Introduction to Computational Physics – Exercise 8

Simon Groß-Bölting, Lorenz Vogel, Sebastian Willenberg

June 19, 2020

Stability Analysis of Many Species Population Dynamics

In this exercise we study the evolution of 6 populations according to the following equations for population dynamics: 3 predator- (P_i) and 3 prey-species (N_i), all parameters positive, always $i, j = 1, \ldots, 3$:

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = N_i\left(a_i - N_i - \sum_j b_{ij}P_j\right) \qquad \text{and} \qquad \frac{\mathrm{d}P_i}{\mathrm{d}t} = P_i\left(\sum_j c_{ij}N_j - d_i\right) \tag{1}$$

The parameters chosen are $a_1 = 56$, $a_2 = 12$, $a_3 = 35$, $d_1 = 85$, $d_2 = 9$ and $d_3 = 35$; the parameters b_{ij} and c_{ij} are given in matrix form here:

$$b_{ij} = \begin{pmatrix} 20 & 30 & 5 \\ 1 & 3 & 7 \\ 4 & 10 & 20 \end{pmatrix} \qquad \text{and} \qquad c_{ij} = \begin{pmatrix} 20 & 30 & 35 \\ 3 & 3 & 3 \\ 7 & 8 & 20 \end{pmatrix} \tag{2}$$

Notices: The unusual feature here in the equations is that the prey populations N_i have a Verhulst style growth limiting factor in their equations, which limits their growth even if there is no predator (model for limited resources even in absence of predators).

- Please do not try to make the equations dimensionless, just use the numbers given here.
- In this mathematical example the solutions can become negative in some cases, which is unphysical for population dynamics. Nevertheless let us please use this case to show the mathematical features.

Task: What are the fixed points (FP) for the system of equations given above? *Hint 1:* No complicated computations are necessary, the idea is that you should guess the fixed points very easily. Compare our previous examples. *Hint 2:* This time there are three fixed points! In addition to our "usual" ones, there is a third one related to the Verhulst growth limiting factor in the first three equations.

condition for fixed points (FP) in the case of interacting populations:

$$n = n^* \text{ stationary point of the system} \qquad \Longleftrightarrow \qquad f(n^*) := \left.\frac{\mathrm{d}n}{\mathrm{d}\tau}\right|_{n=n^*} = 0 \tag{3}$$

stability analysis in the multi-dimensional case: analogous to one-dimensional, but vector form:

$$\vec{f}(\vec{N}, \vec{P}) = \begin{pmatrix} \dot{\vec{N}} \\ \dot{\vec{P}} \end{pmatrix} \tag{4}$$

with

$$\dot{\vec{N}} = \frac{\mathrm{d}\vec{N}}{\mathrm{d}t} = \begin{pmatrix} \dot{N}_1 \\ \dot{N}_2 \\ \dot{N}_3 \end{pmatrix} = \begin{pmatrix} n_1(N_1, \vec{P}) \\ n_2(N_2, \vec{P}) \\ n_3(N_3, \vec{P}) \end{pmatrix} \qquad \text{and} \qquad \dot{\vec{P}} = \frac{\mathrm{d}\vec{P}}{\mathrm{d}t} = \begin{pmatrix} \dot{P}_1 \\ \dot{P}_2 \\ \dot{P}_3 \end{pmatrix} = \begin{pmatrix} p_1(P_1, \vec{N}) \\ p_2(P_2, \vec{N}) \\ p_3(P_3, \vec{N}) \end{pmatrix} \tag{5}$$

$$n_i(N_i, \vec{P}) := \frac{\mathrm{d}N_i}{\mathrm{d}t} = N_i\left(a_i - N_i - \sum_j b_{ij}P_j\right) \tag{6}$$

$$p_i(P_i, \vec{N}) := \frac{\mathrm{d}P_i}{\mathrm{d}t} = P_i\left(\sum_j c_{ij}N_j - d_i\right) \tag{7}$$

$$\begin{cases} \dot{N}_1 = N_1\,(56 - N_1 - 20P_1 - 30P_2 - 5P_3) \\ \dot{N}_2 = N_2\,(12 - N_2 - P_1 - 3P_2 - 7P_3) \\ \dot{N}_3 = N_3\,(35 - N_3 - 4P_1 - 10P_2 - 20P_3) \\ \dot{P}_1 = P_1\,(20N_1 + 30N_2 + 35N_3 - 85) \\ \dot{P}_2 = P_2\,(3N_1 + 3N_2 + 3N_3 - 9) \\ \dot{P}_3 = P_3\,(7N_1 + 8N_2 + 10N_3 - 35) \end{cases} \tag{8}$$

$$d_i = \sum_j c_{ij} N_j^* \quad \Longrightarrow \quad \left\{ \begin{array}{l} 85 = 20N_1^* + 30N_2^* + 35N_3^* \\ 9 = 3N_1^* + 3N_2^* + 3N_3^* \\ 35 = 7N_1^* + 8N_2^* + 10N_3^* \end{array} \right\} \quad \Longrightarrow \quad N_i^* = 1 \tag{9}$$

$$N_i^* = a_i - \sum_j b_{ij} P_j^* \quad \xrightarrow{N_i^* = 1} \quad \left\{ \begin{array}{l} 1 = 56 - (20P_1^* + 30P_2^* + 5P_3^*) \\ 1 = 12 - (P_1^* + 3P_2^* + 7P_3^*) \\ 1 = 35 - (4P_1^* + 10P_2^* + 20P_3^*) \end{array} \right\} \quad \Longrightarrow \quad P_i^* = 1 \tag{10}$$

$$\vec{P}^* = \begin{pmatrix} P_1^* \\ P_2^* \\ P_3^* \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad \Longrightarrow \quad N_i^* = a_i - \sum_j b_{ij} P_j^* = a_i \quad \Longrightarrow \quad \vec{N}^* = \begin{pmatrix} N_1^* \\ N_2^* \\ N_3^* \end{pmatrix} = \begin{pmatrix} 56 \\ 12 \\ 35 \end{pmatrix} \tag{11}$$

"usual" fixed points:

$$N_i^* = P_i^* = 0 \qquad \text{and} \qquad N_i^* = P_i^* = 1 \tag{12}$$

third fixed point:

$$P_i^* = 0 \qquad \text{and} \qquad N_i^* = a_i \tag{13}$$

$$\begin{pmatrix} \vec{N}^* \\ \vec{P}^* \end{pmatrix} \text{ fixed point} \qquad \Longleftrightarrow \qquad \vec{f}(\vec{N}^*,\, \vec{P}^*) = \vec{0} \tag{14}$$

Jacobi matrix:

$$\mathrm{D}\vec{f} = \begin{pmatrix} \frac{\partial n_1}{\partial N_1} & 0 & 0 & \frac{\partial n_1}{\partial P_1} & \frac{\partial n_1}{\partial P_2} & \frac{\partial n_1}{\partial P_3} \\ 0 & \frac{\partial n_2}{\partial N_2} & 0 & \frac{\partial n_2}{\partial P_1} & \frac{\partial n_2}{\partial P_2} & \frac{\partial n_2}{\partial P_3} \\ 0 & 0 & \frac{\partial n_3}{\partial N_3} & \frac{\partial n_3}{\partial P_1} & \frac{\partial n_3}{\partial P_2} & \frac{\partial n_3}{\partial P_3} \\ \frac{\partial p_1}{\partial N_1} & \frac{\partial p_1}{\partial N_2} & \frac{\partial p_1}{\partial N_3} & \frac{\partial p_1}{\partial P_1} & 0 & 0 \\ \frac{\partial p_2}{\partial N_1} & \frac{\partial p_2}{\partial N_2} & \frac{\partial p_2}{\partial N_3} & 0 & \frac{\partial p_2}{\partial P_2} & 0 \\ \frac{\partial p_3}{\partial N_1} & \frac{\partial p_3}{\partial N_2} & \frac{\partial p_3}{\partial N_3} & 0 & 0 & \frac{\partial p_3}{\partial P_3} \end{pmatrix} \tag{15}$$

$$\frac{\partial n_i}{\partial N_i} = \frac{\partial}{\partial N_i}\left[N_i \left(a_i - N_i - \sum_j b_{ij} P_j \right) \right] = \left(a_i - N_i - \sum_j b_{ij} P_j \right) - N_i \tag{16}$$

$$\frac{\partial p_i}{\partial P_i} = \frac{\partial}{\partial P_i}\left[P_i \left(\sum_j c_{ij} N_j - d_i \right) \right] = \sum_j c_{ij} N_j - d_i \tag{17}$$

$$\frac{\partial n_i}{\partial P_j} = \frac{\partial}{\partial P_j}\left[N_i \left(a_i - N_i - \sum_j b_{ij} P_j \right) \right] = -N_i b_{ij} \tag{18}$$

$$\frac{\partial p_i}{\partial N_j} = \frac{\partial}{\partial N_j}\left[P_i \left(\sum_j c_{ij} N_j - d_i \right) \right] = P_i c_{ij} \tag{19}$$

$$\mathbf{A} := \mathrm{D}\vec{f}(N_i^* = P_i^* = 1) = \begin{pmatrix} -1 & 0 & 0 & -20 & -30 & -5 \\ 0 & -1 & 0 & -1 & -3 & -7 \\ 0 & 0 & -1 & -4 & -10 & -20 \\ 20 & 30 & 35 & 0 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 & 0 \\ 7 & 8 & 20 & 0 & 0 & 0 \end{pmatrix} \tag{20}$$

Task: What is the Jacobi matrix **A** at the non-trivial fixed point? (non-trivial FP means here that *all* elements are unequal to zero. There is only one FP with this property)

Task: Determine the eigenvalues λ_i and eigenvectors $\mathbf{v}_i$ ($i = 1, \ldots, 6$) of $\mathbf{A}$ for this fixed point. Choose an initial state

$$\mathbf{n}_0 = \sum_{i=1}^{6} c_i \mathbf{v}_i \qquad \text{with} \qquad c_1 = c_2 = 3\,, \quad c_3 = c_4 = 1\,, \quad c_5 = -5\,, \quad c_6 = 0.1 \tag{21}$$

Plot and discuss the time-dependent evolution of the six populations (given by this linear model); in particular what about oscillations, growth or decay?

Table 1: Eigenvalues and eigenvectors

eigenvalues λ_i	eigenvectors $\mathbf{v}_i$
$-0.5 + 33.626\mathrm{i}$	
$-0.5 - 33.626\mathrm{i}$	
$-0.5 + 7.679\mathrm{i}$	
$-0.5 - 7.679\mathrm{i}$	
-1.136	
$+0.136$	

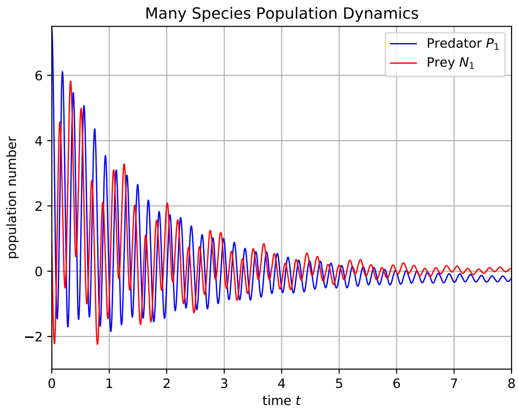
If λ_i is an eigenvalue of $\mathbf{A}$ with the eigenvector $\mathbf{v}_i$, then $\exp(\lambda_i t)$ is eigenvalue of $\exp(\mathbf{A}t)$.

$$\mathbf{n}(t) = \exp(\mathbf{A}t)\mathbf{n}_0 \qquad \text{with} \qquad \mathbf{A} = \mathrm{D}\vec{f} \tag{22}$$

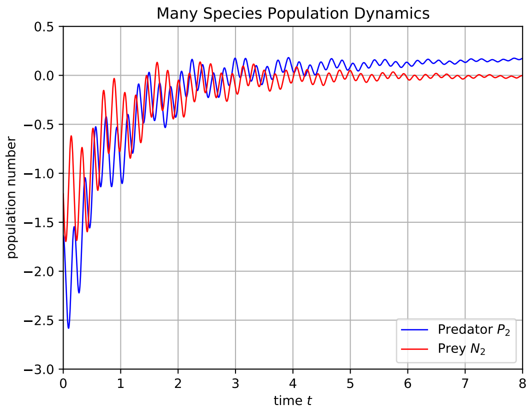
$\mathbf{n}_0$ is our initial condition.

$$\mathbf{n}(t) = \exp(\mathbf{A}t) \sum_{i=1}^{6} c_i \mathbf{v}_i = \sum_{i=1}^{6} c_i \exp(\mathbf{A}t) \mathbf{v}_i = \sum_{i=1}^{6} c_i \exp(\lambda_i t) \mathbf{v}_i \tag{23}$$

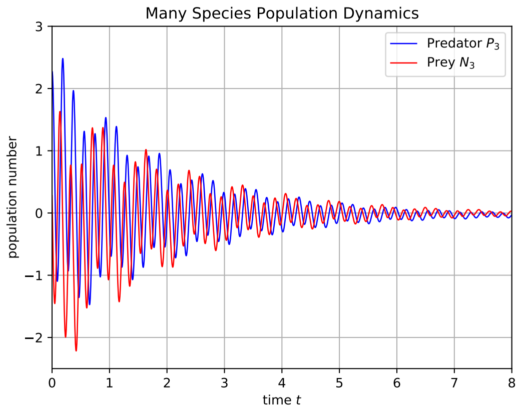
We have an oscillating solution.



(a) Predator P_1 and prey N_1



(b) Predator P_2 and prey N_2



(c) Predator P_3 and prey N_3

Figure 1: Time-dependent evolution of the six populations

Python-Code 1: Numerical determination of the time-dependent evolution of the six populations for a given initial state using the eigenvalues and the eigenvectors

```
# -*- coding: utf-8 -*-
"""
Introduction to Computational Physics
- Exercise 08:  Many Species Population Dynamics
- Group: Simon Groß-Bölting, Lorenz Vogel, Sebastian Willenberg
"""

import numpy as np; import matplotlib.pyplot as plt
import numpy.linalg as linalg

def initial_state(coeff,ev):
    ''' Function to set the initial state for given coefficients
        Input:  coefficients (coeff), eigenvalues (ew) and eigenvectors (ev)'''
    n = np.zeros(len(ev[:,0]))
    for i in range(0,len(n)): n = n+coeff[i]*ev[:,i]

    # if all imaginary part are zero, then only use the real parts
    complex = np.iscomplex(n)
    if complex.all() == False: n = np.real(n)
    return n

def time_evolution(t,coeff,ew,ev):
    ''' Function to compute the time-dependent evolution of the population
        Input: time (t), coefficients (coeff), eigenvalues (ew)
               and eigenvectors (ev) '''
    evolution = np.zeros((len(t),len(ew)))
    for i in range(0,len(t)):
        for j in range(0,len(ew)):
```

```
            evolution[i,:] = evolution[i,:]+coeff[j]*np.exp(ew[j]*t[i])*ev[:,j]
    return evolution

# set the Jacobi matrix A at the non-trivial fixed point
A = np.array([[-1,0,0,-20,-30,-5],[0,-1,0,-1,-3,-7],[0,0,-1,-4,-10,-20],
              [20,30,35,0,0,0],[3,3,3,0,0,0],[7,8,20,0,0,0]])

# determine the eigenvalues (ew) and eigenvectors (ev) of A
eigenvalues, eigenvectors = linalg.eig(A)

# print the results (eigenvalues and eigenvectors)
for i in range(0,len(eigenvalues)):
    print('Eigenvalue:  {}'.format(eigenvalues[i]))
    print('Eigenvector: {}\n'.format(eigenvectors[:,i]))

# set the initial state with the given coefficients
c = np.array([3,3,1,1,-5,0.1])
print(initial_state(c,eigenvectors))

# compute and plot the time-dependent evolution of the six populations
t = np.linspace(0,10,10000) # time
time_evolution = time_evolution(t,c,eigenvalues,eigenvectors)


fig1, ax1 = plt.subplots()
ax1.plot(t,time_evolution[:,3], 'b-', linewidth=1, label=r'Predator $P_1$')
ax1.plot(t,time_evolution[:,0], 'r-', linewidth=1, label=r'Prey $N_1$')
ax1.set_xlim((0,8)); ax1.set_ylim((-3,7.5)); ax1.legend(loc='upper right')

fig2, ax2 = plt.subplots()
ax2.plot(t,time_evolution[:,4], 'b-', linewidth=1, label=r'Predator $P_2$')
ax2.plot(t,time_evolution[:,1], 'r-', linewidth=1, label=r'Prey $N_2$')
ax2.set_xlim((0,8)); ax2.set_ylim((-3,0.5)); ax2.legend(loc='lower right')

fig3, ax3 = plt.subplots()
ax3.plot(t,time_evolution[:,5], 'b-', linewidth=1, label=r'Predator $P_3$')
ax3.plot(t,time_evolution[:,2], 'r-', linewidth=1, label=r'Prey $N_3$')
ax3.set_xlim((0,8)); ax3.set_ylim((-2.5,3)); ax3.legend(loc='upper right')

for ax in [ax1,ax2,ax3]:
    ax.set_title('Many Species Population Dynamics'); ax.grid()
    ax.set_xlabel(r'time $t$'); ax.set_ylabel(r'population number')
fig1.savefig('figures/Population-Time-Evolution-01.pdf', format='pdf')
fig2.savefig('figures/Population-Time-Evolution-02.pdf', format='pdf')
fig3.savefig('figures/Population-Time-Evolution-03.pdf', format='pdf')
plt.show(); plt.clf(); plt.close()
```