

Introduction to Computational Physics – Exercise 8

Simon Groß-Böltung, 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 $i, j = 1, \dots, 3$:

$$\frac{dN_i}{dt} = N_i \left(a_i - N_i - \sum_j b_{ij} P_j \right) \quad \text{and} \quad \frac{dP_i}{dt} = P_i \left(\sum_j c_{ij} N_j - d_i \right) \quad (1)$$

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:

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

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:

$$n = n^* \text{ stationary point of the system} \iff f(n^*) := \frac{dn}{d\tau} \Big|_{n=n^*} = 0 \quad (3)$$

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

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

with

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

$$n_i(N_i, \vec{P}) := \frac{dN_i}{dt} = N_i \left(a_i - N_i - \sum_j b_{ij} P_j \right) \quad (6)$$

$$p_i(P_i, \vec{N}) := \frac{dP_i}{dt} = P_i \left(\sum_j c_{ij} N_j - d_i \right) \quad (7)$$

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

$$d_i = \sum_j c_{ij} N_j^* \implies \begin{cases} 85 = 20N_1^* + 30N_2^* + 35N_3^* \\ 9 = 3N_1^* + 3N_2^* + 3N_3^* \\ 35 = 7N_1^* + 8N_2^* + 10N_3^* \end{cases} \implies N_i^* = 1 \quad (9)$$

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

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

“usual” fixed points:

$$N_i^* = P_i^* = 0 \quad \text{and} \quad N_i^* = P_i^* = 1 \quad (12)$$

third fixed point:

$$P_i^* = 0 \quad \text{and} \quad N_i^* = a_i \quad (13)$$

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

Jacobi matrix:

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

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

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

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

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

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

Task: What is the Jacobi matrix \mathbf{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 ($i = 1, \dots, 6$) of \mathbf{A} for this fixed point. Choose an initial state

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

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
$-0.5 + 33.626i$	
$-0.5 - 33.626i$	
$-0.5 + 7.679i$	
$-0.5 - 7.679i$	
-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 $\exp(\mathbf{A}t)$.

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

\mathbf{n}_0 is our initial condition.

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

We have an oscillating solution.

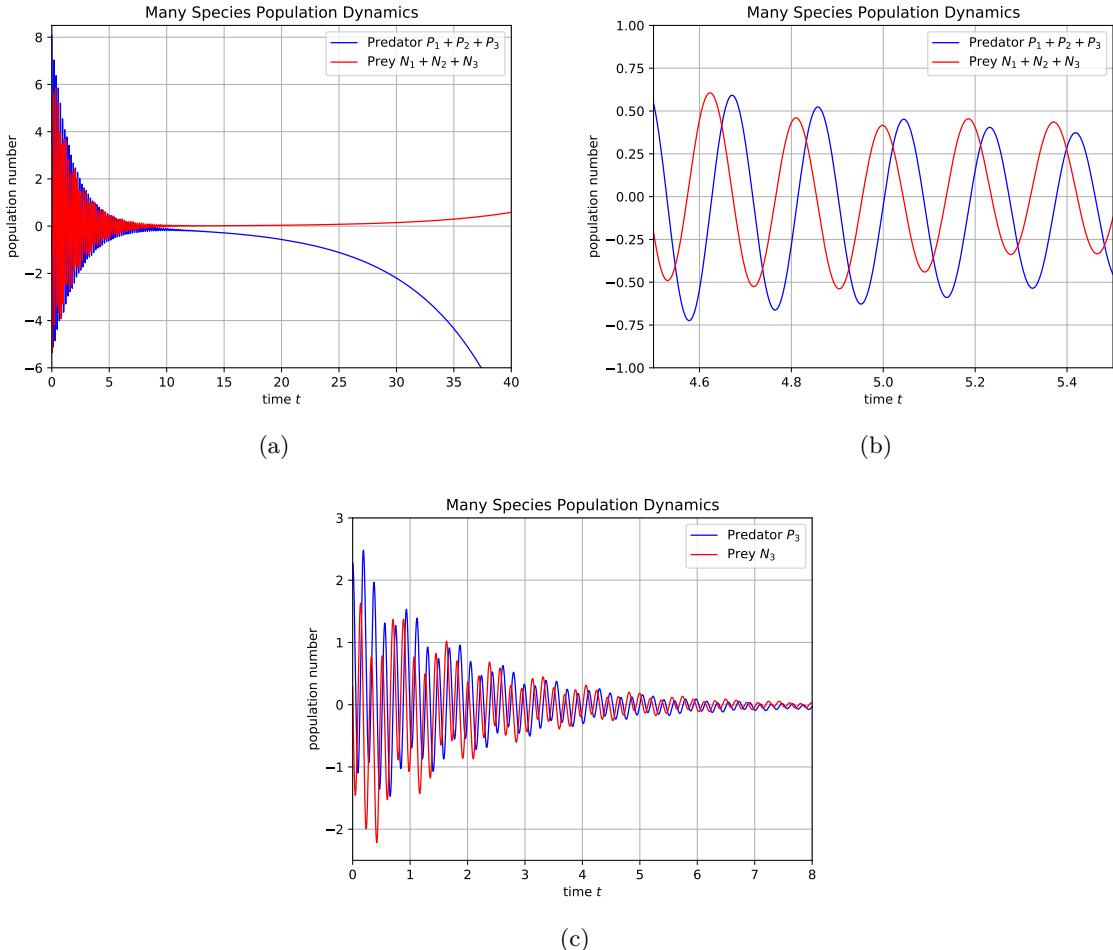


Figure 1: Time-dependent evolution of the six populations

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

```

1 # -*- coding: utf-8 -*-
2 """
3 Introduction to Computational Physics
4 - Exercise 08: Many Species Population Dynamics
5 - Group: Simon Groß-Böltig, Lorenz Vogel, Sebastian Willenberg
6 """
7
8 import numpy as np; import matplotlib.pyplot as plt
9 import numpy.linalg as linalg
10
11 def initial_state(coeff, ev):
12     ''' Function to set the initial state for given coefficients
13         Input: coefficients (coeff), eigenvalues (ew) and eigenvectors (ev) '''
14     n = np.zeros(len(ev[:, 0]))
15     for i in range(0, len(n)): n = n+coeff[i]*ev[:, i]
16
17     # if all imaginary part are zero, then only use the real parts
18     complex = np.iscomplex(n)
19     if complex.all() == False: n = np.real(n)
20     return n
21
22 def time_evolution(t, coeff, ew, ev):
23     ''' Function to compute the time-dependent evolution of the population
24         Input: time (t), coefficients (coeff), eigenvalues (ew)
25             and eigenvectors (ev) '''
26     evolution = np.zeros((len(t), len(ew)))
27     for i in range(0, len(t)):
28         for j in range(0, len(ew)):

```

```

29     evolution[i,:] = evolution[i,:]+coeff[j]*np.exp(ew[j]*t[i])*ev[:,j]
30
31     return evolution
32
33 # set the Jacobi matrix A at the non-trivial fixed point
34 A = np.array([[-1,0,0,-20,-30,-5],[0,-1,0,-1,-3,-7],[0,0,-1,-4,-10,-20],
35               [20,30,35,0,0,0],[3,3,3,0,0,0],[7,8,20,0,0,0]])
36
37 # determine the eigenvalues (ew) and eigenvectors (ev) of A
38 eigenvalues, eigenvectors = linalg.eig(A)
39
40 # print the results (eigenvalues and eigenvectors)
41 for i in range(0,len(eigenvalues)):
42     print('Eigenvalue: {}' .format(eigenvalues[i]))
43     print('Eigenvector: {}' .format(eigenvectors[:,i]))
44
45 # set the initial state with the given coefficients
46 c = np.array([3,3,1,1,-5,0.1])
47 print(initial_state(c,eigenvectors))
48
49 # compute and plot the time-dependent evolution of the six populations
50 t = np.linspace(0,40,10000) # time
51 time_evolution = time_evolution(t,c,eigenvalues,eigenvectors)
52
53 predator = time_evolution[:,3]+time_evolution[:,4]+time_evolution[:,5]
54 prey = time_evolution[:,0]+time_evolution[:,1]+time_evolution[:,2]
55
56 fig1, ax1 = plt.subplots()
57 ax1.plot(t,predator, 'b-', linewidth=1, label=r'Predator $P_1+P_2+P_3$')
58 ax1.plot(t,prey, 'r-', linewidth=1, label=r'Prey $N_1+N_2+N_3$')
59 ax1.set_xlim((0,40)); ax1.set_ylim((-6,8.5))
60
61 fig2, ax2 = plt.subplots()
62 ax2.plot(t,predator, 'b-', linewidth=1, label=r'Predator $P_1+P_2+P_3$')
63 ax2.plot(t,prey, 'r-', linewidth=1, label=r'Prey $N_1+N_2+N_3$')
64 ax2.set_xlim((0,10)); ax2.set_ylim((-6,8.5))
65
66 fig3, ax3 = plt.subplots()
67 ax3.plot(t,predator, 'b-', linewidth=1, label=r'Predator $P_1+P_2+P_3$')
68 ax3.plot(t,prey, 'r-', linewidth=1, label=r'Prey $N_1+N_2+N_3$')
69 ax3.set_xlim((4.5,5.5)); ax3.set_ylim((-1,1))
70
71 for ax in [ax1,ax2,ax3]:
72     ax.set_title('Many Species Population Dynamics')
73     ax.set_xlabel(r'time $t$'); ax.set_ylabel(r'population number')
74     ax.grid(); ax.legend(loc='upper right')
75 fig1.savefig('figures/Population-Time-Evolution -01.pdf', format='pdf')
76 fig2.savefig('figures/Population-Time-Evolution -02.pdf', format='pdf')
77 fig3.savefig('figures/Population-Time-Evolution -02.pdf', format='pdf')
78 plt.show(); plt.clf(); plt.close()

```