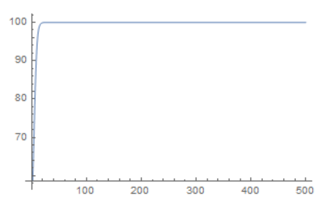
# Problem set 1, Task 1
# Time delayed model with Allee effect
# Computational Biology
# FFR110/FIM740

Mattias Berg, Anita Ullrich

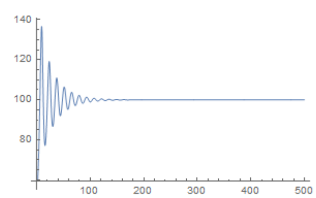
February 11, 2021

## a.)

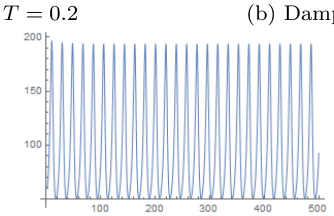
Examples of the three different dynamics obtained in the system (no oscillations, damped oscillations, and stable oscillations), can be seen in Figure 1.




(a) No oscillations, $T = 0.2$




(b) Damped oscillations, $T = 3.3$

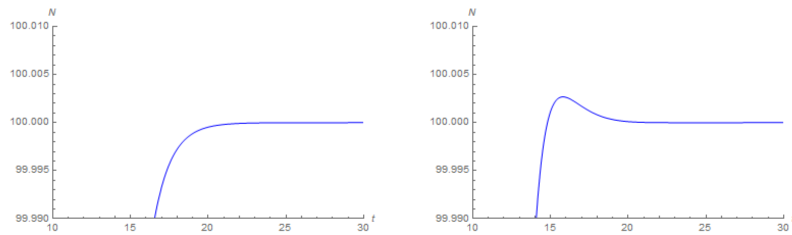



(c) Stable oscillations, $T = 4.4$

Figure 1: Examples for different dynamics, $N(t)$ on the y-axis and $t$ on the x-axis.

## b.)

The estimate of when the system goes from no oscillations to damped oscillations, is between T=0.9 and T=1.0, according to Figure 2. Depending on what you use as a definition of damped oscillations, T=1.0 could also be a critically damped system (only overshooting the target), then the oscillations start between T=1.0 and T=1.1.



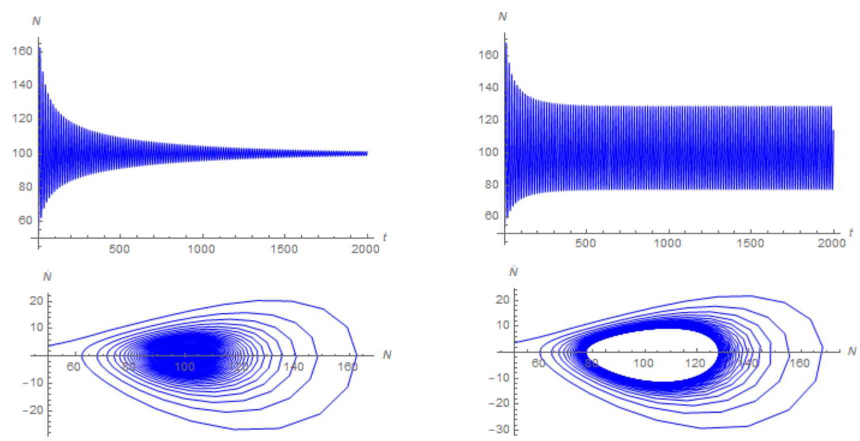

(a) Overdamped system at $T = 0.9$

(b) Underdamped (damped oscillations) system at $T = 1.0$

Figure 2: The oscillations start somewhere between T = 0.9 and T = 1.0

## c.)

From Figure 3, we can see that the Hopf bifurcation occurs some where between T=3.9 and T=4.0.




(a) Damped oscillation at $T = 3.9$

(b) Limit cycle at $T = 4.0$

Figure 3: Between $T = 3.9$ and $T = 4.0$ we have a Hopf-bifurcation as the system changes from damped oscillations to a limit cycle.

## d.)

Consider the steady state $N^* = K$ and assume that

$$N(t) = N^* + \eta(t),$$

where $\eta(t)$ is a small perturbation. Then we get by neglecting higher-order terms

$$\begin{aligned}\dot{N}(t) = \dot{K} + \dot{\eta}(t) = \dot{\eta}(t) = r\left(K + \eta(t)\right)\left(1 - \frac{K + \eta(t-T)}{K}\right)\left(\frac{K + \eta(t)}{A} - 1\right) \\ = r\eta(t-T) - r\frac{K}{A}\eta(t-T) = r\left(1 - \frac{K}{A}\right)\eta(t-T) \\ = -0.4\eta(t-T).\end{aligned}$$

Let $\eta$ have now the form $\eta(t) = c\exp^{\lambda t}$ with $\lambda = \mu + i\omega$ for $\mu, \omega$ real. With this we get

$$\begin{aligned}\lambda &= -0.4\exp^{-\lambda T}, \\ \operatorname{Re}\lambda &= \mu = -0.4\exp^{-\mu T}\cos(\omega T), \\ \operatorname{Im}\lambda &= \omega = 0.4\exp^{-\mu T}\sin(\omega T).\end{aligned}$$

Now we know that a Hopf bifurcation occurs when $\mu$ crosses 0 from negative to positive. If we insert $\mu = 0$ in the equation for the real part we obtain that $\omega T = \frac{\pi}{2}$ as one solution. Of course, cos is 1 for infinitely many solutions, but in this case it is enough to consider the first one. Inserting $\omega T = \frac{\pi}{2}$ into the equation for the imaginary part we obtain $\omega = 0.4$ and hence $T = \frac{\pi}{0.8} \approx 3.92699$. Hence, analytically we obtain $T_H \approx 3.92$.

We found approximately the same $T_H$ with our numerical estimate as we found it lying between 3.9 and 4.0. Therefore, it seems that plotting the results for an appropriate time interval delivers reasonable results in this case.

# Appendix

Course : FFR110/FIM740 Computational Biology

Problem : Problem set 1, Task 1. Time delayed model with Allee effect

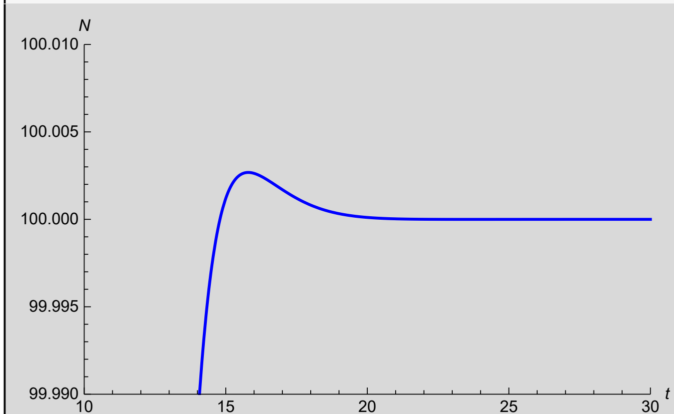
Code : Wolfram Mathematica 12.1

1a/1b)

In[1]:=

```
ClearAll["Global`*"]
Clear[Derivative]

r = 0.1;
A = 20;
K = 100;
T = 1.0;
tmax = 30;

sol = NDSolve[
   {n'[t] == r * n[t] * (1 - n[t - T] / K) * (n[t] / A - 1),
    n[t /; t ≤ 0] == 50,
    n'[0] == 0},
   {n},
   {t, -tmax, tmax}];

n0d[t_] = n[t] /. sol;
n1d[t_] = n'[t] /. sol;

plot0d = Plot[n0d[t], {t, -2, tmax}, AxesLabel → {t, N},
  PlotStyle → Blue, PlotRange → {{10, tmax}, {99.99, 100.01}}]
plot1d = Plot[n1d[t], {t, -2, tmax}, PlotLabels → "N'",
   PlotStyle → Blue, PlotRange → All];

plotparam = ParametricPlot[Evaluate[{n[t], n'[t]} /. sol],
   {t, 0, tmax}, AxesLabel → {N, Ṅ}, PlotStyle → Blue, PlotRange → All];

GraphicsColumn[{plot0d, plotparam}];
```

Out[11]=




1c)

In[15]:=

```
ClearAll["Global`*"]
Clear[Derivative]

r = 0.1;
A = 20;
K = 100;
T = 3.9;
tmax = 2000;

sol = NDSolve[
   {n'[t] == r * n[t] * (1 - n[t - T] / K) * (n[t] / A - 1),
    n[t /; t ≤ 0] == 50,
    n'[0] == 0},
   {n},
   {t, -tmax, tmax}];

n0d[t_] = n[t] /. sol;
n1d[t_] = n'[t] /. sol;

plot0d = Plot[n0d[t], {t, -2, tmax},
   AxesLabel → {t, N}, PlotStyle → Blue, PlotRange → All];
plot1d = Plot[n1d[t], {t, -2, tmax}, PlotLabels → "N'",
   PlotStyle → Blue, PlotRange → All];

plotparam = ParametricPlot[Evaluate[{n[t], n'[t]} /. sol],
   {t, 0, tmax}, AxesLabel → {N, Ṅ}, PlotStyle → Blue, PlotRange → All];

GraphicsColumn[{plot0d, plotparam}]
```

NDSolve: Conditions given at t = 0.` will be interpreted as initial history functions for t /; t ≤ 0..

Out[28]=