COSC 343: homework 4 Runge-Kutta

Micah Sherry

May 5, 2024

1 Lotka-Voltera Model

For class we looked at the populations of two species, a prey denoted by y_1 and a predator denoted by y_2, can be modeled by the autonomous, nonlinear ODE

$$\mathbf{y}' = \begin{pmatrix} y_1' \\ y_2' \end{pmatrix} = \begin{pmatrix} y_1(\alpha_1 - \beta_1 y_2) \\ y_2(-\alpha_2 + \beta_2 y_1) \end{pmatrix} = \mathbf{f}(\mathbf{y})$$

This model is known as the Lotka-Voltera model. The parameters α_1 and α_2 are birth and death rates in isolation for prey and predators respectively. The parametersβ_1 and β_2 determine the effects of the interactions of the two species. Write a program that uses the classical fourth order Runge-Kutta method to solve the Lotka-Voltera system. Integrate from t = 0 to t = 25. For class we will use the parameters $\alpha_1 = 0.5$, $\beta_1 = 0.1$, $\alpha_2 = 1.0$, $\beta_2 = 0.02$, and the initial populations $y_1(0) = 100$, and $y_2(0) = 10$. We plotted each of the two populations as a function of time, and on a separate graph plot the trajectory of the point (y1(t), y2(t)) in the plane as a function of time and created a phase portrait

For this assignment try other initial populations. Observe the results using the same type of graphs used in class. Can you find nonzero initial populations that allow for either of the populations to become extinct? Can you find a non-zero initial population that never changes.

1.1 code:

methods for Runge-Kutta

```
import numpy as np

def simulate_model(f, prey_inital = 100, predator_inital = 10, t0=0, T=25, steps = 2000
    """ a method to simmulate a preditor prey model for a given a forcing fuction """

    # set up model inital conditions
    h = (T-t0)/steps # change in time step
    Y = np.zeros(2).reshape(2,1)
    Y[0] = prey_inital
    Y[1] = predator_inital
    tvec = []
    preyVec = []
    predatorVec = []
    t=t0

    for i in range(steps):
        t = t + h
        Y = RungeKutta4(t,h,Y,f)
        prey = Y[0]
        preditor = Y[1]

        tvec.append(i)
```

```
        preyVec.append(prey)
        predatorVec.append(preditor)
        #if prey <=0 or preditor <=0:
        #    print(prey, preditor)
        #    break
    return tvec, preyVec, predatorVec

def RungeKutta4(t,h,Y,f):
    """a fourth order runge kuta method"""
    K1 = f(t,Y)
    K2 = f(t+.5*h, Y+.5*K1)
    K3 = f(t+.5*h, Y+.5*K2)
    K4 = f(t+h, Y+K3)

    return Y + (h/6)*(K1 + 2*K2 + 2*K3 + K4)
```

methods for Lokta-Voltera

```
import numpy as np
import matplotlib.pyplot as plt
from Runge_Kutta import simulate_model



def Lotka_Voltera_forcing_funtion(a1, a2, b1, b2):
    def f(t,Y):
        Ynew = np.zeros(2).reshape(2,1)
        Ynew[0] = (a1-b1*Y[1])*Y[0]
        Ynew[1] = (-a2+b2*Y[0])*Y[1]

        return Ynew
    return f

# initalize the model with the give parameters
f = Lotka_Voltera_forcing_funtion(.5, 1, .1, .02)


prey = 100
pred = 10
tvec,uvec,vvec =simulate_model(f,prey_inital=prey, predator_inital=pred)
plt.plot(tvec,uvec, "k")
plt.plot(tvec,vvec, "g")
plt.title("Lotka-Voltera with inital Conditions y1 = "+ str(prey)+" and y2 = " + str(pre
plt.show()

plt.plot(uvec,vvec)
plt.title("Lotka-Voltera phase portrait with inital Conditions y1 = "+ str(prey)+" and y
plt.show()
```

1.2 initial condition that causes extinction of a population

consider what would happen as the number of predators approached zero. the growth rate of the prey would look like:

$$y_1' = \alpha_1 y_1$$

this tells us that the prey would grow exponentially. now consider what would happen when the number of prey is going to zero. the death rate of the predators would look like:

$$y_2' = -\alpha_2 y_2$$

this tells us that the predators would die exponentially and as I already established when the number predators goes to zero the number of prey grows exponentially. so what this tells us is there no initial condition that will cause one of the populations to go extinct.

1.3 stable initial condition

to find initial conditions such that both populations remain constant. If a population is constant then its derivative is zero. so if we set $\mathbf{y} = \mathbf{0}$ then solve the system for y_1 and y_2

$$\mathbf{0} = \begin{pmatrix} y_1(\alpha_1 - \beta_1 y_2) \\ y_2(-\alpha_2 + \beta_2 y_1) \end{pmatrix}$$

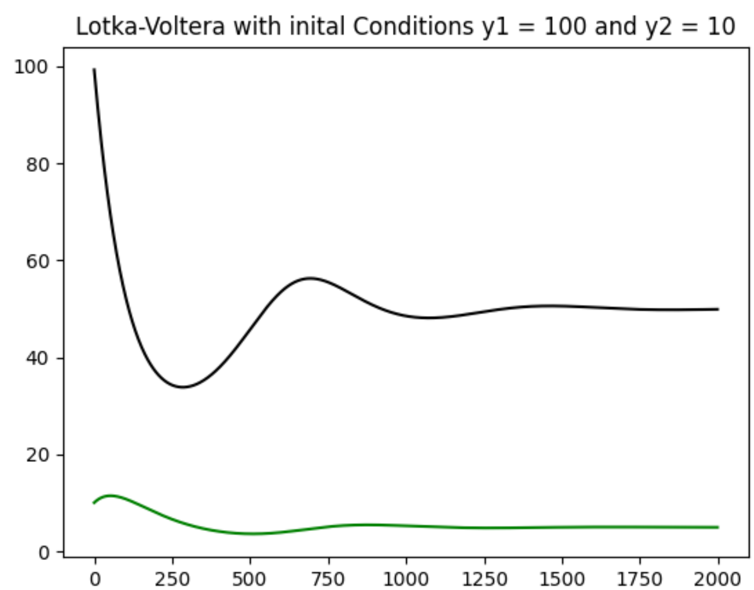
solving this system yields (ignoring the trivial solution):

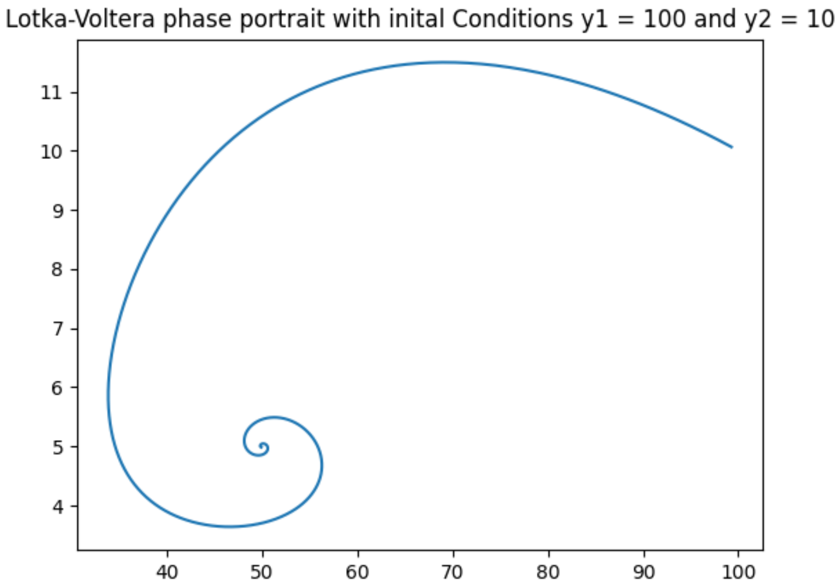
$$y_1 = \frac{\alpha_2}{\beta_2} \text{ and } y_2 = \frac{\alpha_1}{\beta_1}$$

for the parameters specified above. that would make:

$$y_1 = 50 \text{ and } y_2 = 5$$

1.4 plots





2 Leslie-Gower Model

Repeat your work using the Leslie-Gower model:

$$\mathbf{y}' = \begin{pmatrix} y_1' \\ y_2' \end{pmatrix} = \begin{pmatrix} y_1(\alpha_1 - \beta_1 y_2) \\ y_2(\alpha_2 - \frac{\beta_2 y_2}{y_1}) \end{pmatrix} = \mathbf{f}(\mathbf{y})$$

use the same parameters given above with the exception of setting $\beta_2 = 10$. How does the solution differ between the two models?

2.1 code:

this code uses the same methods for Runge-Kutta as lokta-voltera the only thing that changes is the forcing function

```
import numpy as np
import matplotlib.pyplot as plt
from Runge_Kutta import simulate_model
from matplotlib.widgets import Slider


def Leslie_Gower_forcing_funtion(a1, a2, b1, b2):
    def f(t,Y):
        Ynew = np.zeros(2).reshape(2,1)
        Ynew[0] = (a1-b1*Y[1])*Y[0]
        Ynew[1] = Y[1]*(a2-b2*Y[1]/Y[0])

        return Ynew
    return f
f = Leslie_Gower_forcing_funtion(.5, 1, .1, 10)
prey = 100
pred = 10
tvec,uvec,vvec =simulate_model(f,prey_inital=prey, predator_inital=pred)
print(vvec)
```

```
print(uvec)
plt.plot(tvec,uvec, "k")
plt.plot(tvec,vvec, "g")
plt.title("Leslie-Gower with inital Conditions y1 = "+ str(prey)+" and y2 = " + str(pred
plt.show()

plt.plot(uvec,vvec)
plt.title("Leslie-Gower phase portrait with inital Conditions y1 = "+ str(prey)+" and y2
plt.show()
```

2.2 initial condition that causes extinction of a population

for this model i was able to find several that led to the extinction of the predators. typically if i set the ratio (prey to predator) of initial conditions to 1.8 (or less) to 1 .

2.3 stable initial condition

As mentioned while answering this question for the Loktka-Voltera model we need to set its derivative to zero and then solve for y_1 and y_2

$$\mathbf{0} = \begin{pmatrix} y_1(\alpha_1 - \beta_1 y_2) \\ y_2(\alpha_2 - \frac{\beta_2 y_2}{y_1}) \end{pmatrix}$$

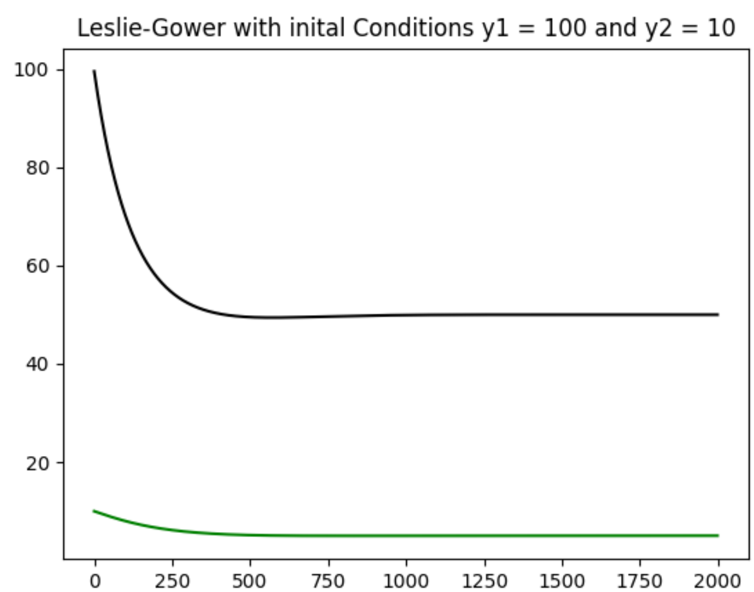
solving this system yields (ignoring the trivial solution):

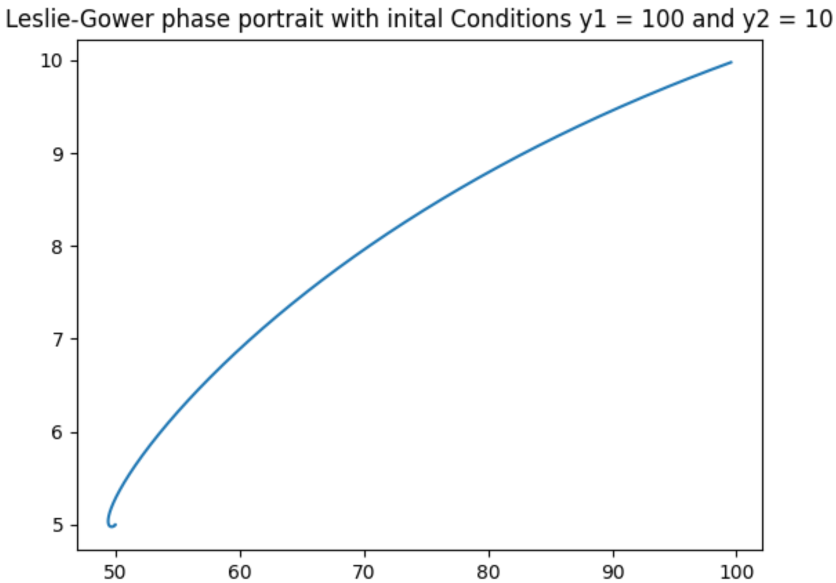
$$y_1 = \frac{\beta_2}{\alpha_2} y_2 = \frac{\beta_2}{\alpha_2}\frac{\alpha_1}{\beta_1} \text{ and } y_2 = \frac{\alpha_1}{\beta_1}$$

and like with Lotka-Voltera model with the given parameters the solution is

$$y_1 = 50 \text{ and } y_2 = 5$$

2.4 plots





2.5 how do the solutions differ between 2 models

In the Lokta-Voltera model the populations oscillate more than the Leslie-Gower model. both Models had an initial condition ($y_1 = 50$ and $y_2 = 5$) that caused the populations to remain constant. But only the Leslie-Gower model had an initial condition that will cause extinction.