机械工程中的数值分析课程报告

2020 至 2021 学年第二学期

项目名称：Final Project

学生姓名：易弘睿

学院专业班级：UC 联合学院 2018 级机械 01 班

学　　号：20186103

任课教师：温罗生

重庆大学数学与统计学院

CQU-UC Joint Co-op Institute (JCI) Student Project Report

Final project of Applied Numerical Method in Engineering and Science

Institution CQU-UC Joint Co-op Institute (JCI)

Grade 2018 **Major** Mechanical Engineering **Class** 01

Student Name Yi, Hongrui **Student Number** 20186103

Academic Year 2021 Summer

Course Code: ME40881 **Instructor** Wen, Luosheng

CQU-UC Joint Co-op Institute

August 2021

Abstract

Isle Royale National Park is a 210-square mile archipelago composed of a single large island and many small islands in Lake Superior. Moose arrived around 1900, and by 1930, their population approached 3000, ravaging vegetation. In 1949, wolves crossed an ice bridge from Ontario. Since the late 1950s, the numbers of the moose and wolves have been tracked.

In the lecture of *Applied Numerical Method in Engineering and Science*, we have learnt various of knowledge of numerical methods including curve fitting, interpolation, numerical integration, numerical differentiation, differential equations and etc.

In this report, knowledge learnt in this lecture is used comprehensively.

In question (1), polynomial fitting method is applied to make the best order polynomial and show the error.

In question (2), numerical differentiation method is applied to give the growth rate of moose.

In question (3), predator-prey models are established and explained.

In question (4), the ordinary differential equations of the system are solved.

In question (5), curve fitting method is used to find four parameters a, b, c and d.

In question (6), spline interpolation method is used to integrate the number of moose eaten by wolves.

In question (7), numerical calculation methods are applied comprehensively to compare the results of different models.

In question (8), shooting method of ODE is used to solve the system of equations with boundary values.

All in all, this report represents the comprehensive application of knowledge, and the skill to use theory to solve practical problems.

Table of Contents

Question 1: Polynomial Fitting

Use the data of the moose from 1959 to 1973, show the polynomial fitting results. You need give the best order polynomial and show the error. And you should give an estimation about the numbers of moose before 1959.

1.1 Problem description and ideas

In previous lectures, we learnt that fitting is a way of substituting existing data into a numerical formula through mathematical methods. Through fitting, the discrete data points can be expressed with the most suitable function, so that data prediction and data processing can be carried out. Most of the fitted models are based on the least square's method. When there are few data points, we can use the method of interpolating the data first and increasing the amount of data before fitting.

I set the range of x to (0:14) and (1959:1973) respectively and got different fitting results.

1.2 Answer and results:

1.2.1 Method I

When the range of x is set to (0:14), the results are as follows.

Plot:

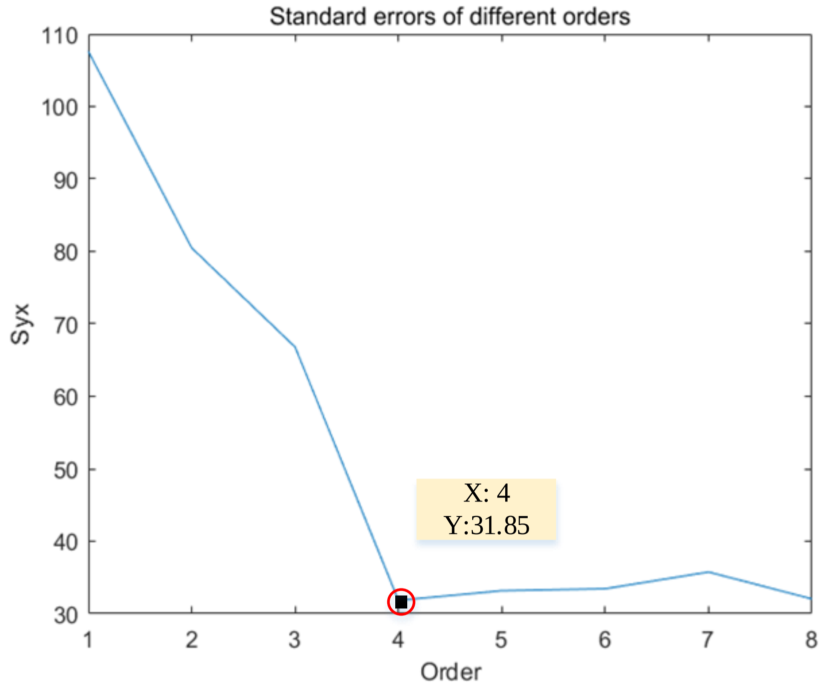

Fig. 1. Standard error of different orders of method I

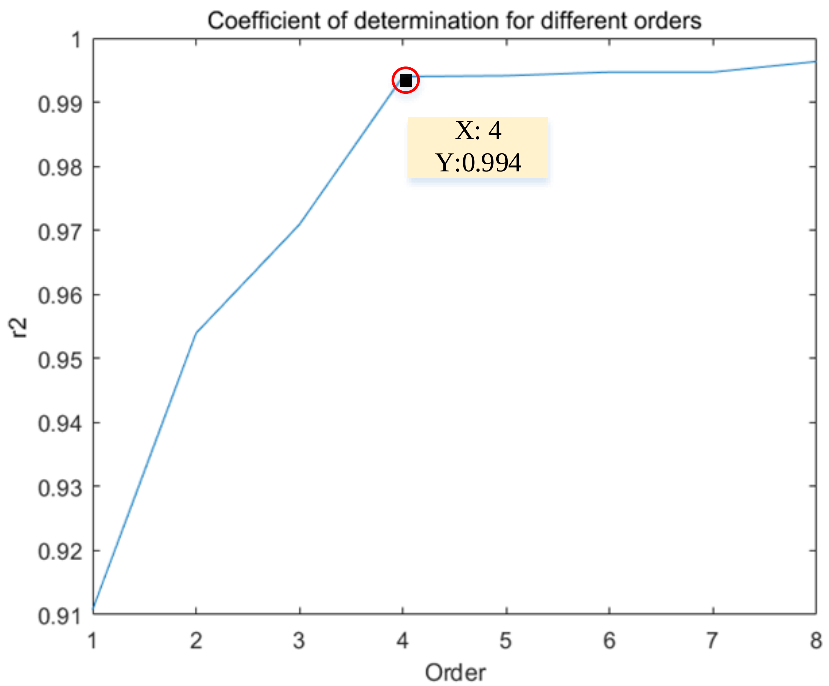

Fig. 2. Coefficient of determination for different orders of method I

By comparing the standard errors and coefficient of determination of different orders, it is finally found that when the order is 4, the standard error S_{yx} = 31.85, which is the smallest. Furthermore, the coefficient of determination r2 will increase very slow after 4th order, which is very close to 1.

Hence, consider comprehensively, the best polynomial order is 4.

The polynomial fitting function is:

y_moose=-0.2264*x.^4+5.6313*x.^3-36.7984*x.^2+95.6669*x.^1+554.4735.

The error now is 31.85. The number of moose before 1959 can be seen in the next second figure.

Plot:

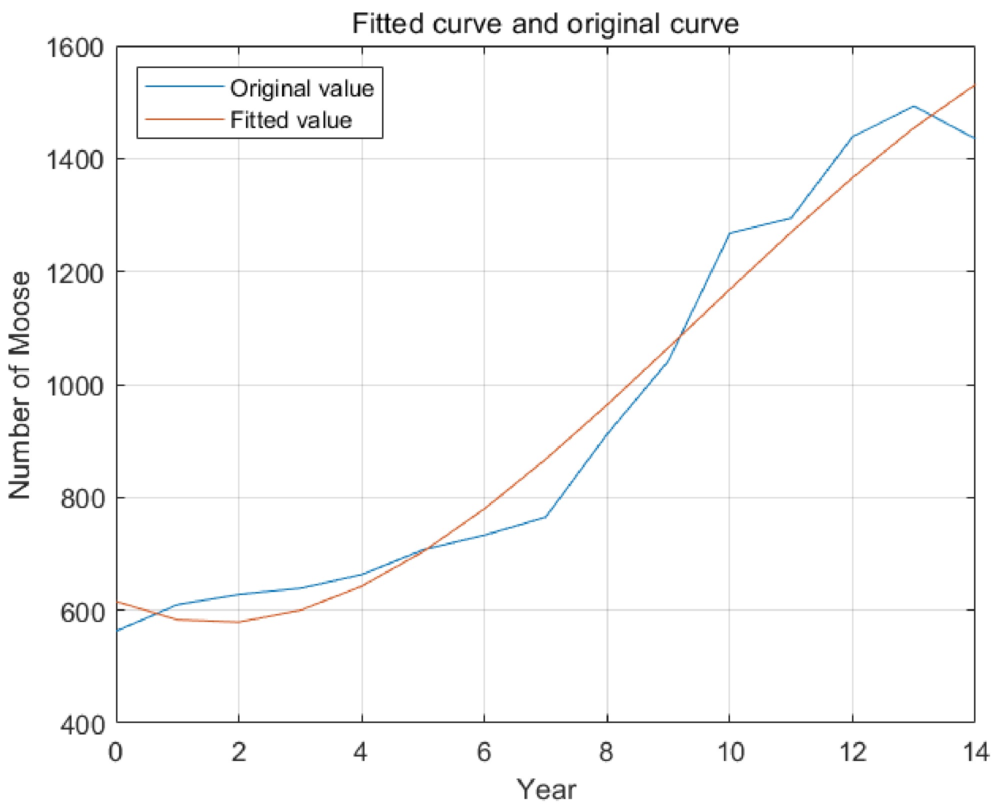

Fig. 3. Fitted curve and original curve of method I

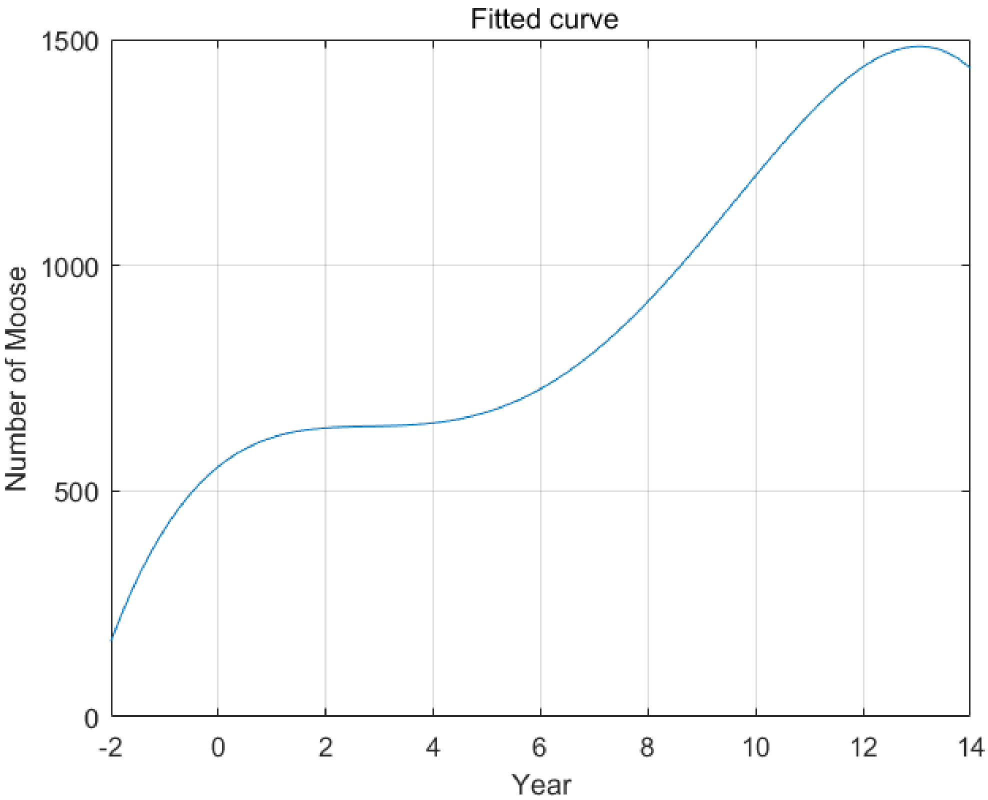

Fig. 4. Estimation of fitted curve of method I

1.2.2 Method II

When the range of x is set to (1959:1973), the results are as follows.

Result:

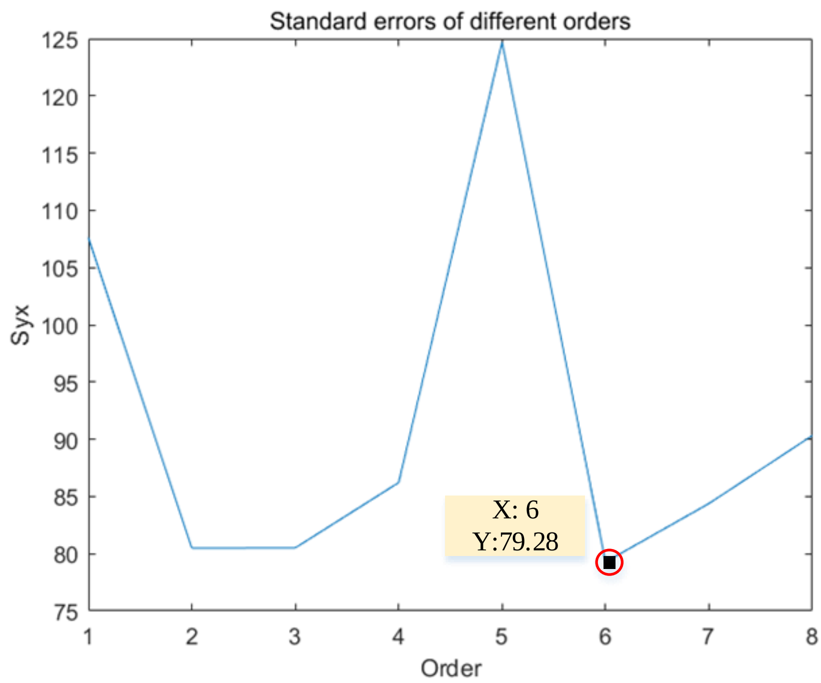

Fig. 5. Standard error of different orders of method II

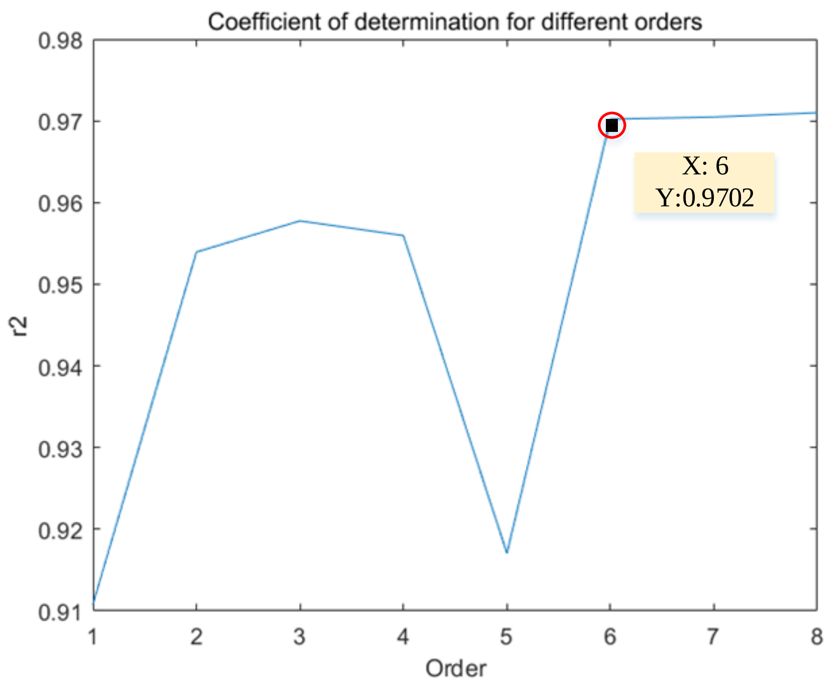

Fig. 6. Coefficient of determination for different orders of method II

By comparing the standard errors and coefficient of determination of different orders, it is finally found that when the order is 6, the standard error S_{yx} = 79.28, which is the smallest. Furthermore, the coefficient of determination r2 will increase very slow after 6th order, which is very close to 1.

Hence, consider comprehensively, the best polynomial order is 6.

The polynomial fitting function is:

y_moose=p1*x.^6+p2*x.^5+p3*x.^4+p4*x.^3+p5*x.^2+ p6*x.^1+p7.

p1 = -8.19232899472328e-12.

p2 = 1.869201746620641e-08.

p3 = -3.04549292149801e-05.

p4 = 0.205419912565625.

p5 = -85.6293750147719.

p6 = -1.072000217493370e+06.

p7 = 1.256588577934436e+09.

The error now is 79.28. The number of moose before 1959 can be seen in the next second figure.

Plot:

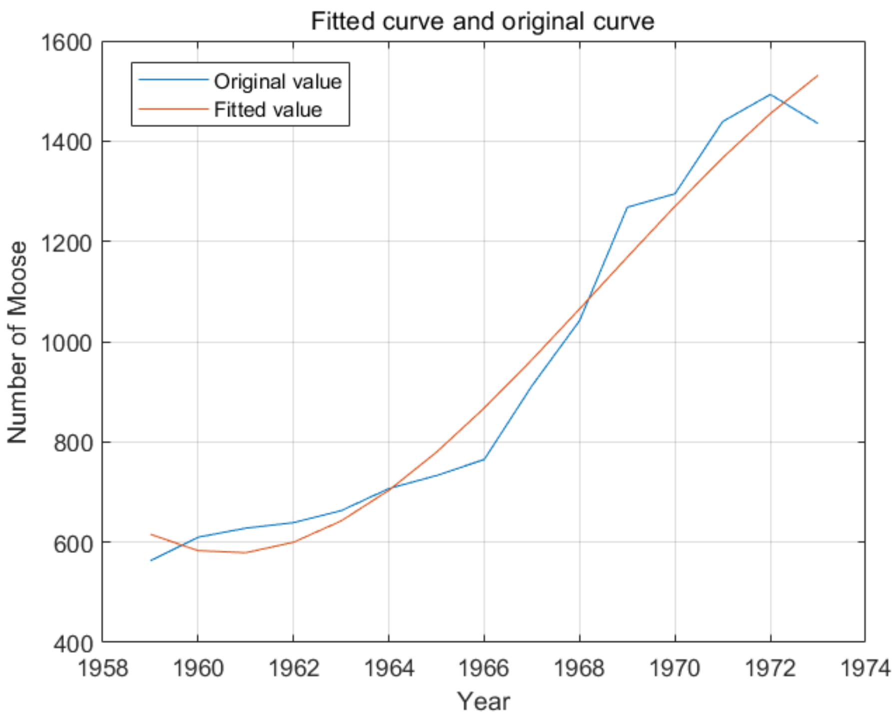

Fig. 7. Fitted curve and original curve of method II

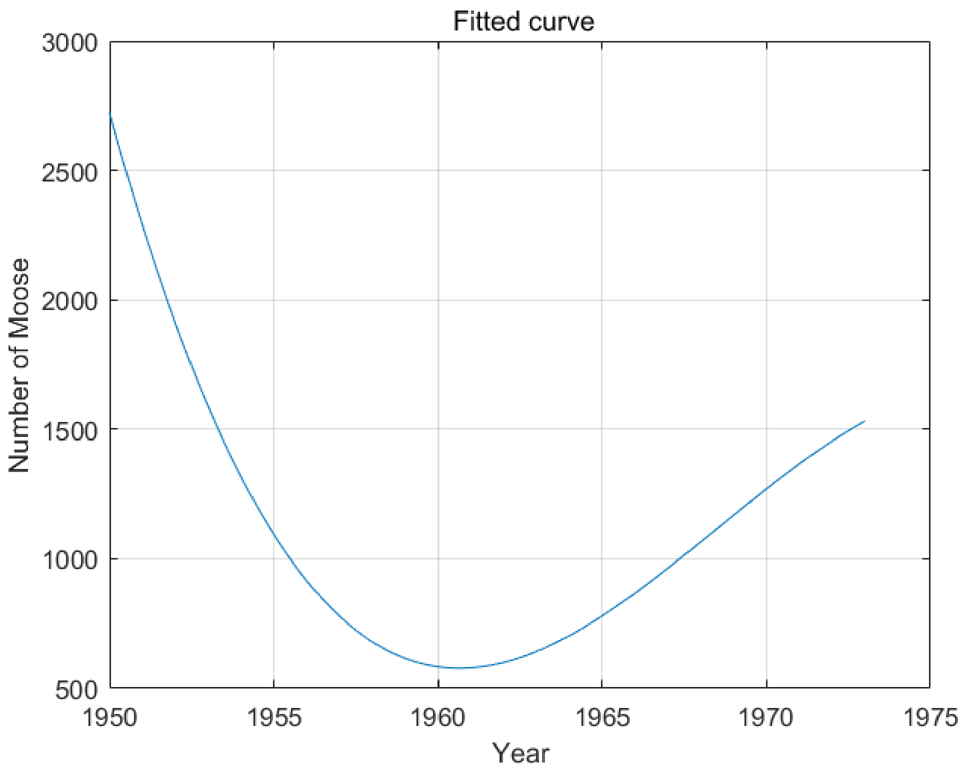

Fig. 8. Estimation of fitted curve of method II

1.3 The main code

```
% data import
Year=1959:2006;
Moose=[563 610 628 639 663 707 733 765 912 1042 1268 1295 1439 1493 1435 1467
1355 1282 1143 1001 1028 910 863 872 932 1038 1115 1192 1268 1335 1397 1216
1313 1590 1879 1770 2422 1163 500 699 750 850 900 1100 900 750,540,450];
Wolves=[20 22 22 23 20 26 28 26 22 22 17 18 20 23 24 31 41 44 34 40 43 50 30 14
23 24 22 20 16 12 12 15 12 12 13 17 16 22 24 14 25 29 19 17 19 29 30 30];

% data processing
% year=(0:14)';
year=(1959:1973)';
moose_interpolation=Moose(1:15)';

figure(1)
plot(year,moose_interpolation)
r2=[];syx=[];

% First order
```

```
Z = [ones(size(year)) year];
a=(Z'*Z)\(Z'*moose_interpolation);
Sr = sum((moose_interpolation-Z*a).^2);
r2(1) = 1-Sr/sum((moose_interpolation-mean(moose_interpolation)).^2);
syx(1) = sqrt(Sr/(length(year)-length(a)));

% Second order
Z = [ones(size(year)) year year.^2 ];
a=(Z'*Z)\(Z'*moose_interpolation);
Sr = sum((moose_interpolation-Z*a).^2);
r2(2) = 1-Sr/sum((moose_interpolation-mean(moose_interpolation)).^2);
syx(2) = sqrt(Sr/(length(year)-length(a)));

% Third order
Z = [ones(size(year)) year year.^2 year.^3];
a=(Z'*Z)\(Z'*moose_interpolation);
Sr = sum((moose_interpolation-Z*a).^2);
r2(3) = 1-Sr/sum((moose_interpolation-mean(moose_interpolation)).^2);
syx(3) = sqrt(Sr/(length(year)-length(a)));

% Fourth order
Z = [ones(size(year)) year year.^2 year.^3 year.^4];
a=(Z'*Z)\(Z'*moose_interpolation);
Sr = sum((moose_interpolation-Z*a).^2);
r2(4) = 1-Sr/sum((moose_interpolation-mean(moose_interpolation)).^2);
syx(4)= sqrt(Sr/(length(year)-length(a)));

% Fifth order
Z = [ones(size(year)) year year.^2 year.^3 year.^4 year.^5];
a=(Z'*Z)\(Z'*moose_interpolation);
Sr = sum((moose_interpolation-Z*a).^2);
r2(5) = 1-Sr/sum((moose_interpolation-mean(moose_interpolation)).^2);
syx(5) = sqrt(Sr/(length(year)-length(a)));

% Sixth order
Z = [ones(size(year)) year year.^2 year.^3 year.^4 year.^5 year.^6];
a=(Z'*Z)\(Z'*moose_interpolation);
Sr = sum((moose_interpolation-Z*a).^2);
r2(6) = 1-Sr/sum((moose_interpolation-mean(moose_interpolation)).^2);
syx(6)= sqrt(Sr/(length(year)-length(a)));

% Seventh order
Z = [ones(size(year)) year year.^2 year.^3 year.^4 year.^5 year.^6 year.^7];
a=(Z'*Z)\(Z'*moose_interpolation);
```

```
Sr = sum((moose_interpolation-Z*a).^2);
r2(7) = 1-Sr/sum((moose_interpolation-mean(moose_interpolation)).^2);
syx(7) = sqrt(Sr/(length(year)-length(a)));

% Eighth order
Z = [ones(size(year)) year year.^2 year.^3 year.^4 year.^5 year.^6 year.^7
year.^8 ];
a=(Z'*Z)\(Z'*moose_interpolation);
Sr = sum((moose_interpolation-Z*a).^2);
r2(8) = 1-Sr/sum((moose_interpolation-mean(moose_interpolation)).^2);
syx(8) = sqrt(Sr/(length(year)-length(a)));

% Fitting result
x_order=(1:8);
figure(2)
plot(x_order,r2);
xlabel('Order');ylabel('r2');title('Coefficient of determination for different
orders')
figure(3)
plot(x_order,syx);
xlabel('Order');ylabel('Syx');title('Standard errors of different orders')
% P=[ -0.2264 5.6313 -36.7984 95.6669 554.4735 ];
x=year;
% y_moose=-0.2264*x.^4+5.6313*x.^3-36.7984*x.^2+95.6669*x.^1+554.4735;

p7 = 1.256588577934436e+09;
p6 = -1.072000217493370e+06;
p5 = -85.6293750147719;
p4 = 0.205419912565625;
p3 = -3.04549292149801e-05;
p2 = 1.869201746620641e-08;
p1 = -8.19232899472328e-12;
y_moose=p1*x.^6+p2*x.^5+p3*x.^4+p4*x.^3+p5*x.^2+ p6*x.^1+p7 ;

figure(4) % Fitted curve and original curve
plot(year,moose_interpolation,year,y_moose);
% plot(year,polyval(P,x))
% plot(year,y_moose)
xlabel('Year');ylabel('Number of Moose');title('Fitted curve and original
curve')
legend('Original value','Fitted value')
grid on
```

Question 2: Numerical Differentiation

Based on the data in question (1), use numerical differential to give the growth rate of moose.

2.1 Problem description and ideas

This problem requires to use the method of numerical differentiation to solve the growth rate of moose based on the last question. Central Difference is used most widely as a Numerical Differentiation.

As a result, I used a simple method, which directly consider the change of one year over the number in last year as the growth rate, and central difference method, to calculate the growth rate of moose, and compare the result.

2.2 Answer and results

Plot:

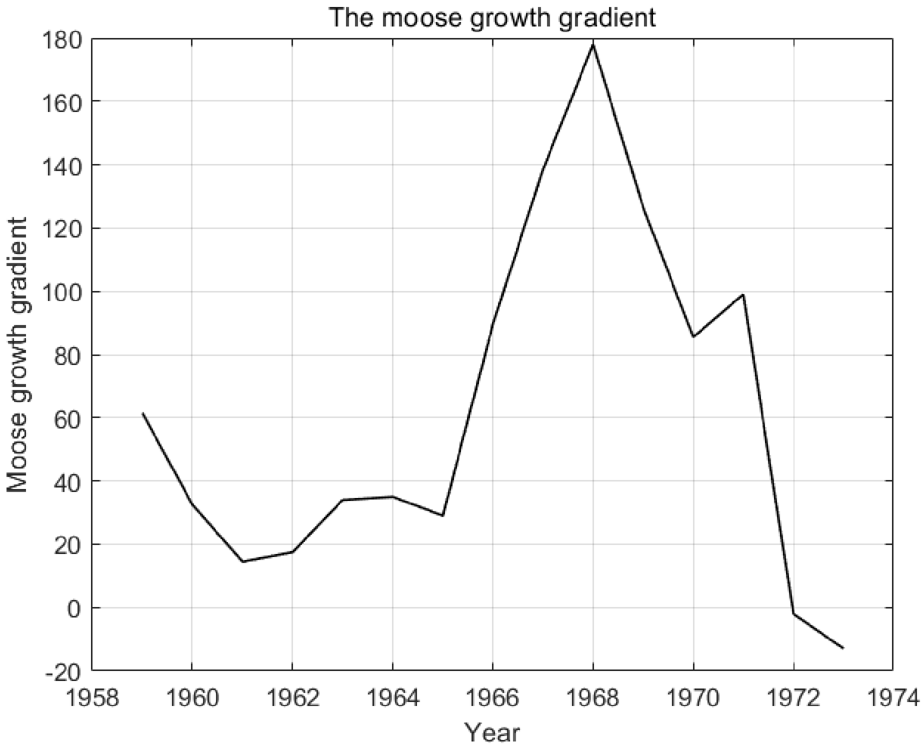

Fig. 9. The moose growth gradient

Fig. 10. The moose growth gradient in simple method and differentiation method

2.3 The main code

```
% the simple way to calculate the growth ratio
r_easy = (Moose(2:length(Moose))-Moose(1:length(Moose)-
1))./(Moose(1:length(Moose)-1))*100;
r_easy = [0 r_easy];
% use numerical differentiation to calcualte the growth ratio
len = length(Moose);
moose_gradient = zeros(1,len);
moose_gradient(1) = (-3*Moose(1)+4*Moose(2)-1*Moose(3))/(Year(3)-Year(1));
moose_gradient(len) = (3*Moose(len)-4*Moose(len-1)+1*Moose(len-2))/(Year(len)-
Year(len-2));
for r = 2:len-1
    moose_gradient(r) = (Moose(r+1)-Moose(r-1))/(Year(r+1)-Year(r-1));
end
```

The output code:

```
figure(6)
% plot(Year(1:length(Moose)),moose_gradient,'-k','linewidth',1,'markersize',5)
```

```
plot(Year(1:15),moose_gradient(1:15),'-k','linewidth',1,'markersize',5)
xlabel('Year');ylabel('Moose growth gradient');title('The moose growth
gradient')
grid on
r_numerical = moose_gradient./Moose*100;

figure(7)
% plot(Year(1:length(Moose)),r_easy,'-b','linewidth',1,'markersize',5)
plot(Year(1:15),r_easy(1:15),'linewidth',1,'markersize',5)
hold on
% plot(Year(1:length(Moose)),r_numerical,'-y','linewidth',1,'markersize',5)
plot(Year(1:15),r_numerical(1:15),'linewidth',1,'markersize',5)
xlabel('Year');ylabel('Moose growth rate(%)');title('The moose growth rate ')
legend('Simple method','Differentiation method')
grid on
```

Question 3: Predator-prey Models

You can use the model (22.49) and (22.50) on Page 578 of textbook to describe the change of the two populations over time t. please show how to create the model and the meanings of all parameters.

3.1 Problem description and ideas

According to the model (22.49) and (22.50) on textbook, the case study focuses on two applications of differential equations. The first relates to predator-prey models that are used to study species interactions. The second are equations derived from fluid dynamics that are used to simulate the atmosphere. Predator-prey models were developed independently in the early part of the twentieth century by the Italian mathematician Vito Volterra and the American biologist Alfred Lotka. These equations are commonly called Lotka-Volterra equations.

I established the mathematical model due to the principle of provided equations in the textbook, represent the population change of moose and wolves, and describe the progress of the establishment of the model.

3.2 Answer and results

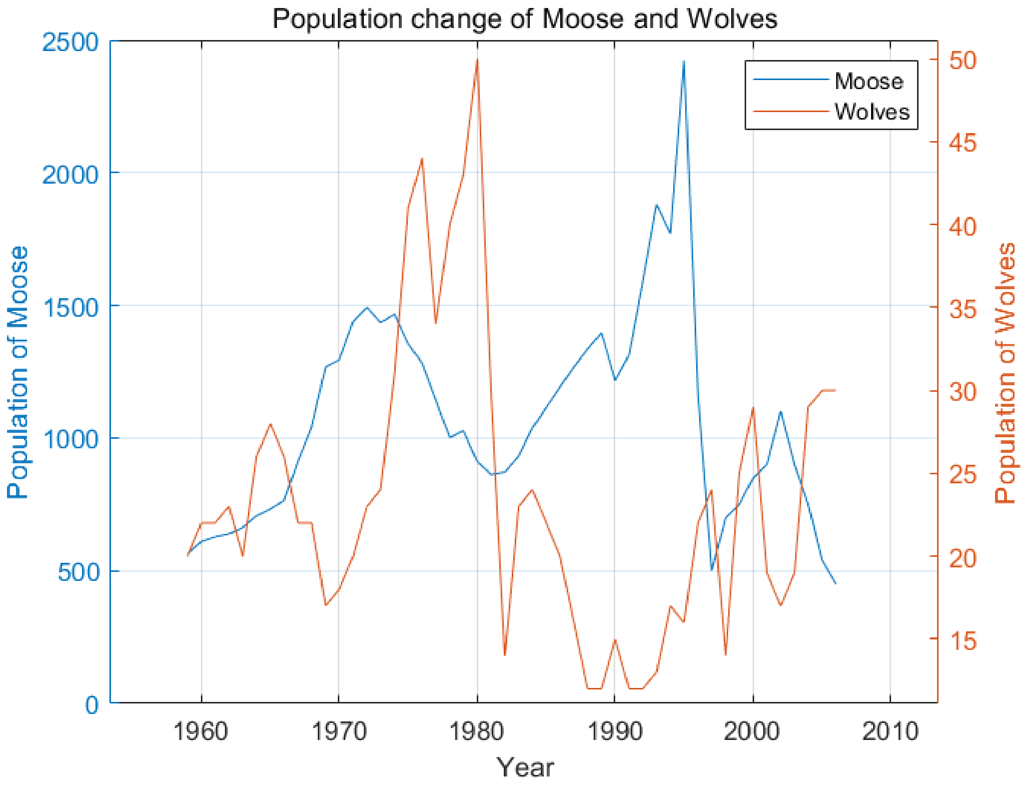

Fig. 11. Population changes of moose and wolves in provided model

3.3 Description of changes of two populations

As can be seen from the curve in the figure, the predator population is initially small, the prey grows exponentially.

At a certain point, the prey become so numerous that the predator population begins to grow.

Then, the increased predators cause the prey to decline. This decrease, in turn, leads to a decrease of the predators.

Eventually, the process repeats. Notice that, as expected, the predator peak lags the prey.

3.4 Model establishment

Table 1: Variables and Parameters

Variables	***Definitions***
x	The number of preys
y	The number of predators
a	The growth rate of the preys
b	The rate that characterizing the effect of the interactions on the prey death
c	The death rate of the predator
d	The rate that characterizing the effect of the interactions on the predator growth

In the model, prey (moose) and predator (wolf) are selected as the research objects, x and y are the number of prey and predators, respectively, a is the prey growth rate, c is the predator death rate, and b and d are the rates characterizing the effect of the predator-prey interactions on the prey death and the predator growth respectively.

Before establishing the model, two assumptions need to be made: First, assume that the predator (wolf) cannot survive without the prey (moose). Second, assume that the Isle Royale National Park is rich in resources and that the prey (moose) grows exponentially when it lives independently.

The prey (moose) grows exponentially when it lives independently, and the relative growth rate of the prey (moose) is a, that is $x' = ax$. The existence of predators reduces the growth rate of prey. Assuming that the degree of reduction is proportional to the number of predators, then $x(t)$ satisfies the equation:

$$\frac{dx}{dt} = x(a - by) = ax - bxy$$

The proportional coefficient a reflects the predator's ability to grab prey.

Because the predator cannot survive without the prey, and the death rate when it lives independently is d, that is, $y' = -dy$, and the existence of the prey provides food for the predator, which is equivalent to reducing the death rate of the predator and promoting its growth. Assuming that this effect is proportional to the number of preys, then y(t) satisfies the equation:

$$\frac{dy}{dt} = y(-c + dx) = -cy + dxy$$

The proportional coefficient d reflects the ability of the prey to feed the predator.

3.5 The main code

```
figure(8)
yyaxis left
plot(Year, Moose)
xlabel('Year');
ylabel('Population of Moose');
yyaxis right
plot(Year, Wolves)
ylabel('Population of Wolves')
grid on
title('Population change of Moose and Wolves');
legend('Moose','Wolves');
```

Question 4: Solution of ODEs

Use the parameters a=0.23, b=0.0133, c=0.4 and d=0.0004, solve the system of ordinary differential equations and compare your simulation with the data and determine the sum of the squares of the residuals between your model and the data for both the moose and wolves.

4.1 Problem description and ideas

There are mainly two methods for solving differential equations, Euler's method and fourth-order RK method.

In this question, the fourth-order RK method is used to solve the system of differential equations. The result of the solution is presented in the form of figures, and the sum of the residual squares of the two populations is calculated separately.

4.2 Answer and results

Plot:

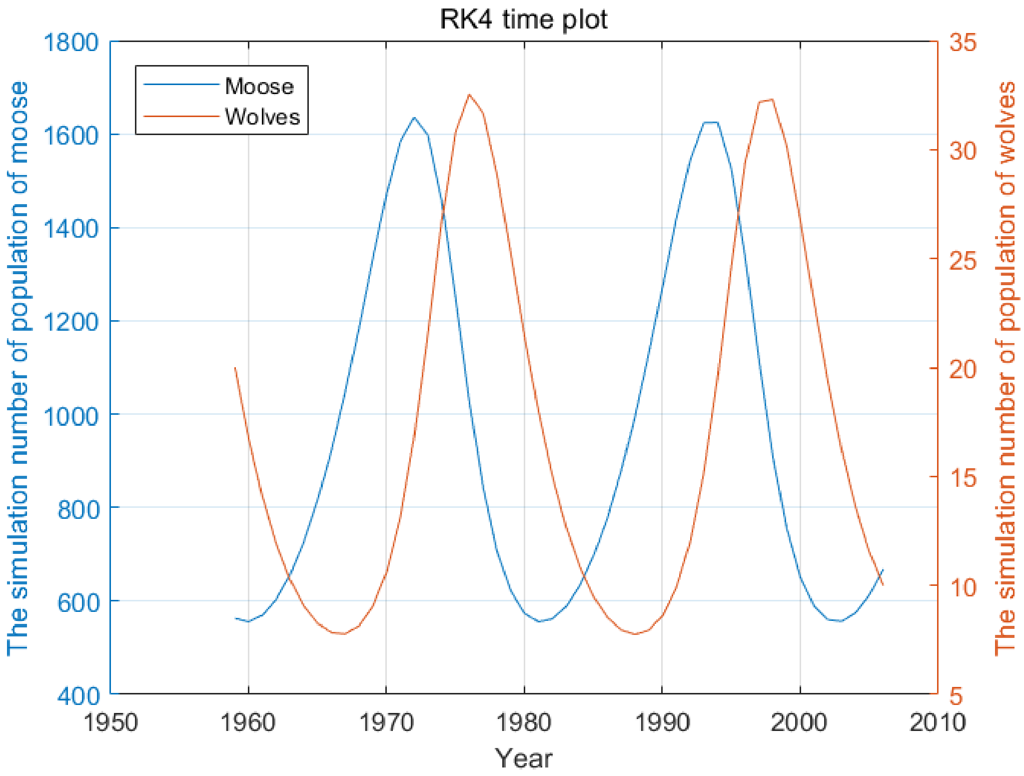

Fig. 12. Population changes of moose and wolves in RK4 method

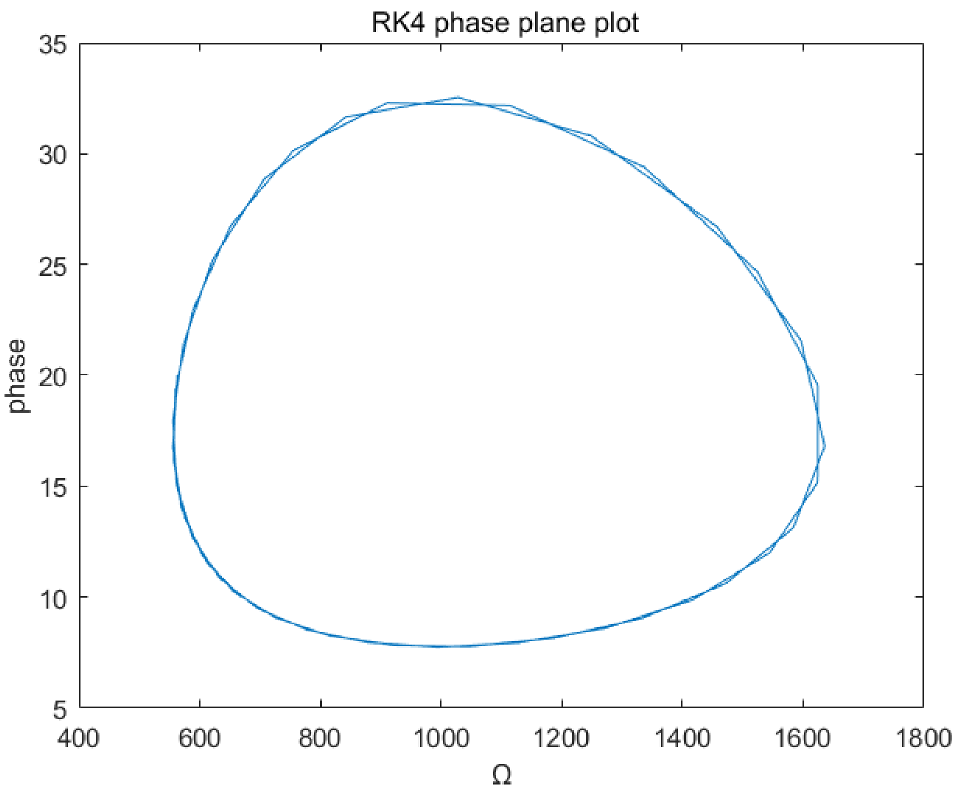

Fig. 13. Phase plane plot of moose and wolves in RK4 method

Value:

For moose, the sum of the squares of the residuals between the model and the data is 3.8661e+06, and for wolves the sum of the squares of the residuals is 5.8701e+03, total error is 3.8719e+06.

4.3 Analysis of original data

When using the origin data to make the simulation, it can be found that the error is relatively large, and also the simulation of wolves population has quite difference from the origin data. The assumption in the previous part is that wolves cannot survive without the existence of moose, and it can only eat moose. Thus, if population of moose is small, population of the wolves will decrease because of lack of food. In the origin data, moose number is small which means population of wolves should also decrease. However, the actual number of wolves is increasing which contradicts to our assumption.

Hence, the optimized data could be:

Moose=[563 610 628 639 663 707 733 765 912 1042 1268 1295 1439 1493 1435 1467 1355 1282 1143 1001 1028 910 863 872 932 1038 1115 1192 1268 1335 1397 1216 1313 1590 1879 1770 1500 1163 500 699 750 850 900 1100 900 750 540 450];

Wolves=[20 18 16 14 12 14 14 16 18 18 17 18 20 23 24 31 35 38 35 33 30 27 25 14 23 24 22 20 16 12 12 15 12 12 13 17 16 22 24 14 25 29 19 17 19 29 30 30];

4.4 The main code

```
function yp = predprey(t,y,a,b,c,d)
yp = [a*y(1)-b*y(1)*y(2);-c*y(2)+d*y(1)*y(2)];
end
h=1;tspan=[1959 2006];y0=[563 20];
a=0.23;b=0.0133;c=0.4;d=0.0004; %Q4 data
a2=0.2550;b2=0.0107;c2=0.2251;d2=0.0002;%Q5 data
[time, num] = rk4sys(@predprey,tspan,y0,h,a,b,c,d);
[time2, num2] = rk4sys(@predprey,tspan,y0,h,a2,b2,c2,d2);
```

The output code

```
figure(9)
yyaxis left
plot(time, num(:,1))
xlabel('Year');
ylabel('The simulation number of population of moose');
yyaxis right
plot(time, num(:,2))
ylabel('The simulation number of population of wolves')
grid on
title('RK4 time plot');
legend('Moose','Wolves');
grid on

figure(10)
plot(num(:,1),num(:,2))
xlabel('Moose');ylabel('Wolves');title('RK4 phase plane plot')
```

4.5 Function used in this part

Fourth-order RK method:

```
function [tp,yp] = rk4sys(dydt,tspan,y0,h,varargin)
% rk4sys: fourth-order Runge-Kutta for a system of ODEs
%   [t,y] = rk4sys(dydt,tspan,y0,h,p1,p2,...): integrates
%           a system of ODEs with fourth-order RK method
% input:
%   dydt = name of the M-file that evaluates the ODEs
%   tspan = [ti, tf]; initial and final times with output
%           generated at interval of h, or
%   = [t0 t1 ... tf]; specific times where solution output
%   y0 = initial values of dependent variables
%   h = step size
%   p1,p2,... = additional parameters used by dydt
% output:
%   tp = vector of independent variable
%   yp = vector of solution for dependent variables
if nargin<4,error('at least 4 input arguments required'), end
if any(diff(tspan)<=0),error('tspan not ascending order'), end
n = length(tspan);
ti = tspan(1);tf = tspan(n);
if n == 2
t = (ti:h:tf)'; n = length(t);
if t(n)<tf
t(n+1) = tf;
```

```
n = n+1;
end
else
t = tspan;
end
tt = ti; y(1,:) = y0;
np = 1; tp(np) = tt; yp(np,:) = y(1,:);
i=1;
while(1)
tend = t(np+1);
hh = t(np+1) - t(np);
if hh>h,hh = h;end
while(1)
if tt+hh>tend,hh = tend-tt;end
k1 = dydt(tt,y(i,:),varargin{:})';
ymid = y(i,:) + k1.*hh./2;
k2 = dydt(tt+hh/2,ymid,varargin{:})';
ymid = y(i,:) + k2*hh/2;
k3 = dydt(tt+hh/2,ymid,varargin{:})';
yend = y(i,:) + k3*hh;
k4 = dydt(tt+hh,yend,varargin{:})';
phi = (k1+2*(k2+k3)+k4)/6;
y(i+1,:) = y(i,:) + phi*hh;
tt = tt+hh;
i=i+1;
if tt>=tend,break,end
end
np = np+1; tp(np) = tt; yp(np,:) = y(i,:);
if tt>=tf,break,end
end
end
```

Question 5: Curve fitting method

Develop a curve fitting mothed based on the data and model to determine the four parameters a, b, c and d. please show your idea, method, algorithm, code and results.

5.1 Problem description and ideas

For a system of nonlinear multivariate differential equations in a given format, determining the parameters is the inverse process of solving the equations, which is very complicated to implement.

Therefore, it is necessary to make some changes to the equations first, then use the least square method and other practical methods to estimate the parameters of the changed equations.

5.2 Method

The formula of the prey-predator model is as follows:

$$\begin{cases} \dfrac{dx}{dt} = ax(t) - bx(t)y(t) \\ \dfrac{dy}{dt} = -cy(t) + dx(t)y(t) \end{cases}$$

Transform the formula:

$$\begin{cases} \dfrac{x'(t)}{x(t)} = a - by(t) \\ \dfrac{y'(t)}{y(t)} = -c + dx(t) \end{cases}$$

It can be taken to:

$$Z_1(t) = \frac{x'(t)}{x(t)}, Z_2(t) = \frac{y'(t)}{y(t)}$$

So the original formula can be reduced to:

$$\begin{cases} Z_1(t) = a - by(t) \\ Z_2(t) = -c + dx(t) \end{cases}$$

x(t) and y(t) can be obtained by numerical differentiation based on the given values of x and y. In question 2, x(t) has been obtained, and y(t) can be obtained in the same way. After calculating x(t) and y(t), Z_1(t) and Z_2(t) can be obtained, so that the estimation of the parameter a, b, c, d is transformed into two linear least squares estimation problems.

The objective functions of the two linear least squares are:

$$\begin{cases} f1 = [a - by(t)] - Z_1 \\ f2 = [-c + dx(t)] - Z_2 \end{cases}$$

Then, the value of the parameter a, b, c, d can be obtained using linear regression.

5.3 Algorithm

In programming, the algorithm *linregr* in the textbook is used for curve fitting, and its principle is least squares fit of straight line to data by solving the normal equations, the input of the algorithm is the independent variable and the dependent variable, and the output is the slope, intercept and coefficient of determination.

5.4 Answer and results

The result of the algorithm is (a,b,c,d)=(0.1144,0.0051,0.0804,0.0001), and the 95% confidence interval of each value calculated by another fitting method is (-0.0262—0.2550,-0.0107 —0.0005, -0.2251—0.0642, -0.0000—0.0002).

By comparing the error and trying to bring in the value, it can be found that the error obtained by taking the edge value of the confidence interval for each value is the smallest, and the curve can also maintain periodic oscillation (i.e. the phase plane plot shown below is closed-loop).

Hence, after comparing the effects of different values of each parameter, taking into account errors and oscillations, the optimized result is (a,b,c,d)=(0.2550, 0.0107, 0.2251, 0.0002), total error is 1.6492e+06.

The following four pictures respectively show: the curve fitting graph of parameters a and b, the curve fitting graph of parameters c and d, the population change, and the phase graph obtained by the model.

Plot:

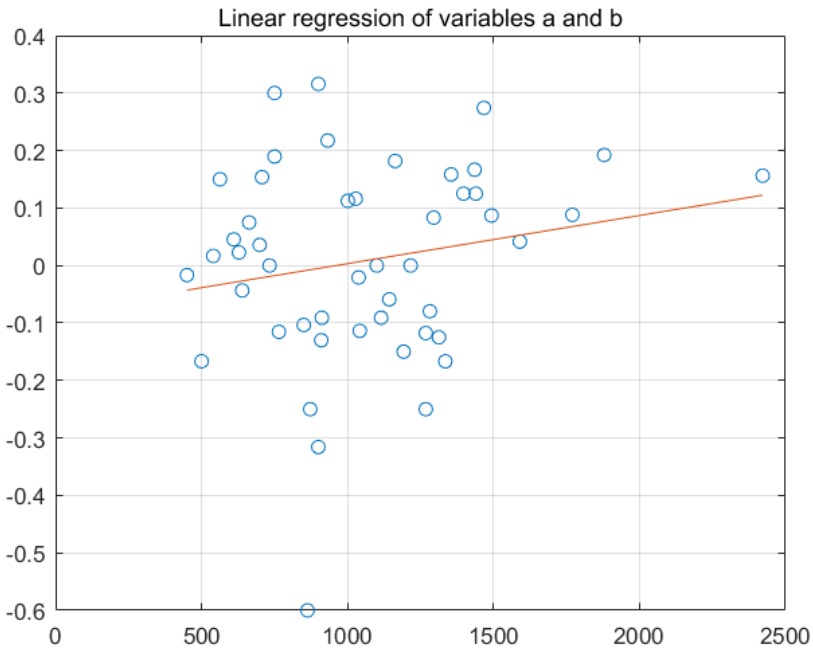

Fig. 14. Linear regression of variables a and b

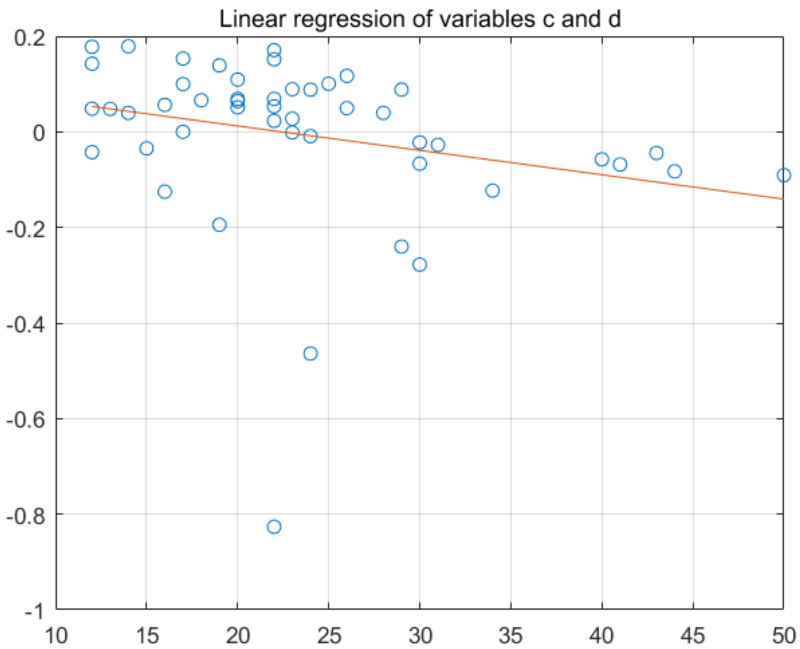

Fig. 15. Linear regression of variables c and d

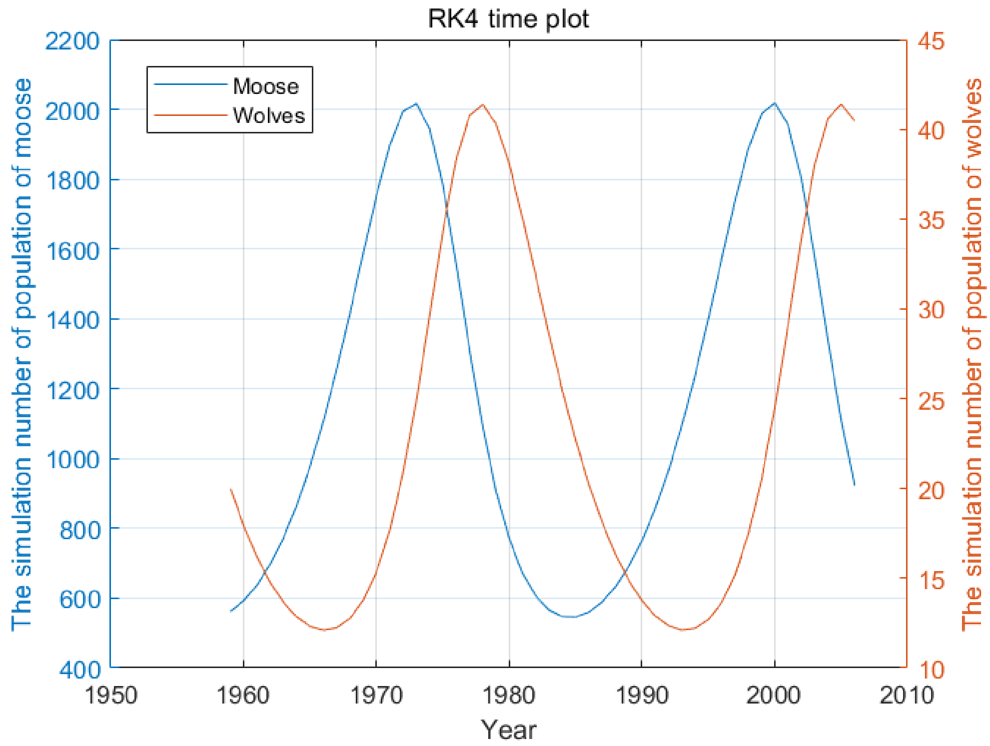

Fig. 16. Population changes of moose and wolves with optimized parameters

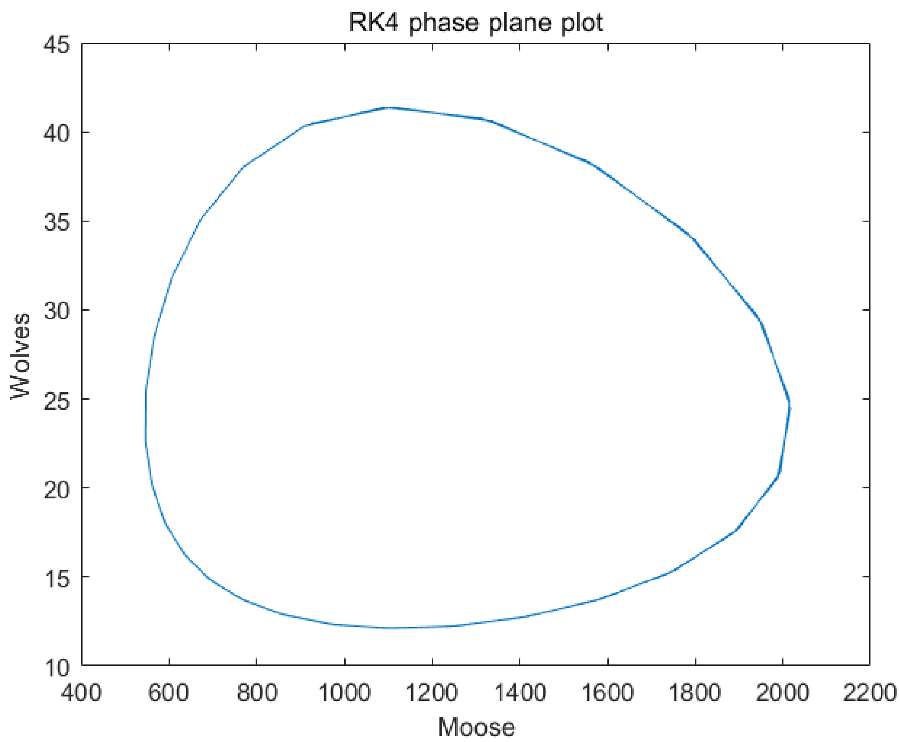

Fig. 17. Phase plane plot of moose and wolves with optimized parameters

5.5 The main code

```
x=Year;y=Moose;y2=Wolves;
% use numerical differentiation to calcualte the growth ratio of wolves
n = length(y2);
wolves_gradient = zeros(1,n);
wolves_gradient(1) = (-3*y2(1)+4*y2(2)-1*y2(3))/(x(3)-x(1));
wolves_gradient(n) = (3*y2(n)-4*y2(n-1)+1*y2(n-2))/(x(n)-x(n-2));
for i = 2:n-1
    wolves_gradient(i) = (y2(i+1)-y2(i-1))/(x(i+1)-x(i-1));
end
[a, r2] = linregr(y2,(moose_gradient./y));
% [a1, r22] = linregr(y,(wolves_gradient./y2));
```

5.6 Function used in this part:

Linear regression curve fitting:

```
function [a, r2] = linregr(x,y)
% linregr: linear regression curve fitting
%  [a, r2] = linregr(x,y): Least squares fit of straight
% line to data by solving the normal equations
% input:
%  x = independent variable
%  y = dependent variable
% output:
%  a = vector of slope, a(1), and intercept, a(2)
%  r2 = coefficient of determination
n = length(x);
if length(y)~=n, error('x and y must be same length'); end
x = x(:); y = y(:);  % convert to column vectors
sx = sum(x); sy = sum(y);
sx2 = sum(x.*x); sxy = sum(x.*y); sy2 = sum(y.*y);
a(1) = (n*sxy-sx*sy)/(n*sx2-sx^2);
a(2) = sy/n-a(1)*sx/n;
r2 = ((n*sxy-sx*sy)/sqrt(n*sx2-sx^2)/sqrt(n*sy2-sy^2))^2;

% create plot of data and best fit line
xp = linspace(min(x),max(x),2);
yp = a(1)*xp+a(2);
figure
plot(x,y,'o',xp,yp);
title('Linear regression of variables c and d');
```

```
grid on
end
```

Question 6: Spline interpolation

In (22.49), the second part of the rhs term –bxy can be viewed as the number of moose eaten by wolves, please show the total number of moose eaten from 1959 to 1989.

6.1 Problem description and ideas

The formula (22.49) can be deformed to obtain:

$$\mathrm{bxy} = \mathrm{ax} - \mathrm{x}'(\mathrm{t})$$

In this way, the integral of bxy can be transformed into the integral of $\mathrm{ax} - \mathrm{x}'(\mathrm{t})$. x' can be calculated using the numerical differentiation method in question 2. x and x' cannot be derived from the analytical formula of the function, so they cannot be directly integrated. This article first performs spline interpolation on x and x', and then integrates the data after spline interpolation.

6.2 Answer and results

By calculation, the number of moose eaten by wolves from 1959 to 1989 was 6,806. The figure below reflects the fit of the spline interpolation method to the data.

Plot:

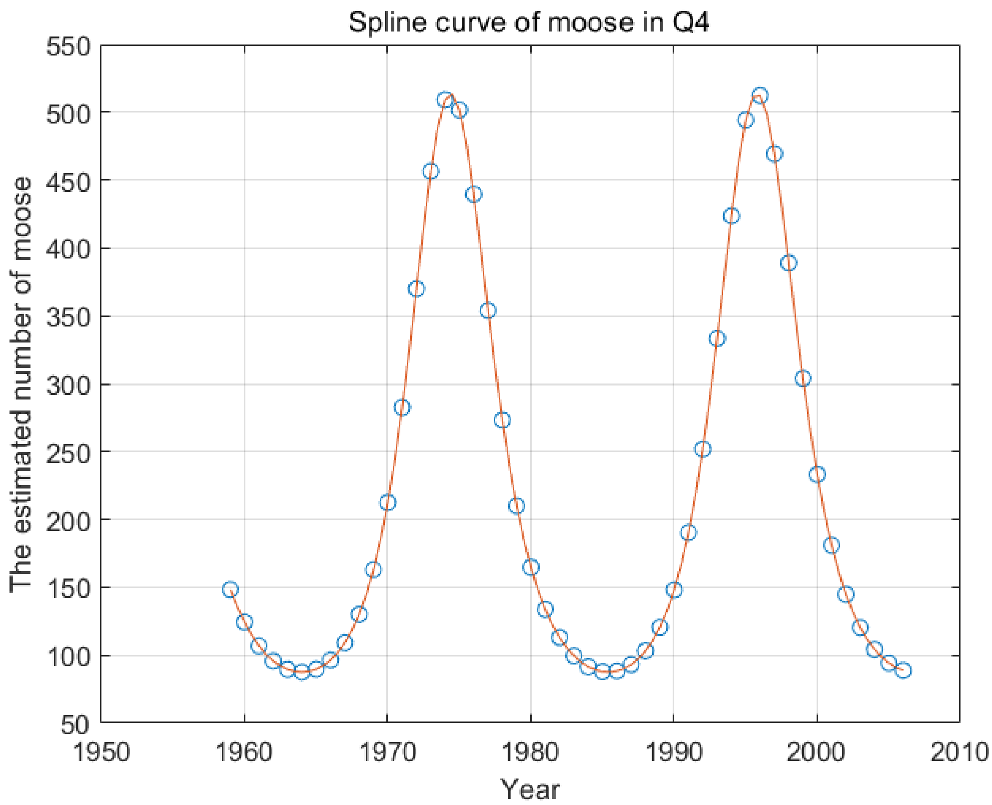

Fig. 18. Spline curve of moose in question 4

6.3 The main code

```
% Find the differential of moose in Q4
moose_Q4_gradient = zeros(1,length(Year));
moose_Q4_gradient(1) = (-3*Moose_Q4(1)+4*Moose_Q4(2)-1*Moose_Q4(3))/(Year(3)-
Year(1));
moose_Q4_gradient(length(Year)) = (3*Moose_Q4(length(Year))-
4*Moose_Q4(length(Year)-1)+1*Moose_Q4(length(Year)-2))/(Year(length(Year))-
Year(length(Year)-2));
for s = 2:length(Year)-1
    moose_Q4_gradient(s) = (Moose_Q4(s+1)-Moose_Q4(s-1))/(Year(s+1)-Year(s-1));
end
bxy=0.23.*Moose_Q4-moose_Q4_gradient;
% Interpolation + integration
% bxy_interp=interp1(Year,bxy,'o',spline)
spline_moose_bxy_x=(1959:0.5:2006);
spline_moose_bxy=spline(Year,bxy,spline_moose_bxy_x);
figure(21)
```

```
plot(Year,bxy,'o',spline_moose_bxy_x,spline_moose_bxy);
xlabel('Year');ylabel('The estimated number of moose');title('Spline curve of
moose in Q4');
grid on;
spline_bxy=spline(Year,bxy);
diff(fnval(fnint(spline_bxy),[1959 1989]));
```

Question 7: Numeral Calculations

Use the result of question (4), you can get the period T of the periodical solution. Select the interval [1975, 1975+T], give the mean numbers of moose and wolves in the period. Use data in the period, compute the maximum and minimum numbers of moose and wolves in the period. Compare between the estimation and the estimation numbers by question (5).

7.1 Problem description and ideas

This problem requires to use numeral calculation methods to achieve some special values of the model and make compassions for different models in question 4 and question 5.

7.2 Answer and results

Table 2: Variables and Parameters

Period	Numerical value	Moose	Wolves
T_Q4	Average	1005	18
	Maximum	1636	33
22	Minimum	555	8
T_Q5	Average	1123	20
	Maximum	2019	41
27	Minimum	545	12

Plot:

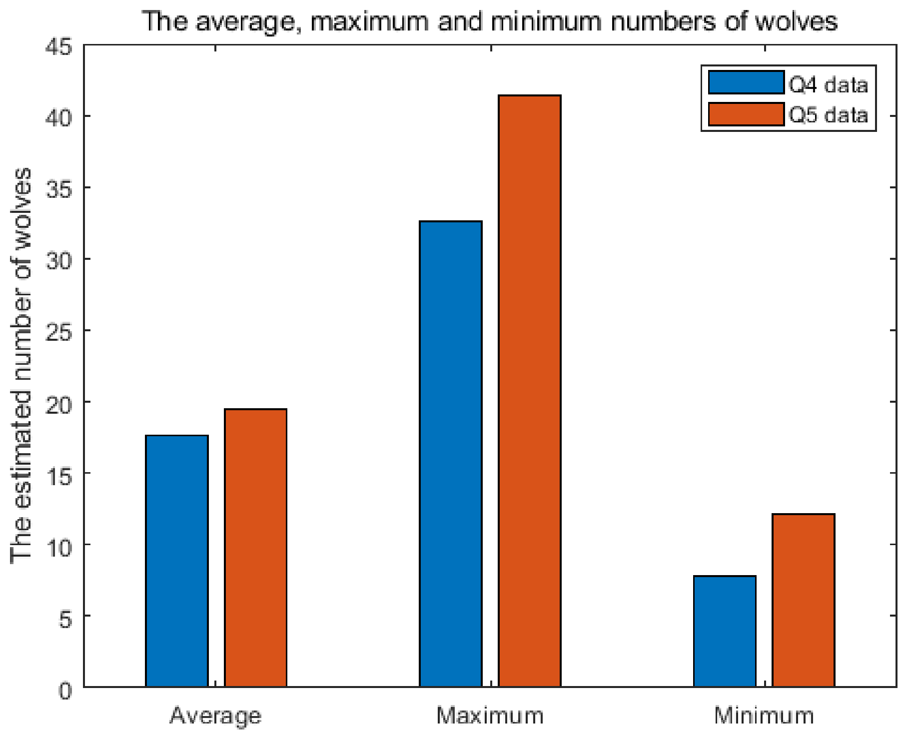

Fig. 19. The average, maximum and minimum numbers of wolves

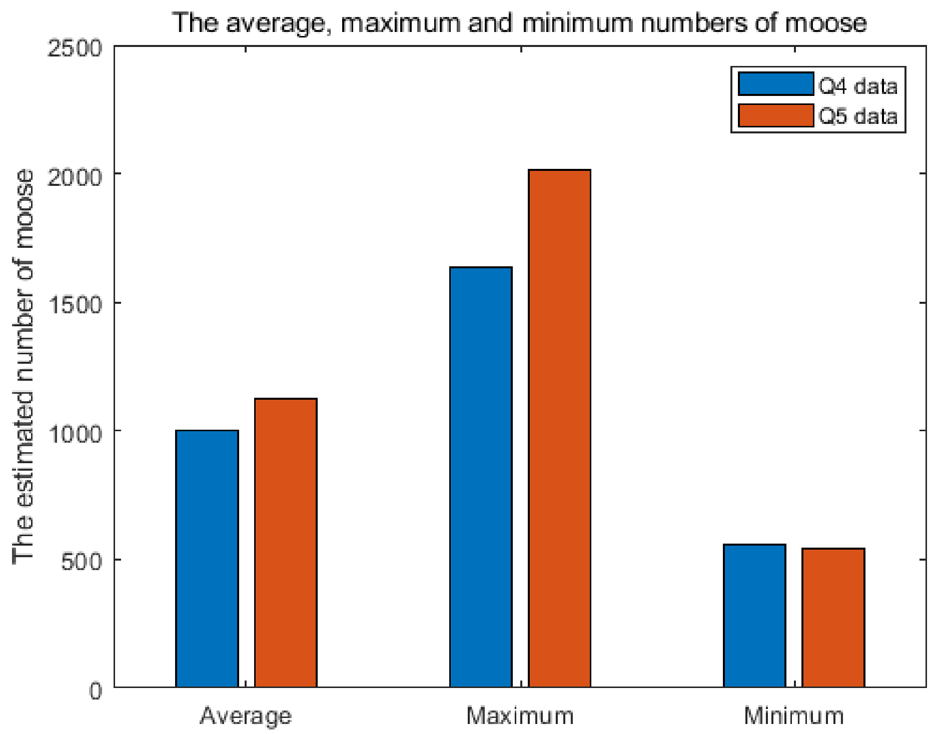

Fig. 20. The average, maximum and minimum numbers of moose

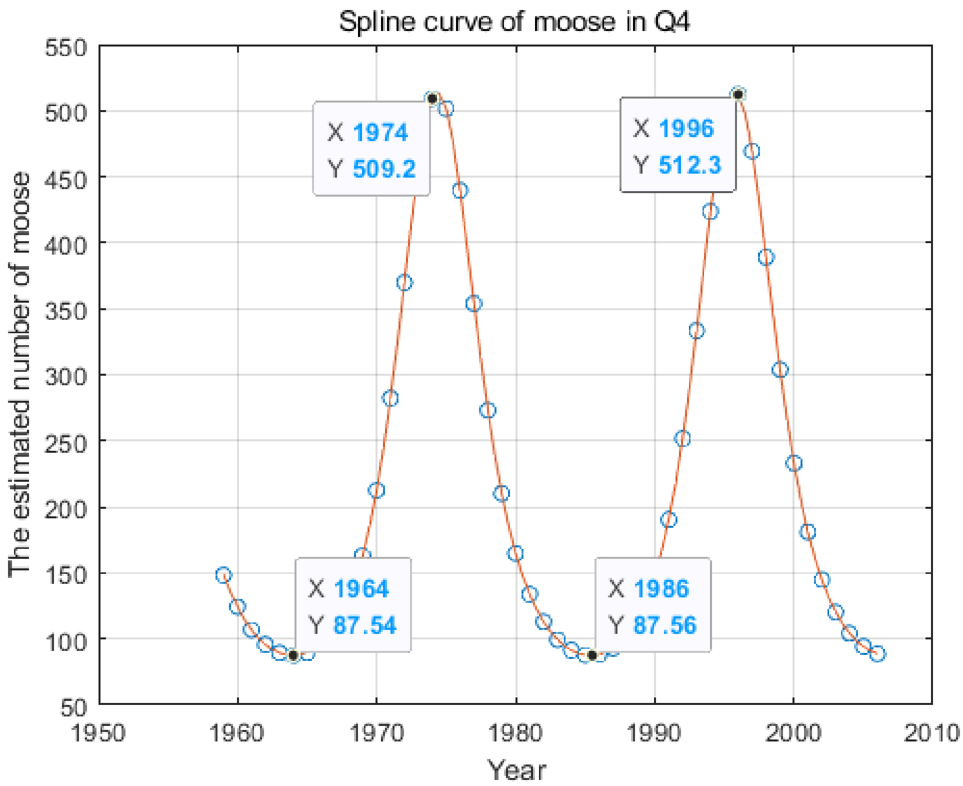

Fig. 21. Spline curve of moose in question 4

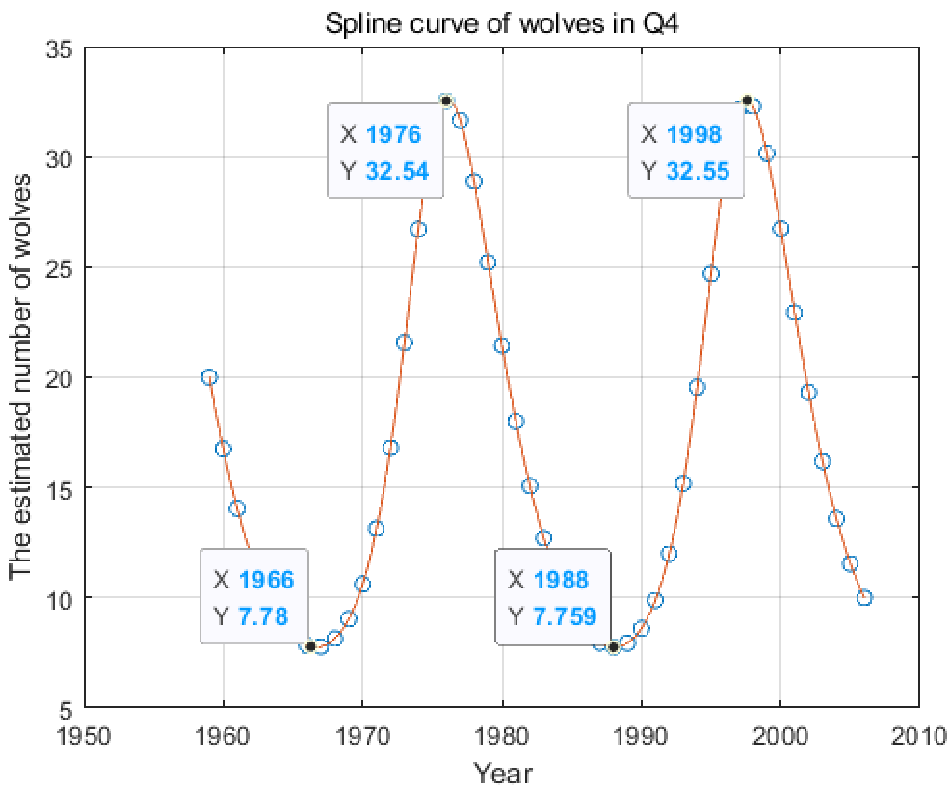

Fig. 22. Spline curve of wolves in question 4

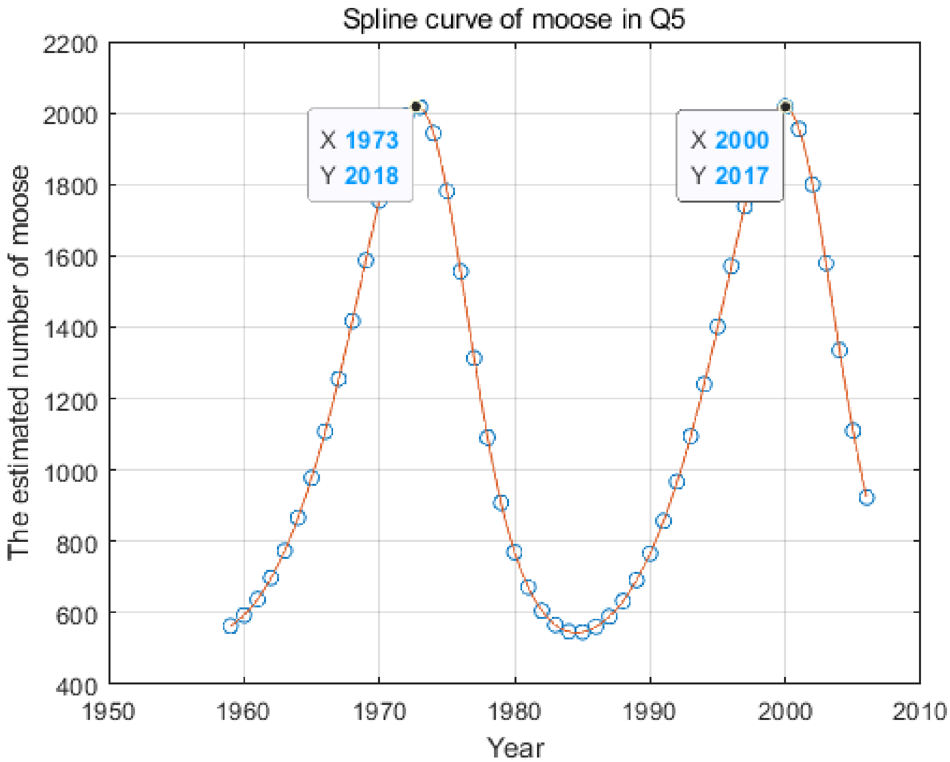

Fig. 23. Spline curve of moose in question 5

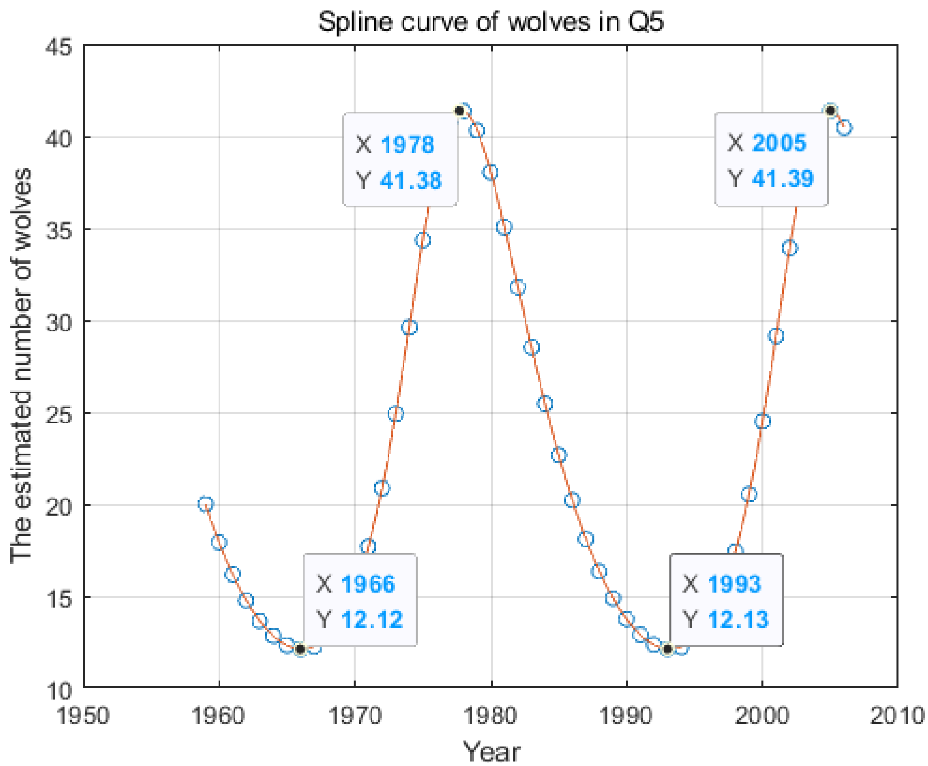

Fig. 24. Spline curve of wolves in question 5

7.3 The main code:

```
% Q4 moose
T_Q4=22;
% average
spline_moose_Q4a=spline(Year,Moose_Q4);
moose_average_Q4=diff(fnval(fnint(spline_moose_Q4a),[1975 1975+T_Q4]))/T_Q4 ;
% max min
spline_moose_Q4_x=(1959:0.1:2006);
spline_moose_Q4=spline(Year,Moose_Q4,spline_moose_Q4_x);
max(spline_moose_Q4);
min(spline_moose_Q4);
% Q4 wolves
% average
spline_wolves_Q4a=spline(Year,Wolves_Q4);
wolves_average_Q4=diff(fnval(fnint(spline_wolves_Q4a),[1975 1975+T_Q4]))/T_Q4 ;
% max min
spline_wolves_Q4_x=(1959:0.1:2006);
spline_wolves_Q4=spline(Year,Wolves_Q4,spline_wolves_Q4_x);

figure(11)
plot(Year,Wolves_Q4,'o',spline_wolves_Q4_x,spline_wolves_Q4)
xlabel('Year');ylabel('The estimated number of wolves');title('Spline curve of
wolves in Q4');
grid on;
max(spline_wolves_Q4);
min(spline_wolves_Q4);
% Q5 moose
T_Q5=27;
% average
spline_moose_Q5a=spline(Year,Moose_Q5);
moose_average_Q5=diff(fnval(fnint(spline_moose_Q5a),[1975 1975+T_Q5]))/T_Q5 ;
% max min
spline_moose_Q5_x=(1959:0.1:2006);
spline_moose_Q5=spline(Year,Moose_Q5,spline_moose_Q5_x);

figure(12);
plot(Year,Moose_Q5,'o',spline_moose_Q5_x,spline_moose_Q5);
xlabel('Year');ylabel('The estimated number of moose');title('Spline curve of
moose in Q5');
grid on;
max(spline_moose_Q5);
min(spline_moose_Q5);
% Q5wolves
```

```
% average
spline_wolves_Q5a=spline(Year,Wolves_Q5);
wolves_average_Q5=diff(fnval(fnint(spline_wolves_Q5a),[1975 1975+T_Q4]))/T_Q5 ;
% max min
spline_wolves_Q5_x=(1959:0.1:2006);
spline_wolves_Q5=spline(Year,Wolves_Q5,spline_wolves_Q5_x);

figure(13);
plot(Year,Wolves_Q5,'o',spline_wolves_Q5_x,spline_wolves_Q5)
xlabel('Year');ylabel('The estimated number of wolves');title('Spline curve of
wolves in Q5');
grid on;
max(spline_wolves_Q5);
min(spline_wolves_Q5);
% bar wolves
y_wolves_Q6=[wolves_average_Q4,wolves_average_Q5;max(spline_wolves_Q4),max(spli
ne_wolves_Q5);
             min(spline_wolves_Q4),min(spline_wolves_Q5)];
x_wolves_Q6={'Average';'max';'min'};
figure(14)
bar(y_wolves_Q6)
set(gca,'XTickLabel',x_wolves_Q6);
legend('Q4 data','Q5 data')
ylabel('The estimated number of wolves');title('The average, maximum and
minimum numbers of wolves');
% bar moose
y_moose_Q6=[moose_average_Q4,moose_average_Q5;max(spline_moose_Q4),max(spline_m
oose_Q5);
             min(spline_moose_Q4),min(spline_moose_Q5)];
x_moose_Q6={'Average';'max';'min'};

figure(15)
bar(y_moose_Q6)
set(gca,'XTickLabel',x_moose_Q6);
legend('Q4 data','Q5 data')
ylabel('The estimated number of moose');title('The average, maximum and minimum
numbers of moose');
```

Question 8: Shooting method

For the ODE with given parameters in the question (4), use the shooting method of ODE solve the system of equations with boundary value moose(1959) = 1028 and wolves(2010) = 25.

8.1 Problem description and ideas

The shooting method is based on converting the boundary-value problem into an equivalent initial-value problem. A trial-and-error approach is then implemented to develop a solution for the initial-value version that satisfies the given boundary conditions.

This method matches this system, as it is nicely illustrated for a second-order linear ODE such, though the method can also be employed for higher-order and nonlinear equations.

8.2 Answer and results

The following figure respectively reflects the number of wolves and moose, and the phase diagram calculated under the shooting method.

Plot:

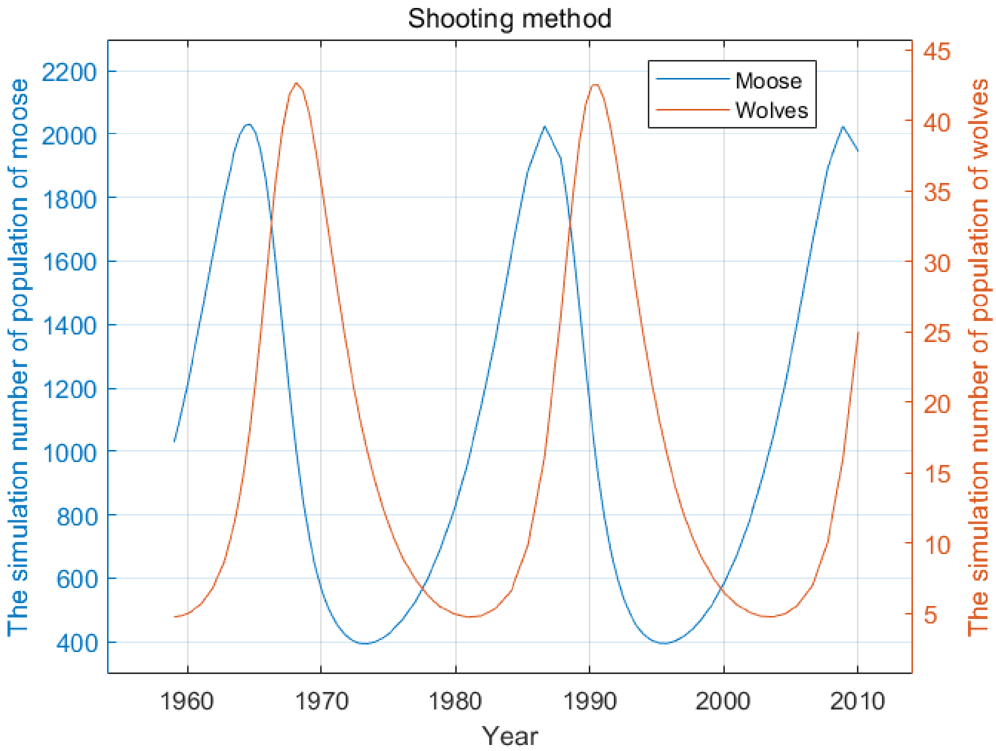

Fig. 25. The population changes of moose and wolves in shooting method

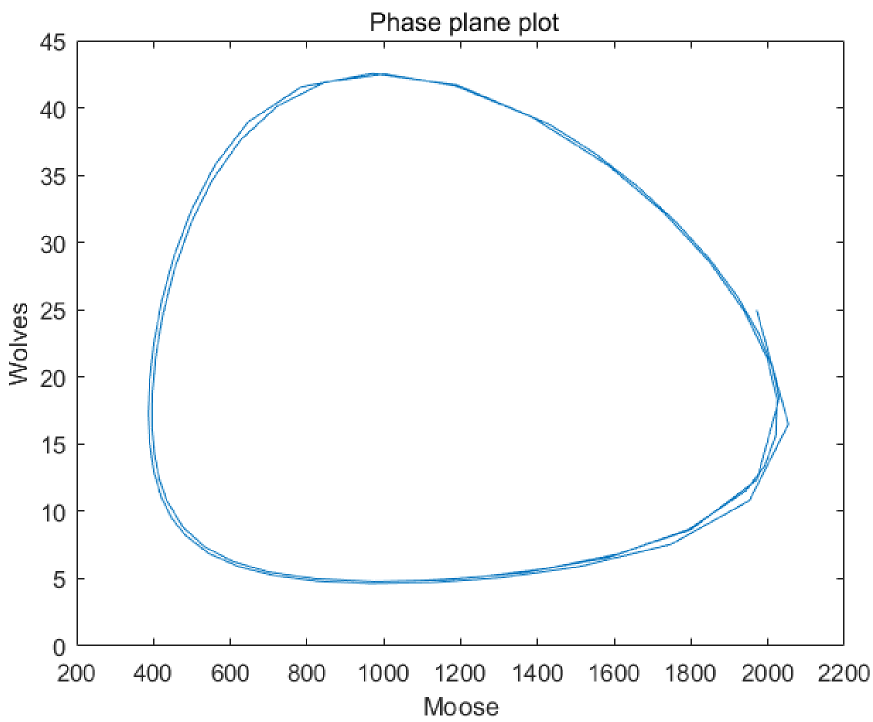

Fig. 26. The phase plane plot of moose and wolves in shooting method

8.3 The main code:

```
    function dy=EX2402(t,y)
        dy=[0.23*y(1)-0.0133*y(1)*y(2);-
0.4*y(2)+0.0004*y(1)*y(2)];
    end
    function r=res(za)
        [t1,y1]=ode45(@EX2402,[1959 2010],[1028,za]);
        r=y1(length(y1),2)-25;
    end
fzero(@res,1);
[time,y]=ode45(@EX2402,[1959 2010],[1028 fzero(@res,1)]);

figure(16)
plot(time,y(:,2))
xlabel('Year');ylabel('The simulation number of
wolves');title('Shooting method')
grid on

figure(17)
yyaxis left
plot(time, y(:,1))
xlabel('Year');
ylabel('The simulation number of population of moose');
yyaxis right
plot(time, y(:,2))
ylabel('The simulation number of population of wolves')
grid on
title('Shooting method');
legend('Moose','Wolves');
grid on

figure(18)
plot(y(:,1),y(:,2))
xlabel('Moose');ylabel('Wolves');title('Phase plane plot')
```

Appendix

Matlab Code:

```
function MATLABcode
% data import
Year=1959:2006;
Moose=[563 610 628 639 663 707 733 765 912 1042 1268 1295 1439 1493 1435 1467
1355 1282 1143 1001 1028 910 863 872 932 1038 1115 1192 1268 1335 1397 1216
1313 1590 1879 1770 2422 1163 500 699 750 850 900 1100 900 750,540,450];
Wolves=[20 22 22 23 20 26 28 26 22 22 17 18 20 23 24 31 41 44 34 40 43 50 30 14
23 24 22 20 16 12 12 15 12 12 13 17 16 22 24 14 25 29 19 17 19 29 30 30];

%% question 1 Polynomial fitting
% data processing
% year=(0:14)';
year=(1959:1973)';
moose_interpolation=Moose(1:15)';

figure(1)
plot(year,moose_interpolation)
r2=[];syx=[];

% First order
Z = [ones(size(year)) year];
a=(Z'*Z)\(Z'*moose_interpolation);
Sr = sum((moose_interpolation-Z*a).^2);
r2(1) = 1-Sr/sum((moose_interpolation-mean(moose_interpolation)).^2);
syx(1) = sqrt(Sr/(length(year)-length(a)));

% Second order
Z = [ones(size(year)) year year.^2 ];
a=(Z'*Z)\(Z'*moose_interpolation);
Sr = sum((moose_interpolation-Z*a).^2);
r2(2) = 1-Sr/sum((moose_interpolation-mean(moose_interpolation)).^2);
syx(2) = sqrt(Sr/(length(year)-length(a)));

% Third order
Z = [ones(size(year)) year year.^2 year.^3];
a=(Z'*Z)\(Z'*moose_interpolation);
Sr = sum((moose_interpolation-Z*a).^2);
r2(3) = 1-Sr/sum((moose_interpolation-mean(moose_interpolation)).^2);
syx(3) = sqrt(Sr/(length(year)-length(a)));
```

```
% Fourth order
Z = [ones(size(year)) year year.^2 year.^3 year.^4];
a=(Z'*Z)\(Z'*moose_interpolation);
Sr = sum((moose_interpolation-Z*a).^2);
r2(4) = 1-Sr/sum((moose_interpolation-mean(moose_interpolation)).^2);
syx(4)= sqrt(Sr/(length(year)-length(a)));

% Fifth order
Z = [ones(size(year)) year year.^2 year.^3 year.^4 year.^5];
a=(Z'*Z)\(Z'*moose_interpolation);
Sr = sum((moose_interpolation-Z*a).^2);
r2(5) = 1-Sr/sum((moose_interpolation-mean(moose_interpolation)).^2);
syx(5) = sqrt(Sr/(length(year)-length(a)));

% Sixth order
Z = [ones(size(year)) year year.^2 year.^3 year.^4 year.^5 year.^6];
a=(Z'*Z)\(Z'*moose_interpolation);
Sr = sum((moose_interpolation-Z*a).^2);
r2(6) = 1-Sr/sum((moose_interpolation-mean(moose_interpolation)).^2);
syx(6)= sqrt(Sr/(length(year)-length(a)));

% Seventh order
Z = [ones(size(year)) year year.^2 year.^3 year.^4 year.^5 year.^6 year.^7];
a=(Z'*Z)\(Z'*moose_interpolation);
Sr = sum((moose_interpolation-Z*a).^2);
r2(7) = 1-Sr/sum((moose_interpolation-mean(moose_interpolation)).^2);
syx(7) = sqrt(Sr/(length(year)-length(a)));

% Eighth order
Z = [ones(size(year)) year year.^2 year.^3 year.^4 year.^5 year.^6 year.^7
year.^8 ];
a=(Z'*Z)\(Z'*moose_interpolation);
Sr = sum((moose_interpolation-Z*a).^2);
r2(8) = 1-Sr/sum((moose_interpolation-mean(moose_interpolation)).^2);
syx(8) = sqrt(Sr/(length(year)-length(a)));

% Fitting result
x_order=(1:8);
figure(2)
plot(x_order,r2);
xlabel('Order');ylabel('r2');title('Coefficient of determination for different
orders')
figure(3)
plot(x_order,syx);
```

```
xlabel('Order');ylabel('Syx');title('Standard errors of different orders')
% P=[ -0.2264 5.6313 -36.7984 95.6669 554.4735 ];
x=year;
% y_moose=-0.2264*x.^4+5.6313*x.^3-36.7984*x.^2+95.6669*x.^1+554.4735;

p7 = 1.256588577934436e+09;
p6 = -1.072000217493370e+06;
p5 = -85.6293750147719;
p4 = 0.205419912565625;
p3 = -3.04549292149801e-05;
p2 = 1.869201746620641e-08;
p1 = -8.19232899472328e-12;
y_moose=p1*x.^6+p2*x.^5+p3*x.^4+p4*x.^3+p5*x.^2+ p6*x.^1+p7 ;

figure(4) % Fitted curve and original curve
% plot((0:14),moose_interpolation,(0:14),y_moose);
plot(year,moose_interpolation,year,y_moose);
% plot(year,polyval(P,x))
% plot(year,y_moose)
xlabel('Year');ylabel('Number of Moose');title('Fitted curve and original
curve')
legend('Original value','Fitted value')
grid on

figure(5)
x=(1950:0.1:1973)';
% x=(-2:0.1:14);
% y_moose=-0.2264*x.^4+5.6313*x.^3-36.7984*x.^2+95.6669*x.^1+554.4735;
y_moose=p1*x.^6+p2*x.^5+p3*x.^4+p4*x.^3+p5*x.^2+ p6*x.^1+p7 ;
plot(x,y_moose);
xlabel('Year');ylabel('Number of Moose');title('Fitted curve');
grid on
%% question 2 numerical differential
% the simple way to calculate the growth ratio
r_easy = (Moose(2:length(Moose))-Moose(1:length(Moose)-
1))./(Moose(1:length(Moose)-1))*100;
r_easy = [0 r_easy];
% use numerical differentiation to calcualte the growth ratio
len = length(Moose);
moose_gradient = zeros(1,len);
moose_gradient(1) = (-3*Moose(1)+4*Moose(2)-1*Moose(3))/(Year(3)-Year(1));
moose_gradient(len) = (3*Moose(len)-4*Moose(len-1)+1*Moose(len-2))/(Year(len)-
Year(len-2));
for r = 2:len-1
```

```
    moose_gradient(r) = (Moose(r+1)-Moose(r-1))/(Year(r+1)-Year(r-1));
end

figure(6)
% plot(Year(1:length(Moose)),moose_gradient,'-k','linewidth',1,'markersize',5)
plot(Year(1:15),moose_gradient(1:15),'-k','linewidth',1,'markersize',5)
xlabel('Year');ylabel('Moose growth gradient');title('The moose growth
gradient')
grid on
r_numerical = moose_gradient./Moose*100;

figure(7)
% plot(Year(1:length(Moose)),r_easy,'-b','linewidth',1,'markersize',5)
plot(Year(1:15),r_easy(1:15),'linewidth',1,'markersize',5)
hold on
% plot(Year(1:length(Moose)),r_numerical,'-y','linewidth',1,'markersize',5)
plot(Year(1:15),r_numerical(1:15),'linewidth',1,'markersize',5)
xlabel('Year');ylabel('Moose growth rate(%)');title('The moose growth rate ')
legend('Simple method','Differentiation method')
grid on
%%  question 3 predator-prey model
figure(8)
yyaxis left
plot(Year, Moose)
xlabel('Year');
ylabel('Population of Moose');
yyaxis right
plot(Year, Wolves)
ylabel('Population of Wolves')
grid on
title('Population change of Moose and Wolves');
legend('Moose','Wolves');
%% question 4 Solution of ODEs
% fourth-order RK method
function [tp,yp] = rk4sys(dydt,tspan,y0,h,varargin)
% rk4sys: fourth-order Runge-Kutta for a system of ODEs
%   [t,y] = rk4sys(dydt,tspan,y0,h,p1,p2,...): integrates
%           a system of ODEs with fourth-order RK method
% input:
%   dydt = name of the M-file that evaluates the ODEs
%   tspan = [ti, tf]; initial and final times with output
%            generated at interval of h, or
%   = [t0 t1 ... tf]; specific times where solution output
%   y0 = initial values of dependent variables
```

```
%   h = step size
%   p1,p2,... = additional parameters used by dydt
% output:
%   tp = vector of independent variable
%   yp = vector of solution for dependent variables
if nargin<4,error('at least 4 input arguments required'), end
if any(diff(tspan)<=0),error('tspan not ascending order'), end
n = length(tspan);
ti = tspan(1);tf = tspan(n);
if n == 2
t = (ti:h:tf)'; n = length(t);
if t(n)<tf
t(n+1) = tf;
n = n+1;
end
else
t = tspan;
end
tt = ti; y(1,:) = y0;
np = 1; tp(np) = tt; yp(np,:) = y(1,:);
i=1;
while(1)
tend = t(np+1);
hh = t(np+1) - t(np);
if hh>h,hh = h;end
while(1)
if tt+hh>tend,hh = tend-tt;end
k1 = dydt(tt,y(i,:),varargin{:})';
ymid = y(i,:) + k1.*hh./2;
k2 = dydt(tt+hh/2,ymid,varargin{:})';
ymid = y(i,:) + k2*hh/2;
k3 = dydt(tt+hh/2,ymid,varargin{:})';
yend = y(i,:) + k3*hh;
k4 = dydt(tt+hh,yend,varargin{:})';
phi = (k1+2*(k2+k3)+k4)/6;
y(i+1,:) = y(i,:) + phi*hh;
tt = tt+hh;
i=i+1;
if tt>=tend,break,end
end
np = np+1; tp(np) = tt; yp(np,:) = y(i,:);
if tt>=tf,break,end
end
end
```

```
% differential equations
function yp = predprey(t,y,a,b,c,d)
yp = [a*y(1)-b*y(1)*y(2);-c*y(2)+d*y(1)*y(2)];
end
h=1;tspan=[1959 2006];y0=[563 20];
a=0.23;b=0.0133;c=0.4;d=0.0004; %Q4 data
a2=0.2550;b2=0.0107;c2=0.2251;d2=0.0002;%Q5 data
[time, num] = rk4sys(@predprey,tspan,y0,h,a,b,c,d);
[time2, num2] = rk4sys(@predprey,tspan,y0,h,a2,b2,c2,d2);
% number of Q4 and Q5
Moose_Q4=num(:,1)';Wolves_Q4=num(:,2)';
Moose_Q5= num2(:,1)';Wolves_Q5= num2(:,2)';

% figure(8)
% plot(time,num(:,1),time,num(:,2),'--')
% xlabel('Year');ylabel('The simulation number of two populations');title('RK4
time plot')
% legend('Moose','Wolves')
% grid on

figure(9)
yyaxis left
plot(time, num(:,1))
xlabel('Year');
ylabel('The simulation number of population of moose');
yyaxis right
plot(time, num(:,2))
ylabel('The simulation number of population of wolves')
grid on
title('RK4 time plot');
legend('Moose','Wolves');
grid on

figure(10)
plot(num(:,1),num(:,2))
xlabel('Moose');ylabel('Wolves');title('RK4 phase plane plot')

figure(19)
yyaxis left
plot(time2, num2(:,1))
xlabel('Year');
ylabel('The simulation number of population of moose');
yyaxis right
plot(time2, num2(:,2))
```

```
ylabel('The simulation number of population of wolves')
grid on
title('RK4 time plot');
legend('Moose','Wolves');
grid on

figure(20)
plot(num2(:,1),num2(:,2))
xlabel('Moose');ylabel('Wolves');title('RK4 phase plane plot')

% residual sum of squares
SSR1=sum((num(:,1)'-Moose).^2);
SSR2=sum((num(:,2)'-Wolves).^2);
TotalError=SSR1+SSR2;
%%  question 5 Curve Fitting
x=Year;y=Moose;y2=Wolves;
% use numerical differentiation to calcualte the growth ratio of wolves
n = length(y2);
wolves_gradient = zeros(1,n);
wolves_gradient(1) = (-3*y2(1)+4*y2(2)-1*y2(3))/(x(3)-x(1));
wolves_gradient(n) = (3*y2(n)-4*y2(n-1)+1*y2(n-2))/(x(n)-x(n-2));
for i = 2:n-1
    wolves_gradient(i) = (y2(i+1)-y2(i-1))/(x(i+1)-x(i-1));
end
% regress
function [a, r2] = linregr(x,y)
% linregr: linear regression curve fitting
%  [a, r2] = linregr(x,y): Least squares fit of straight
% line to data by solving the normal equations
% input:
%  x = independent variable
%  y = dependent variable
% output:
%  a = vector of slope, a(1), and intercept, a(2)
%  r2 = coefficient of determination
n = length(x);
if length(y)~=n, error('x and y must be same length'); end
x = x(:); y = y(:);  % convert to column vectors
sx = sum(x); sy = sum(y);
sx2 = sum(x.*x); sxy = sum(x.*y); sy2 = sum(y.*y);
a(1) = (n*sxy-sx*sy)/(n*sx2-sx^2);
a(2) = sy/n-a(1)*sx/n;
r2 = ((n*sxy-sx*sy)/sqrt(n*sx2-sx^2)/sqrt(n*sy2-sy^2))^2;
```

```
% create plot of data and best fit line
xp = linspace(min(x),max(x),2);
yp = a(1)*xp+a(2);
figure
plot(x,y,'o',xp,yp);
title('Linear regression of variables c and d');
grid on
end
[a, r2] = linregr(y2,(moose_gradient./y));
% [a1, r22] = linregr(y,(wolves_gradient./y2));
% another way to  regress
[b1,bint1,r1,rint1,stats1]=regress((moose_gradient./y)',[ones(n,1) y2']);
[b2,bint2,r2,rint2,stats2]=regress((wolves_gradient./y2)',[ones(n,1) y']);
%%  question 6
% Find the differential of moose in Q4
moose_Q4_gradient = zeros(1,length(Year));
moose_Q4_gradient(1) = (-3*Moose_Q4(1)+4*Moose_Q4(2)-1*Moose_Q4(3))/(Year(3)-
Year(1));
moose_Q4_gradient(length(Year)) = (3*Moose_Q4(length(Year))-
4*Moose_Q4(length(Year)-1)+1*Moose_Q4(length(Year)-2))/(Year(length(Year))-
Year(length(Year)-2));
for s = 2:length(Year)-1
    moose_Q4_gradient(s) = (Moose_Q4(s+1)-Moose_Q4(s-1))/(Year(s+1)-Year(s-1));
end
bxy=0.23.*Moose_Q4-moose_Q4_gradient;
% Interpolation + integration
% bxy_interp=interp1(Year,bxy,'o',spline)
spline_moose_bxy_x=(1959:0.5:2006);
spline_moose_bxy=spline(Year,bxy,spline_moose_bxy_x);
figure(21)
plot(Year,bxy,'o',spline_moose_bxy_x,spline_moose_bxy);
xlabel('Year');ylabel('The estimated number of moose');title('Spline curve of
moose in Q4');
grid on;
spline_bxy=spline(Year,bxy);
diff(fnval(fnint(spline_bxy),[1959 1989]));

%%  question 7
% Q4 moose
T_Q4=22;
% average
spline_moose_Q4a=spline(Year,Moose_Q4);
moose_average_Q4=diff(fnval(fnint(spline_moose_Q4a),[1975 1975+T_Q4]))/T_Q4 ;
% max min
```

```
spline_moose_Q4_x=(1959:0.1:2006);
spline_moose_Q4=spline(Year,Moose_Q4,spline_moose_Q4_x);
max(spline_moose_Q4);
min(spline_moose_Q4);
% Q4 wolves
% average
spline_wolves_Q4a=spline(Year,Wolves_Q4);
wolves_average_Q4=diff(fnval(fnint(spline_wolves_Q4a),[1975 1975+T_Q4]))/T_Q4 ;
% max min
spline_wolves_Q4_x=(1959:0.1:2006);
spline_wolves_Q4=spline(Year,Wolves_Q4,spline_wolves_Q4_x);

figure(11)
plot(Year,Wolves_Q4,'o',spline_wolves_Q4_x,spline_wolves_Q4)
xlabel('Year');ylabel('The estimated number of wolves');title('Spline curve of wolves in Q4');
grid on;
max(spline_wolves_Q4);
min(spline_wolves_Q4);
% Q5 moose
T_Q5=27;
% average
spline_moose_Q5a=spline(Year,Moose_Q5);
moose_average_Q5=diff(fnval(fnint(spline_moose_Q5a),[1975 1975+T_Q5]))/T_Q5 ;
% max min
spline_moose_Q5_x=(1959:0.1:2006);
spline_moose_Q5=spline(Year,Moose_Q5,spline_moose_Q5_x);

figure(12);
plot(Year,Moose_Q5,'o',spline_moose_Q5_x,spline_moose_Q5);
xlabel('Year');ylabel('The estimated number of moose');title('Spline curve of moose in Q5');
grid on;
max(spline_moose_Q5);
min(spline_moose_Q5);
% Q5wolves
% average
spline_wolves_Q5a=spline(Year,Wolves_Q5);
wolves_average_Q5=diff(fnval(fnint(spline_wolves_Q5a),[1975 1975+T_Q4]))/T_Q5 ;
% max min
spline_wolves_Q5_x=(1959:0.1:2006);
spline_wolves_Q5=spline(Year,Wolves_Q5,spline_wolves_Q5_x);

figure(13);
```

```
plot(Year,Wolves_Q5,'o',spline_wolves_Q5_x,spline_wolves_Q5)
xlabel('Year');ylabel('The estimated number of wolves');title('Spline curve of
wolves in Q5');
grid on;
max(spline_wolves_Q5);
min(spline_wolves_Q5);
% bar wolves
y_wolves_Q6=[wolves_average_Q4,wolves_average_Q5;max(spline_wolves_Q4),max(spli
ne_wolves_Q5);
             min(spline_wolves_Q4),min(spline_wolves_Q5)];
x_wolves_Q6={'Average';'max';'min'};
figure(14)
bar(y_wolves_Q6)
set(gca,'XTickLabel',x_wolves_Q6);
legend('Q4 data','Q5 data')
ylabel('The estimated number of wolves');title('The average, maximum and
minimum numbers of wolves');
% bar moose
y_moose_Q6=[moose_average_Q4,moose_average_Q5;max(spline_moose_Q4),max(spline_m
oose_Q5);
             min(spline_moose_Q4),min(spline_moose_Q5)];
x_moose_Q6={'Average';'max';'min'};

figure(15)
bar(y_moose_Q6)
set(gca,'XTickLabel',x_moose_Q6);
legend('Q4 data','Q5 data')
ylabel('The estimated number of moose');title('The average, maximum and minimum
numbers of moose');
%%  question 8
    function dy=EX2402(t,y)
        dy=[0.23*y(1)-0.0133*y(1)*y(2);-0.4*y(2)+0.0004*y(1)*y(2)];
    end
    function r=res(za)
        [t1,y1]=ode45(@EX2402,[1959 2010],[1028,za]);
        r=y1(length(y1),2)-25;
    end
fzero(@res,1);
[time,y]=ode45(@EX2402,[1959 2010],[1028 fzero(@res,1)]);

figure(16)
plot(time,y(:,2))
xlabel('Year');ylabel('The simulation number of wolves');title('Shooting
method')
```

```
grid on

% figure(17)
% plot(time,y(:,1),time,y(:,2),'--')
% xlabel('Year');ylabel('The simulation number of two
populations');title('Shooting method')
% legend('Moose','Wolves')
% grid on

figure(17)
yyaxis left
plot(time, y(:,1))
xlabel('Year');
ylabel('The simulation number of population of moose');
yyaxis right
plot(time, y(:,2))
ylabel('The simulation number of population of wolves')
grid on
title('Shooting method');
legend('Moose','Wolves');
grid on

figure(18)
plot(y(:,1),y(:,2))
xlabel('Moose');ylabel('Wolves');title('Phase plane plot')
end
```