TTK4225 2023 - Assignment 02

January 1, 2024

1 TTK4225 Assignment 2

1.0.1 Note

Some of the points below may be either solved by pen and paper, or – for who feels brave but also learn something that will be very useful in your theses – via symbolic computing (e.g., sympy: https://docs.sympy.org/latest/tutorials/intro-tutorial/solvers.html).

The most important thing in this assignment is to understand the logics behind each procedural step you take. Thus for the teachers it does not matter which way you choose, the important is that you understand what you do.

1.0.2 Introduction

Consider a Lotka Volterra system with inputs, i.e., a system defined by: - the state variables y_{prey} (preys) and y_{pred} (predators), - the inputs u_{prey} (human intervention on the preys population) and u_{pred} (human intervention on the predators population), - the dynamics

$\dot{y}_{prey} = \alpha y_{prey} - \beta y_{prey} y_{pred} - u_{prey}$

$\dot{y}_{pred} = -\gamma y_{pred} + \delta y_{prey} y_{pred} - u_{pred}$

1.0.3 Question 1

Observing these two species of animals in the forests, you note that when there is no food around, the amount of time it takes for the predators to decrease their population of a factor e is 3 time units.

Compute then that γ for which the time constant of the extintion rate of the predators (i.e., the dynamics of the predators under no-preys assumptions) is 3 time units.

- If you are solving this point via pen and paper, then write the procedure you followed to compute the solution in the cell below, using it as a `markdown` cell, and the solution itself.
- If you are solving this point via symbolic computing, then use the cell below as a `code` cell to write the code that computes the solution.

When there are no prey and without human intervention, the dynamics of the predators is given by:

$\dot{y}_{pred} = -\gamma y_{pred}$

[ ]:
```
import numpy as np
import sympy as sp
```

[ ]:
```
gamma, t = sp.symbols('gamma, t')
y_pred = sp.symbols('y_pred', cls=sp.Function)

ode = sp.Eq(sp.Derivative(y_pred(t)), -gamma * y_pred(t))
ode
```

[ ]:
$$\frac{d}{dt}y_{pred}(t) = -\gamma y_{pred}(t)$$

[ ]:
```
initial_condition = 1
cont = sp.dsolve(ode, y_pred(t), ics={y_pred(0): initial_condition})
cont
```

[ ]:
$$y_{pred}(t) = e^{-\gamma t}$$

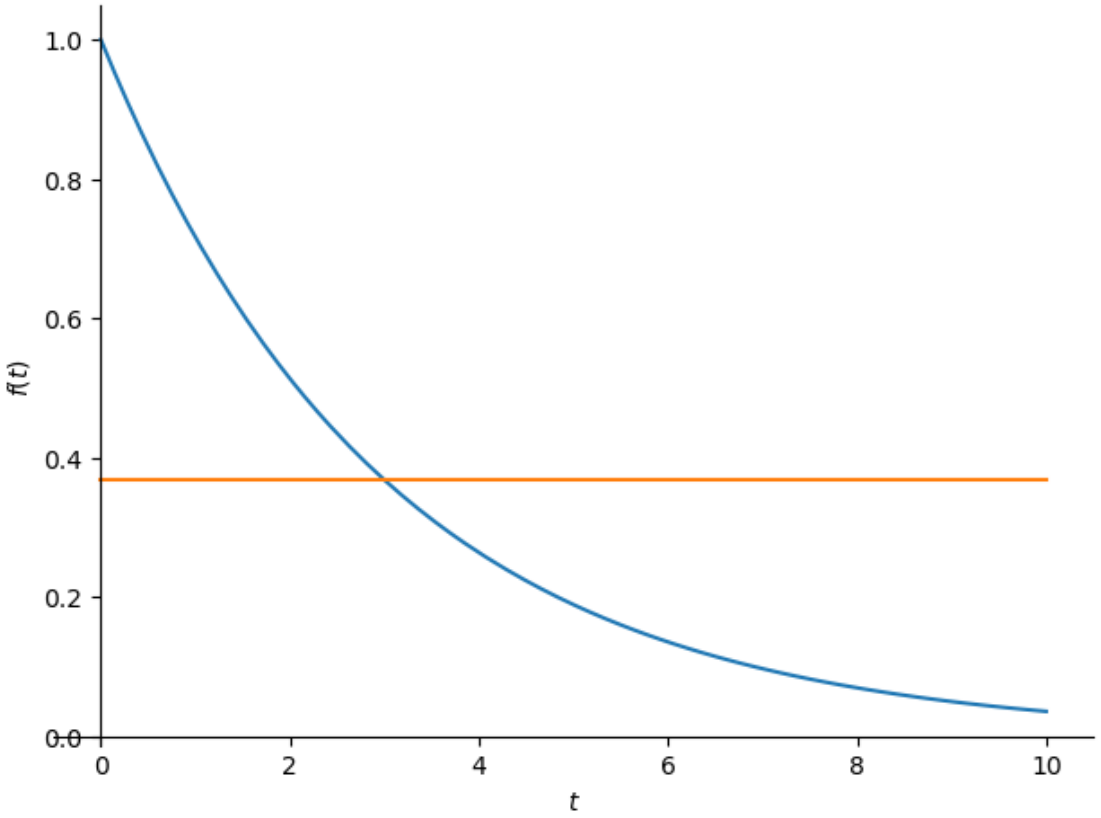
[ ]:
```
gamma_sol = sp.solve(cont.rhs - 1/sp.E, gamma)[0].subs(t, 3)
sp.Eq(gamma, gamma_sol)
```

[ ]:
$$\gamma = \frac{1}{3}$$

[ ]:
```
final_cont = cont.subs(gamma, gamma_sol)
final_cont
```

[ ]:
$$y_{pred}(t) = e^{-\frac{t}{3}}$$

[ ]:
```
sp.plot(final_cont.rhs, 1/sp.E, (t, 0, 10))
```



[ ]: <sympy.plotting.plot.Plot at 0x211dba5fb50>

1.0.4 Question 2

You also noted that when there is nobody eating them around, the amount of time required for the preys to increase their population of a factor e is 5 time units.

Compute then that α for which the time constant of the Malthusian growth of the preys (i.e., the dynamics of the preys under no-predators assumptions) is 5 time units.

- If you are solving this point via pen and paper, then write the procedure you followed to compute the solution in the cell below, using it as a `markdown` cell, and the solution itself.
- If you are solving this point via symbolic computing, then use the cell below as a `code` cell to write the code that computes the solution.

When there are no predators and without human intervention, the dynamics of the prey is given by:

$\dot{y}_{prey} = \alpha y_{prey}$

[ ]:
```
alpha, t = sp.symbols('alpha, t')
y_prey = sp.symbols('y_prey', cls=sp.Function)
```

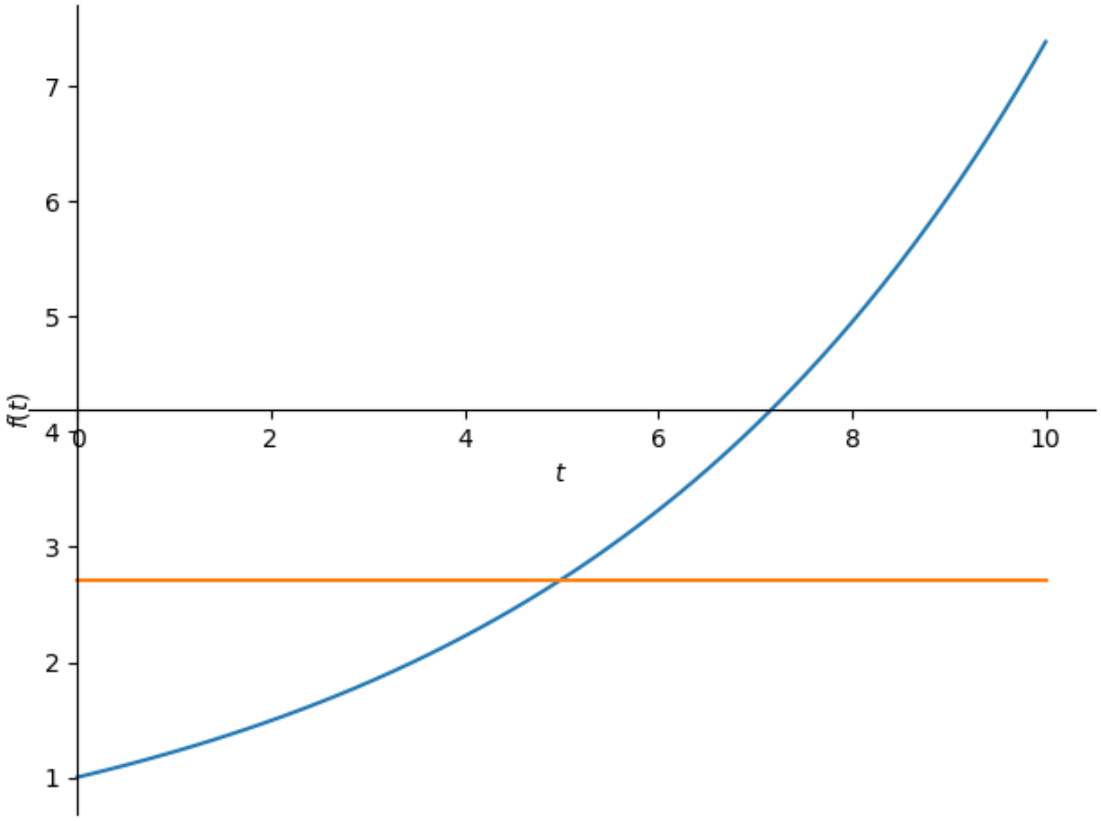
```
ode = sp.Eq(sp.Derivative(y_prey(t)), alpha * y_prey(t))
initial_condition = 1
cont = sp.dsolve(ode, y_prey(t), ics={y_prey(0): initial_condition})
alpha_sol = sp.solve(cont.rhs - sp.E, alpha)[0].subs(t, 5)
sp.Eq(alpha, alpha_sol)
```

[ ]: $\alpha = \frac{1}{5}$

[ ]:
```
final_cont = cont.subs(alpha, alpha_sol)
final_cont
```

[ ]: $y_{prey}(t) = e^{\frac{t}{5}}$

[ ]:
```
sp.plot(final_cont.rhs, sp.E, (t, 0, 10))
```



[ ]: <sympy.plotting.plot.Plot at 0x211dba6f950>

1.0.5 Question 3

You also noted that when nobody poaches these animals, the system seems to be in equilibrium if there are 10 units of preys and 2 units of predators.

Compute then that β and δ so that the system is in equilibrium if there are 10 units of preys and 2 units of predators, and both the u's are set to 0. Obviously let α and γ be that ones you computed above.

- If you are solving this point via pen and paper, then write the procedure you followed to compute the solution in the cell below, using it as a `markdown` cell, and the solution itself.
- If you are solving this point via symbolic computing, then use the cell below as a `code` cell to write the code that computes the solution.

$0 = \frac{1}{5} y_{prey} - \beta y_{prey} y_{pred}$

$0 = -\frac{1}{3} y_{pred} + \delta y_{prey} y_{pred}$

[ ]:
```
delta, beta, y_prey, y_pred = sp.symbols('delta, beta, y_prey, y_pred')
```

[ ]:
```
alpha = alpha_sol
gamma = gamma_sol
```

[ ]:
```
eqs = sp.Array([
    sp.Eq(0, alpha*y_prey - beta*y_prey*y_pred),
    sp.Eq(0, -gamma*y_pred + delta*y_prey*y_pred),
])
eqs
```

[ ]: $\left[0 = -\beta y_{pred} y_{prey} + \frac{y_{prey}}{5} \quad 0 = \delta y_{pred} y_{prey} - \frac{y_{pred}}{3}\right]$

[ ]:
```
eqs_10_2 = eqs.subs({y_prey: 10, y_pred: 2})
eqs_10_2
```

[ ]: $\left[0 = 2 - 20\beta \quad 0 = 20\delta - \frac{2}{3}\right]$

[ ]:
```
solutions = sp.solve(eqs_10_2, [delta, beta])
```

[ ]:
```
sp.Eq(delta, solutions[delta])
```

[ ]: $\delta = \frac{1}{30}$

[ ]:
```
sp.Eq(beta, solutions[beta])
```

[ ]: $\beta = \frac{1}{10}$

[ ]:
```
final_model = eqs.subs({delta: solutions[delta], beta: solutions[beta]})
final_model
```

[ ]: $\left[0 = -\frac{y_{pred} y_{prey}}{10} + \frac{y_{prey}}{5} \quad 0 = \frac{y_{pred} y_{prey}}{30} - \frac{y_{pred}}{3}\right]$

[ ]:
```
final_model[0].simplify()
```

[ ]:

$$\frac{y_{prey}\left(2 - y_{pred}\right)}{10} = 0$$

[ ]:
```
final_model[1].simplify()
```

[ ]:
$$\frac{y_{pred}\left(y_{prey} - 10\right)}{30} = 0$$

1.0.6 Question 4

Somebody in the government says that we need to control these two populations, and there is thus the need to decides how much human intervention there should be. The government is thinking at two strategies, i.e.: * **strategy a**: set $u_{prey} = 1$ and $u_{pred} = 0.2$; * **strategy b**: set $u_{prey} = 1$ and $u_{pred} = 0$.

To help the government, compute and compare the two equilibria that would result by using these two different intervention levels.

- If you are solving this point via pen and paper, then write the procedure you followed to compute the solution in the cell below, using it as a `markdown` cell, and the solution itself.
- If you are solving this point via symbolic computing, then use the cell below as a `code` cell to write the code that computes the solution.

[ ]:
```
u_prey, u_pred = sp.symbols('u_prey, u_pred')

driven_model = sp.Array([
    sp.Eq(0, final_model[0].rhs - u_prey),
    sp.Eq(0, final_model[1].rhs - u_pred),
])

driven_model
```

[ ]:
$$\left[0 = -u_{prey} - \frac{y_{pred}y_{prey}}{10} + \frac{y_{prey}}{5} \quad 0 = -u_{pred} + \frac{y_{pred}y_{prey}}{30} - \frac{y_{pred}}{3}\right]$$

[ ]:
```
print("The equilibria  for u=(1, 0.2) are:")
for eq in sp.solve(driven_model.subs({u_prey: 1, u_pred: 0.2})):
    print(eq)
```

```
The equilibria  for u=(1, 0.2) are:
{y_pred: -0.913552872566004, y_prey: 3.43223563716998}
{y_pred: 1.31355287256600, y_prey: 14.5677643628300}
```

[ ]:
```
print("The equilibria  for u=(1, 0) are:")
for eq in sp.solve(driven_model.subs({u_prey: 1, u_pred: 0})):
    print(eq)
```

```
The equilibria  for u=(1, 0) are:
{y_pred: 0, y_prey: 5}
{y_pred: 1, y_prey: 10}
```

1.0.7 Question 5

You are worried that computing the equilibria is not enough to take a good decision, and you want to help the government by coding a simulator that can be used to draw predictions of how the population levels will change in time, depending on the interventions that humans may want to take in the future (and thus the u signals in time).

Code thus an Euler solver that computes the trajectories of the model, and plot the corresponding results, starting from generic initial conditions, and considering generic actuation signals u. Add as many cells you like to organize your code.

Tip: use as a starting point the code you find in https://github.com/damianovar/TTK4225-2023.

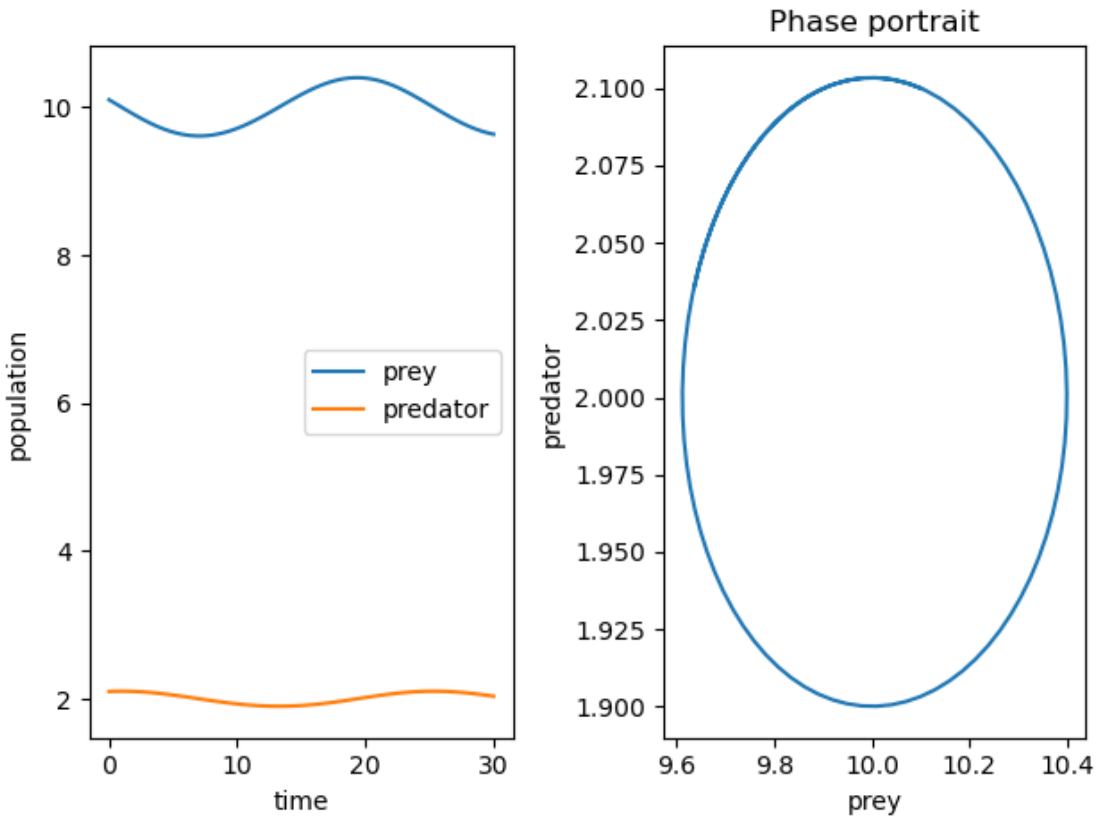
[ ]:
```
driven_model.applyfunc(lambda e: e.rhs)
```

[ ]:
$$\begin{bmatrix} -u_{prey} - \frac{y_{pred}y_{prey}}{10} + \frac{y_{prey}}{5} & -u_{pred} + \frac{y_{pred}y_{prey}}{30} - \frac{y_{pred}}{3} \end{bmatrix}$$

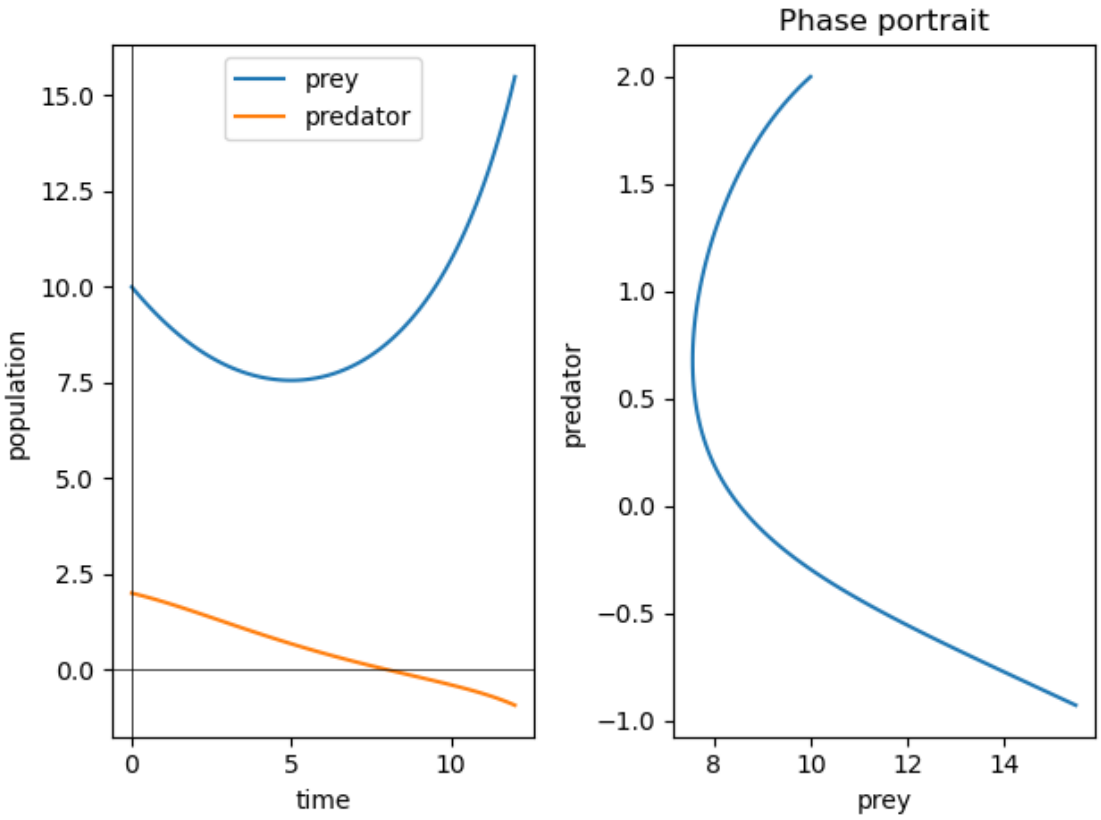
[ ]:
```
import matplotlib.pyplot as plt
from scipy.integrate import odeint


def create_model(u_prey, u_pred):

    def model(y, t):
        alpha = 1/5
        gamma = 1/3
        delta = 1/30
        beta = 1/10

        y_prey, y_pred = y
        return np.array(
            [
                alpha * y_prey - beta * y_prey * y_pred - u_prey,
                -gamma * y_pred + delta * y_prey * y_pred - u_pred,
            ]
        )

    return model


def plot_model(model, y0, duration, plot_axes=False):
    t = np.linspace(0, duration, 100)

    y = odeint(model, y0, t)

    y1, y2 = y.T

    plt.subplot(1, 2, 1)
    plt.plot(t, y1, label="prey")
```

```
    plt.plot(t, y2, label="predator")
    plt.xlabel("time")
    plt.ylabel("population")
    if plot_axes:
        plt.axhline(0, color="black", linewidth=0.5)
        plt.axvline(0, color="black", linewidth=0.5)
    plt.legend()

    plt.subplot(1, 2, 2)
    plt.title("Phase portrait")
    plt.plot(y1, y2)
    plt.xlabel("prey")
    plt.ylabel("predator")
    plt.tight_layout()
```

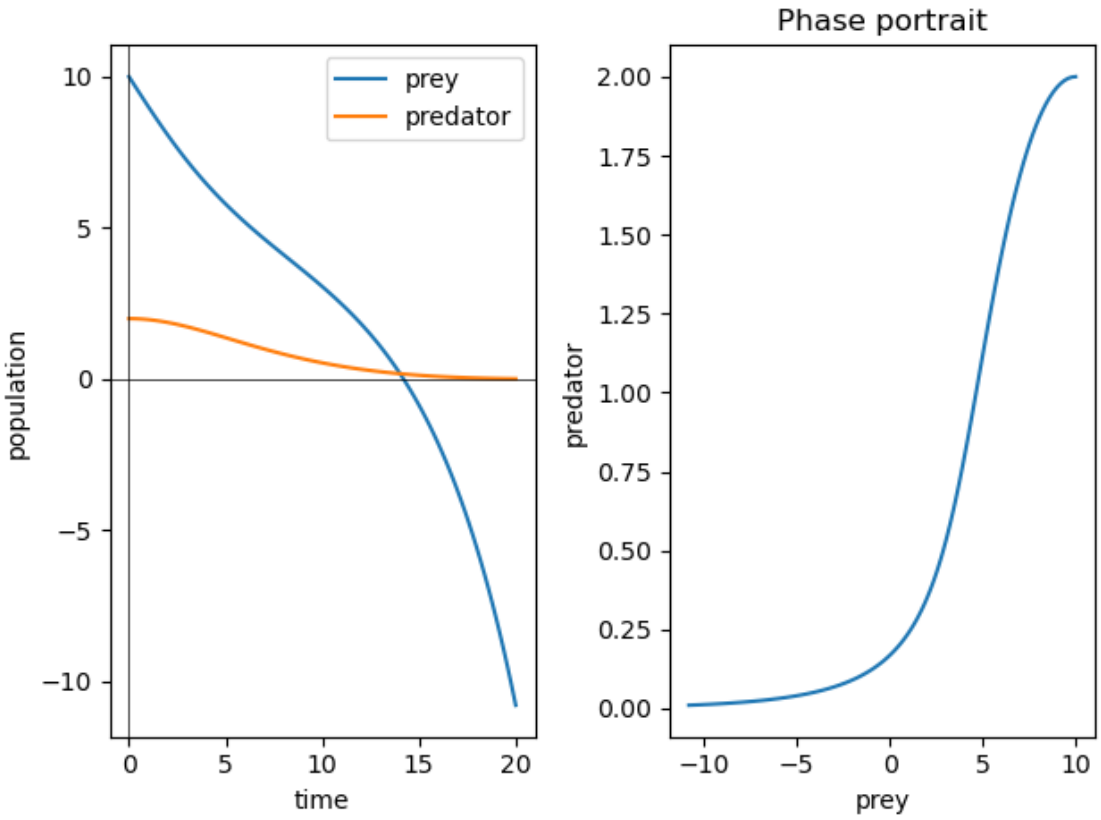
[ ]:
```
plot_model(create_model(u_prey=0, u_pred=0), y0=[10.1, 2.1], duration=30)
```



[ ]:
```
plot_model(create_model(u_prey=1, u_pred=0.2), y0=[10, 2], duration=12,␣
↪plot_axes=True)
```



[ ]:
```
plot_model(create_model(u_prey=1, u_pred=0), y0=[10, 2], duration=20,␣
↪plot_axes=True)
```



1.0.8 Question 6

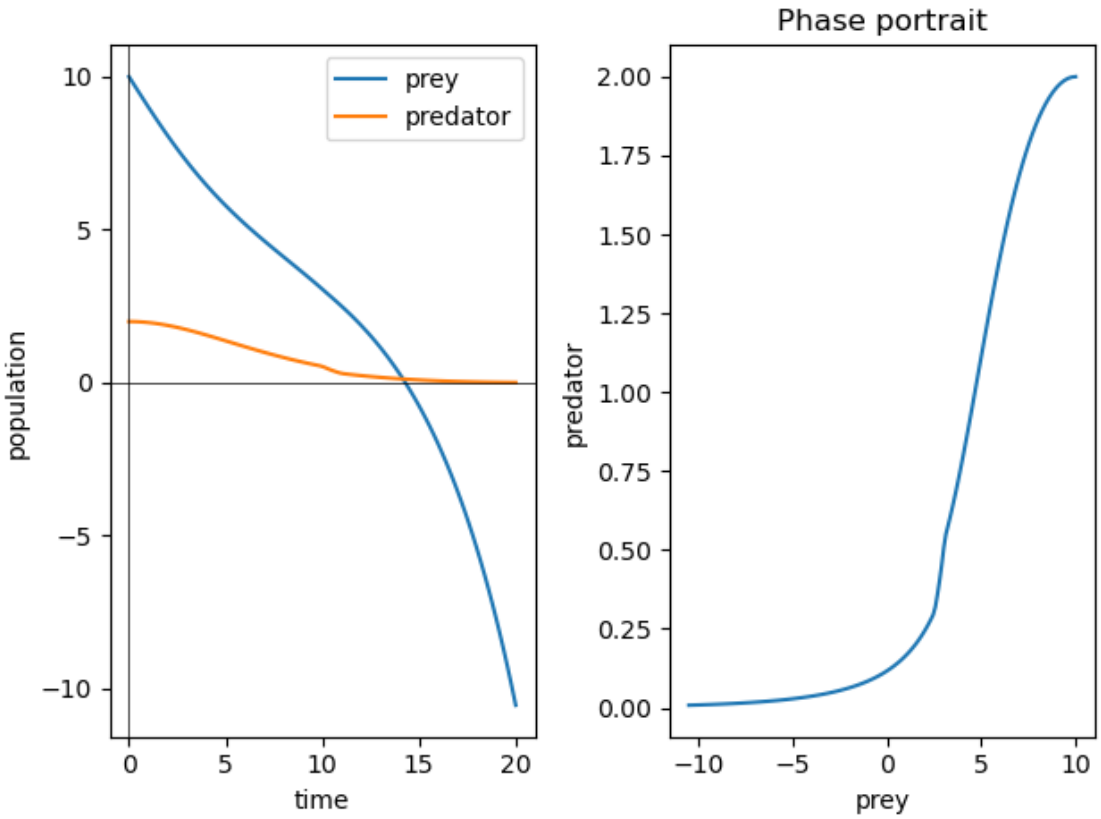
Argh, covid is hitting the predators too! You need to modify your simulator if you want to be able to compute meaningful forecasts. For this, create another version of the ODEs defining your simulator (i.e., assuming you are using names similar to the ones in the github repository above, create another version of the `myModel` function) so that between time $T_1 = 10$ and $T_2 = 11$ the pandemic event affects the predators so that, during that period, their deaths rate γ is twice its original value.

Tip: always give meaningful names to the variables and functions you code.

[ ]:
```
def create_covid_model(u_prey, u_pred):

    def model(y, t):
        alpha = 1/5
        gamma = 2/3 if 10 < t < 11 else 1/3
        delta = 1/30
        beta = 1/10

        y_prey, y_pred = y
        return np.array(
```

```
            [
                alpha * y_prey - beta * y_prey * y_pred - u_prey,
                -gamma * y_pred + delta * y_prey * y_pred - u_pred,
            ]
        )

    return model
```

[ ]:
```
plot_model(create_covid_model(u_prey=1, u_pred=0), y0=[10, 2], duration=20,␣
↪plot_axes=True)
```



1.0.9 Question 7

To see if this pandemic actually affects the populations a lot or not, you need to compare the two versions of the ODEs, i.e., compute and plot opportunely some trajectories using the original dynamics and some trajectories using the "pandemic-affected" ones.

To aid interpretability, you decide to do five different simulations starting from the 5 different initial conditions, each defined by $y(0) = (1 + 0.1k)y_{eq,u}$ with $k = 1, ..., 5.$, always using **strategy a** as an intervention.

Create in the cell below (or in as many cells you need) some code that produces 5 plots, one for

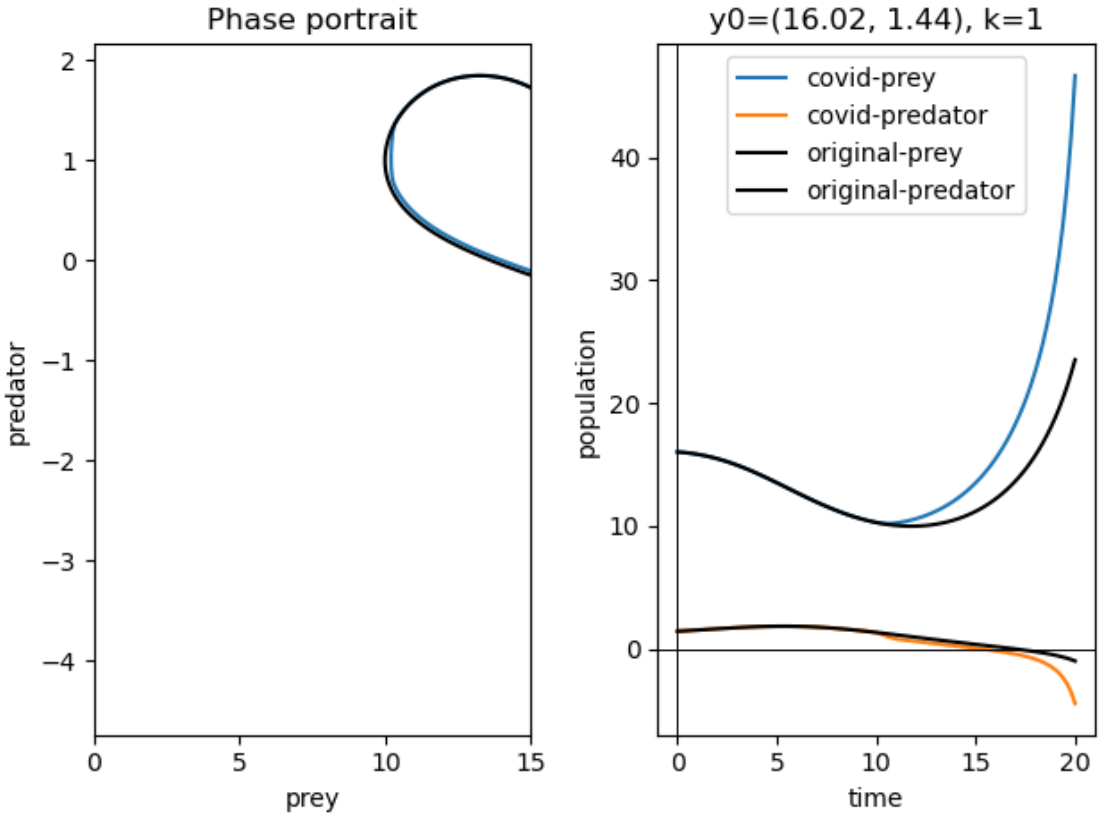
each initial condition, and each divided in 2 parts: * on the left, the two trajectories (original vs. pandemic) in the phase space, * on the right, the two trajectories (original vs. pandemic) as time signals.

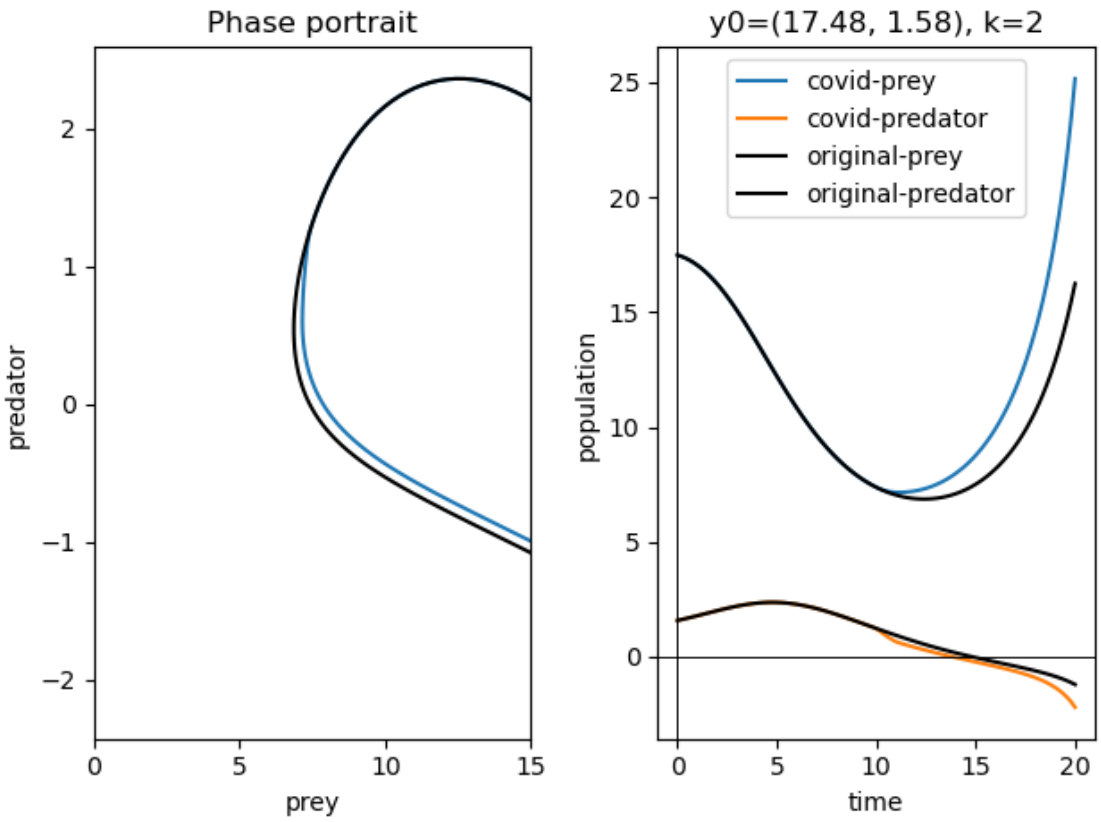
Tip: do so that the axes across the 5 figures of the same type are identical (e.g., the `x_axis_max` of the various phase portraits are the same).

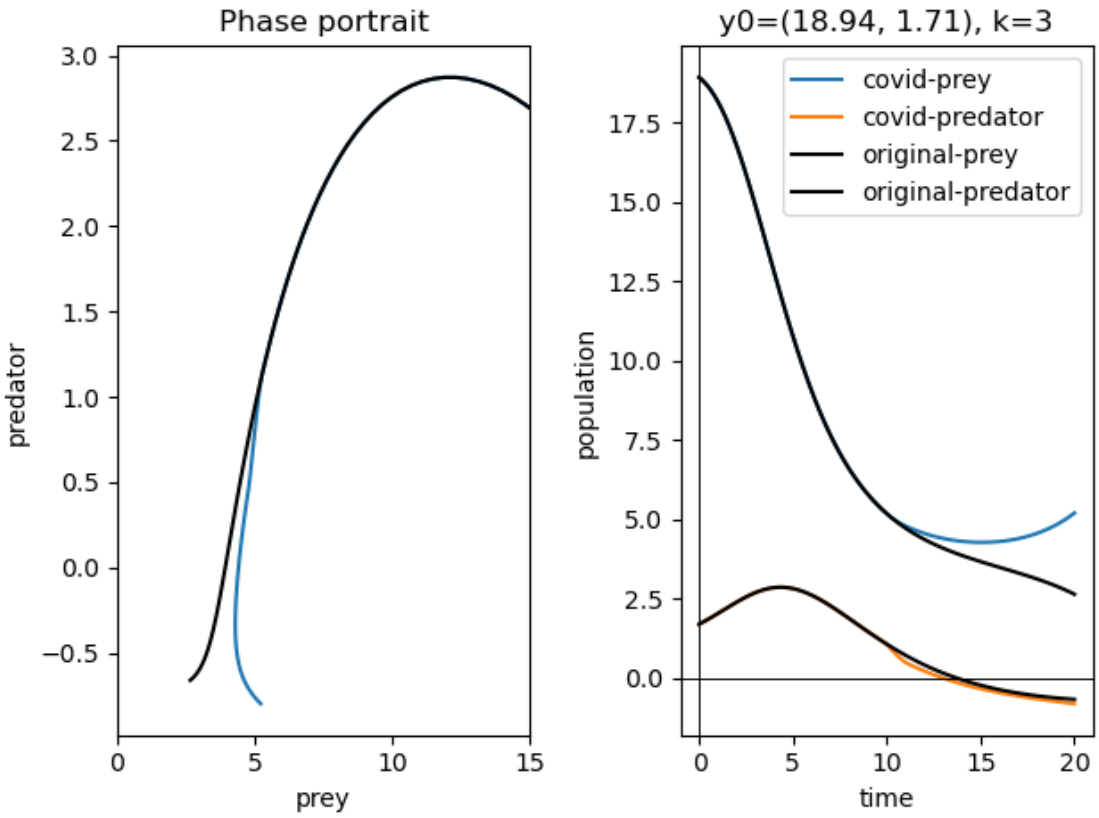
[ ]:
```
def plot_model2(model, y0, duration, plot_axes=False, original=False):
    t = np.linspace(0, duration, 100)

    y = odeint(model, y0, t)

    y1, y2 = y.T

    color = "black" if original else None
    label_prefix = "original-" if original else "covid-"

    plt.subplot(1, 2, 1)
    plt.title("Phase portrait")
    plt.plot(y1, y2, color=color)
    plt.xlabel("prey")
    plt.ylabel("predator")
    plt.xlim(0, 15)

    plt.subplot(1, 2, 2)
    plt.plot(t, y1, label=label_prefix+"prey", color=color)
    plt.plot(t, y2, label=label_prefix+"predator", color=color)
    plt.xlabel("time")
    plt.ylabel("population")
    if plot_axes:
        plt.axhline(0, color="black", linewidth=0.5)
        plt.axvline(0, color="black", linewidth=0.5)
    plt.legend()
    plt.tight_layout()
```

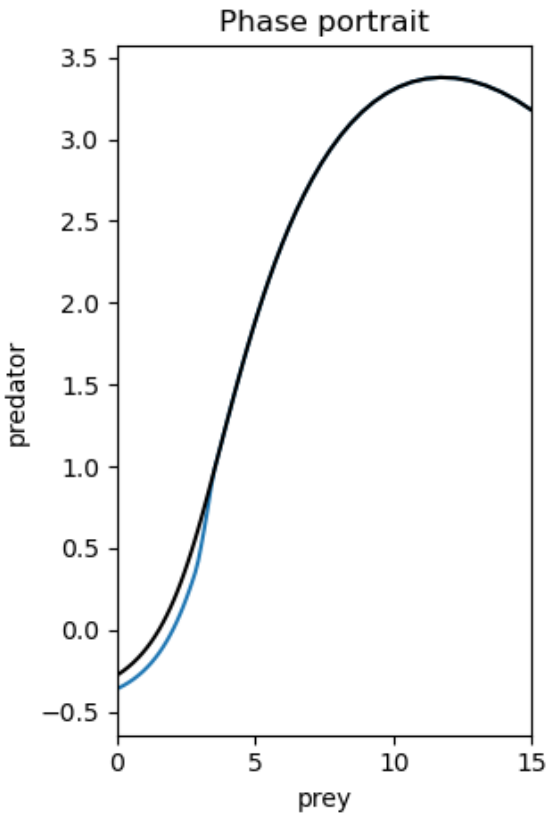
[ ]:
```
y00 = np.array([14.5677643628300, 1.31355287256600])
for k in range(1, 6):
    y0 = y00 * (1 + 0.1 * k)
    plot_model2(
        create_covid_model(u_prey=1, u_pred=0.2), y0=y0, duration=20,␣
 ↪plot_axes=True
    )

    plot_model2(create_model(u_prey=1, u_pred=0.2), y0=y0, duration=20,␣
 ↪plot_axes=True, original=True)
    plt.title(f"y0=({y0[0]:.2f}, {y0[1]:.2f}), k={k}")
    plt.show()
```

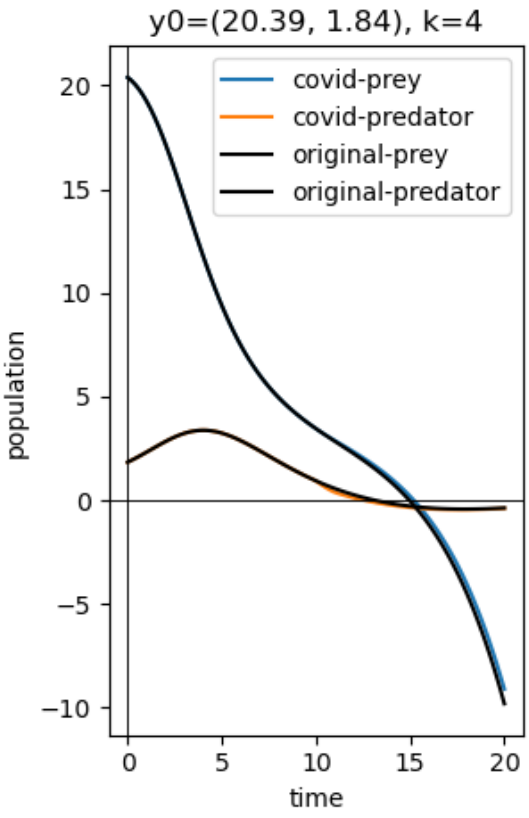
```
C:\Users\noskn\AppData\Local\Temp\ipykernel_44412\2579807086.py:27: UserWarning:
The figure layout has changed to tight
  plt.tight_layout()
```

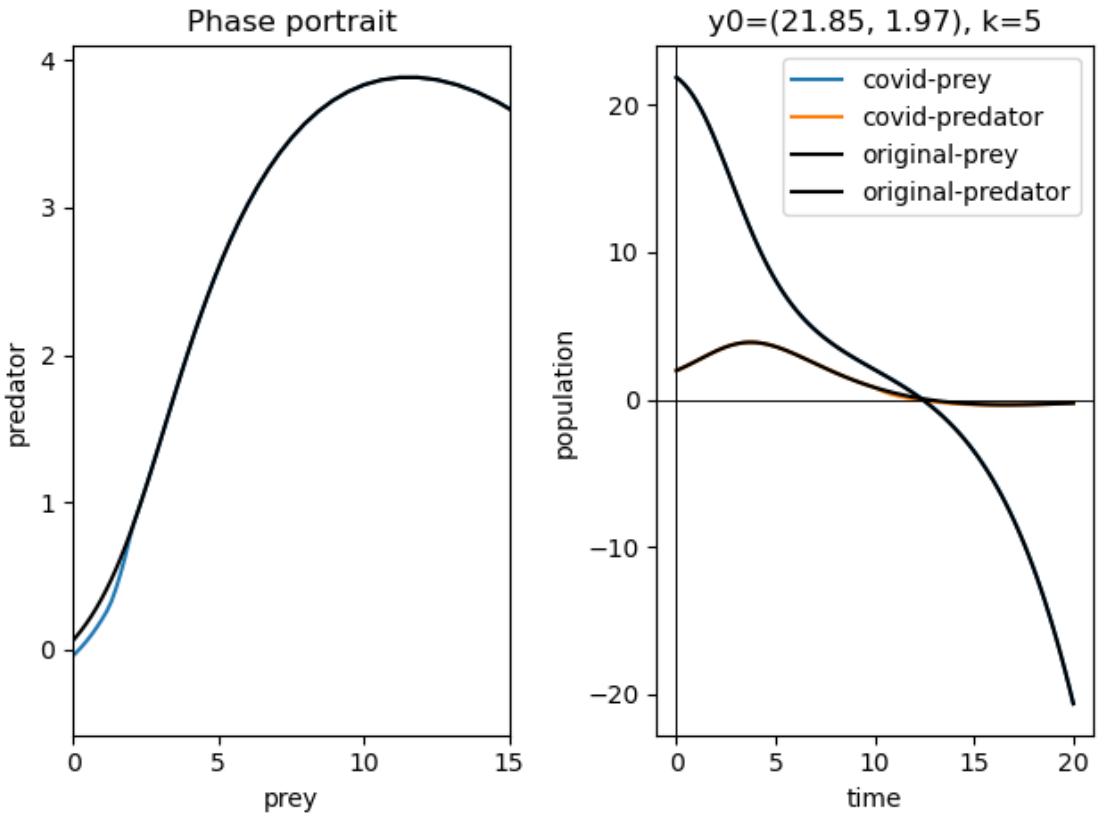



Phase portrait
predator
prey
2
1
0
−1
−2
0
5
10
15
y0=(17.48, 1.58), k=2
population
time
covid-prey
covid-predator
original-prey
original-predator
25
20
15
10
5
0
0
5
10
15
20


Phase portrait
predator
prey
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
0
5
10
15
y0=(18.94, 1.71), k=3
population
time
covid-prey
covid-predator
original-prey
original-predator
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0
5
10
15
20


Phase portrait
predator
prey
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
0
5
10
15


y0=(20.39, 1.84), k=4
population
time
covid-prey
covid-predator
original-prey
original-predator
20
15
10
5
0
−5
−10
0
5
10
15
20



1.0.10 Question 8

How much do you think the pandemic affects this ecological system? Write your thoughts in the cell below, using it as a `markdown` cell.

The pandemic has very little effect on this system. It appears to only have a meaningful effect on very small populations.