Problem set 1, Task 2
Discrete growth models
Computational Biology
FFR110/FIM740

Mattias Berg, Anita Ullrich

February 11, 2021

a.)

General form of discrete models is $N_{t+1} = N_t + \Delta N$. For a steady state we need $\Delta N = 0$. Hence, we get

$$\Delta N = \frac{(r+1)N_t}{1 + \left(\frac{N_t}{K}\right)^b} - N_t = 0 \Leftrightarrow N_1^* = 0 \text{ and } N_2^* = \sqrt[b]{r}K.$$

Depending on b, N_2^* can have negative solutions. Hence, the non-negative steady states are $N_2^* = |\sqrt[b]{r}|K$ as $K, r > 0$.

b.)

Let now $F(N_T) = N_{T+1}$. We assume that $N_T = N^* + v_T$ for $|v_T| << 1$ a small perturbation. Tailor expansion delivers

$$N^* + v_{T+1} = F(N^* + v_T) = F(N^*) + v_T F'(N^*) + O(v_T^2).$$

We can neglect the second and higher order terms as $|v_T| << 1$. Since N^* is a fixed point, we also have $F(N^*) = N^*$ and we use $\lambda = F'(N^*)$. This yields

$$F(N^*) + v_{T+1} = F(N^*) + v_T F'(N^*) \Leftrightarrow v_{T+1} = v_T F'(N^*) = \lambda v_T.$$

Considering the limit $t \to \infty$ we have for $|\lambda| < 1$ that $v_{T+1} = \lambda^T v_0 \to 0$, whereas for $|\lambda| > 1$ that $v_{T+1} = \lambda^T v_0 \to \pm\infty$. Hence, we see that N^* is stable if $-1 < F'(N^*) < 1$ and unstable if $|F'(N^*)| > 1$. As we have as derivation

$$\begin{aligned} F'(N_T) &= \frac{r+1}{1 + (\frac{N_T}{K})^b} - \frac{(r+1)b(\frac{N_T}{K})^b}{(1 + (\frac{N_T}{K})^b)^2} = \frac{(r+1)(1 + (\frac{N_T}{K})^b - b(\frac{N_T}{K})^b)}{(1 + (\frac{N_T}{K})^b)^2} \\ &= -\frac{(r+1)((b-1)(\frac{N_T}{K})^b - 1)}{((\frac{N_T}{K})^b + 1)^2}, \end{aligned}$$

we get

$$\begin{aligned}
\Rightarrow F'(N_1^*) &= -\frac{(r+1)((b-1)(\frac{0}{K})^b-1)}{((\frac{0}{K})^b+1)^2}\\
&= -\frac{(r+1)((b-1)0-1)}{(1)^2}\\
&= -\frac{(r+1)(-1)}{1}\\
&= -(-r-1) = r+1,\\
\Rightarrow F'(N_2^*) &= -\frac{(r+1)((b-1)r-1)}{(r+1)^2} = -\frac{(b-1)r-1}{r+1}.
\end{aligned}$$

With this, N_1^* is a stable fixed point if

$$\begin{aligned}
-1 < r+1 < 1 &\Leftrightarrow -1-1 < r < 1-1\\
&\Leftrightarrow -2 < r < 0
\end{aligned}$$

and unstable otherwise. N_2^* is a stable fixed point if

$$\begin{aligned}
-1 < -\frac{(b-1)r-1}{r+1} < 1 &\Leftrightarrow -(r+1) < -((b-1)r-1) < r+1\\
&\Leftrightarrow r+1 > ((b-1)r-1) > -r-1\\
&\Leftrightarrow r+1 > (b-1)r > -r\\
&\Leftrightarrow \frac{r+2}{r} > b-1 > -1 \Leftrightarrow 0 < b < 2+\frac{2}{r}.
\end{aligned}$$

and unstable otherwise.

c.)

A bifurcation from a stable to an unstable steady state or vice versa occurs when λ is passing -1 or 1. As we have two fixed points we need to investigate the cases $\lambda = \pm 1$ for both fixed points. Lets consider the first fixed point N_1^*:

$$\lambda = F'(N_1^*) = 1 \Leftrightarrow r+1 = 1 \Leftrightarrow r = 0,$$

$$\lambda = F'(N_1^*) = -1 \Leftrightarrow r+1 = -1 \Leftrightarrow r = -2.$$

To summarize for N_1^*, a bifurcation from a stable to an unstable steady state occurs when either $r = 0$ or $r = -2$.
Now let us consider the second fixed point N_2^*:

$$\begin{aligned}
\lambda = F'(N_2^*) = 1 \Leftrightarrow -\frac{(b-1)r-1}{r+1} = 1 &\Leftrightarrow -((b-1)r-1) = (r+1)1 \Leftrightarrow\\
&\Leftrightarrow (1-b)r+1 = r+1 \Leftrightarrow (1-b)r = r \Leftrightarrow 1-b = 1 \Leftrightarrow b = 0
\end{aligned}$$

As we have the assumption that $b \geq 1$, we conclude that $F'(N_2^*)$ cannot take the value 1.

$$\begin{aligned}\lambda = F'(N_2^*) = -1 \Leftrightarrow -\frac{(b-1)r-1}{r+1} = -1 \Leftrightarrow -((b-1)r-1) = -(r+1)1 \Leftrightarrow \\ \Leftrightarrow (1-b)r+1 = -r-1 \Leftrightarrow (1-b)r = -r-2 \\ \Leftrightarrow 1-b = \frac{-r-2}{r} \Leftrightarrow \\ \Leftrightarrow -b = \frac{-r-2}{r} - 1 \Leftrightarrow b = \frac{r+2}{r} + 1\end{aligned}$$

To summarize for N_2^*, a bifurcation from a stable to an unstable steady state only occurs when $b = \frac{r+2}{r} + 1$.

d.)

See Figure 1. Linear part uses the following equation

$$N_{\tau+1} = N_\tau * F'(N_0^*)$$

With N_0 set to different values depending on start point.

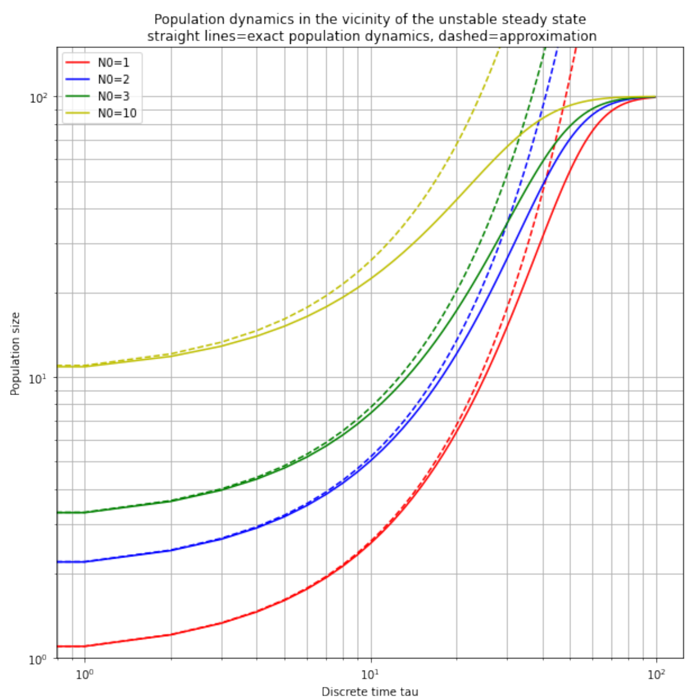

Figure 1

e.)

In the beginning of the simulation the approximations are very close to the exact solution for all starting values. But the larger the starting value the earlier the approximation and exact solution diverge and we get a worse approximation. Especially when the exact solution starts to flatten and converge to the stable steady state, the approximation is not close at all as we continue having exponential growth. So the approximation catches the long term behavior of our population dynamics not at all which suggests that it is not a good approximation. And as already mentioned, the starting value influences how early and how much the exact solution and the approximation diverge, the larger the starting value the faster and stronger is the divergence.

f.)

See Figure 2. Linear part uses the following equation

$$N_{\tau+1} = N_2^* + N_\tau * F'(N_2^*)$$

With N_0 set to different values depending on start point.

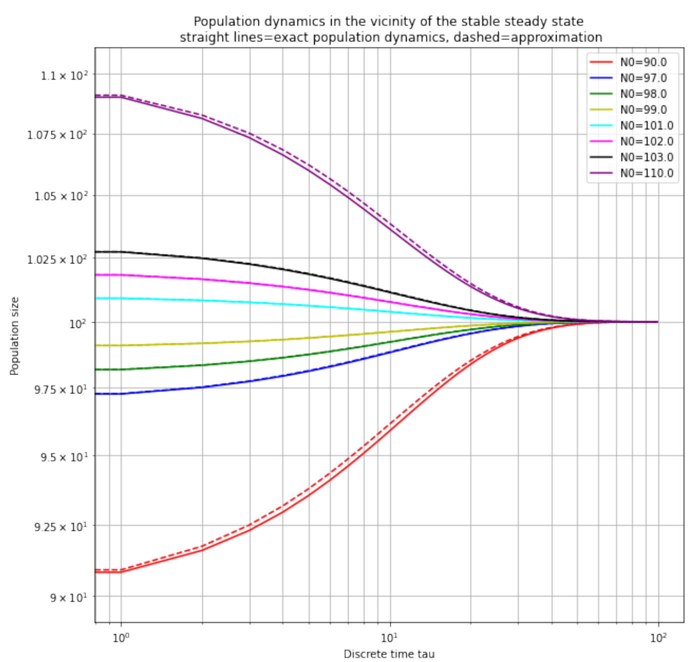

Figure 2

How does the initial perturbation influence the approximation?: For large initial perturbations ($|\delta N_0| = 10$) we can see that the approximation is more off from the exact solution than in the other cases of initial perturbation. But in the end, they both converge to the same population size, regardless of initial perturbation.

Appendix

Code for figure 1

```python
import numpy as np
import matplotlib.pyplot as plt

K = 1000
r = 0.1
b = 1
N0s = [1,2,3,10]
max_t = 100

def f(N,i,t) :
    return ((r+1)*N[i,t])/(1+(N[i,t]/K)**b)

dt = 0.01

Ns = np.ones((len(N0s),max_t))
Lin = np.ones((len(N0s),max_t))
ts = [t for t in range(max_t)]
for i in range(len(N0s)) :
    Ns[i,0] *= N0s[i]
    Lin[i,0] *= N0s[i]

for i in range(len(N0s)) :
    for t in range(1,max_t) :
        Ns[i,t] = f(Ns,i,t-1)
        Lin[i,t] = (r+1) * Lin[i,t-1]

plt.figure(figsize=(10,10))
cols = ['r','b','g','y']
for i in range(len(N0s)) :
    plt.plot(ts,Ns[i,:],label="N0="+str(N0s[i]),c=cols[i])
    plt.plot(ts,Lin[i,:],'--',c=cols[i])
plt.yscale('log')
plt.xscale('log')
plt.ylim(bottom=1,top=150)
plt.xlabel('Discrete time tau')
plt.ylabel('Population size')
plt.title('Population dynamics in the vicinity of the unstable steady state\n st
plt.legend()
plt.grid(True, which="both", ls="-")
plt.show()
```

Code for figure 2

```
import numpy as np
import matplotlib.pyplot as plt

K = 1000
r = 0.1
b = 1
max_t = 100

def f(N,i,t) :
    return ((r+1)*N[i,t])/(1+(N[i,t]/K)**b)

del_N0s = [-10,-3,-2,-1,1,2,3,10]
dt = 0.01
N2 = r**(1/b)*K

Ns = np.ones((len(del_N0s),max_t))
Lin = np.ones((len(del_N0s),max_t))
ts = [t for t in range(max_t)]
for i in range(len(del_N0s)) :
    Ns[i,0] *= N2 + del_N0s[i]
    Lin[i,0] *=  del_N0s[i]

for i in range(len(del_N0s)) :
    for t in range(1,max_t) :
        Ns[i,t] = f(Ns,i,t-1)
        Lin[i,t] = -((b-1)*r-1)/(r+1) * Lin[i,t-1]

Lin += N2

plt.figure(figsize=(10,10))
cols = ['r','b','g','y','cyan','magenta','black','purple']
for i in range(len(del_N0s)) :
    plt.plot(ts,Ns[i,:],label="N0="+str(N2+del_N0s[i]),c=cols[i])
    plt.plot(ts,Lin[i,:],'--',c=cols[i])
plt.yscale('log')
plt.xscale('log')
plt.xlabel('Discrete time tau')
plt.ylabel('Population size')
plt.title('Population dynamics in the vicinity of the stable steady state\n stra
plt.legend()
plt.grid(True, which="both", ls="-")
plt.show()
```