Problem 2

Collaborators: Erik Norlin & Hannes Nilsson

a)

For a steady state N^*, we have that

$$F(N^*) = \frac{(r+1)N^*}{1+\left(\frac{N^*}{K}\right)^b} = N^*. \tag{1}$$

Here, we see that $N_1^* = 0$, or by dividing by N^* on both sides:

$$\frac{r+1}{1+\left(\frac{N^*}{K}\right)^b} = 1$$

$$\implies r+1 = 1+\left(\frac{N^*}{K}\right)^b \implies r = \left(\frac{N^*}{K}\right)^b \implies N^* = Kr^{1/b} = N_2^*.$$

b)

$$\Lambda(N) = F'(N) = \frac{\left(1+\left(\frac{N}{K}\right)^b\right)(r+1) - (r+1)Nb\left(\frac{N}{K}\right)^{b-1}\frac{1}{K}}{1+2\left(\frac{N}{K}\right)^b + \left(\frac{N}{K}\right)^2 b}.$$

For $N_1^* = 0$, we get:

$$\Lambda(N_1^* = 0) = r+1,$$

and for $N_2^* = Kr^{1/b}$, what we obtain is:

$$\begin{aligned}
\Lambda(N_2^* = Kr^{1/b}) &= \frac{(1+r)(r+1) - (r+1)Kr^{1/b}br^{(b-1)/b}\frac{1}{K}}{(r+1)^2} \\
&= 1 - \frac{r^{1/b}br^{(b-1)/b}}{r+1} \\
&= 1 - \frac{br^{(b-1+1)/b}}{r+1} \\
&= 1 - \frac{br}{r+1}.
\end{aligned}$$

Since $r > 0$, $\Lambda(N_1^*) > 1$, and so the fixed point will always be unstable without oscillations. Now for the other fixed point, $\Lambda(N_2^*) < 1$, since b and r are both positive, so we will never have an unstable fixed point without oscillations here. However, $\Lambda(N_2^*)$ can take on any value below 1, and so we will observe bifurcations at $\Lambda(N_2^*) = 0$ and $\Lambda(N_2^*) = -1$. For relatively small b, the negative term will be rather small, so that $0 < \Lambda(N_2^*) < 1$ and we have a stable fixed point without oscillations. As the ratio of b/r increases, $\Lambda(N_2^*)$ will pass zero and oscillations start to appear. Finally, for large enough ratio b/r, $\Lambda(N_2^*)$ becomes less than 1 and we observe the fixed point bifurcate into an unstable fixed point with oscillations.

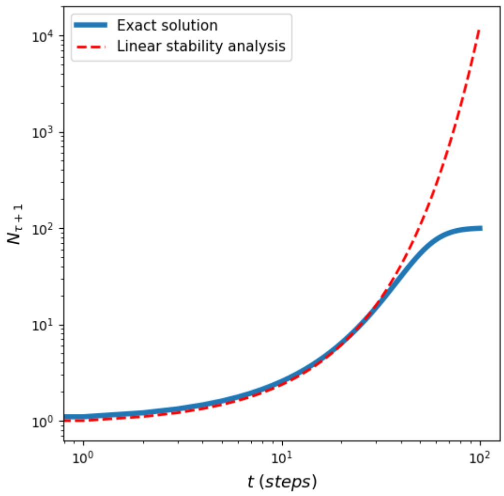
c)

As discussed in part b), only N_2^* will ever be stable, and the transition between stable and unstable occur at $\Lambda(N_2^*) = -1$:

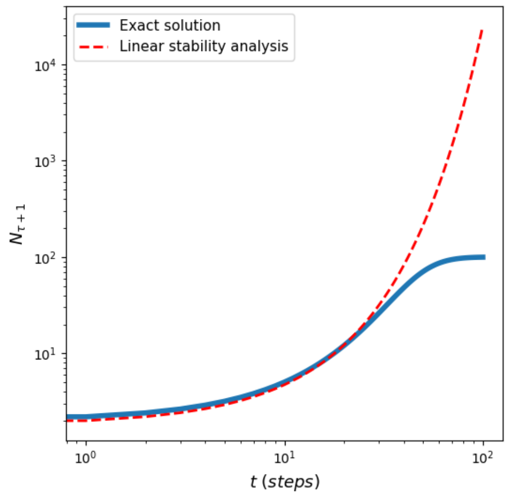
$$1 - \frac{br}{r+1} = -1 \implies \frac{br}{r+1} = 2 \implies b = \frac{2r+2}{r} = 2\left(1 + \frac{1}{r}\right) \iff r = \frac{1}{\frac{b}{2} - 1}.$$

Hence, the transition from a stable to an unstable state happens either as b is increased above $2(1+1/r)$, or as r is decreased below $1/(b/2-1)$.

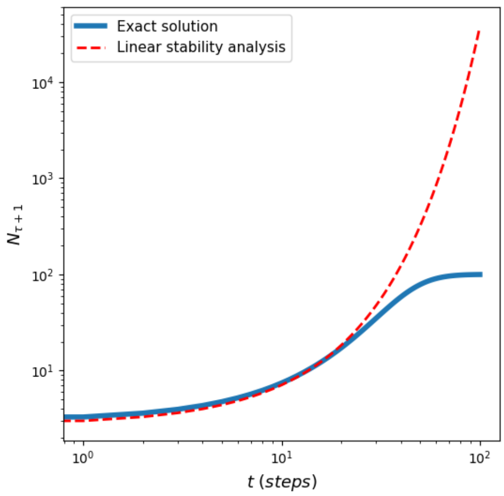
d)



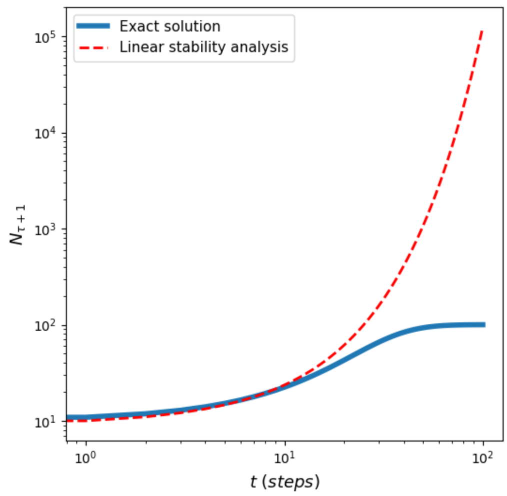
(a) $N_0 = 1$



(b) $N_0 = 2$



(c) $N_0 = 3$



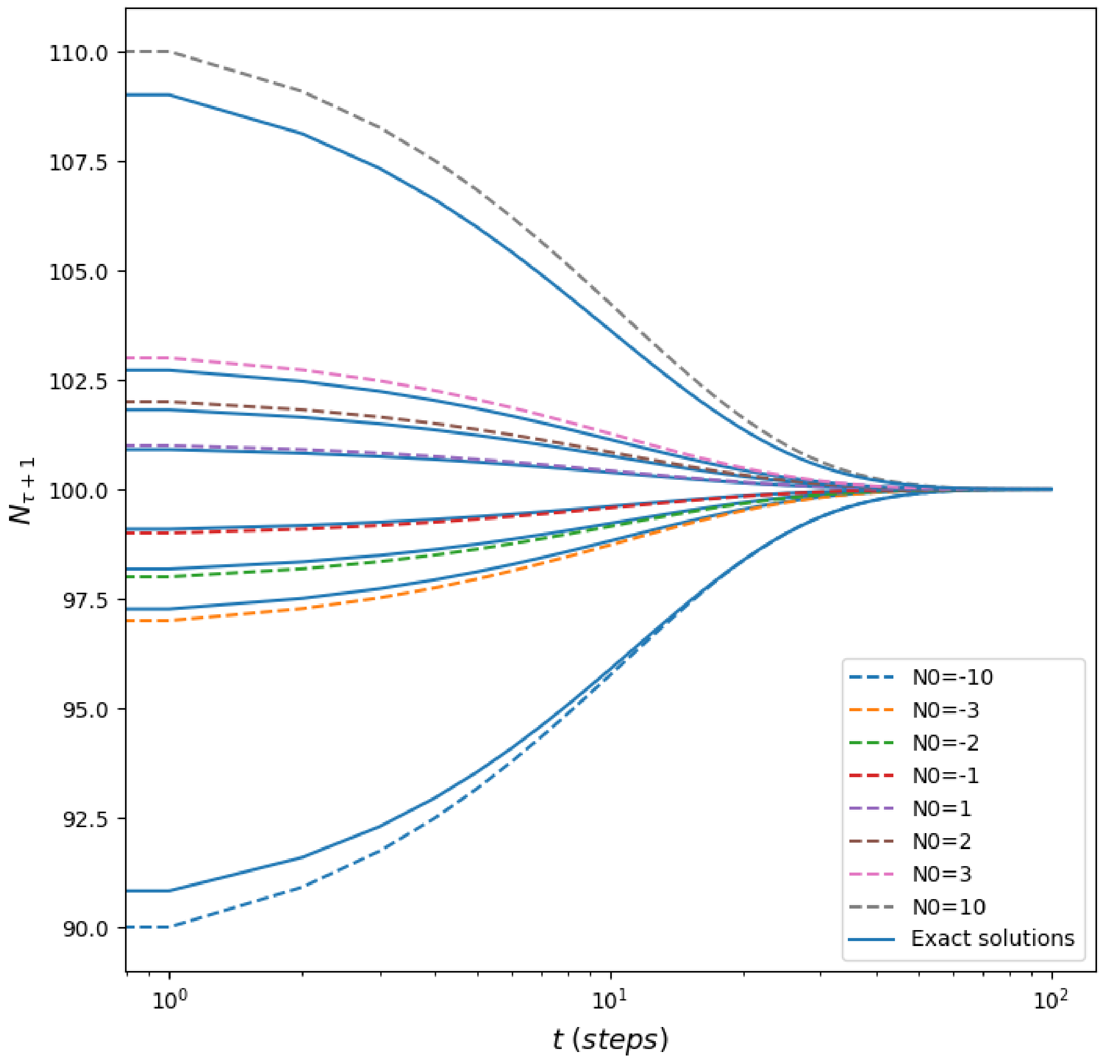
(d) $N_0 = 10$

Figur (1) Approximation of eq. 1 using linear stability analysis around the non-steady state ($N^* = 0$) for different initial conditions versus the exact solution of eq. 1.

e)

The linear stability analysis approximates the true solution somewhat well for around 10-30 time steps (depending on the initial condition) as can be seen in fig. 1. Since higher order terms are neglected the approximation won't follow the true solution as it tapers off at the end, instead the approximation will continue towards infinity. The reason why the approximation follows the true solution at the beginning instead of at the end is because the linearisation was done around $N^* = 0$, i.e. at the very start of the dynamics. As can be seen in the same figure, the initial condition has influence of how well the linear stability analysis approximates the exact solution. As the value of the initial condition is increased, the approximation becomes worse. This is because the approximation grows exponentially and is multiplied with the initial condition. This means that as the value of the initial condition is increased, the exponential term will be multiplied with a greater and greater number. Thus, the approximation will grow and deviate faster from the exact solution as the initial condition increases.

f)



Figur (2) Approximation of eq. 1 using linear stability analysis around the steady state ($N^* = Kr^{1/b} = 100$) for different initial conditions versus the exact solutions of eq. 1.

The linear stability analysis around the steady-state fixed point approximates the exact solution better as the initial condition becomes smaller, similarly to what happened in (d). The same reasoning goes here as before, that the approximation deviates as it goes further away from where it has been linearized, in this case $N^* = Kr^{1/b} = 100$.

Appendix

Code for simulations (Python)

```
import numpy as np
import matplotlib.pyplot as plt
from ddeint import ddeint
import sys
import matplotlib
# matplotlib.use('qtSagg')

T = 100
K = 10**3
r = 0.1
b = 1
eig_1 = 1-b*r/(1+r)
# eig_2 = 1+r
N_star = K*r**(1/b)
delta_N0 = np.array([-10, -3, -2, -1, 1, 2, 3, 10])
# delta_N0 = 10
N0 = N_star + delta_N0
N = N0
N_list = [N0]
N1 = [N0]
N_star_list = [0]

fig, ax = plt.subplots(figsize=(8,8))

for delta_N0_i in delta_N0:

    N0 = N_star + delta_N0_i
    N = N0
    N_list = [N0]
    N1 = [N0]
```

```
    for t in range(T):

        if delta_N0_i != 10:
            N_next = (r+1)*N / (1+(N/K)**b)
            N = N_next
            N_list.append(N)

        N1_next = delta_N0_i*eig_1**t + N_star
        N1.append(N1_next)

        N_star_list.append(t+1)

    ax.plot(N_list, '-', lw=1.5, color='tab:blue')
    ax.plot(N1, '--', lw=1.5, label='N0={}'.format(delta_N0_i))

N0 = N_star + 10
for t in range(T):
    N_next = (r+1)*N / (1+(N/K)**b)
    N = N_next
    N_list.append(N)

ax.plot(N_list, '-', lw=1.5, color='tab:blue', label='Exact solutions')

# for i in range(len(N_list)-1):
#     ax.plot(N_list,lw=1, color='tab:blue')
#     ax.plot(N1[i,:], '--', lw=1, label='$\delta N_0 = {}$'.format(delta_N0[i]))


# ax.plot(N_list, '-', lw=1, color='tab:blue', label='Exact solution')
# ax.plot(N1, '--', lw=1, label='N0={}'.format(delta_N0[0]))

# ax.plot(N_star_list, '--', color='black', label='$N^{*}$ ($Kr^{1/b}$)')
ax.set_xlabel('$t$ $(steps)$', fontsize=13)
ax.set_ylabel(r'$N_{\tau+1}$', fontsize=13)
ax.set_xscale('log')
# ax.set_yscale('log')
# ax.set_ylim([0,105])
ax.legend(loc='lower right', prop={'size': 10})
title = '/1.2f'
```

```
location = r'C:\Users\erikn\OneDrive - Chalmers\Computational Biology\CB HW 1'
plt.savefig(location+title+'.png')
plt.show()
```