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 $i, j = 1, \ldots, 3$:

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

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:

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

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.

First of all, we abbreviate the functions occurring in Eq. (1) with $n_i(N_i, \vec{P})$ and $p_i(P_i, \vec{N})$:

$$n_i(N_i, \vec{P}) := \frac{\mathrm{d}N_i}{\mathrm{d}t} = N_i\left(a_i - N_i - \sum_j b_{ij}P_j\right) \qquad \text{with} \qquad \vec{P} = \begin{pmatrix} P_1 & P_2 & P_3 \end{pmatrix}^T \qquad (3)$$

$$p_i(P_i, \vec{N}) := \frac{\mathrm{d}P_i}{\mathrm{d}t} = P_i\left(\sum_j c_{ij}N_j - d_i\right) \qquad \text{with} \qquad \vec{N} = \begin{pmatrix} N_1 & N_2 & N_3 \end{pmatrix}^T \qquad (4)$$

We now define the vectorial function $\vec{f}(\vec{N}, \vec{P})$ for arguments $\vec{N}$ (prey-species) and $\vec{P}$ (predator-species):

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

The condition for fixed points (FP) in the case of interacting populations (multi-dimensional case) is more or less analogous to the one-dimensional case, but in vector form:

$$\begin{pmatrix} \vec{N}^* \\ \vec{P}^* \end{pmatrix} \text{ fixed point} \quad \Longleftrightarrow \quad \vec{f}(\vec{N}^*, \vec{P}^*) = \vec{0} \quad \Longleftrightarrow \quad \left.\frac{\mathrm{d}\vec{N}}{\mathrm{d}t}\right|_{\vec{N}=\vec{N}^*} = \vec{0} \quad \wedge \quad \left.\frac{\mathrm{d}\vec{P}}{\mathrm{d}t}\right|_{\vec{P}=\vec{P}^*} = \vec{0} \qquad (6)$$

If we use the values given by the matrices b_{ij} and c_{ij}, the function $\vec{f}(\vec{N}, \vec{P})$ is given by the following system of equations:

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

If we look at Eq. (1), the **first fixed point** is obviously given by $N_i^* = P_i^* = 0$, or in vector form:

$$N_i^* = P_i^* = 0 \qquad \Longleftrightarrow \qquad \begin{pmatrix} \vec{N}^* & \vec{P}^* \end{pmatrix}^T = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T \tag{8}$$

We get the second fixed point by first looking at for which vector $\vec{N}^*$ the system of equations $\dot{\vec{P}}$ becomes zero (except for $\vec{P}^* = 0$):

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

With the condition $\vec{N}^* = 1$ we get for the components of $\vec{P}^*$:

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

The **second fixed point** is therefore given by $N_i^* = P_i^* = 1$, or in vector form:

$$N_i^* = P_i^* = 1 \qquad \Longleftrightarrow \qquad \begin{pmatrix} \vec{N}^* & \vec{P}^* \end{pmatrix}^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}^T \tag{11}$$

We have now received our two "usual" results for fixed points ($N_i^* = P_i^* = 0$ and $N_i^* = P_i^* = 1$). By setting $\vec{P}^* = 0$ and looking at the condition in Eq. (10) again, we get the corresponding vector $\vec{N}^*$:

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

Our result for the **third fixed point** is:

$$N_i^* = a_i \quad \wedge \quad P_i^* = 0 \qquad \Longleftrightarrow \qquad \begin{pmatrix} \vec{N}^* & \vec{P}^* \end{pmatrix}^T = \begin{pmatrix} 56 & 12 & 35 & 0 & 0 & 0 \end{pmatrix}^T \tag{13}$$

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)

Because of

$$\frac{\partial n_i}{\partial N_j} = 0 \qquad \text{and} \qquad \frac{\partial p_i}{\partial P_j} = 0 \qquad \text{for} \qquad i \neq j \tag{14}$$

we get for the Jacobi matrix of the given system of equations for population dynamics:

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

In this case, the entries on the main diagonal are given by:

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

and

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

We receive the following values for the remaining entries:

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

and

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

The only non-trivial fixed point (*all* elements are unequal to zero) of this system is:

$$N_i^* = P_i^* = 1 \qquad \Longleftrightarrow \qquad \left(\vec{N}^* \quad \vec{P}^*\right)^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}^T \tag{20}$$

The Jacobi matrix $\mathbf{A}$ at the non-trivial fixed point therefore is:

$$\mathbf{A} := \mathrm{D}\vec{f}(N_i^* = P_i^* = 1) = \begin{pmatrix} -1 & 0 & 0 & -20 & -30 & -5 \\ 0 & -1 & 0 & -1 & -3 & -7 \\ 0 & 0 & -1 & -4 & -10 & -20 \\ 20 & 30 & 35 & 0 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 & 0 \\ 7 & 8 & 20 & 0 & 0 & 0 \end{pmatrix} \tag{21}$$

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

$$\mathbf{n}_0 = \sum_{i=1}^{6} c_i \mathbf{v}_i \qquad \text{with} \qquad c_1 = c_2 = 3\,, \quad c_3 = c_4 = 1\,, \quad c_5 = -5\,, \quad c_6 = 0.1 \tag{22}$$

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

From the previous task we know that the Jacobi matrix $\mathbf{A}$ at the non-trivial fixed point is given by the expression in Eq. (21). We can numerically determine the eigenvalues λ_i and eigenvectors $\mathbf{v}_i$ of the matrix $\mathbf{A}$ using the `SciPy` function `linalg.eig` (see Python-Code 1). The results are shown in Table 1.

Table 1: Numerically determined eigenvalues and eigenvectors

eigenvalues λ_i	eigenvectors $\mathbf{v}_i^T$
$-0.5 + 33.626\mathrm{i}$	$\begin{pmatrix} -0.008 + 0.543\mathrm{i} & -0.001 + 0.093\mathrm{i} & -0.004 + 0.294\mathrm{i} & 0.712 & 0.083 & 0.310 \end{pmatrix}$
$-0.5 - 33.626\mathrm{i}$	$\begin{pmatrix} -0.008 - 0.543\mathrm{i} & -0.001 - 0.093\mathrm{i} & -0.004 - 0.294\mathrm{i} & 0.712 & 0.083 & 0.310 \end{pmatrix}$
$-0.5 + 7.679\mathrm{i}$	$\begin{pmatrix} -0.869 & 0.138 & 0.342 & 0.011 + 0.167\mathrm{i} & 0.010 + 0.151\mathrm{i} & -0.016 - 0.240\mathrm{i} \end{pmatrix}$
$-0.5 - 7.679\mathrm{i}$	$\begin{pmatrix} -0.869 & 0.138 & 0.342 & 0.011 - 0.167\mathrm{i} & 0.010 - 0.151\mathrm{i} & -0.016 + 0.240\mathrm{i} \end{pmatrix}$
-1.136	$\begin{pmatrix} -0.538 & 0.295 & 0.075 & -0.644 & 0.442 & -0.092 \end{pmatrix}$
$+0.136$	$\begin{pmatrix} 0.081 & -0.045 & -0.011 & -0.815 & 0.560 & -0.116 \end{pmatrix}$

We now choose the initial state:

$$\mathbf{n}_0 := \mathbf{n}(t = 0) = \sum_{i=1}^{6} c_i \mathbf{v}_i \qquad \text{with} \qquad c_1 = c_2 = 3\,, \quad c_3 = c_4 = 1\,, \quad c_5 = -5\,, \quad c_6 = 0.1 \tag{23}$$

By inserting the numerically determined eigenvectors $\mathbf{v}_i$ (see Table 1) we get:

$$\mathbf{n}_0 = \begin{pmatrix} 0.909 & -1.210 & 0.278 & 7.431 & -1.637 & 2.275 \end{pmatrix}^T \tag{24}$$
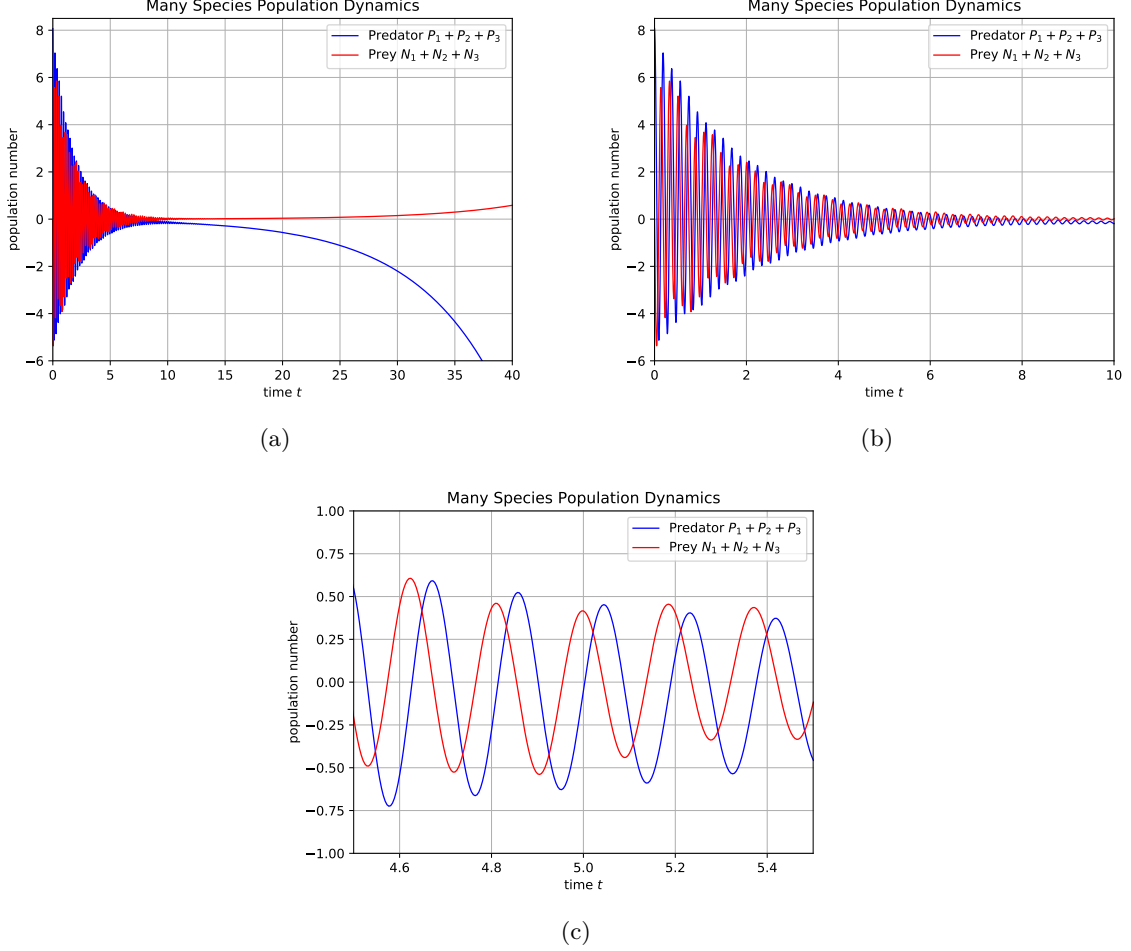
(a)

(b)

(c)

Figure 1: Time-dependent evolution of the six populations in different time intervals. We have plotted the sum of the three predator species and the sum of the three prey species against the time t.

The time-dependent evolution $\mathbf{n}(t)$ of our initial condition $\mathbf{n}_0$ is given by:

$$\mathbf{n}(t) = \exp(\mathbf{A}t)\mathbf{n}_0 \qquad \text{with} \qquad \mathbf{A} = \mathrm{D}\vec{f}(N_i^* = P_i^* = 1) \tag{25}$$

We know that, if λ_i is an eigenvalue of $\mathbf{A}$ with the eigenvector $\mathbf{v}_i$, then $\exp(\lambda_i t)$ is an eigenvalue of $\exp(\mathbf{A}t)$. With this we get for the time-dependent evolution $\mathbf{n}(t)$ of $\mathbf{n}_0$:

$$\mathbf{n}(t) = \exp(\mathbf{A}t) \sum_{i=1}^{6} c_i \mathbf{v}_i = \sum_{i=1}^{6} c_i \exp(\mathbf{A}t) \mathbf{v}_i = \sum_{i=1}^{6} c_i \exp(\lambda_i t) \mathbf{v}_i \tag{26}$$

The time-dependent evolution of the six populations given by this linear model is shown in Fig. 1. Both populations of the predators and the prey fluctuate around zero in a damped harmonic pattern. The oscillating behavior occurs because some of the eigenvalues λ_i are complex. In both cases the fluctuation decreases over time. After about $t = 10$ both populations arrive at zero.

Both populations fluctuate because they are influencing each other. If the prey population is at a maximum there is more food for the predators and therefore the predator population will rise and the prey population will fall. Because the prey population is now falling there are more predators than prey and therefore the predator population will fall again. This causes the prey population to increase because of less predators and the cycle continues.

This fluctuation pattern is caused by influence of the prey (or predator) population in the predator (or prey) defining function. That causes the prey (or predator) population to depend on the predator (or prey) population. The fact that the population numbers become negative is unphysical for population dynamics and caused by the model we use. The next thing we notice is that both populations decrease over time. This is due to limited resources which are introduced by the prey population depending on

4

its own population and causing it to decrease. In reality this is always the case because limited resources cause the population to reach a limit and therefore stop growing.

Shortly after $t = 10$ the prey population begins to rise exponentially and the predator population begins to fall exponentially. This effect has no physical meaning because the predator population is decreasing to negative infinity which is not possible and the prey population is rising to positive infinity. This effect is caused by the predator population having no positive extremum at some point ($t = 0$) and therefore increasing the prey population. This causes the predator population to decrease even more because of the way its derivative is defined.

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

```python
1  # -*- coding: utf-8 -*-
2  """
3  Introduction to Computational Physics
4  - Exercise 08:  Many Species Population Dynamics
5  - Group: Simon Groß-Bölting, 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) 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     return evolution
31
32 # set the Jacobi matrix A at the non-trivial fixed point
33 A = np.array([[-1,0,0,-20,-30,-5],[0,-1,0,-1,-3,-7],[0,0,-1,-4,-10,-20],
34               [20,30,35,0,0,0],[3,3,3,0,0,0],[7,8,20,0,0,0]])
35
36 # determine the eigenvalues (ew) and eigenvectors (ev) of A
37 eigenvalues, eigenvectors = linalg.eig(A)
38
39 # print the results (eigenvalues and eigenvectors)
40 for i in range(0,len(eigenvalues)):
41     print('Eigenvalue:  {}'.format(eigenvalues[i]))
42     print('Eigenvector: {}\n'.format(eigenvectors[:,i]))
43
44 # set the initial state with the given coefficients
45 c = np.array([3,3,1,1,-5,0.1])
46 print('Initial state: {}'.format(initial_state(c,eigenvectors)))
47
48 # compute and plot the time-dependent evolution of the six populations
49 t = np.linspace(0,40,10000) # time
50 time_evolution = time_evolution(t,c,eigenvalues,eigenvectors)
51
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 $P_1+P_2+P_3$')
```

```python
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-03.pdf', format='pdf')
78  plt.show(); plt.clf(); plt.close()
```