first four samples (corresponding to the sections at  $t = 0, 2\pi, 4\pi$  and  $6\pi$ ):

```
(%i83) transient:map(lambda([x],rest(x)),
    makelist(pduffing2[i],i,5,miter))$
(%i84) wxdraw2d(point_type=filled_circle,color=magenta,xtics=1,
    ytics=1,xrange=[-4,1.5],yrange=[0,4],points(transient));
```

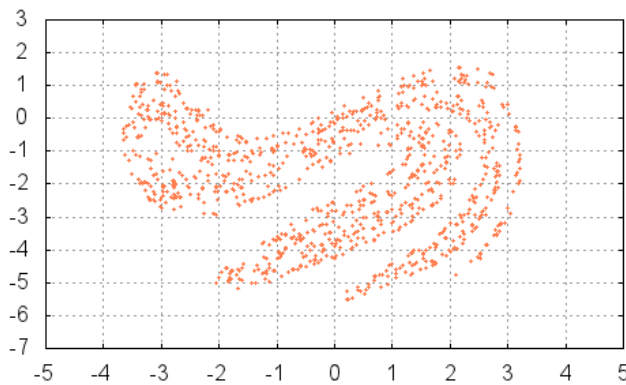


(%t84)

(%o84)

Of course, increasing the number of samples we get a more illustrative picture. But the computational effort is not worth in this case. There are more interesting situations, as it is the case of the Duffing oscillator with forcing term  $2.5 \sin(2t)$ . Now the period corresponds to the frequency  $\omega = 2$ , that is,  $T = \frac{2\pi}{2} = \pi$ . We let the system evolve for a long time in order to get a large sample via the Poincaré sections:

```
(%i85) duffing3:[v,-v/10+x-x^3/4+2.5*sin(2*t)]$
(%i86) icduffing3:[0,0]$
(%i87) maxiter:1000$
(%i88) sduffing3:rk(duffing3,[x,v],icduffing3,
    [t,0,maxiter*%tau,%tau/30])$
(%i89) pduffing3:
    create_list(sduffing3[i],i,makelist(i*30,i,1,maxiter))$
(%i90) poinduffing3:map(lambda([x],rest(x)),
    makelist(pduffing3[i],i,1,maxiter))$
(%i91) wxdraw2d(point_size=0.3,point_type=circle,
    xrange=[-5,5],yrange=[-7,3],xticks=1,yticks=1,
    color=coral,points(poinduffing3),grid=true);
```



(%t91)

(%o91)

## 10 Fractal dimension

As we have pointed out in previous sections, the appearance of *strange attractors* is a usual feature of chaotic systems. These are sets around which the orbits move asymptotically (thus justifying the name of attractor) and frequently have fractal dimension. We will introduce a particular notion of fractal dimension (there are other possibilities) in the case of a two dimensional dynamical system, so the attracting set  $A$  will be a subset of the plane. An example of this situation is given by the Duffing

oscillator. The *box-counting dimension* of  $A$  is defined as follows: we start with a square of side length  $l$  enclosing  $A$ . In the next step, we divide the square in  $r$  sub-squares, each having side length  $l/r$ , and so on. In the  $k$ -step we have a grid with  $r^{k-1}$  squares, each with side length  $\delta_k = \frac{l}{r^{k-1}}$ , that cover the set  $A$  as in (91). For each step  $k$ , let  $N(k)$  be the number of squares containing at least some point of  $A$ .

Let us observe that if  $A$  were a smooth curve (and, therefore, homeomorphic to a segment of  $\mathbb{R}$ , to which we would assign a topological dimension 1), for a small enough value of the side of the squares each of them would contain a piece of the curve with length as close as we wish to the side length of the square. So, in the case of a smooth curve, we would have

$$\lim_{k \rightarrow \infty} N(k) \cdot \delta_k = L,$$

with  $L$  the length of the curve. In this case we say that the scale of  $N(k)$  goes as  $\delta_k^{-1}$ , that is,  $N(k) \simeq \delta_k^{-1}$ . Analogously, if  $A$  were a measurable set on the plane (a set having an area and, thus, of dimension 2), we would have  $N(k) \simeq \delta_k^{-2}$ . Let us note that in these cases, that can be called “regular” ones, the exponent  $D$  in the relation  $N(k) \simeq \delta_k^{-D}$  equals the topological dimension of the set. Then, a *fractal set* can be defined as a set for which a relation of the kind  $N(k) \simeq \delta_k^{-D}$  holds, with  $D$  a positive real number (not necessarily an integer). In general, we call this number  $D$  the *box counting dimension* of the set, and remark that it satisfies

$$D = \lim_{k \rightarrow \infty} \frac{\log N(k)}{-\log \delta_k}.$$

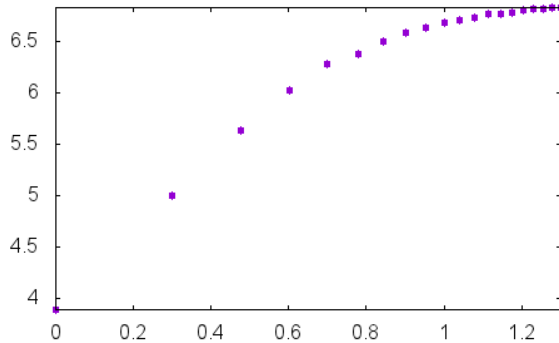
This is a constructive definition that suggests how to perform the calculation of the box-counting dimension for a given set. Explicitly, we can plot  $\log N(k)$  versus  $-\log \delta_k$ , then we can fit the data and finally estimate  $D$  as the slope of the regression line. As an example, let us calculate the dimension of the Duffing attractor (85) with this procedure. We note that the Duffing attractor is contained in the box  $[-5, 5] \times [-7, 3]$ . We divide this box in  $10 \times 10, 20 \times 20, \dots, 200 \times 200$  boxes, counting how many points are inside them. As a detail, we first “normalize” the coordinates in such a way that they lie in the box  $[0, 10] \times [0, 10]$ . The algorithm starts with a matrix filled with zeros and changes the scale at successive steps multiplying these lengths by 2, 3, ..., 20:

```
(%i92) resolution:20$
(%i93) for n:1 thru resolution do
    (Z[n]:substpart(" ", zeromatrix(10*n, 10*n), 0),
    boxcount[n]:0,
    for k:1 thru maxiter do (
        ix:floor(n*(poinduffing3[k][1]+5)),
        iy:floor(n*(poinduffing3[k][2]+7)),
        if is(Z[n][ix][iy]=0) then
            (Z[n][ix][iy]:1, boxcount[n]:boxcount[n]+1)))$
```

Here is the plot of the pairs of numbers obtained.

```
(%i94) fitdim:makelist(
    [log(n)/log(10), log(boxcount[n])], n, 1, resolution), numer$
```

```
(%i95) wxdraw2d(point_size=1,point_type=filled_circle,
color=dark_violet,points(fitdim));
```



(%t95)

(%o95)

The first values are “outliers” (since the boxes are in fact too big), so we can safely discard them. In the same way, for the last values the boxes are too small, counting nearly one point per box, so again we can discard them. This situation, of data that gives a wrong contribution to the estimation of the box-counting dimension, is a well known problem discussed in the literature (see [Kli 94]); in fact, there are several algorithms aimed to automatically selecting the most representative data, but for our (limited) purposes, we just use the visual information in the graph (%o95). According with this, it seems a good choice to eliminate the first 7 values and the two last ones to make the fitting. The result is:

```
(%i96) fitdimension:rest(rest(fitdim,7),-2)$
```

```
(%i97) load(stats)$
```

```
(%i98) model:linear_regression(fitdimension)$
```

```
(%i99) fiteqn:take_inference('b_estimation,model);
```

```
(%o99) [6.021481801936183,.6534032420835236]
```

```
(%i100) fractaldimension:second(%);
```

```
(%o100) .6534032420835236
```

Since our approximation is very rough, we can round-off the value so obtained. Therefore, we can say that the fractal dimension of the Duffing attractor is approximately 0.65 (between 0 and 1, as it should be expected).

## 11 Conclusions

Let us finish with some comments about our experiences using of this material. When asking the students for their impressions about the course, we were surprised by the (practical) absence of CAS-related complains: its usage was considered as “very easy”. That was a surprise because we ourselves

found it difficult to figure out the actual implementation for some questions arising in the development of the work. We even had a case in which we wanted to do *in situ* modifications to the commands and blundered into a dead end, until the way out was found by a clever student by digging in the documentation. No doubt, this kind of interaction with the students at the calculus lab, the mutual learning environment, is one of the major benefits of this approach. Aside from this, the motivation of the techniques by means of physical models was well received (we took the opportunity to stress the close relationship of Mathematics to other disciplines, such as Physics and Engineering).

Maybe, the lack of criticism about the CAS was due to the fact that it was the first exposure to one of them for many of our students. Surely it was a rewarding experience for them (we were generating the same figures appearing in the textbooks!), but we teachers noticed some drawbacks of Maxima. The main one is related to its numerical capabilities: Maxima is way too slow when compared with commercial software, such as Maple or Mathematica, especially when running on a laptop (which we used when teaching outside our campus). But we think that its free and open-source character, and the ease of its syntax, compensates for that. The other major concern was the quality of the graphics, which clearly could be improved. Anyway, for the limited purposes of visualization of the mathematical phenomena leading to chaos, we found Gnuplot acceptable.

As mentioned in the introduction, some topics were not touched upon in the lectures or in the practical sessions. Those theoretical results depending on the use of irrational numbers are an example, other were assigned as “research projects” to the students. We finish these brief comments with a partial list of possible assignments:

1. The techniques explained in the notes can be applied to any dynamical system of the form  $\frac{dx}{dt} = F(x, t)$  or  $x_{n+1} = F(r, x_n)$ . Seek in the literature other standard systems and study them. For example,  $x_{n+1} = k + x_n^2$ , or  $x_{n+1} = \cos(k \cdot x_n)$ , etc.
2. Get a copy of K. Briggs paper [Bri 89] (it is freely available in his web page). Study it and write a Maxima script for computing the value of Feigenbaum’s constant.
3. It is interesting to study the behavior of the Lyapunov exponents as functions of the logistic map’s parameter, that is, the function  $\lambda = \lambda(k)$ . Write the necessary code to plot in the same graph  $\lambda(k)$  and the bifurcations of the logistic map (it is a very popular picture). What could be deduced from the result?.
4. Study the orbits and attractors of the Rössler system. Compare them to those of the Lorentz system.
5. Determine the attractor of other systems that are described by a second order differential equation, in the same way that was done for the Duffing oscillator (for example, read about the elastic pendulum, also known as the spring pendulum). Then, calculate its fractal dimension.
6. Generalize the algorithm for computing the box-counting dimension to the 3D–case. Compute the fractal dimension of the Lorentz and Rössler attractors.

**Acknowledgements:** This work was partially supported by a grant from the Mexican Consejo Nacional de Ciencia y Tecnología, project code CB-2012 179115 (JAV).



## References

- [Ban 92] J. Banks et al.: *On Devaney's definition of chaos*. The American Mathematical Monthly **99** 4 (1992) 332-334.
- [Bri 89] K. Briggs: *How to calculate the Feigenbaum's constants in your PC*. Australian Mathematical Society Gazette **16** n.4 (1989) 89–92.
- [Bru 98] M. Bruschi: *Determinantal solutions of the logistic maps*. Journal of Physics A: Mathematical and General **31** (1998) L153–155.
- [Dev 89] R. L. Devaney: *An introduction to chaotic dynamical systems*. Addison-Wesley (1989).
- [GOPY 84] C. Grebogi, E. Ott, S. Pelikan, J. A. Yorke: *Strange attractors that are not chaotic*. Physica D **13** (1984) 261-268.
- [Kli 94] B. Klinkenberg: *A review of methods used to determine the fractal dimension of linear features*. Mathematical Geology **26** 1 (1994) 23-46.
- [Lor 63] E. N. Lorenz: *Deterministic non-periodic flow*. Journal of the atmospheric sciences **20** (1963) 130-141.
- [May 76] R. M. May: *Simple mathematical models with very complicated dynamics*. Nature **261** (1976) 459-467.
- [Max 11] Maxima.sourceforge.net. Maxima, a Computer Algebra System. Version 5.24.0 (2011). <http://maxima.sourceforge.net/>.
- [SL 00] J. C. Sprott, S. J. Linz: *Algebraically simple chaotic flows*. International Journal of chaos theory and applications **5** 2 (2000) 3-22.
- [VB 94] M. Vellekoop, R. Berglund: *On intervals, transitivity = chaos*. The American Mathematical Monthly **101** 4 (1994) 353-355.