Annex A: Sourcecode for the Program implementing Fourth Order Runga-Kutta method

### A.1 SOURCECODE OF FIGURE 1

```matlab
% func.m
% Author: Jitendra Shrestha
% 
function diffy = func(t,y)
K = 10000; % Carrying Capacity Population
T = 1000; % Minimum Threshold Population
e = 4; % Population Growth Rate
r = 0.04; % Fishing Effort Rate
if y/T <= e
    diffy = -r*y*(1-y/T)*(1-y/K) - E*y; % Equation (6.2 & 6.3)
else
diffy = r*y*(1-y/K) - E*y; % Equation (6.2 & 6.3)
end
end

% rungekutta1.m
% Fourth Order Runge-Kutta Method
% Author: Jitendra Shrestha
% 
clear all

t0 = 0; % Initial time
T = 520; % Final time = 520 weeks = approx. 10 years
h = 1; % Step size = 1 week
n = (T-t0)/h+1; % Number of Steps
y = zeros(n); % Vector Initialization
y(1) = Pop; % Population Initialization

for i = 1:n-1
    k1 = h*func(t(i),y(i)); % Runge-Kutta Method
    k2 = h*func(t(i)+0.5*h,y(i)+0.5*k1);
    k3 = h*func(t(i)+0.5*h,y(i)+0.5*k2);
    k4 = h*func(t(i)+h,y(i)+k3);
    y(i+1) = y(i) + (k1+2*k2+2*k3+k4)/6;
    t(i+1) = t(i) + h;
    fprintf('%10.2f %20.0f', t(i), y(i));
end % Inner for loop

figure (1) % Hold the figure to juxtapose various graphs
hold on
if (j==1) plot (t, y, 'b')
```

### rungekutta1.m
% Fourth Order Runge-Kutta Method
% Author: Jitendra Shrestha
% 
clear all

t0 = 0; % Initial time
T = 520; % Final time = 520 weeks = approx. 10 years
h = 1; % Step size = 1 week
n = (T-t0)/h+1; % Number of Steps
y = zeros(n); % Vector Initialization
y(1) = Pop; % Population Initialization

for i = 1:n-1
    k1 = h*func(t(i),y(i)); % Runge-Kutta Method
    k2 = h*func(t(i)+0.5*h,y(i)+0.5*k1);
    k3 = h*func(t(i)+0.5*h,y(i)+0.5*k2);
    k4 = h*func(t(i)+h,y(i)+k3);
    y(i+1) = y(i) + (k1+2*k2+2*k3+k4)/6;
    t(i+1) = t(i) + h;
    fprintf('%10.2f %20.0f', t(i), y(i));
end % Inner for loop

figure (1) % Hold the figure to juxtapose various graphs
hold on
if (j==1) plot (t, y, 'b')
elseif (j==2);plot (t,y,'c')
elseif (j==3);plot (t,y,'g')
elseif (j==4);plot (t,y,'k')
elseif (j==5);plot (t,y,'m')
elseif (j==6);plot (t,y,'k')
elseif (j==7);plot (t,y,'r')
end  %If Loop
end  %Inner for loop

%FOllowing code inserts various text on the graph
text(300,11000, 'K=10000,T=1000,r=0.04');
title('Carp Population Growth Model with Eq.(5)');
xlabel('Time in Weeks');
ylabel('Fish population density per mile');

A.2 SOURCECODE OF FIGURE 3

%func.m
%Author: Jitendra Shrestha

function diffy = func(t,y)
K = 10000;  %Carrying Capacity Population
T = 1000;   %Minimum Threshold Population
e = K/T;    %Function Behavior Constant
r = 0.04;   %Population Growth Rate
E = 0.0737; %Fishing Effort Rate
if y/T <= e
    diffy = -r*y*(1-y/T)*(1-y/K)-E*y; %Equation (6.2 & 6.3)
else
    diffy = r*y*1*(1-y/K)-E*y; %Equation (6.2 & 6.3)
end

%rungekutta1.m
%Fourth Order Runge-Kutta Method
%Author: Jitendra Shrestha
%clc
clear all

t0 =0;   %Initial time
t1 =520; %Final time - 520 weeks = approx. 10 years
h = 1;   %Step size - 1 week
n = (t1-t0)/h+1; %Number of Steps
t = zeros(n); %Time Calibration
y = zeros(n); %Vector Initialization

t(1) = t0; %Initial time
Pop = [12000; 10000; 7000; 5500; 4000; 1000; 500]; %Initial Population Vector

for j =1:size(Pop)
y(1) = Pop(j); %Population Initialization
for i = 1:n-1
    k1 = h*func(t(i),y(i));            % Runge-Kutta Method
    k2 = h*func(t(i)+0.5*h,y(i)+0.5*k1);
    k3 = h*func(t(i)+0.5*h,y(i)+0.5*k2);
    k4 = h*func(t(i)+h,y(i)+k3);
    y(i+1) = y(i)+(k1+2*k2+2*k3+k4)/6;
    t(i+1) = t(i)+h;
    fprintf('n%10.2f%20.0f',t(i),y(i));
end %Inner for loop

figure (1) %Hold the figure to juxtapose various graphs
hold on
if (j==1);plot (t,y,'b')
elseif (j==2);plot (t,y,'c')
elseif (j==3);plot (t,y,'g')
elseif (j==4);plot (t,y,'k')
elseif (j==5);plot (t,y,'m')
elseif (j==6);plot (t,y,'k')
elseif (j==7);plot (t,y,'r')
end %If Loop
end %Inner for loop

%FOLLOWING CODE INSERTS VARIOUS TEXT ON THE GRAPH
text(250,11000,'K=10000,T=1000,r=0.04,E=0.0737');
title('Carp Population Growth Model with Eq.(6.2)');
xlabel('Time in Weeks');
ylabel('Fish population density per mile');

Sourcecode for other figures require slight variation in the input parameter in the code above.