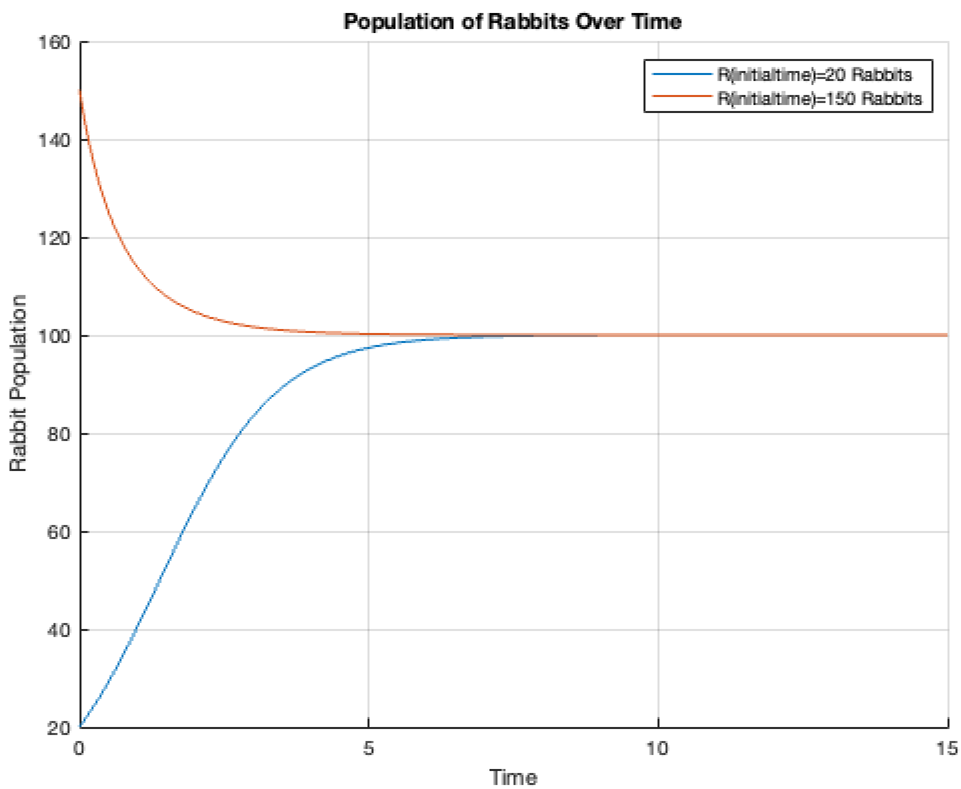
```
%Kaitlyn Kirt, CMOR 220, Spring 2024, Population Model Project
%Project4.m
%This script is a project on differential equations
%Last modified: February 19, 2024

%driver
function Project4
%Logistic Growth Model
disp("Logistic Function")
Logistic

%Predator-Prey population Model
disp("Predator-Prey Model")
PredatorPrey
k1=3; k2=3*10.^-3; k3=6*10.^-4; k4=0.5;
Solver(k1,k2,k3,k4,0,15,1000,500)
end

function Logistic
%inputs: none
%outputs: none
%description: this function adds inputs and outputs to make the function
%more variable
r=1;
K=100;
dRdelta=@(time,R) r*R*(1-(R/K)); %anoymous function for logistic ODE

delta=0.01;
initialtime=0;
finaltime=15;
R0=20;
R01=150;
[time,R]=RabbitModel(delta,initialtime,R0,finaltime,dRdelta); %creates a
function with initial condition as 20
[time1,R1]=RabbitModel(delta,initialtime,R01,finaltime,dRdelta); %creates a
function with initial condition as 150

figure(1);
hold on; grid on;
plot(time,R)
plot(time1,R1)
title("Population of Rabbits Over Time")
xlabel("Time")
ylabel("Rabbit Population")
legend("R(initialtime)=20 Rabbits","R(initialtime)=150 Rabbits")

%Given any positive initial condition, the rabbit population would increase
or decrease to its carrying capacity (100)
end

function [time,R]=RabbitModel(delta,initialtime,R0,finaltime,dRdelta)
%inputs: delta,initialtime,R0,finaltime,dRdelta
```

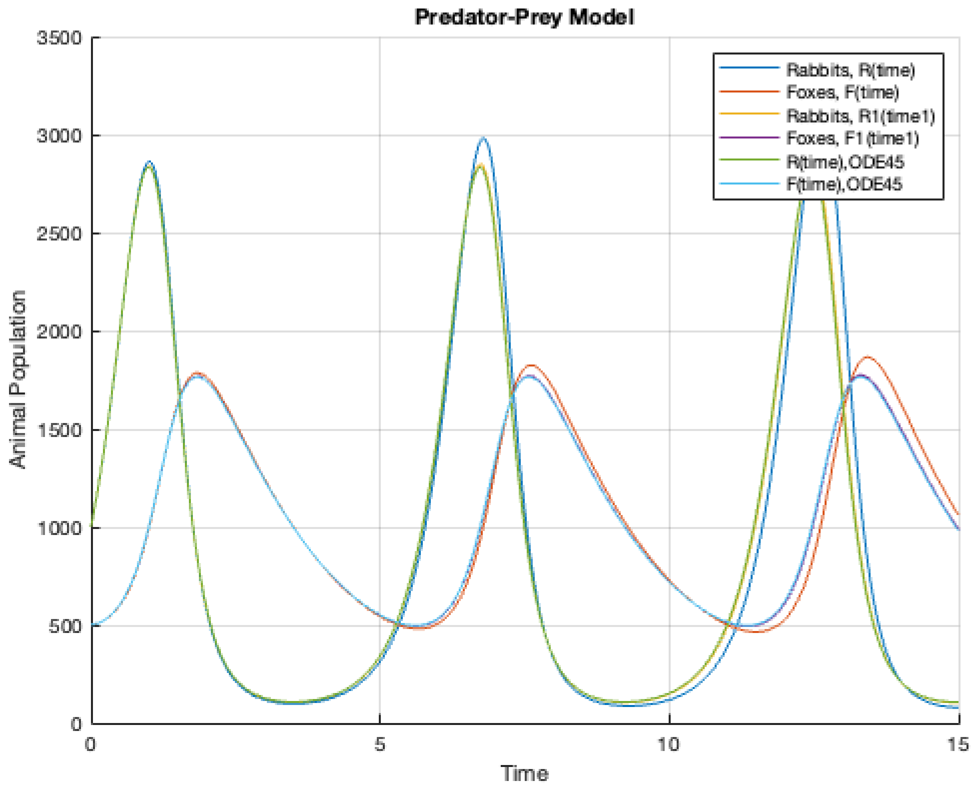
```
%outputs: time,R
%description: this function uses Euler's method to solve the ODE
time=initialtime:delta:finaltime; %denotes time interval with step
R=zeros(1,length(time)); %preallocates R
R(1)=R0; %assigns the first R value as initial population
    for n=1:length(time)-1 %runs for length(time)-1 times
        dR=(dRdelta(time(n),R(n)))*delta; %solve dR using time and R
        R(n+1)=R(n)+dR; %adds dR to R and redefines R
    end
end

%Predator-Prey Models
function PredatorPrey
%inputs: none
%outputs: none
%description: this function adds inputs and outputs to make the function
%more variable
k1=3; k2=3*10.^-3; k3=6*10.^-4; k4=0.5;
R0=1000; F0=500; delta=0.01; delta1=0.001;
initialtime=0; finaltime=15;
dRdelta=@(time,R,F) k1*R-k2*R*F; %anonymous function for rabbit equation
dFdelta=@(time,R,F) k3*R*F-k4*F; %anonymous function for fox equation
[time,R,F]=Model(delta,initialtime,finaltime,R0,F0,dRdelta,dFdelta); %creates
a function with 0.01 delta
[time1,R1,F1]=Model(delta1,initialtime,finaltime,R0,F0,dRdelta,dFdelta);
%creates a function with 0.001 delta

figure(2);
hold on; grid on;
plot(time,R)
plot(time,F)
plot(time1,R1)
plot(time1,F1)
%The solutions using the original ODE method and the ode45 method are very
%similar on patterns and overall shape of the graph. The ode45 method tends
%to have greater population of sizes than the original ODE method. The
%patterns tell me that the population of foxes and rabbits increase and
%decrease throughout time. Before the the rabbit population peak, the
%rabbit population is greater than the fox population until the population
%of rabbits decreases. After a short period, the fox population is greater
%than the rabbit population.

end

function [time,R,F]=Model(delta,initialtime,finaltime,R0,F0,dRdelta,dFdelta)
%inputs: delta,initialtime,finaltime,R0,F0,dRdelta,dFdelta
%outputs: time,R,F
%description: this function solves a system of ODEs
time=initialtime:delta:finaltime; %denotes time interval with steps
R=zeros(1,length(time)); %preallocates R
F=zeros(1,length(time)); %preallocates F
R(1)=R0; %assigns the first R value as initial population
F(1)=F0; %assigns the first F value as initial population
    for n=1:length(time)-1 %runs code for length(time)-1 times
```

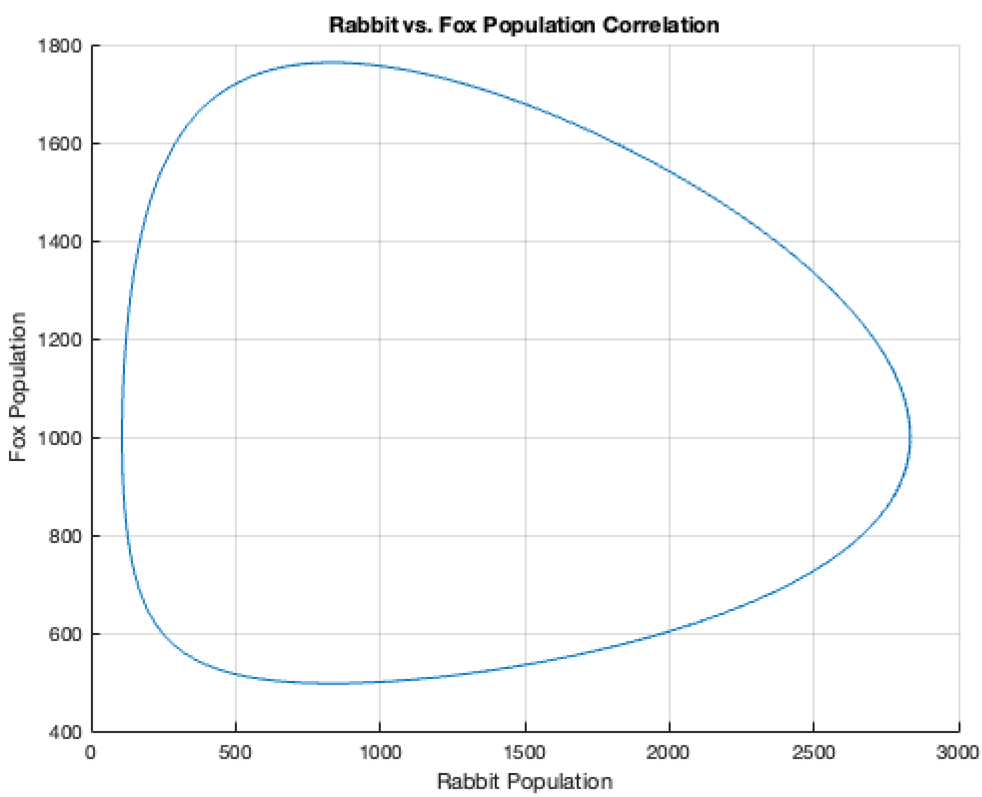
```
        dR=(dRdelta(time(n),R(n),F(n)))*delta; %solve dR using time, R, and F
        dF=(dFdelta(time(n),R(n),F(n)))*delta; %solve dF using time, R, and F
        R(n+1)=R(n)+dR; %adds dR to R and redefines
        F(n+1)=F(n)+dF; %adds dF to F and redefines
    end
end

function Solver(k1,k2,k3,k4,initialtime,finaltime,R0,F0)
%inputs: k1,k2,k3,k4,initialtime,finaltime,R0,F0
%outputs: none
%descripton: this function uses ode45 to solve a system of ODEs
dRdelta=@(time,R) [k1*R(1)-k2*R(1)*R(2); %creates an anonymous function for
both ODEs (rabbits and foxes)
                   k3*R(1)*R(2)-k4*R(2)]

time=[initialtime finaltime]; %denotes time interval
initialconditions=[R0 F0]; %denotes initial population for rabbits and foxes
options=odeset('RelTol',1e-6); %helps the ODE tolerance
[time,R]=ode45(dRdelta,time,initialconditions,options); %creates function for
ode45()

plot(time,R(:,1))
plot(time,R(:,2))
title("Predator-Prey Model")
ylabel("Animal Population")
xlabel("Time")
legend("Rabbits, R(time)","Foxes, F(time)","Rabbits, R1(time1)","Foxes,
F1(time1)","R(time),ODE45","F(time),ODE45")

figure(3);
hold on; grid on;
plot(R(:,1),R(:,2))
title("Rabbit vs. Fox Population Correlation")
xlabel("Rabbit Population")
ylabel("Fox Population")
%In this figure, as the rabbit population increases, the fox population
%tends to decrease. This relates to the last figure because it showed the
%population of rabbits and foxes at its peak. As we move leftward on figure
3, we notice that the population of foxes increases and the population of
rabbits decreases. The Predator-Prey model exhibits this after the peak in
population.

figure
PopulationVid=VideoWriter("LimitCircle.avi");
open(PopulationVid)
    for n=1:length(time)
        hold on; grid on;
        plot(R(:,1),R(:,2))
        plot(R(n,1),R(n,2),"o")
        title(["Rabbits and Foxes Population at t="+time(n)])
        legend("Population Boundary","Populations at time")
        xlabel("Rabbit Population")
        ylabel("Fox Population")
        writeVideo(PopulationVid,getframe(gcf));
```

```
        clf
    end
close(PopulationVid)
end
```

```
Logistic Function
Predator-Prey Model

dRdelta =

  function_handle with value:

    @(time,R)[k1*R(1)-k2*R(1)*R(2);k3*R(1)*R(2)-k4*R(2)]
```




Predator-Prey Model
Rabbits, R(time)
Foxes, F(time)
Rabbits, R1(time1)
Foxes, F1(time1)
R(time),ODE45
F(time),ODE45
Animal Population
Time
0
500
1000
1500
2000
2500
3000
3500
5
10
15


Rabbit vs. Fox Population Correlation
Fox Population
Rabbit Population
400
600
800
1000
1200
1400
1600
1800
0
500
1000
1500
2000
2500
3000

Published with MATLAB® R2023b