Exercise 4

1. **Toggle-switch**

   Use the script provided below to draw a phase portrait of the toggle-switch

```
eq1 = 2/(1+y^3) -x +0
eq2 =2/(1+x^3) -y  + 0
sp=StreamPlot[{eq1,eq2}, {x,-0,3},{y,-0,3}];
p1 = ContourPlot[Evaluate[{0==eq1}], {x,0,3},{y,0,3}, ContourStyle->Magenta];
p2 = ContourPlot[Evaluate[{0==eq2}], {x,0,3},{y,0,3}, ContourStyle->Blue];
Show[sp,p1,p2]
```

a. How many fixpoints does the system have?
b. How many are attractive/stable?

c. Change the model to have a hill coefficient of 1 for one of the repressive functions. Do you still get a toggle-switch? Explain the shape that the nullclines must have two create a toggle switch.

d. Change the model to be encoded by two activating genes. Do you still get a toggle-switch? How does the steady-state depend on the history of the activation? What is the difference to the double negative feedback?

2. **Oscillations:**

The code below described the relaxation oscillator we discussed in the lecture.

a. Run the code in Mathematica and validate that it is oscillating
    Each period can be split into two time domains:
    - A time domain where x is high and y increases.
b. A time domain where x is low and y decreasescreases.
c. Change the degradation rate µ of y.
    - How does the ratio between the duration of each time domain change when you increase µ? When you decrease µ?
    - Explain the intuition in words of why you see this phenomenon with the help of the phase portrait.
    - How does the degradation rate of y determine the oscillatory period?
d. Change the parameter µ to very high and very low levels, until oscillations stop. How are the two bifurcation points and the resulting steady-states different from each other? (us terminology of X high, y low, etc.)

```
params = {β->8, γ ->0.3, σ ->10, ρ->50, μ->0.1}
xlim = 5;
ylim = 30;
xdot = β (1+ρ (x)^2)/(1+(x)^2)-x(μ + σ y)
 ydot = γ (1+ρ (x)^2)/(1+(x)^2)-μ y
sp=StreamPlot[{ xdot, ydot}/.params, {x,0,xlim},{y,0,ylim},StreamColorFunction->"Rainbow"];
p1 = ContourPlot[Evaluate[{0== xdot}/.params], {x,0,xlim},{y,0,ylim}, ContourStyle->Magenta]; (* nullcline x'[t] = 0*)
p2 = ContourPlot[Evaluate[{0==ydot}/.params], {x,0,xlim},{y,0,ylim}, ContourStyle->Blue]; (* nullcline y'[t] = 0*)
tr1 = ParametricPlot[Evaluate[First[{x[t],y[t]}/.NDSolve[{x'[t] ==(xdot/.{x->x[t], y->y[t]}), y'[t] == (ydot/.{x->x[t], y->y[t]}),x[0]==
1,y[0]==1}/.params, {x,y},{t,0,100}]]],{t,0,100},PlotStyle->Red, PlotRange->All ];
tr2 = ParametricPlot[Evaluate[First[{x[t],y[t]}/.NDSolve[{x'[t] ==(xdot/.{x->x[t], y->y[t]}), y'[t] == (ydot/.{x->x[t], y->y[t]}),x[0]==
1,y[0]==1}/.params, {x,y},{t,0,100}]]],{t,0,100},PlotStyle->Red, PlotRange->All ];

sol = NDSolve[{x'[t] ==(xdot/.{x->x[t], y->y[t]}), y'[t] == (ydot/.{x->x[t], y->y[t]}),x[0]== 1,y[0]==1}/.params, {x,y},{t,0,100}];
Plot[Evaluate[{x[t],y[t]}/.sol],{t,0,100}, PlotLegends->{"x","y"}]

Show[p1, p2, sp, tr1, tr2]
```