NONLINEAR SYSTEMS, CHAOS AND CONTROL IN ENGINEERING

ASSIGNMENTS

Student: YI QIANG JI ZHANG

Professor: DR. CRISTINA MASOLLER, DR. ANTONIO PONS

Aerospace Engineering
The School of Industrial, Aerospace and Audiovisual Engineering of Terrassa
Polytechnic University of Catalonia | BarcelonaTech

April 22, 2021

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa

Contents

1 Logistic Map

The logistic map

$$x(i+1) = rx(i)[1 - x(i)] \tag{1}$$

a) For $r = 3.5$, calculate the 10 values, $x(i)$ with $i = i...10$, that follow $x(0) = 0.2$. Plot $x(i)$ vs i.

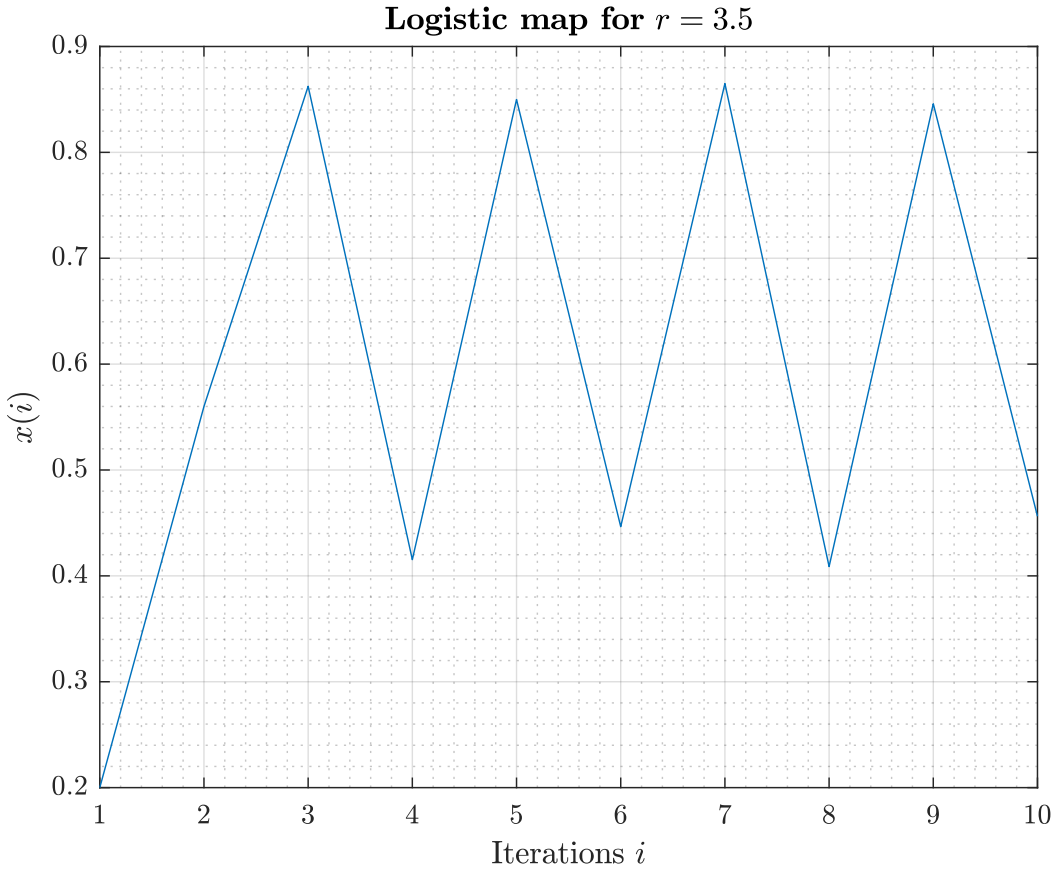

Figure 1 Logistic map for $r = 3.5$. Source Own.

The above picture shows the logistic map for $r = 3.5$. Notice how the the function x does not remain constant as the number of iterations i the value of x is fluctuating. If we consider r as the growth rate, and expression (1) as a function that represents a population, if the growth rate r decreases, the equilibrium population will decrease as well, and if r goes to 1, then, eventually the population will go down to 0, that is, extinct. Once $r > 3$, no matter how many times it iterates the equation it never settles down to a constant value. Instead, it oscillates back and forward between two values.

b) Plot the bifurcation diagram for r in the interval (3.5 , 4).

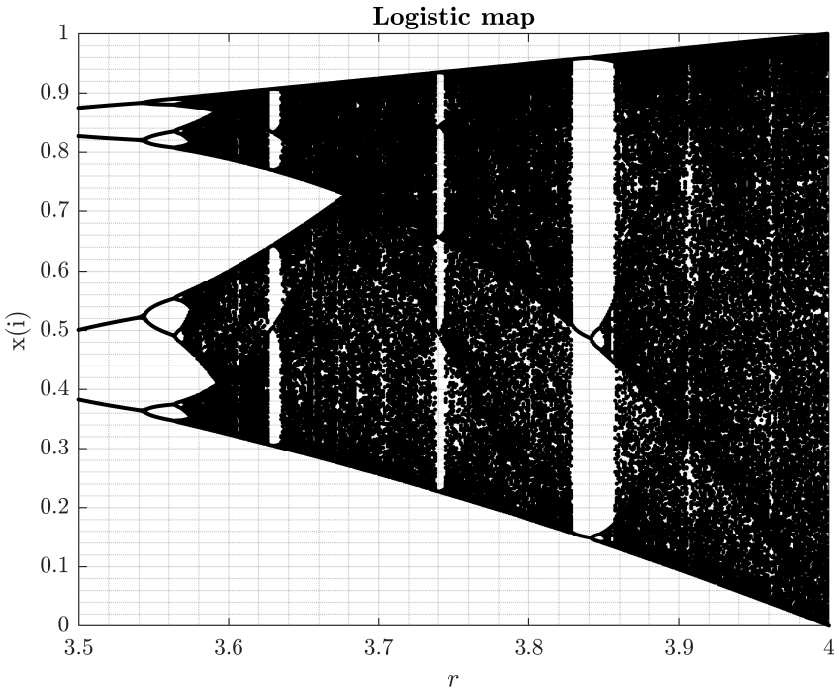

Figure 2 Logistic map for $r \in [3.5 - 4]$. Source Own.

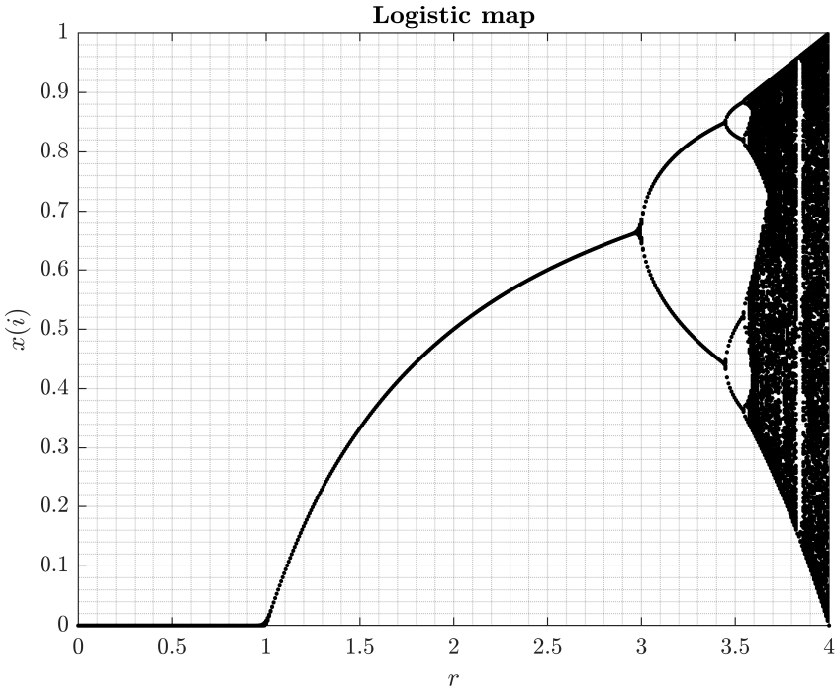

Figure 3 Logistic map for $r \in [0 - 4]$. Source Own.

Figures 2 and 3 depicts the phenomenon of increasing values of r. What is interesting is that as

r increases, the variation increases as well and when r reaches 3, the population does not cycle back and forth but population goes through a four year cycle before repeating. These are known as period doubling bifurcation. Besides, if r continues to increase the pattern repeats and it splits again.

c) Estimate $\delta = \dfrac{r_2 - r_1}{r_3 - r_2}$.

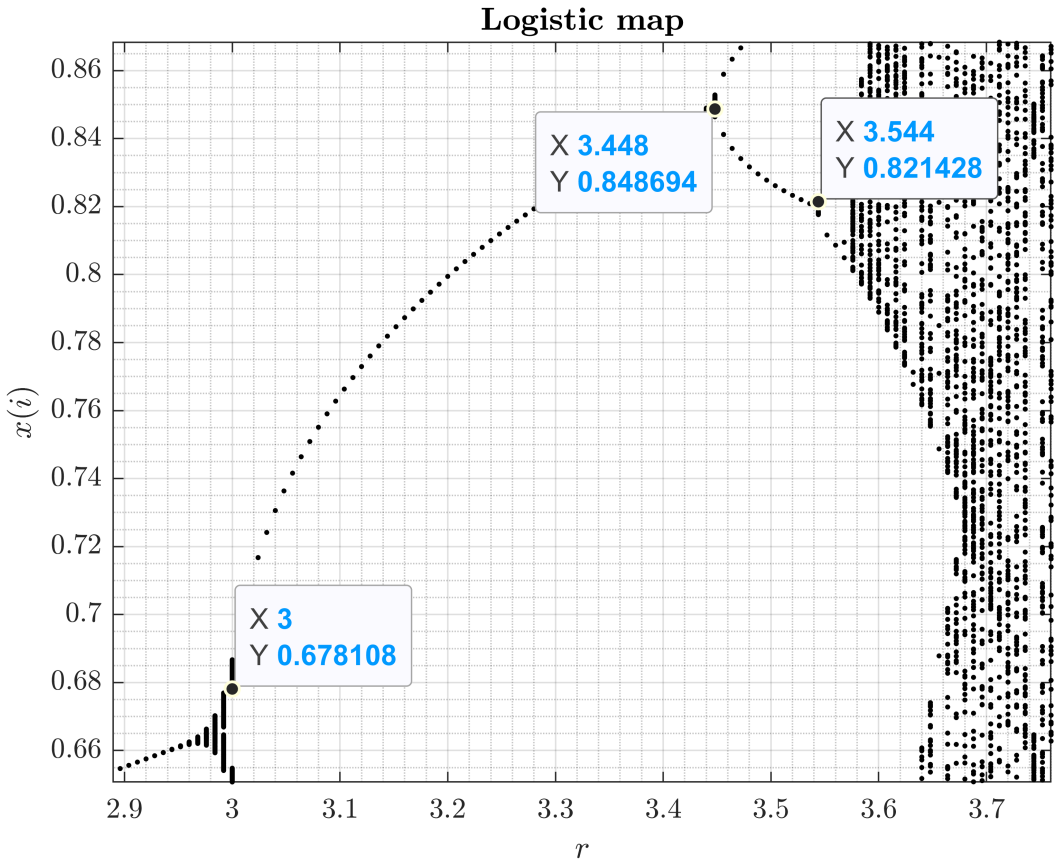

Figure 4 Bifurcation points approximation. Source Own.

The Feigenbaum constant is expressed as:

The Feigenbaum constant δ is a universal constant for functions approaching chaos via period doubling. It was discovered by Feigenbaum in 1975 while studying the fixed points of the iterated function $f(x) = 1 - \mu(x)^r$, and characterizes the geometric approach of the bifurcation parameter to its limiting value as the parameter μ is increased for fixed x [1].

The constant can be approximated using the following expression:

$$\delta = \lim_{n \to \infty} \frac{\lambda_{n+1} - \lambda_n}{\lambda_{n+2} - \lambda_{n+1}} \tag{2}$$

However, the above expression can be approximated as:

$$\delta \approx \frac{r_2 - r_1}{r_3 - r_2} = \frac{3.478 - 3}{3.544 - 3.448} \approx \boxed{4.979167} \tag{3}$$

The relative error between the above result and the theoretical value is:

$$\epsilon(\%) = \frac{4.979167 - 4.669202}{4.669202} = \boxed{6.64\ \%} \tag{4}$$

This result indicates the resultant value is correct and with a smaller step size it will tend towards the theoretical value.

2 Euler Approximation

a) Integrate analytically $\dot{x} = -x$ with the initial condition $x(0) = 1$. Which is the value of $x(1)$?
Give the following differential equation, the first step is to rewrite the expression as follows:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = -x \tag{5}$$

Next, it is a separable equation, developing the values:

$$-\frac{\mathrm{d}x}{x} = \mathrm{d}t \tag{6}$$

Integrating both sides of the equation:

$$-\int \frac{\mathrm{d}x}{x} = \int \mathrm{d}t \tag{7}$$

Thus,

$$-\ln x = t + C_1 \tag{8}$$

where C_1 is an arbitrary constant and $t_0 = 0$. Thereby,

$$\ln x = -t - C_1 \tag{9}$$

Finally, the solution of the differential equation is:

$$x(t) = K \cdot e^{-t} \tag{10}$$

where K is an arbitrary constant.
Applying the initial conditions, $x(0) = 1$, the constant K can be found:

$$x(0) = K \cdot e^{-0} = K = 1 \tag{11}$$

So, the value of $x(1)$ is:

$$x(1) = 1 \cdot e^{-1} = \boxed{\frac{1}{e}} \tag{12}$$

b) Use Euler formula with $dt = 1$ to estimate $x(1)$. Using the Euler formula for $dt = 1$, the estimated value of $x(1) = 1$. With this step size, the program does only one step which contributes to a significant error.

c) Repeat for $dt = 0.1, 0.01, 0.001, 0.0001$, complete the table and plot (in $log - log$ scale) the *Error* vs *dt*.

The real value of $x(1) = \frac{1}{e}$ is 0.367879 approximately. So, using the same procedure as the above the data is summarized to the following table:

dt	**x(t = 1)**	**Error | Real - Estimated |**
1	1	$3.6788 \cdot 10^{-1}$
0.1	0.3487	$1.9201 \cdot 10^{-2}$
0.01	0.3660	$1.8471 \cdot 10^{-3}$
0.001	0.3677	$1.8402 \cdot 10^{-4}$
0.0001	0.3679	$-1.8393 \cdot 10^{-5}$

Table 1

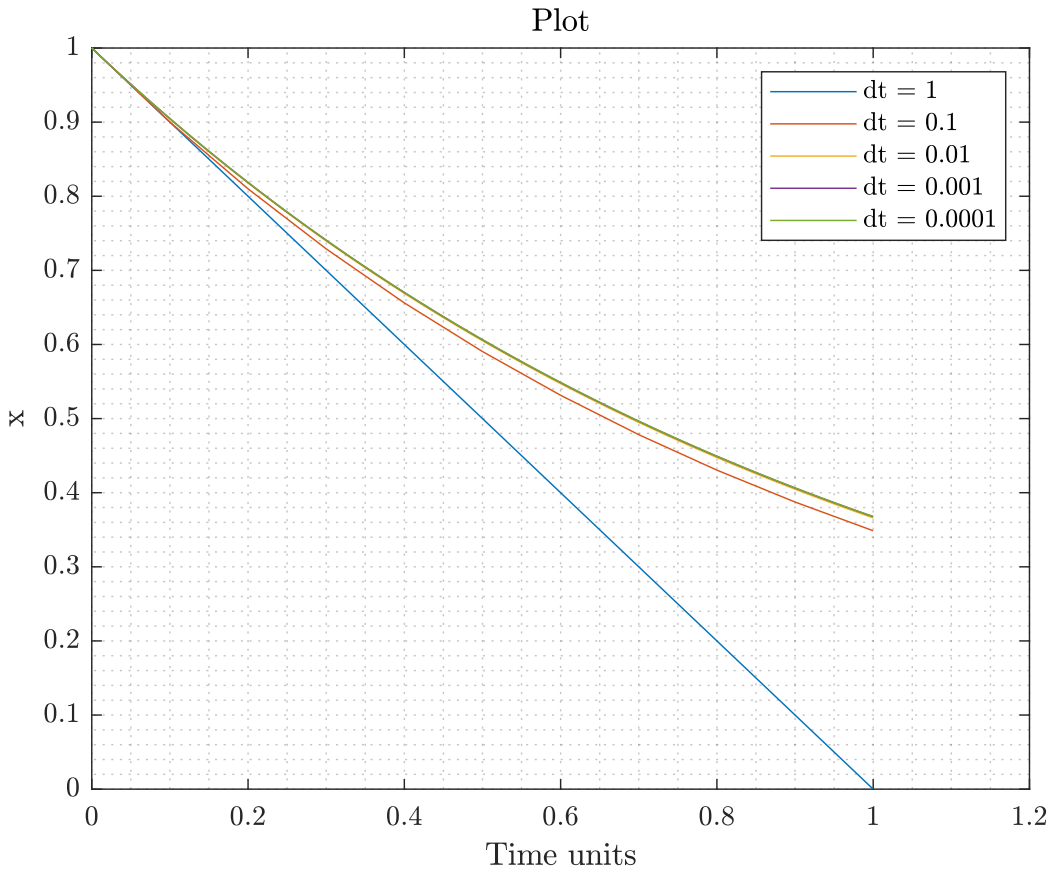

Figure 5 Numerical Euler Approximation for different *dt*. Source Own.

Below is presented the plot of the error in $log - log$ scale. Notice how the error decreases as the step size *dt* is smaller.

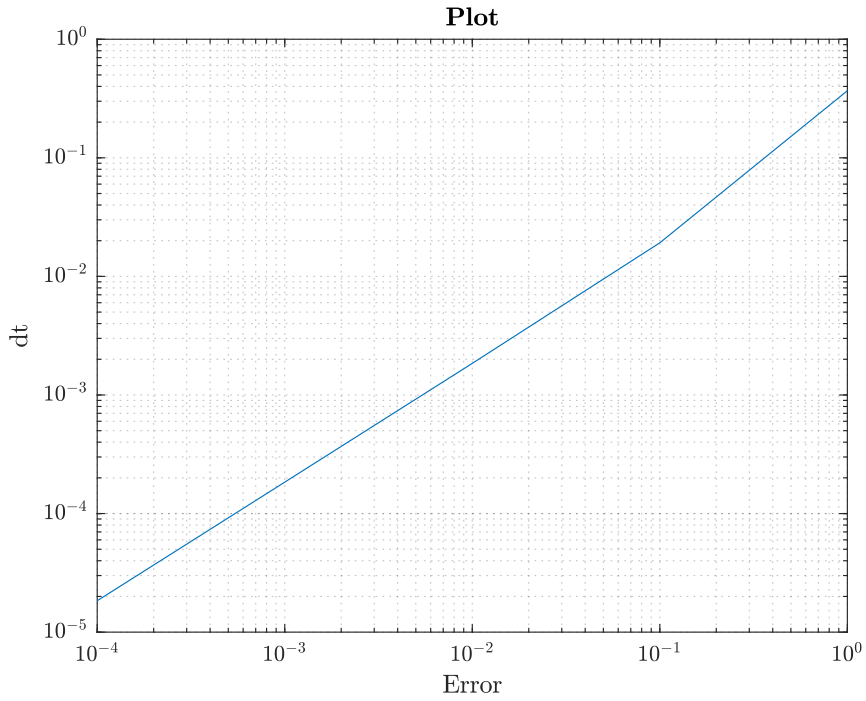

Figure 6 Step size vs Error. Source Own.

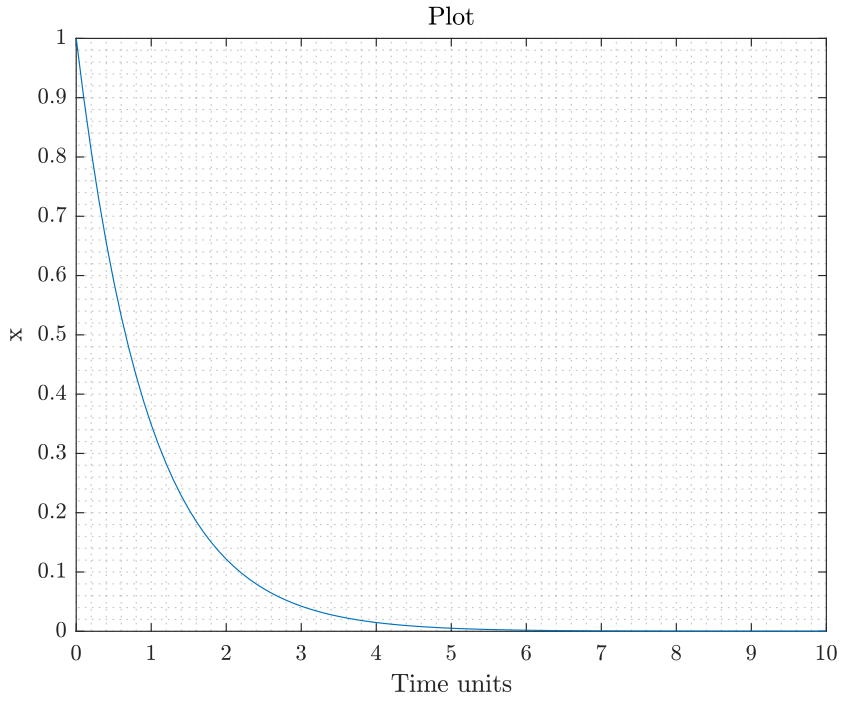

Figure 7 Analytical plot of the differential equation. Source Own.

Above is presented the analytical result of the differential equation.

3 Fixed Points and Stability analysis

Given

$$\frac{\mathrm{d}N}{\mathrm{d}t} = -aN\ln(bN) \tag{13}$$

a) Calculate the fixed points and their linear stability.

In order to calculate the fixed points, we set $\frac{\mathrm{d}N}{\mathrm{d}t} = 0$. Thus,

$$0 = -aN\ln(bN) \tag{14}$$

Eventually, two fixed points can be extracted from the above expression:

$$\dot{N} = 0 \begin{cases} N^* = 0 \\ N^* = \frac{1}{b} \end{cases} \tag{15}$$

Next, to calculate the linear stability, the first expression is derived over N as follows,

$$\frac{\mathrm{d}\dot{N}}{\mathrm{d}N} = -a\ln(bN) - aN\frac{1}{bN}b \tag{16}$$

After cancelling out terms,

$$\frac{\mathrm{d}\dot{N}}{\mathrm{d}N} = -a\left(\ln bN + 1\right) \tag{17}$$

Evaluating the above expression,

$$\begin{cases} \left.\frac{\mathrm{d}\dot{N}}{\mathrm{d}N}\right|_{N^*=0} = +\infty > 0 \\ \left.\frac{\mathrm{d}\dot{N}}{\mathrm{d}N}\right|_{N^*=\frac{1}{b}} = -a < 0 \end{cases} \tag{18}$$

where clearly $N* = \frac{1}{b}$ is a stable fixed point and $N^* = 0$ is an unstable fixed point.

b) Demonstrate that when $\boldsymbol{a} = \boldsymbol{b}$ the analytical solution of $\frac{dN}{dt} = -bN\ln(bN)$ is $N(t) = \frac{e^{ce^{bt}}}{b}$ where $\boldsymbol{c}$ is a constant that depends on the initial condition.

Particularizing the above expression to the case when $a = b$, the expression goes,

$$\frac{\mathrm{d}\dot{N}}{\mathrm{d}N} = -bN\ln(bN) \tag{19}$$

Rearranging the terms,

$$\frac{\frac{1}{N}}{-b\ln(bN)}\mathrm{d}N = \mathrm{d}t \tag{20}$$

Integrating both sides,

$$-\frac{1}{b}\int \frac{\frac{1}{Nb}}{\ln(bN)}\mathrm{d}N = \int \mathrm{d}t \tag{21}$$

Next, we integrate both sides of the equation,

$$-\frac{1}{b}\ln\left[\ln\left(bN\right)\right]\mathrm{d}N = t + C_1 \tag{22}$$

Developing the resultant terms, the final result can be extracted:

$$\ln[\ln\left(bN\right)] = -bt + C_1 \tag{23}$$

Finally, from the above expression $N(t)$ shall be isolated,

$$e^{-bt} \cdot C_2 = \ln(bN) \tag{24}$$

Therefore,

$$\boxed{dN(t) = \frac{e^{ke^{-bt}}}{b}} \tag{25}$$

c) Integrate the equation numerically and compare the numerical and analytical solutions.

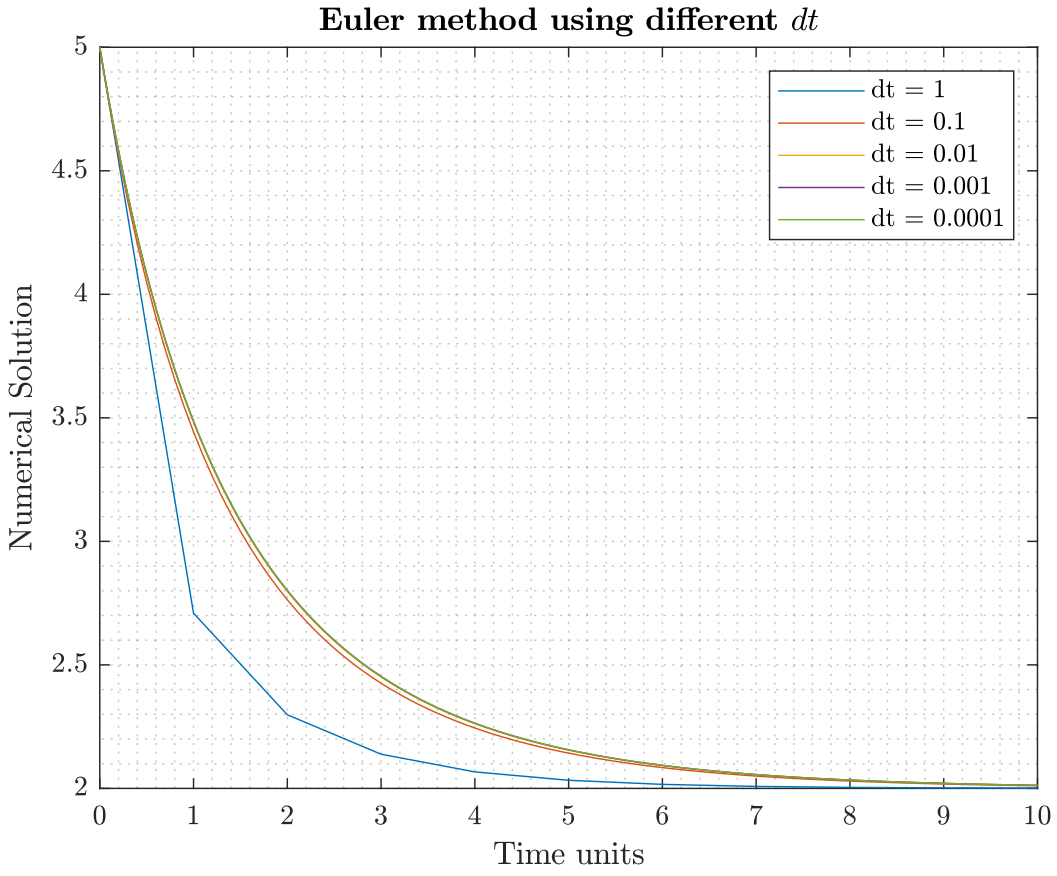

Figure 8 Euler Method using different dt. Source Own.

Above can be seen the numerical integration with Euler approximation. As the step size diminishes the plot resembles more the analytical solution (see Figure 10).

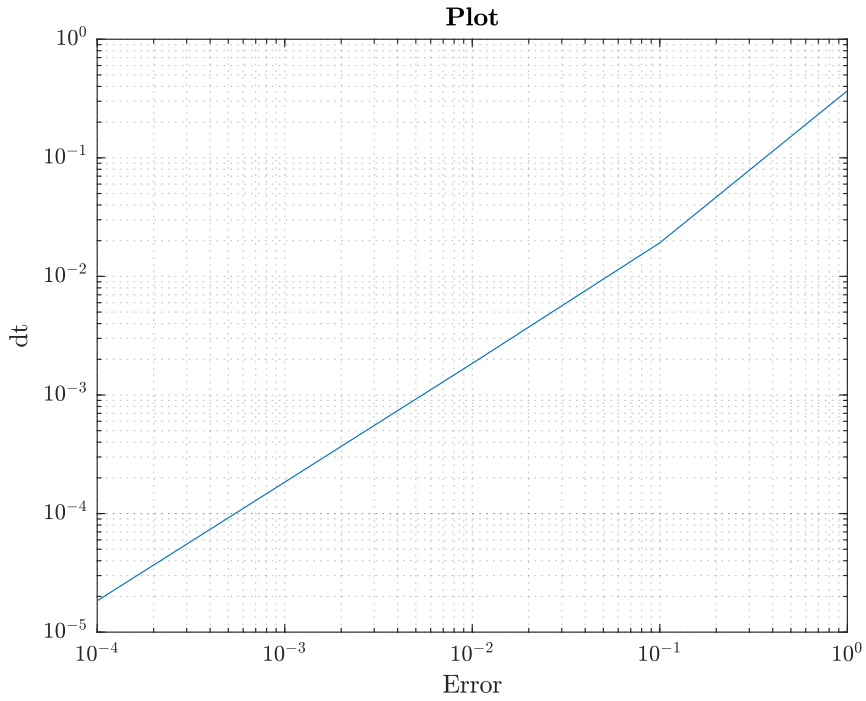

Figure 9 Step size vs dt. Source Own.

Another time, as the step size is minor, the error diminishes. However, using an extremely reduced step size will increase computation time and can increase error due to the own numerical limitations. Below is presented the analytical solution.

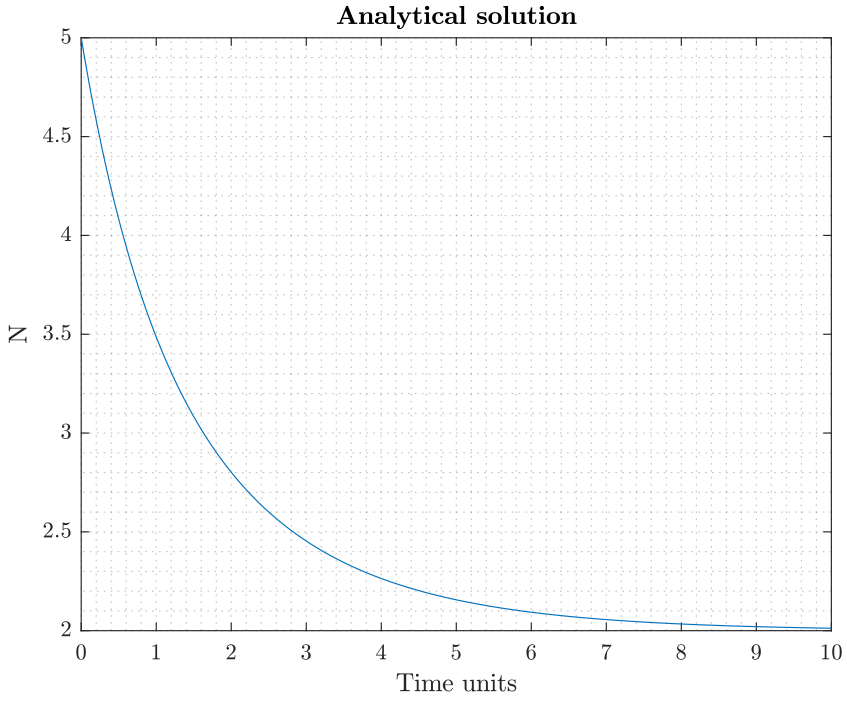

Figure 10 Analytical solution. Source Own.

3.1 Transcritical bifurcation

For the following ODE,

$$\dot{x} = r \ln x + x - 1 \tag{26}$$

show that a trans-critical bifurcation occurs near $x = 1$.

Hint: consider $u = x - 1$ small.

The first step is to consider teh following change of variable:

$$u = x - 1 \tag{27}$$

The derivative is expressed as,

$$\dot{u} = \dot{x} \tag{28}$$

Replacing the expression into the initial equation:

$$\dot{u} = r \ln (u + 1) + u \tag{29}$$

Then, as the statement mentioned, it can be supposed that $u = x - 1 << 1$. Thus, *Taylor's series* for the natural logarithm is

$$\ln(u + 1) = u - \frac{u^2}{2} + \frac{u^3}{3} + O\left(u^4\right) \tag{30}$$

Grabbing the first two terms of the prior series, and replacing them into the initial equation,

$$\dot{u} = r\left(u - \frac{u^2}{2}\right) + u \tag{31}$$

Rewriting the equation,

$$\dot{u} = -\frac{r}{2}u^2 + (1 + r)u \tag{32}$$

Rewriting the above expression,

$$\dot{u} = -\frac{r}{2}u^2 + (1 + r)u \approx (1 + r)u \tag{33}$$

Finally, if the change of variable is undone,

$$\dot{x} \approx (1 + r)(x - 1) \tag{34}$$

It confirms the transcritical bifurcation occurs near $x = 1$.

4 Fixed Points and Stability analysis

Find the fixed points and compute their stability

$$\dot{x} = rx + x^3 - x^5 \tag{35}$$

First, let's find the associated fixed points of the above expression. Let $f(x)$ be:

$$f(x) = rx + x^3 - x^5 \tag{36}$$

Then, to find the fixed points we shall $f(x) = 0$,

$$0 = rx + x^3 - x^5 \tag{37}$$

Notice how it is possible to extract the common factor and end up with a 4th order equation.

$$0 = x(r + x^2 - x^4) \tag{38}$$

There are two possible solutions at this point, either $x = 0$ or the term in parenthesis $r+x^2-x^4 = 0$. That is, the first fixed point is:

$$\boxed{x_1 = 0} \tag{39}$$

By assaying carefully, the term in parenthesis can be identified as a bi-quadratic equation, which easily be rewritten using a change of variables:

$$t = x^2 \qquad\qquad t^2 = x^4 \tag{40}$$

Thus, by means of the change of variables, the resultant expression goes:

$$0 = r + t - t^2 \tag{41}$$

Solving for this second order equation using the quadratic formula:

$$t = \frac{-1 \pm \sqrt{1^2 - 4 \cdot (-1) \cdot r}}{2 \cdot (-1)} = \frac{-1 \pm \sqrt{1 + 4r}}{-2} \tag{42}$$

Thus, Real solutions only exists when $\boxed{r \leq -\frac{1}{4}}$.

By taking $\mathbf{t_1}$, x_2 and x_3 can be found:

$$t_1 = \boxed{\begin{cases} x_2 &= \dfrac{+1 - \sqrt{1 + 4r}}{2} \\ x_3 &= \dfrac{+1 + \sqrt{1 + 4r}}{2} \end{cases}} \tag{43}$$

By taking $\mathbf{t_2}$, the resultant two solutions can be obtained:

$$t_2 = \boxed{\begin{cases} x_4 &= \dfrac{-1 + \sqrt{1 + 4r}}{2} \\ x_5 &= \dfrac{-1 - \sqrt{1 + 4r}}{2} \end{cases}} \tag{44}$$

Notice that an hysteresis cycle appears, in the way that if there is an initial condition such that we are in a stable solution 0, a certain point there will be a critical transition (up or down).

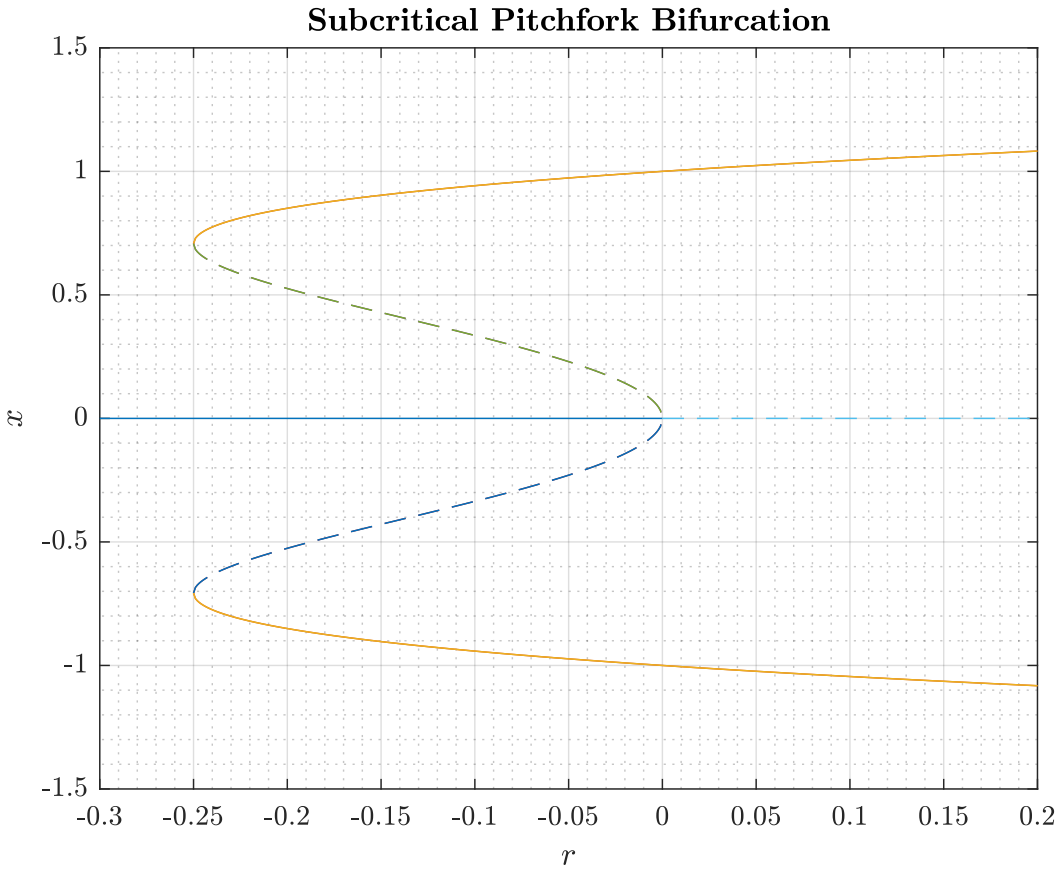

Figure 11 Subcritical Pitchfork Bifurcation. Source Own.

Dotted lines represent unstable solutions while straight line represents a stable solution.

5 Neuron Model Simulation

Simulate the neuron model with different values of the control parameter I and/or different initial conditions.

The neuron model can be simulated using the following expression:

$$\begin{gathered} C\dot{V} = I - g_{\mathrm{L}}\left(V - E_{\mathrm{L}}\right) - \overbrace{g_{\mathrm{Na}} m_{\infty}(V)\left(V - E_{\mathrm{Na}}\right)}^{\text{instantaneous } I_{\mathrm{Na,p}}} \\ m_{\infty}(V) = 1/\left(1 + \exp\left\{\left(V_{1/2} - V\right)/k\right\}\right) \end{gathered} \tag{45}$$

and the following parameters

$$\begin{array}{llll} C = 10\mu \text{ F}, & I = 0 \text{ pA}, & g_{\mathrm{L}} = 19 \text{ mS}, & E_{\mathrm{L}} = -67 \text{ mV} \\ g_{\mathrm{Na}} = 74 \text{ mS}, & V_{1/2} = 1.5 \text{ mV}, & k = 16 \text{ mV}, & E_{\mathrm{Na}} = 60 \text{ mV} \end{array} \tag{46}$$

The above expression is a simplified model for the potential of a neuron and it depends on different parameters that are listed above. There is no oscillatory motion since there is only one variable, however, the external input is represented by I.

Figure 12 shows the Neuron model for $V_0 = -80$ and different vales of I.

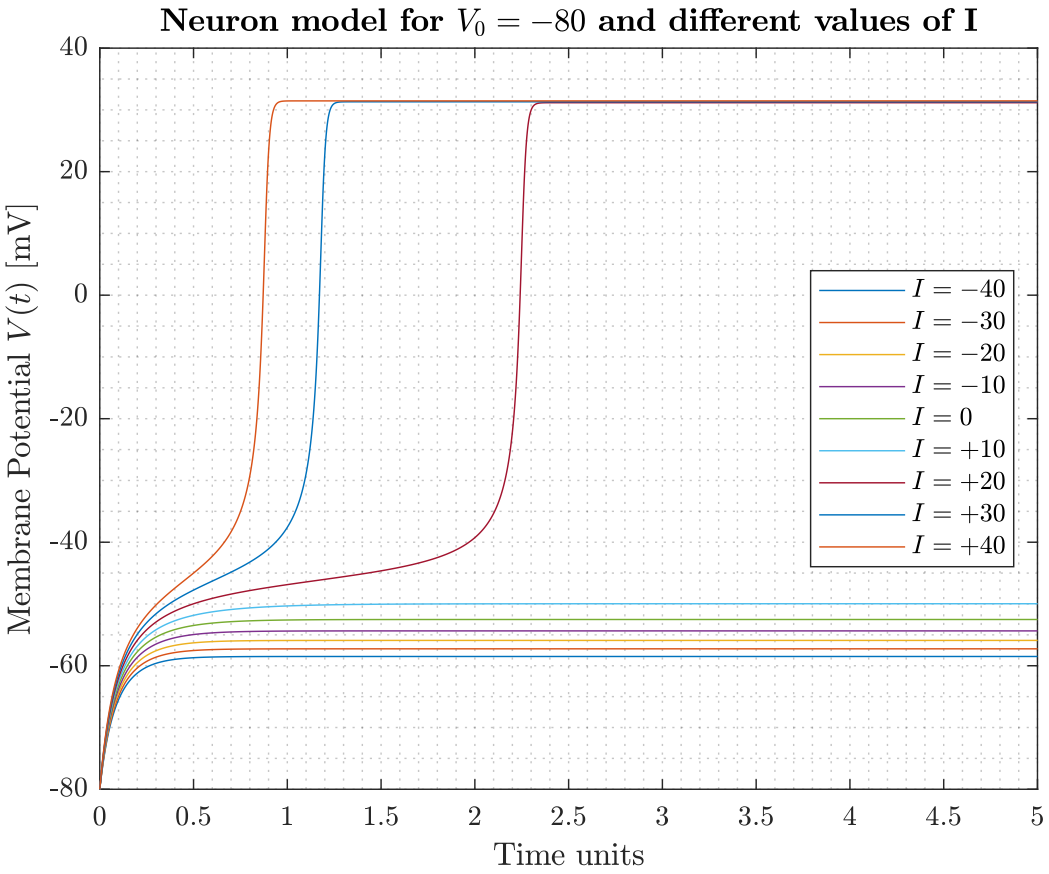

Figure 12 Neuron model simulation. Source Own.

Figure 13 shows the Neuron model for $I = 10$ and different values of V_0.

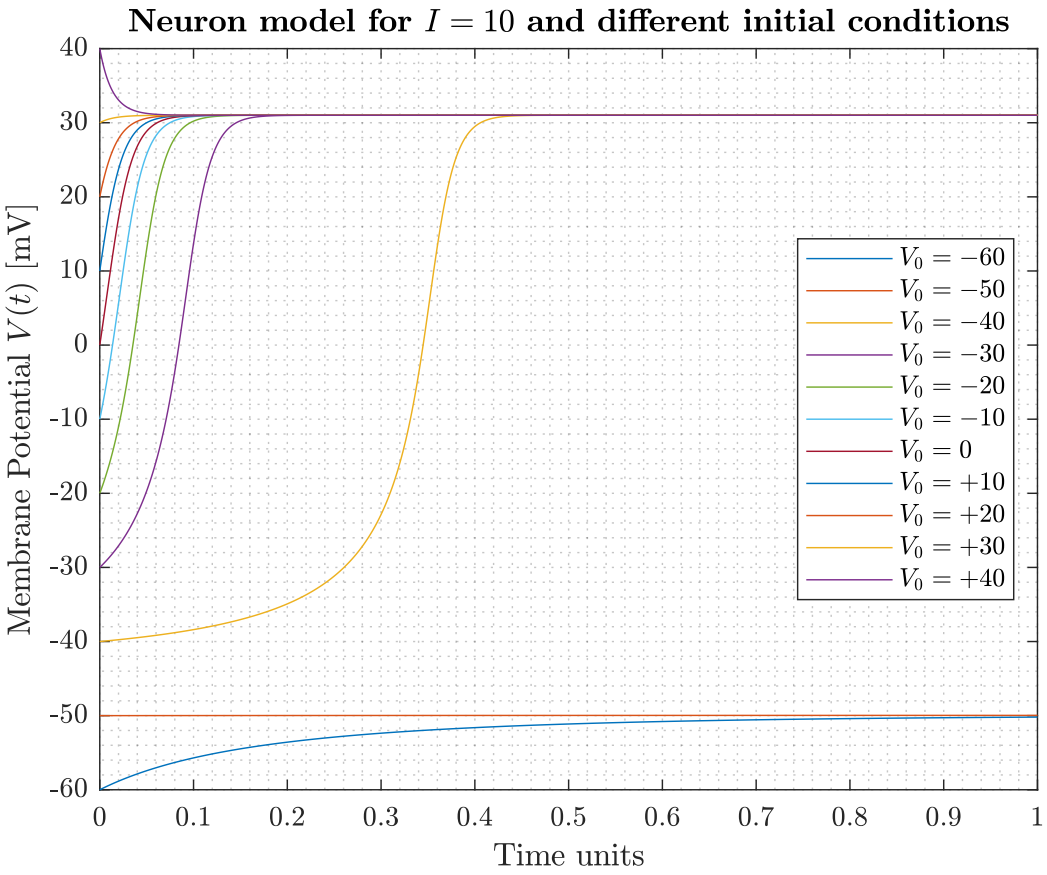

Figure 13 Neuron model simulation. Source Own.

Next page, the neuron model expression is evaluated with different initial conditions (see Figures 14 and 15).

In Figure 14 a), we see two states, a "rest" state, where the voltage is very small, and if the system is perturbed above the threshold, the system will move to a fixed point with large potential called "excited state". It is clearly seen the threshold is around -40 mV. With $I = 60$ (Figure 14 c)) there is no rest state, whatever initial condition it goes directly into the excited state. A similar phenomenon happens when $I = -1000$, where now, for any initial conditions, the neuron's potential V goes into the rest state.

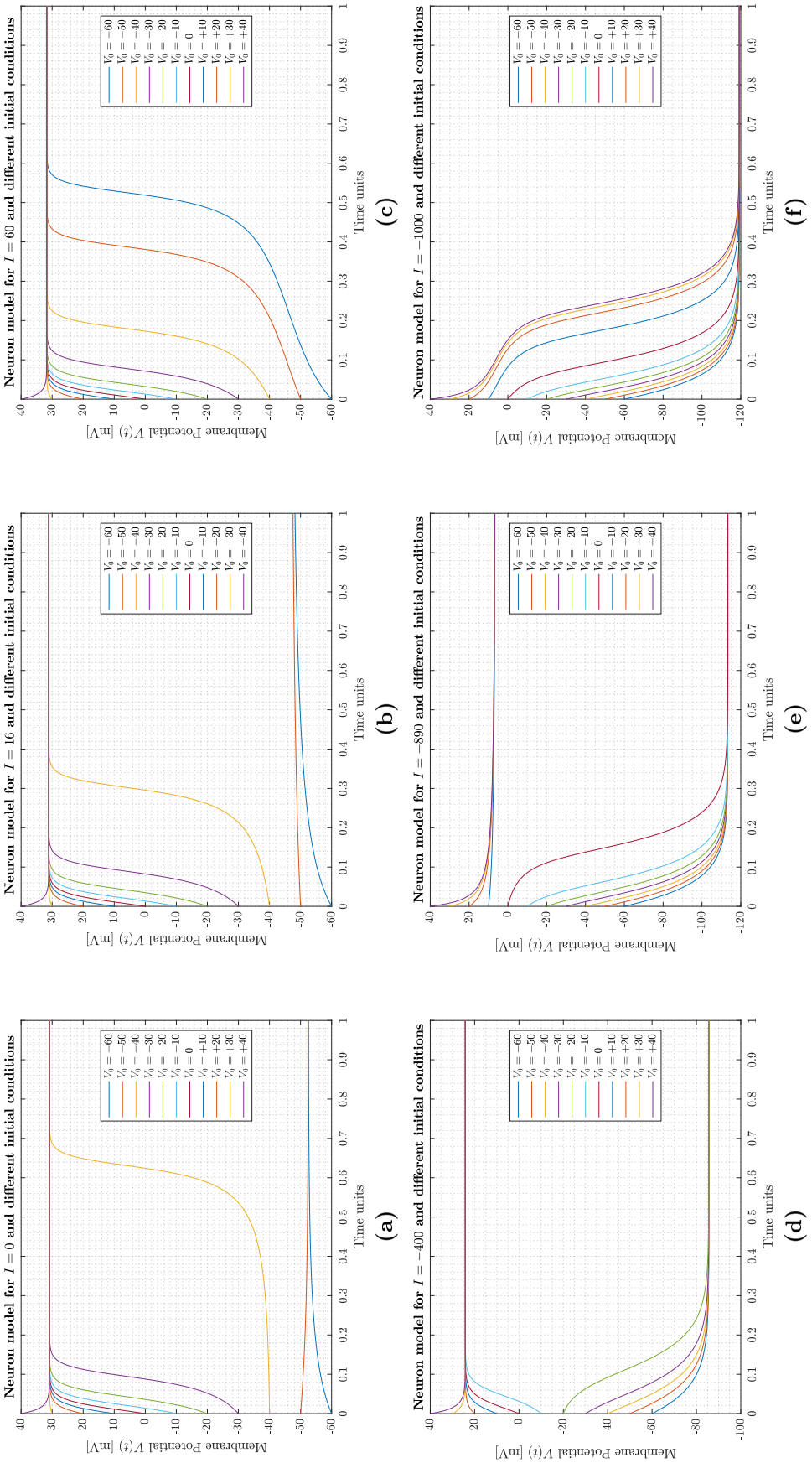

Figure 14 Neuron model simulation for different conditions. Source: Own.

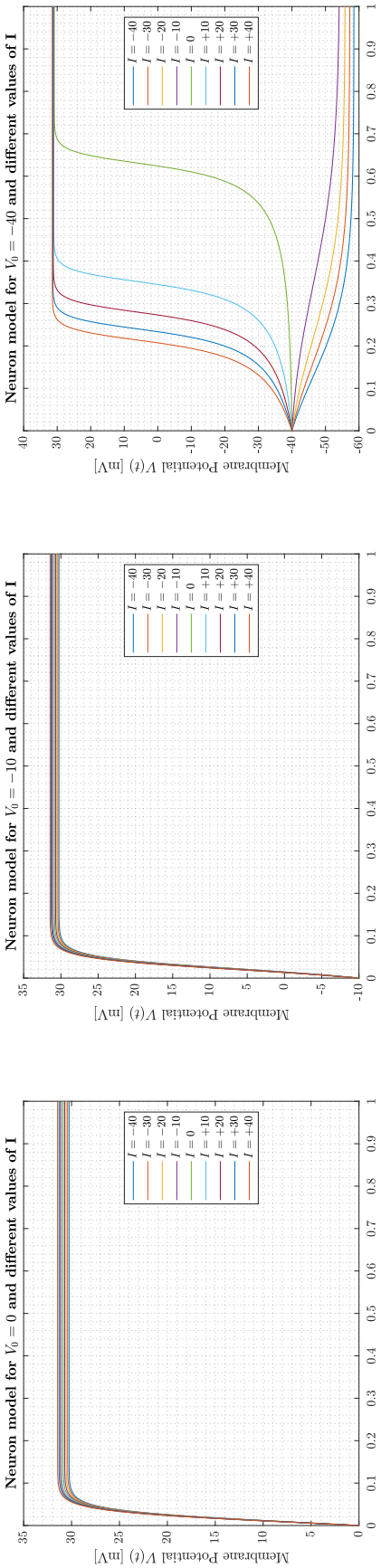

Figure 15 Neuron model for different conditions.

6 Laser turn on simulation

$$\dot{x} = rx - x^2 \tag{47}$$

a) Simulate the "turn on" when r is constant, $r > r^* = 0$

$$r(t) = r$$
$$x_0 = 0.01$$

The idea is to integrate the equation (47) above the threshold (in this case the threshold is $r^* = 0$. And take an inital condition very close to zero (since 0 is a fixed point).

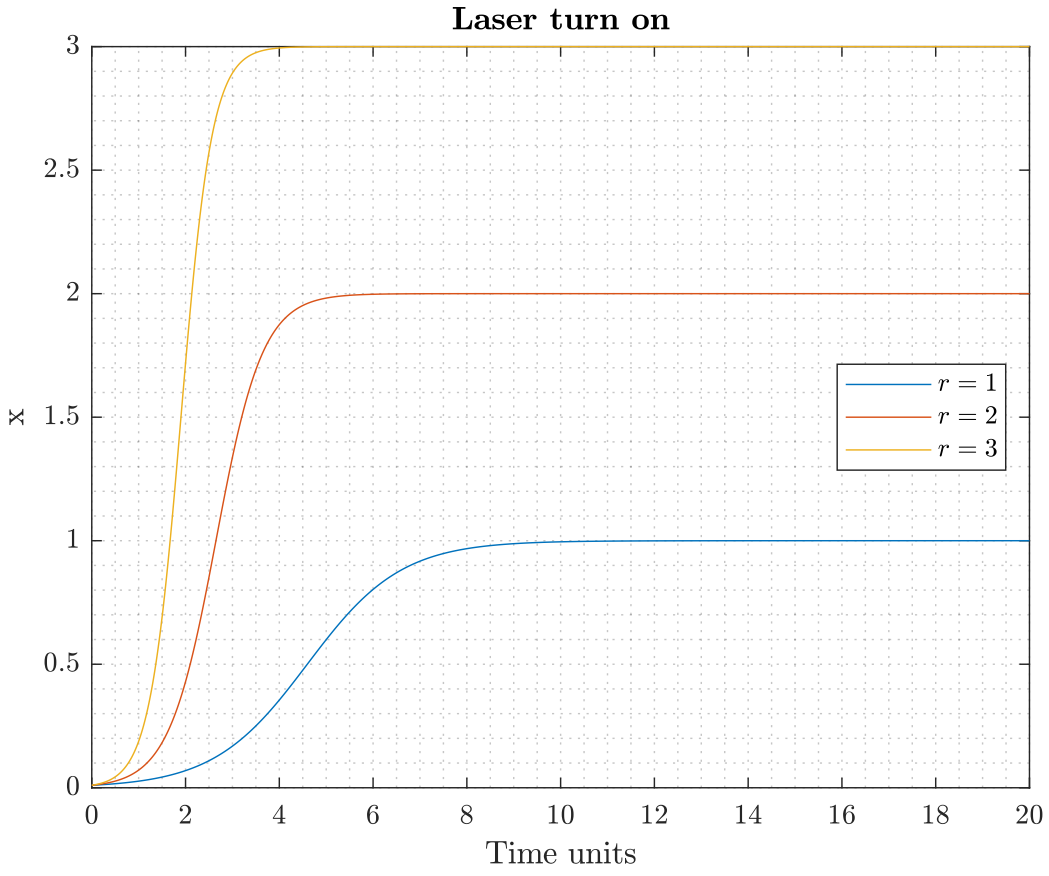

Figure 16 Analytical solution. Source Own.

As seen in the resultant plot, the output increases with respect to the parameter.

b) Calculate the bifurcation diagram by plotting $x(t = 50)$ vs r

$$\dot{x} = rx - x^2 + h$$

In this second part, the idea is to plot the final time with respect to r. The small parameter h represents the noise. That makes the bifurcation smooth. This final state, disregarded the transient behaviour.

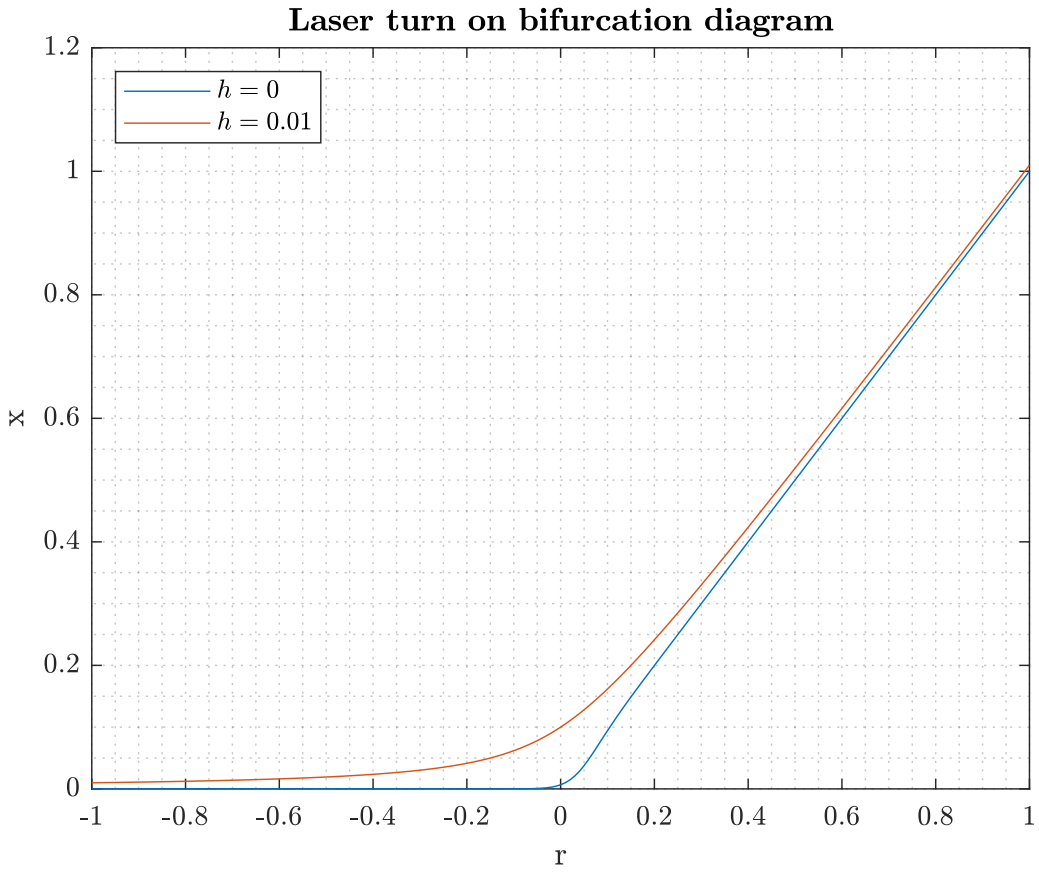

Figure 17 Analytical solution. Source Own.

c) Simulate the equation with r increasing linearly in time. Consider different variation rate (v) and/or different initial value of the parameter (r_0)

$$r(t) = r_0 + vt$$
$$x_0 = 0.01$$

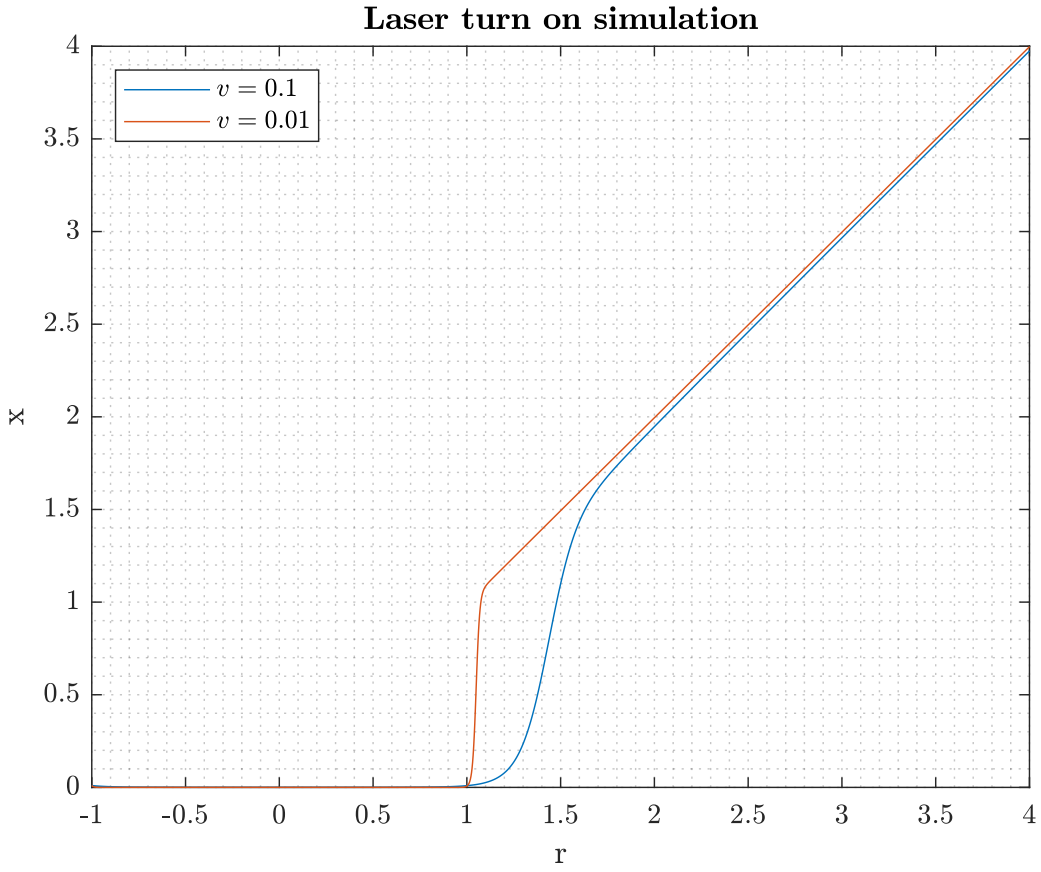

Figure 18 Analytical solution. Source Own.

$$v = 0.1$$
$$x_0 = 0.01$$

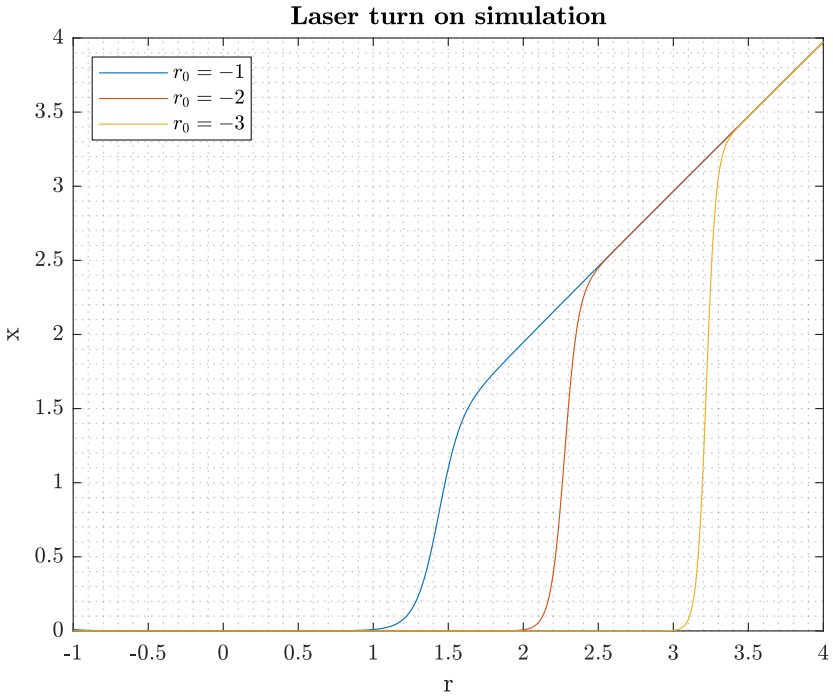

Figure 19 Analytical solution. Source Own.

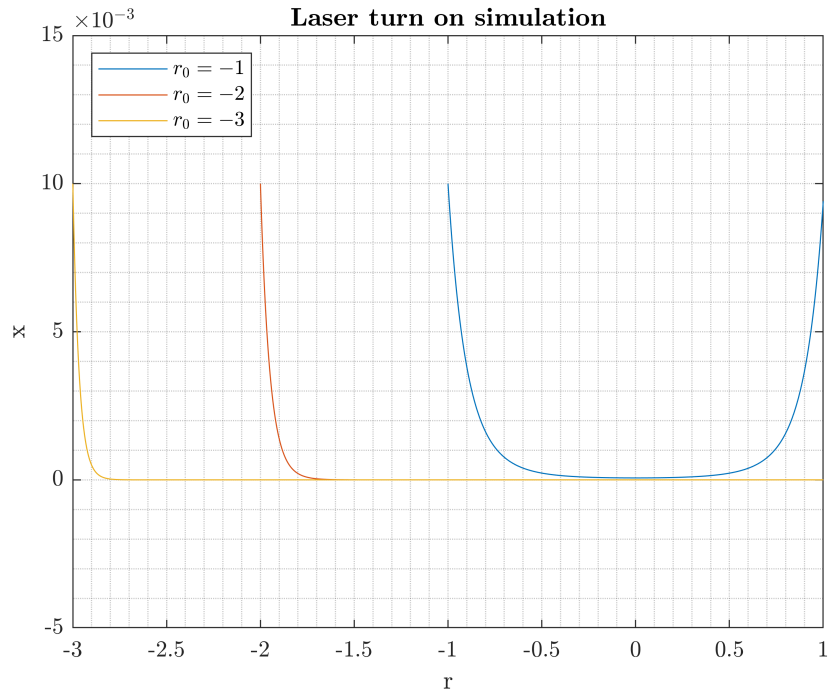

Figure 20 Analytical solution. Source Own.

d) Consider that the control parameter r increases and then decreases linearly in time: plot x and r vs. t and plot x vs. r.

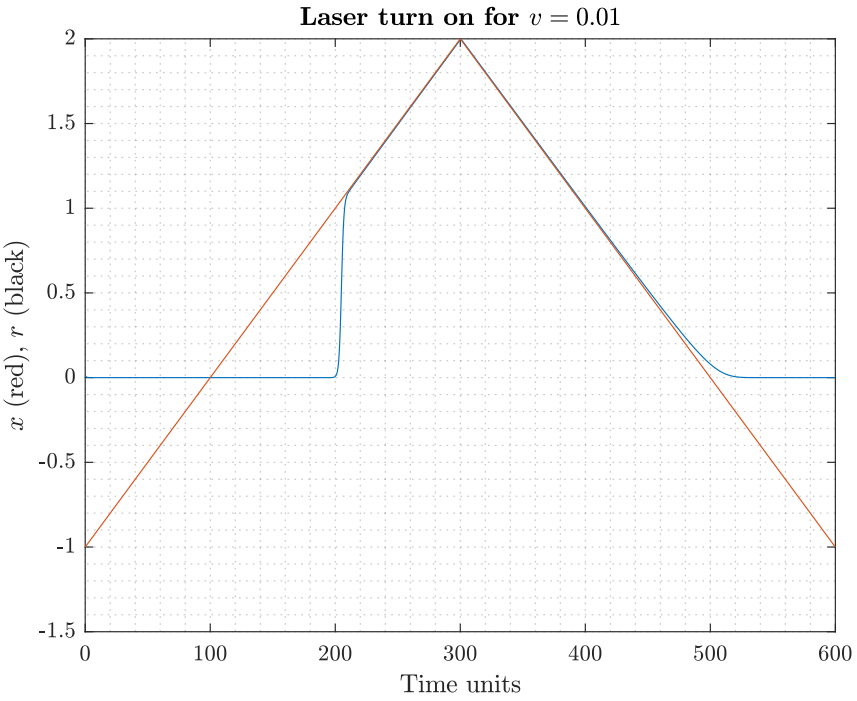

Figure 21 Analytical solution. Source Own.

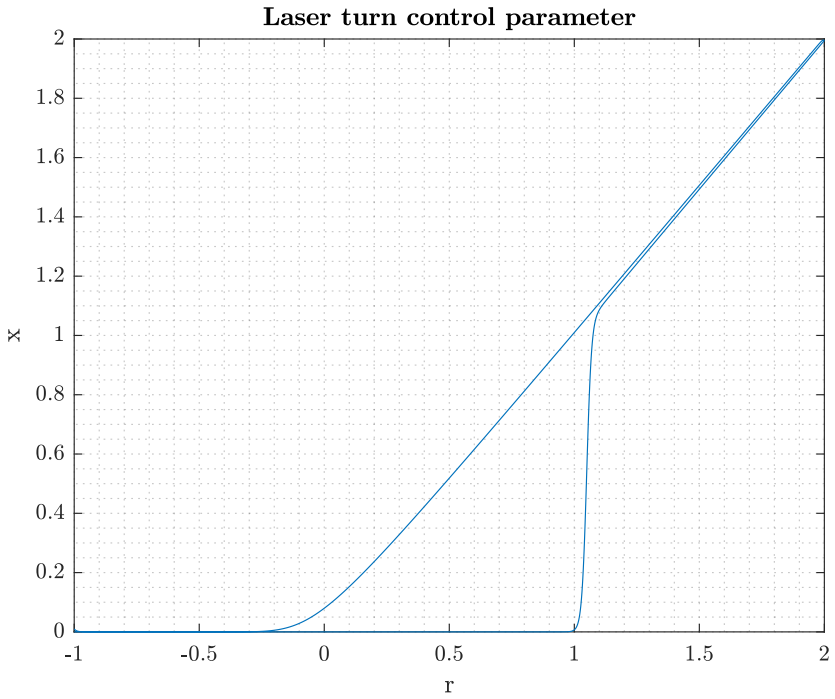

Figure 22 Analytical solution. Source Own.

7 Imperfect bifurcations

Calculate the analytical expression of $h_c(r)$ Tip: use the equations of

- The fixed points: $f(x^\star) = 0$
- Of the saddle node bifurcation: $f'(x^\star) = 0$

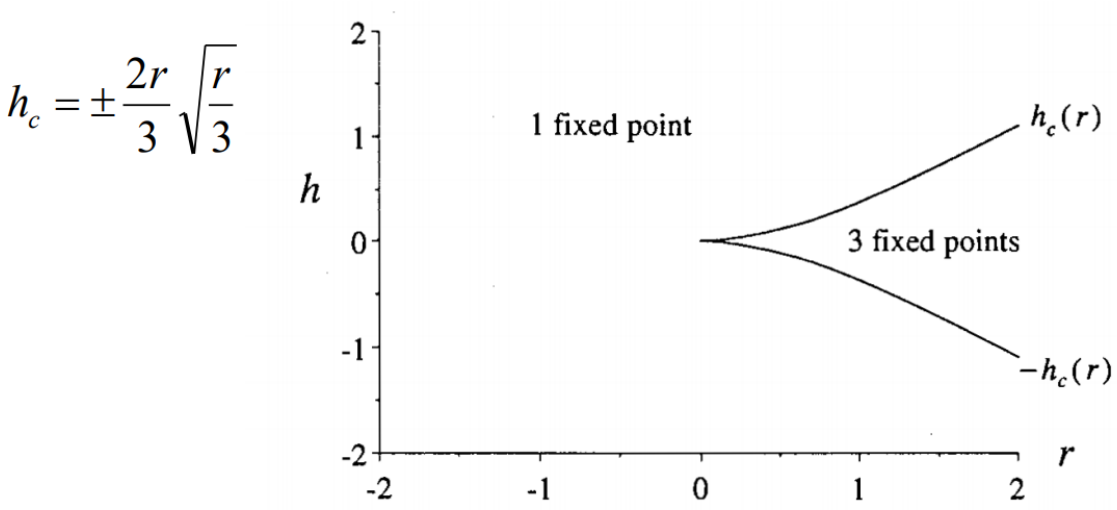

Figure 23 Scheme. Source [2].

Given the equation:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = h + rx - x^3 \tag{48}$$

Imposing the two conditions stated above, we have:

$$\begin{cases} 0 &= h + rx^* - x^* \\ 0 &= r - 3x^{*2} \end{cases} \tag{49}$$

From the second expression the fixed points can be solved:

$$x^* = \pm\sqrt{\frac{r}{3}} \tag{50}$$

Thereby, there are two cases to be studied.

- First let's compute with $x^* = +\sqrt{\frac{r}{3}}$: Replacing the fixed point $x*$ into the first expression,

$$h_c = x^{*3} - r^* = \sqrt{\left(\frac{r}{3}\right)^3} - r\sqrt{\frac{r}{3}} \tag{51}$$

  Developing the above stated expression, h_c can be found:

$$h_c = \frac{r^{\frac{3}{2}}}{3\sqrt{3}} - \frac{r^{\frac{3}{2}}}{\sqrt{3}} = \boxed{-\frac{2r}{3}\sqrt{\frac{r}{3}}} \tag{52}$$

- Analogously, for $x^* = -\sqrt{\frac{r}{3}}$:

  Replacing the fixed point $x*$ into the first expression,

$$h_c = x^{*3} - r^* = \sqrt{\left(-\frac{r}{3}\right)^3} + r\sqrt{\frac{r}{3}} \tag{53}$$

  Developing the above stated expression, h_c can be found:

$$h_c = -\frac{r^{\frac{3}{2}}}{3\sqrt{3}} + \frac{r^{\frac{3}{2}}}{\sqrt{3}} = \boxed{+\frac{2r}{3}\sqrt{\frac{r}{3}}} \tag{54}$$

8 Insect outbreak

Show that $x* = 0$ is always unstable.

Given the

$$\dot{N} = RN\left(1 - \frac{N}{K}\right) - p(N) \tag{55}$$

where

- Budworms population grows logistically ($R > 0$ grow rate) $p(N)$: dead rate due to predation
- If no budworms ($N \approx 0$) : no predation: birds look for food elsewhere
- If N large, $p(N)$ saturates: birds eat as much as they can.

The dimensionless expression is the following:

$$x = N/A \quad \tau = \frac{Bt}{A}, \quad r = \frac{RA}{B}, \quad k = \frac{K}{A}$$

and

$$\frac{dx}{d\tau} = rx\left(1 - \frac{x}{k}\right) - \frac{x^2}{1+x^2} \tag{56}$$

- Fixed point: $x^* = 0$
- Other FPs are the solutions of $r\left(1 - \frac{x}{k}\right) = \frac{x}{1+x^2}$

Fist, we calculate the fixed points by setting $\frac{\mathrm{d}x}{\mathrm{d}t} = f(\tau) = 0$, thus,

$$f(x) = 0 = rx\left(1 - \frac{x}{k}\right) - \frac{x^2}{1+x^2} \tag{57}$$

Then we derive the above expression:

$$f'(x) = r - \frac{2rx}{k} - \frac{2x(1+x^2) - x^2(2x)}{(1+x^2)^2} \tag{58}$$

Rearranging the obtained expression:

$$-\frac{2r}{k}x^5 + x^4 + \frac{4r}{k}x^3 + 2rx^2 + \left(-\frac{2r}{k} - 2\right)xr = 0 \tag{59}$$

For the case of $x = 0$, it is easy to see that the result would be $r = 0$. As a conclusion, the system is always unstable at this point.

9 Adler's equation

$$\dot{\theta} = \omega - a \sin \theta \tag{60}$$

a) Solve Adler's equation with $\omega = 1$, a= 0.99 and $\theta(0) = \pi/2$.

$$T = \frac{2\pi}{\sqrt{\omega^2 - a^2}} \tag{61}$$

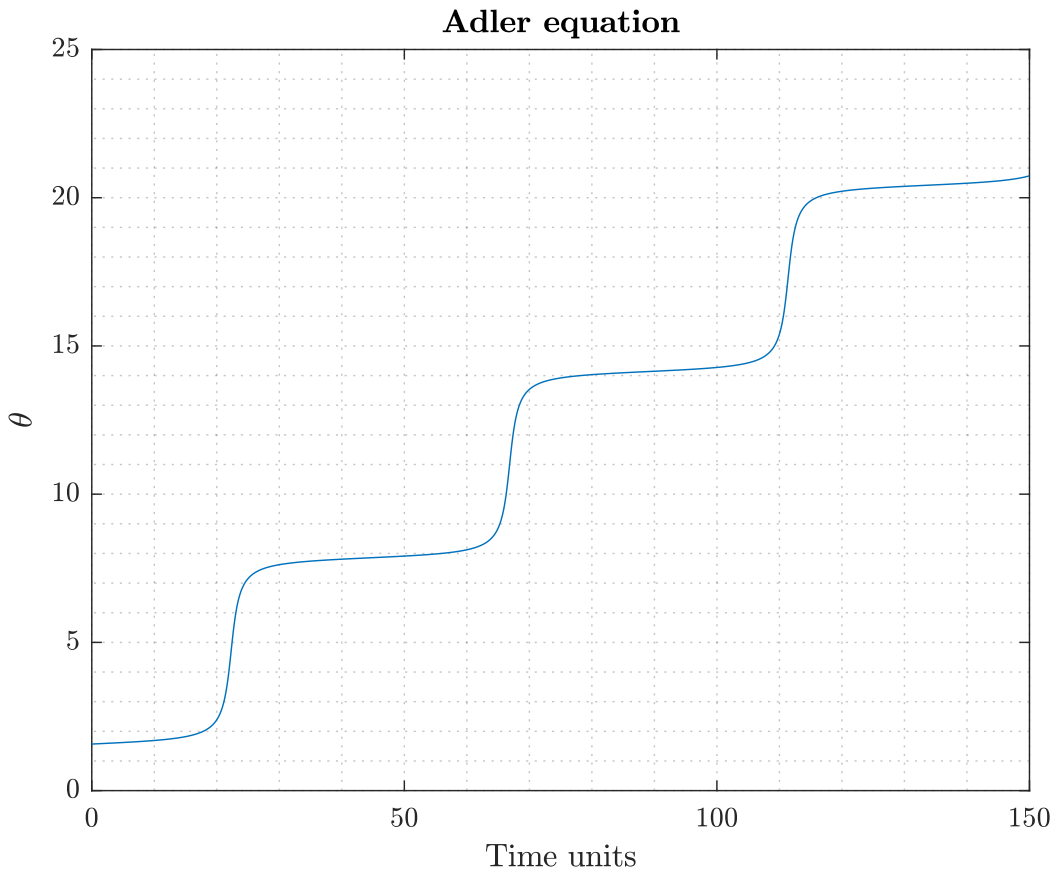

Figure 24 Analytical solution. Source Own.

Plotting the Adler's equation, it can be seen in Figure 24 that the growth is with small steps. This is due to the bottleneck in the wayt that the velocity is very small in a certain amount of range but then jumps a lot.

b) With $\omega = 1$, calculate the average oscillation period and compare with the analytical expression.

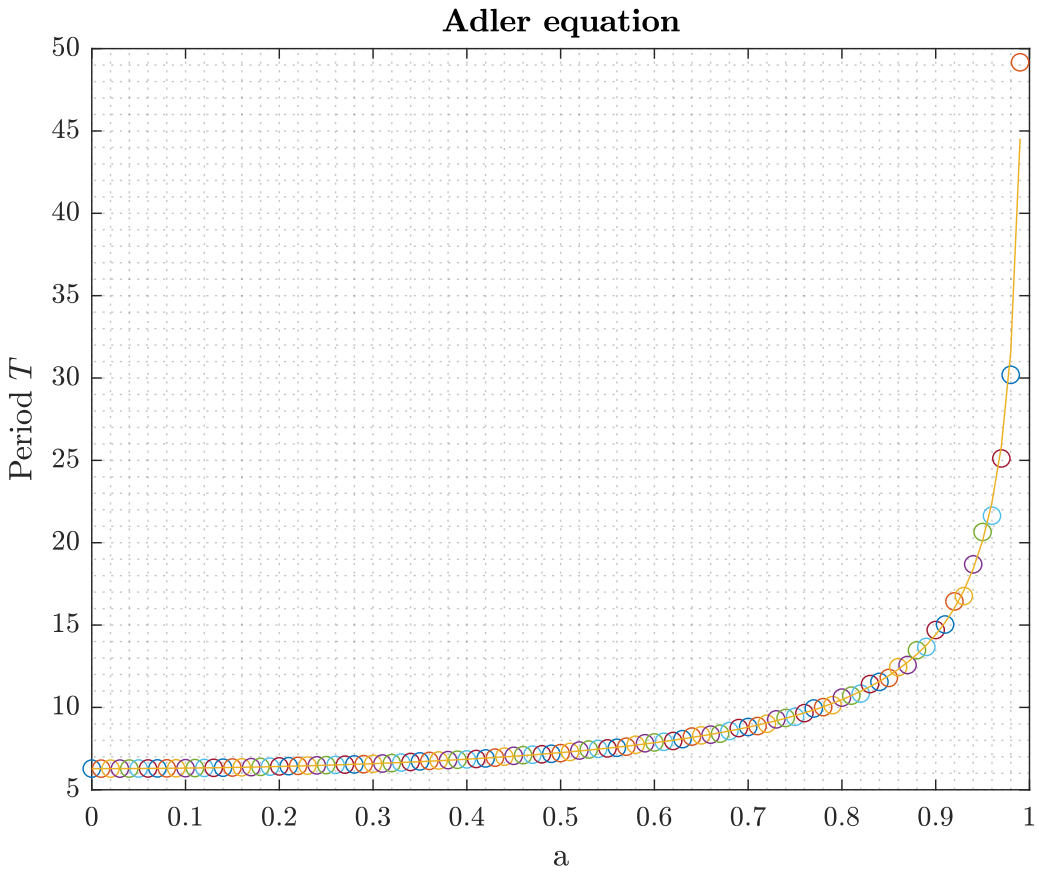

Figure 25 Analytical solution. Source Own.

This second part, the plot shows the average oscillation period for different values of "a". Comparing with the analytical solution, we can see that the Period calculated numerically does not differ so much with the analytical result.

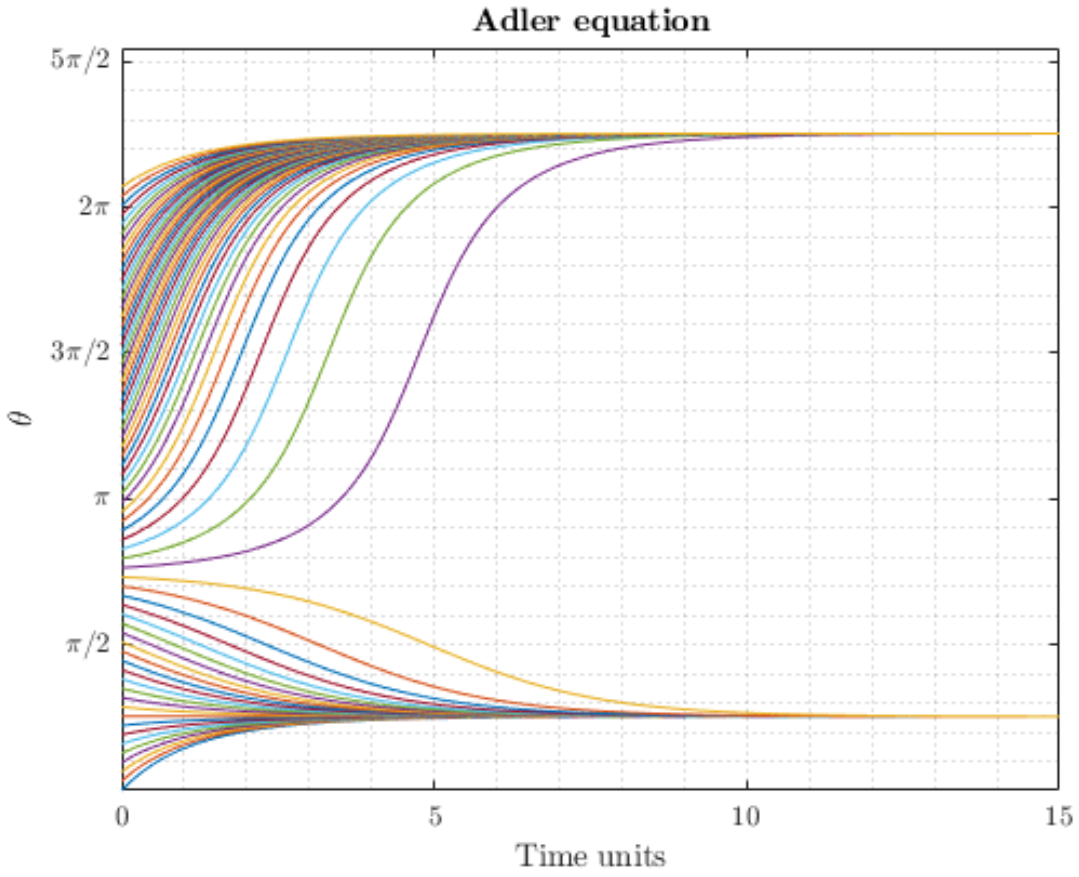

Figure 26 Analytical solution. Source Own.

Finally, below is plotted the trajectory for an arbitrary initial condition. It is interesting to see how both of the solutions are separated with a distance of 2π. Both of them are unstable and actually, they are the same solution. The only difference is how they are approached.

10 Delayed logistic equation

$$\frac{dy}{dt} = \lambda y(1 - y(t-1)) \tag{62}$$

a) Simulate the delayed logistic equation with different values of the delay τ.

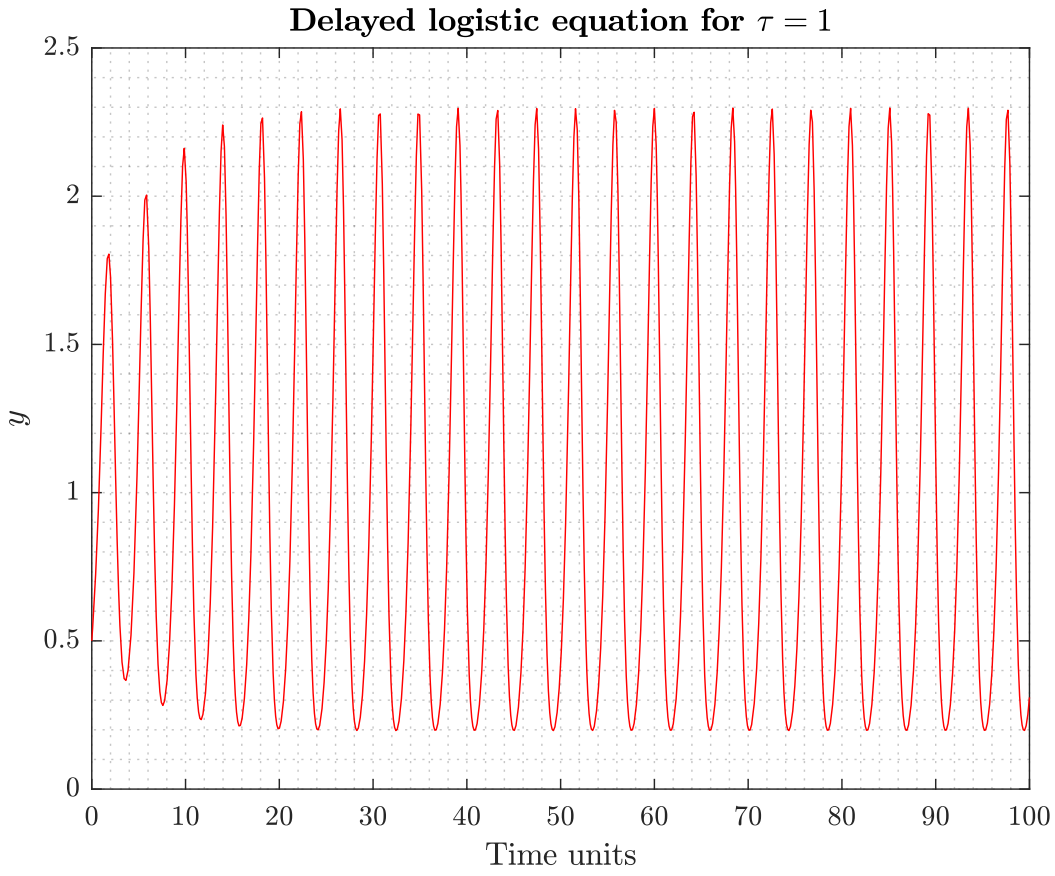

Figure 27 Delayed Logistic map. Source Own.

A feedback control mechanism usually involves a time delay. In a 2D system delayed feedback can reduce oscillations, but in a 1D system it can induce oscillations.

In the above expression, the delay is $\tau = 1$. The initial condition is constant $y_0 = 0.5$.

As seen in the plot above, for a 1D system if the delay generates a set of oscillations to the system and it does not recover or diminish its effect. It cannot be controlled. This delay induces long transients.

A time-delayed feedback loop can stabilize a system, but it can also induce a set of oscillations.

11 Stability diagram

$$\frac{dx}{dt} = x - x^3 + cx(t-\tau) + \sqrt{2D}\xi \tag{63}$$

Test the stability diagram by simulating the equation with D = 0 and different values of the parameters c and τ.

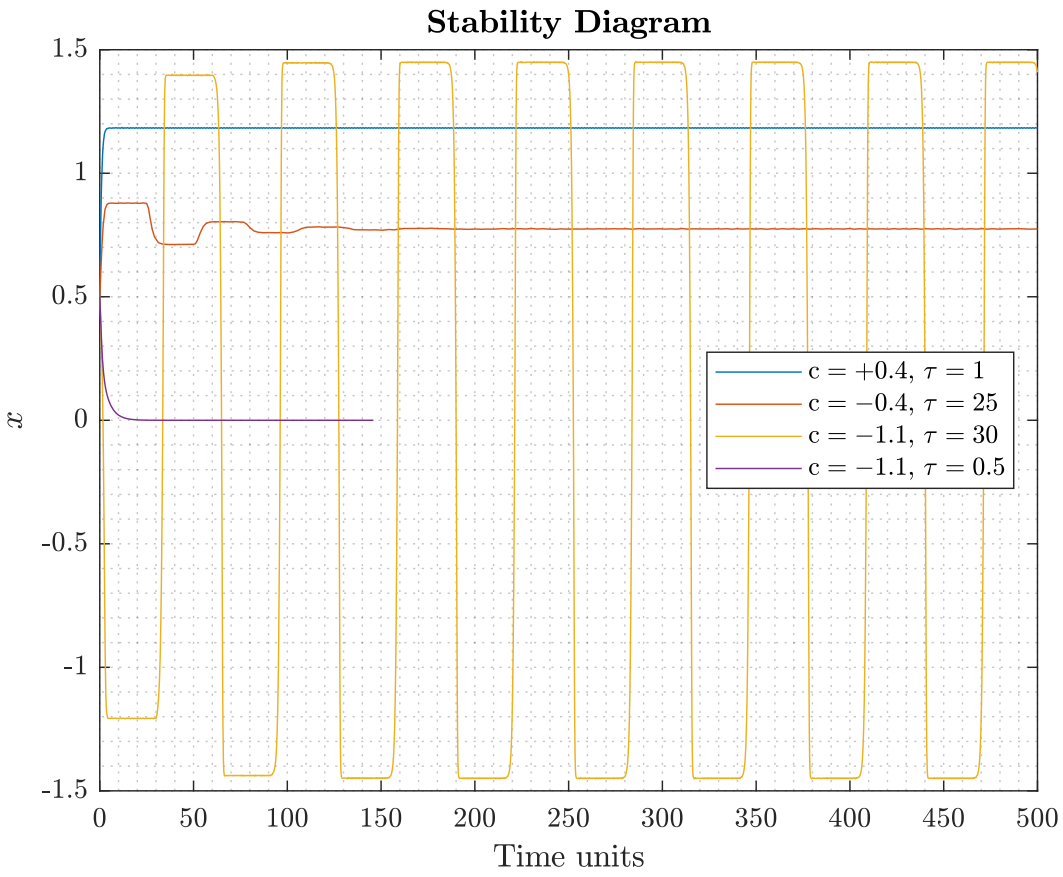

Figure 28 Stability Diagram. Source Own.

This is a simple model to understand two-state systems.

With appropriated parameters delay feedback can suppress fluctuations and stabilize the system in one state (the laser emits a stable output with stable polarization)

The equation above depends strongly on the parameter c. If $c > 0$ it is stable for all τ. Howeever, if $c < 0$ the stability depends on (c, τ).

There are two fixed points, one at $x = -1$ and $x = 1$. With a positive initial condition the system will go to $x = 1$ but with a negative initial condition the system will end up in $x = -1$.

Appendices

A Logistic Map for $r = 3.5$

```
%% Logistic map
%
%--------------------------------------------------------------------------%
% Exercise 1: Logistic map for r=3.5
%--------------------------------------------------------------------------%

% Date: 25/01/2021
% Author/s: Yi Qiang Ji Zhang
% Subject: Nonlinear Systems, Chaos and Control in Engineering
% Professor: Antonio Pons & Cristina Masoller

% Clear workspace, command window and close windows
clear;
close all;
clc;

% Set interpreter to latex
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');

%% Logistic map

r = 3.5; % Parameter r
it = 10; % Number of iterations
x = zeros(1,it); % Initialize vector x

% Initial condition
x(1) = 0.2;

% Iterate function
for i = 2:it
    x(i) = r*x(i-1)*(1-x(i-1));
end

plot_pdf = figure(1);
plot(x);
title("\textbf{Logistic map for} $r=3.5$");
xlabel("Iterations $i$");
ylabel("$x(i)$");
grid on;
grid minor;

% Save pdf
set(plot_pdf, 'Units', 'Centimeters');
pos = get(plot_pdf, 'Position');
set(plot_pdf, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Centimeters', ...
    'PaperSize',[pos(3), pos(4)]);
print(plot_pdf, 'logistic_map_r35.pdf', '-dpdf', '-r0');

% Save png
print(gcf,'logistic_map_r35.png','-dpng','-r600');
```

B Logistic Map for $r = 3.5 - 4$

```
%% Exercise 1: Logistic map

%--------------------------------------------------------------------------%
% Logistic map
%--------------------------------------------------------------------------%

% Date: 27/02/2021
% Author/s: Yi Qiang Ji Zhang
% Subject: Nonlinear Systems, Chaos and Control in Engineering
% Professor: Antonio Pons & Cristina Masoller

% Clear workspace, command window and close windows
clear;
close all;
clc;
% Set interpreter to latex
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');

%% Logistic map plot

% Declare function
logisticMap = @(x,r) r*x*(1-x);

% Number of iterations for each r
numIter = 400;
% Range of r
rMin = 3.5;
rMax = 4;
% Number of points of R
numR = 501;
% Initial condition
x0 = 0.5;

% Last 30% constant values (when it is in steady state)
k = 0.3*numIter;

% Create vector of R's
R = linspace(rMin, rMax, numR);
% Length of R
m = size(R, 2);

% Define points
points = zeros(numIter-k, m);

% Loop through each R
for i = 1:m
    % Calculate logistic map points
    x = iterateFunction(x0, @(x) logisticMap(x,R(i)), numIter);
    % Matrix of points (for each colum R, get the points of logistic map)
    points(:,i) = x(k+1:end);
end

plot_pdf = figure(1);
hold on;
title("\textbf{Logistic map}");
```

```
plot(R, points, '.k');
box on
grid on
grid minor
xlabel("$r$");
ylabel("x(i)");
hold off;


% Save pdf
set(plot_pdf, 'Units', 'Centimeters');
pos = get(plot_pdf, 'Position');
set(plot_pdf, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Centimeters', ...
    'PaperSize',[pos(3), pos(4)]);
print(plot_pdf, 'logistic_map_r_35_4.pdf', '-dpdf', '-r1000');

% Save png
print(gcf,'logistic_map_r_35_4.png','-dpng','-r600');
```

```
function x = iterateFunction(x0, logisticMap, n)

% Define solution vector
x = zeros(n,1);
% Set initial condition (each R)
x(1) = x0;

% Iterate logistic map function
for i = 2:n
    x(i) = logisticMap(x(i-1));
end

end
```

C EXERCISE 4

D Neuron model simulation

D.1 Neuron model for different values of I and initial condition $V_0 = -80$

```
%% Exercise 5: Neuron model
%
%-------------------------------------------------------------------------%
% Simulation of a Neuron model with different I
%-------------------------------------------------------------------------%

% Date: 02/03/2021
% Author/s: Yi Qiang Ji Zhang
% Subject: Nonlinear Systems, Chaos and Control in Engineering
% Professor: Antonio Pons & Cristina Masoller

% Clear workspace, command window and close windows
clear;
close all;
```

```
clc;

% Set interpreter to latex
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');

%% Neuron model

% Constants and Parameters
C = 10;
g_Na = 74;
I = -40:10:40;
V_1_2 = 1.5;
g_L = 19;
k = 16;
E_L = -67;
E_Na = 60;

% Functions
f2 = @(V) (1 + exp((V_1_2 - V)/k))^(-1);
f1 = @(t,V,I) I - g_L*(V - E_L) - g_Na*f2(V)*(V - E_Na);

% Numerical data
h = [0.001]; % Time steps (dt)
t_final = 10; % time units

for j=1:length(h)
% N_steps(j) = ceil(t_final/h(j)); % Number of steps, rounds up with ceil
N_steps(j) = length(0:h(j):t_final)-1;
end

% Initial conditions
initial_cond = -80;

% For each h (dt)
for j=1:length(h)

    t(j,1) = 0; % We begin at t=0 s
    for I_counter = 1:length(I)
    for k = 1:length(initial_cond)
    V(j,1) = initial_cond(k); % x(0) = 1

        % Euler method's update loop
        for i=1:N_steps(j)
            t(j,i+1) = t(j,i) + h(j);
            V(j,i+1) = V(j,i) + f1(t(j,i),V(j,i),I(I_counter))*h(j);
        end
    end
    plot_pdf = figure(1);
    plot(t(j,:),V(j,:));
    hold on;
    end
end
xlim([0 5])
xlabel('Time units')
ylabel('Numerical Solution')
title('\textbf{Neuron model for $V_0=-80$ and different values of I}')
legend('$I=-40$', '$I=-30$', '$I=-20$', '$I=-10$', '$I=0$', '$I=+10$', ...
    '$I=+20$', ...
    '$I=+30$', '$I=+40$', 'location','east');
```

```
box on
grid minor
hold off;

% Save pdf
set(plot_pdf, 'Units', 'Centimeters');
pos = get(plot_pdf, 'Position');
set(plot_pdf, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Centimeters', ...
    'PaperSize',[pos(3), pos(4)]);
print(plot_pdf, 'neuron_model_V0_m80.pdf', '-dpdf', '-r0');

% Save png
print(gcf,'neuron_model_V0_m80.png','-dpng','-r600');
```

E Neuron model for different values of V_0 and parameter $I = 10$

```
%% Exercise 5: Neuron model
%
%--------------------------------------------------------------------------%
% Simulation of a Neuron model with different initial conditions
%--------------------------------------------------------------------------%

% Date: 02/03/2021
% Author/s: Yi Qiang Ji Zhang
% Subject: Nonlinear Systems, Chaos and Control in Engineering
% Professor: Antonio Pons & Cristina Masoller

% Clear workspace, command window and close windows
clear;
close all;
clc;

% Set interpreter to latex
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');

%% Neuron model

% Constants and Parameters
C = 10;
g_Na = 74;
I = 10;
V_1_2 = 1.5;
g_L = 19;
k = 16;
E_L = -67;
E_Na = 60;

% Functions
f2 = @(V) (1 + exp((V_1_2 - V)/k))^(-1);
f1 = @(t,V) I - g_L*(V - E_L) - g_Na*f2(V)*(V - E_Na);

% Numerical data
h = [0.001]; % Time steps (dt)
t_final = 1; % time units

```

```
for j=1:length(h)
% N_steps(j) = ceil(t_final/h(j)); % Number of steps, rounds up with ceil
N_steps(j) = length(0:h(j):t_final)-1;
end

% Initial conditions
initial_cond = -60:10:40;

% For each h (dt)
for j=1:length(h)

    t(j,1) = 0; % We begin at t=0 s
    for k = 1:length(initial_cond)
    V(j,1) = initial_cond(k); % x(0) = 1

        % Euler method's update loop
        for i=1:N_steps(j)
            t(j,i+1) = t(j,i) + h(j);
            V(j,i+1) = V(j,i) + f1(t(j,i),V(j,i))*h(j);
        end
    plot_pdf = figure(1);
    plot(t(j,:),V(j,:));
    hold on;
    end
end
xlim([0 1])
xlabel('Time units')
ylabel('Numerical Solution')
title('\textbf{Neuron model for $I=10$ and different initial conditions}')
legend('$V_0=-60$', '$V_0=-50$', '$V_0=-40$', '$V_0=-30$', '$V_0=-20$', ...
    '$V_0=-10$', '$V_0=0$', '$V_0=+10$', '$V_0=+20$', '$V_0=+30$', ...
        '$V_0=+40$', ...
    'location','east');
box on
grid minor
hold off;

% Save pdf
set(plot_pdf, 'Units', 'Centimeters');
pos = get(plot_pdf, 'Position');
set(plot_pdf, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Centimeters', ...
    'PaperSize',[pos(3), pos(4)]);
print(plot_pdf, 'neuron_model_I_10.pdf', '-dpdf', '-r0');

% Save png
print(gcf,'neuron_model_I_10.png','-dpng','-r600');
```

F Laser turn on simulation

F.1 Neuron model for different values of I and initial condition $V_0 = -80$

```
%% Laser turn on simulation

%-------------------------------------------------------------------------%
% Exercise 6_1: Simulate the 'turn on' when r is constant, r>r star=0.
%-------------------------------------------------------------------------%
```

```
6
7  % Date: 09/03/2021
8  % Author/s: Yi Qiang Ji Zhang
9  % Subject: Nonlinear Systems, Chaos and Control in Engineering
10 % Professor: Antonio Pons & Cristina Masoller
11
12 % Clear workspace, command window and close windows
13 clear;
14 close all;
15 clc;
16 % Set interpreter to latex
17 set(groot,'defaultAxesTickLabelInterpreter','latex');
18 set(groot,'defaulttextinterpreter','latex');
19 set(groot,'defaultLegendInterpreter','latex');
20
21 %% Laser turn on
22
23 % Constants and Parameters
24 r = 1:1:3;
25
26 % Function handle
27 f1 = @(t,x,r) r*x - x^2;
28
29 % 1.1 Numerical data
30 dt = [0.001]; % Time steps (dt)
31 t_final = 20; % time units
32
33 V_steps = zeros(1,length(dt));
34
35 for j=1:length(dt)
36 % N_steps(j) = ceil(t_final/h(j)); % Number of steps, rounds up with ceil
37 N_steps(j) = length(0:dt:t_final)-1;
38 end
39
40 % Initial conditions
41 x0 = 0.01;
42
43 % For each h (dt)
44 for j=1:length(dt)
45
46     t(j,1) = 0; % We begin at t=0 s
47     for k = 1:length(x0)
48     x(j,1) = x0(k); % x(0) = 1
49
50     for k=1:length(r)
51         % Euler method's update loop
52         for i=1:N_steps(j)
53             t(j,i+1) = t(j,i) + dt(j);
54             x(j,i+1) = x(j,i) + f1(t(j,i),x(j,i),r(k))*dt(j);
55         end
56         plot_pdf = figure(1);
57         plot(t(j,:),x(j,:));
58         hold on;
59     end
60     end
61 end
62 xlim([0 20])
63 xlabel('Time units')
64 ylabel('x')
65 title('\textbf{Laser turn on}')
66 legend('$r=1$', '$r=2$', '$r=3$','location','east')
```

```
box on
grid minor
hold off;

% Save pdf
set(plot_pdf, 'Units', 'Centimeters');
pos = get(plot_pdf, 'Position');
set(plot_pdf, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Centimeters', ...
    'PaperSize',[pos(3), pos(4)]);
print(plot_pdf, 'laser_turn_on_v.pdf', '-dpdf', '-r0');

% Save png
print(gcf,'laser_turn_on_v.png','-dpng','-r600');
```

```
%% Laser turn on simulation

%--------------------------------------------------------------------------%
% Exercise 6_2: Bifuracation diagram by plotting x(t=50) vs r.
%--------------------------------------------------------------------------%

% Date: 09/03/2021
% Author/s: Yi Qiang Ji Zhang
% Subject: Nonlinear Systems, Chaos and Control in Engineering
% Professor: Antonio Pons & Cristina Masoller

% Clear workspace, command window and close windows
clear;
close all;
clc;
% Set interpreter to latex
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');

%% Plot 2

% Constants and Parameters
r = -1:0.01:1;
h = [0 0.01];

t0 = 0;
dt = [0.1]; % Time steps (dt)
t_final = 50; % time units

N_steps = length(t0:dt:t_final)-1;

f1 = @(t,x,r,h) r*x - x^2 + h;

t(1:length(r),1) = 0;
x(1:length(r),1) = 0.01;

% Loop through each h
for k = 1:length(h)
    % Loop through each r
    for j=1:length(r)
        % Loop through time steps
        for i=1:N_steps
            t(j,i+1) = t(j,i) + dt;
            x(j,i+1) = x(j,i) + f1(t(j,i),x(j,i),r(j),h(k))*dt;
```

```
        end
    x_r(k,j)=x(j,end);
    end
plot_pdf = figure(1);
plot(r,x_r(k,:));
hold on
end

xlabel('r')
ylabel('x')
title('\textbf{Laser turn on bifurcation diagram}')
legend('$h=0$', '$h=0.01$', 'location', 'northwest')
box on
grid minor
hold off;

% Save pdf
set(plot_pdf, 'Units', 'Centimeters');
pos = get(plot_pdf, 'Position');
set(plot_pdf, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Centimeters', ...
    'PaperSize',[pos(3), pos(4)]);
print(plot_pdf, 'laser_turn_on_bifurcation_diagram_h.pdf', '-dpdf', '-r0');

% Save png
print(gcf,'laser_turn_on_bifurcation_diagram_h.png','-dpng','-r600');
```

```
%% Laser turn on simulation
%-------------------------------------------------------------------------%
% Exercise 6_3: Simulate the equation with r increasing linearly in time.
% Consider different variation rate (v) and/or different
% initial value of the parameter (r0).
%-------------------------------------------------------------------------%

% Date: 09/03/2021
% Author/s: Yi Qiang Ji Zhang
% Subject: Nonlinear Systems, Chaos and Control in Engineering
% Professor: Antonio Pons & Cristina Masoller

% Clear workspace, command window and close windows
clear;
close all;
clc;
% Set interpreter to latex
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');

%% Laser turn on

% Constants and Parameters
r0 = -1;
v = [0.1 0.01]; % Variation rate

% Time domain
t0 = 0; % Initial time
dt = [0.01]; % Time steps (dt)
t_final = 1000; % time units
N_steps = length(t0:dt:t_final)-1; % Number of time steps

```

```
% Function handle
f1 = @(t,x,r) r*x - x^2;

% Initial conditions
t(1:length(v),1) = 0;
x(1:length(v),1) = 0.01;
r_x(1:length(v),1) = r0;

% Loop through each h
for j = 1:length(v)
    % Loop through each r

    for i=1:N_steps
        t(j,i+1) = t(j,i) + dt;
        r_x(j,i+1) = r_x(j,i) + v(j)*dt;
        x(j,i+1) = x(j,i) + f1(t(j,i),x(j,i),r_x(j,i))*dt;
    end
plot_pdf = figure(1);
plot(r_x(j,:),x(j,:));
hold on
end

xlim([-1 4])
xlabel('r')
ylabel('x')
title('\textbf{Laser turn on simulation}')
legend('$v=0.1$','$v=0.01$','location','northwest')
box on
grid minor
hold off;

% Save pdf
set(plot_pdf, 'Units', 'Centimeters');
pos = get(plot_pdf, 'Position');
set(plot_pdf, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Centimeters', ...
    'PaperSize',[pos(3), pos(4)]);
print(plot_pdf, 'laser_turn_on_v_linear.pdf', '-dpdf', '-r0');

% Save png
print(gcf,'laser_turn_on_v_linear.png','-dpng','-r600');
```

```
%% Laser turn on simulation
%-------------------------------------------------------------------------%
% Exercise 6_3: Simulate the equation with r increasing linearly in time.
% Consider different variation rate (v) and/or different
% initial value of the parameter (r0).
%-------------------------------------------------------------------------%

% Date: 09/03/2021
% Author/s: Yi Qiang Ji Zhang
% Subject: Nonlinear Systems, Chaos and Control in Engineering
% Professor: Antonio Pons & Cristina Masoller

% Clear workspace, command window and close windows
clear;
close all;
clc;
% Set interpreter to latex
set(groot,'defaultAxesTickLabelInterpreter','latex');
```

```
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');

%% Plot 2

% Constants and Parameters
r0 = [-1 -2 -3];
v = 0.1; % Variation rate


% Time domain
t0 = 0; % Initial time
dt = [0.01]; % Time steps (dt)
t_final = 1000; % time units
N_steps = length(t0:dt:t_final)-1; % Number of time steps

% Function handle
f1 = @(t,x,r) r*x - x^2;

% Initial conditions
t(1:length(r0),1) = 0;
x(1:length(r0),1) = 0.01;

% Loop through each h
for j = 1:length(r0)
    % Loop through each r
    r_x(1:length(r0),1) = r0(j);
    for i=1:N_steps
        t(j,i+1) = t(j,i) + dt;
        r_x(j,i+1) = r_x(j,i) + v*dt;
        x(j,i+1) = x(j,i) + f1(t(j,i),x(j,i),r_x(j,i))*dt;
    end
plot_pdf = figure(1);
plot(r_x(j,:),x(j,:));
hold on
end

xlim([-1 4])
xlabel('r')
ylabel('x')
title('\textbf{Laser turn on simulation}')
legend('$r_0=-1$','$r_0=-2$', '$r_0=-3$','location','northwest')
box on
grid minor
hold off;

% Save pdf
set(plot_pdf, 'Units', 'Centimeters');
pos = get(plot_pdf, 'Position');
set(plot_pdf, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Centimeters', ...
    'PaperSize',[pos(3), pos(4)]);
print(plot_pdf, 'laser_turn_on_r0.pdf', '-dpdf', '-r0');

% Save png
print(gcf,'laser_turn_on_r0.png','-dpng','-r600');

% Duplicate plot
a1 = gca;
f2 = figure(2);
a2 = copyobj(a1,f2);
xlabel('r')
```

```
ylabel('x')
title('\textbf{Laser turn on simulation}')
legend('$r_0=-1$','$r_0=-2$', '$r_0=-3$','location','northwest')
box on
xlim([-3 1])
ylim([-5e-3 15e-3])

% Save pdf
set(f2, 'Units', 'Centimeters');
pos = get(f2, 'Position');
set(f2, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Centimeters', ...
    'PaperSize',[pos(3), pos(4)]);
print(f2, 'laser_turn_on_r0_zoom.pdf', '-dpdf', '-r0');

% Save png
print(gcf,'laser_turn_on_r0_zoom.png','-dpng','-r600');
```

G Adler equation

G.1 Neuron model for different values of I and initial condition $V_0 = -80$

```
%% Adler Equation

%-------------------------------------------------------------------------%
% Exercise 9 : Solve Adler's equation with w=1, a=0.99 and theta(0)=pi/2
%-------------------------------------------------------------------------%

% Date: 12/03/2021
% Author/s: Yi Qiang Ji Zhang
% Subject: Nonlinear Systems, Chaos and Control in Engineering
% Professor: Antonio Pons & Cristina Masoller

% Clear workspace, command window and close windows
clear;
close all;
clc;
% Set interpreter to latex
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');

%% Adler equation

% Constants and Parameters
w = 1;
a = 0.99;

% Function handle
f1 = @(t,x,a,w) w-a*sin(x);

% 1.1 Numerical data
dt = [0.01]; % Time steps (dt)
t_final = 150; % time units

for j=1:length(dt)
% N_steps(j) = ceil(t_final/h(j)); % Number of steps, rounds up with ceil
```

```
N_steps(j) = length(0:dt:t_final)-1;
end

% Initial conditions
x0 = pi/2;

% For each h (dt)
for j=1:length(dt)
    t(j,1) = 0; % We begin at t=0 s
    for k = 1:length(x0)
    x(j,1) = x0(k); % x(0) = 1

        % Euler method's update loop
        for i=1:N_steps(j)
            t(j,i+1) = t(j,i) + dt(j);
            x(j,i+1) = x(j,i) + f1(t(j,i),x(j,i),a,w)*dt(j);
        end
        plot_pdf = figure(1);
        plot(t(j,:),x(j,:));
        hold on;
    end
end
xlim([0 150])
xlabel('Time units')
ylabel('$\theta$')
title('\textbf{Adler equation}')
box on
grid minor
hold off;

% Save pdf
set(plot_pdf, 'Units', 'Centimeters');
pos = get(plot_pdf, 'Position');
set(plot_pdf, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Centimeters', ...
    'PaperSize',[pos(3), pos(4)]);
print(plot_pdf, 'adler_eq_t_theta.pdf', '-dpdf', '-r0');

% Save png
print(gcf,'adler_eq_theta_t_theta.png','-dpng','-r600');
```

```
%% Adler Equation

%-------------------------------------------------------------------------%
% Exercise 9 : Solve Adler's equation with w=1, a=0.99 and theta(0)=pi/2
%-------------------------------------------------------------------------%

% Date: 12/03/2021
% Author/s: Yi Qiang Ji Zhang
% Subject: Nonlinear Systems, Chaos and Control in Engineering
% Professor: Antonio Pons & Cristina Masoller

% Clear workspace, command window and close windows
clear;
close all;
clc;
% Set interpreter to latex
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');
```

```
20
21 %% Adler equation
22
23 % Constants and Parameters
24 w = 1;
25 a = 0:0.01:0.99;
26
27 % Function handle
28 f1 = @(t,x,a,w) w-a*sin(x);
29
30 % 1.1 Numerical data
31 dt = [0.01]; % Time steps (dt)
32 t_final = 150; % time units
33
34 for j=1:length(dt)
35 % N_steps(j) = ceil(t_final/h(j)); % Number of steps, rounds up with ceil
36 N_steps(j) = length(0:dt:t_final)-1;
37 end
38
39 % Initial conditions
40 x0 = pi/2;
41
42 for j = 1:length(a)
43     x(j,1) = x0; % x(0) = 1
44     t(j,1) = 0; % We begin at t=0 s
45     % Euler method's update loop
46     for i=1:N_steps
47         t(j,i+1) = t(j,i) + dt;
48         x(j,i+1) = x(j,i) + f1(t(j,i),x(j,i),a(j),w)*dt;
49     end
50     T(j) = ((t(j,end) -t(j,1))/(x(j,end) - x(j,1)))*(2*pi);
51     plot_pdf = figure(1);
52     figure(1)
53     plot(a(j),T(j),'o');
54     hold on;
55 end
56
57 xlabel('a')
58 ylabel('Period $T$')
59 title('\textbf{Adler equation}')
60 box on
61 grid minor
62
63 % Analytical
64 T_analytical = 2*pi./sqrt(w^2 - a.^2);
65 plot(a,T_analytical)
66
67 % Save pdf
68 set(plot_pdf, 'Units', 'Centimeters');
69 pos = get(plot_pdf, 'Position');
70 set(plot_pdf, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Centimeters', ...
71     'PaperSize',[pos(3), pos(4)]);
72 print(plot_pdf, 'adler_eq_a_T.pdf', '-dpdf', '-r0');
73
74 % Save png
75 print(gcf,'adler_eq_a_T.png','-dpng','-r600');
```

```
1 %% Adler Equation
2
```

```
%--------------------------------------------------------------------------%
% Exercise 9 : With w=1/sqrt(2), calculate the trajectory for an arbitrary
% initial condition.
%--------------------------------------------------------------------------%

% Date: 12/03/2021
% Author/s: Yi Qiang Ji Zhang
% Subject: Nonlinear Systems, Chaos and Control in Engineering
% Professor: Antonio Pons & Cristina Masoller

% Clear workspace, command window and close windows
clear;
close all;
clc;
% Set interpreter to latex
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');

%% Plot 1

% Constants and Parameters
w = 1/sqrt(2);
a = 0.99;

% Function handle
f1 = @(t,x,a,w) w-a*sin(x);

% 1.1 Numerical data
dt = [0.01]; % Time steps (dt)
t_final = 150; % time units

for j=1:length(dt)
% N_steps(j) = ceil(t_final/h(j)); % Number of steps, rounds up with ceil
N_steps(j) = length(0:dt:t_final)-1;
end

% Initial conditions
x0 = [0:0.1:6.5];

% For each h (dt)
for j=1:length(dt)
    t(j,1) = 0; % We begin at t=0 s
    for k = 1:length(x0)

        x(j,1) = x0(k); % x(0) = 1

        % Euler method's update loop
        for i=1:N_steps(j)
            t(j,i+1) = t(j,i) + dt(j);
            x(j,i+1) = x(j,i) + f1(t(j,i),x(j,i),a,w)*dt(j);
        end
        plot_pdf = figure(1);
        plot(t(j,:),x(j,:));
        hold on;

    end
end

xlim([0 15])
ylim([0 8])
```

```
set(gca,'YTick',0:pi/2:3*pi)
set(gca,'YTickLabel',{'$04','$\pi/2$','$\pi$','$3\pi/2$','$2\pi$','$5\pi/2$'})
xlabel('Time units')
ylabel('$\theta$')
title('\textbf{Adler equation}')
% legend()
box on
grid minor
hold off;

% Save pdf
set(plot_pdf, 'Units', 'Centimeters');
pos = get(plot_pdf, 'Position');
set(plot_pdf, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Centimeters', ...
    'PaperSize',[pos(3), pos(4)]);
print(plot_pdf, 'adler_eq_theta_time_init_cond.pdf', '-dpdf', '-r0');

% Save png
print(gcf,'adler_eq_theta_time_init_cond.png','-dpng','-r600');
```

H Delayed Logistic Equation

```
%% Exercise 10

%-------------------------------------------------------------------------%
% Delayed logistic equation
%-------------------------------------------------------------------------%

% Date: 12/03/2021
% Author/s: Yi Qiang Ji Zhang
% Subject: Nonlinear Systems, Chaos and Control in Engineering
% Professor: Antonio Pons & Cristina Masoller

% Clear workspace, command window and close windows
clear;
close all;
clc;
% Set interpreter to latex
set(groot,'defaultAxesTickLabelInterpreter','latex');
set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultLegendInterpreter','latex');

%% Delayed Adler equation

% Paramter tau
tau = [1];

% For different tau values
for i=1:1:length(tau)
    solve_delay(tau(i));
end


% Plot
plot_pdf = figure(1);
xlabel('Time units')
ylabel('$y$')
```

```
36 title('\textbf{Delayed logistic equation for $\tau = 1$}')
37 % legend()
38 box on
39 grid minor
40
41 % Save pdf
42 set(plot_pdf, 'Units', 'Centimeters');
43 pos = get(plot_pdf, 'Position');
44 set(plot_pdf, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Centimeters', ...
45     'PaperSize',[pos(3), pos(4)]);
46 print(plot_pdf, 'delayed_logistic_eq.pdf', '-dpdf', '-r0');
47
48 % Save png
49 print(gcf,'delayed_logistic_eq.png','-dpng','-r600');
50
51
52 % Function definition
53 function solve_delay(tau)
54     ic = [0.5];
55     tspan = [0 100];
56     lambda = 1.8;
57     sol = dde23(@f,tau,ic,tspan);
58     plot(sol.x,sol.y(1,:),'r-')
59     hold on;
60     function v=f(t,y,Z)
61         v = [lambda*y(1).*(1-Z(1))];
62     end
63 end
```

I Stability Diagram

```
1  %% Exercise 11
2
3  %-------------------------------------------------------------------------%
4  % Stability Diagram:
5  % Test the stability diagram by simulating the equation with D=0
6  % and different values of the parameters c and tau.
7  %-------------------------------------------------------------------------%
8
9  % Date: 16/03/2021
10 % Author/s: Yi Qiang Ji Zhang
11 % Subject: Nonlinear Systems, Chaos and Control in Engineering
12 % Professor: Antonio Pons & Cristina Masoller
13
14 % Clear workspace, command window and close windows
15 clear;
16 close all;
17 clc;
18 % Set interpreter to latex
19 set(groot,'defaultAxesTickLabelInterpreter','latex');
20 set(groot,'defaulttextinterpreter','latex');
21 set(groot,'defaultLegendInterpreter','latex');
22
23 %% Delayed Adler equation
24
25 % Paramter tau
26 c = [0.4, -0.4, -1.1, -1.1];
```

```
tau = [1, 25, 30, 0.5];

% For different tau values
for i=1:1:length(tau)
    solve_delay(tau(i),c(i));
end


% Plot
plot_pdf = figure(1);
xlabel('Time units')
ylabel('$x$')
title('\textbf{Stability Diagram}')
legend('$\mathrm{c} = +0.4$, $\tau = 1$', '$\mathrm{c} = -0.4$, $\tau = 25$', ...
    '$\mathrm{c} = -1.1$, $\tau = 30$', '$\mathrm{c} = -1.1$, $\tau = 0.5$', ...
    'location', 'east')
box on
grid minor

% Save pdf
set(plot_pdf, 'Units', 'Centimeters');
pos = get(plot_pdf, 'Position');
set(plot_pdf, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Centimeters', ...
    'PaperSize',[pos(3), pos(4)]);
print(plot_pdf, 'delayed_logistic_eq.pdf', '-dpdf', '-r0');

% Save png
print(gcf,'delayed_logistic_eq.png','-dpng','-r600');


% Function definition
function solve_delay(tau,c)
    ic = [0.5];
    tspan = [0 500];
    lambda = 1.8;
    sol = dde23(@f,tau,ic,tspan);
    plot(sol.x,sol.y(1,:))
    hold on;
    function v=f(t,y,Z)
        v = [y(1)-y(1).^3 + c.*Z(1)];
    end
end
```

References

[1] Borwein, Jonathan M and Bailey, David H and Bailey, David. (2004). Mathematics by experiment: Plausible reasoning in the 21st century. AK Peters Natick, MA.

[2] Masoller, Cristina. (2021a). Bifurcations in 1D systems: Saddle-node, Transcritical, Pitchfork. UPC.

[3] Masoller, Cristina. (2021b). Introduction to dynamical systems, fixed points, linear stability, and numerical solutions of ordinary differential equations. UPC.