Hung Le (John) & Anh Nguyen (Ryan)

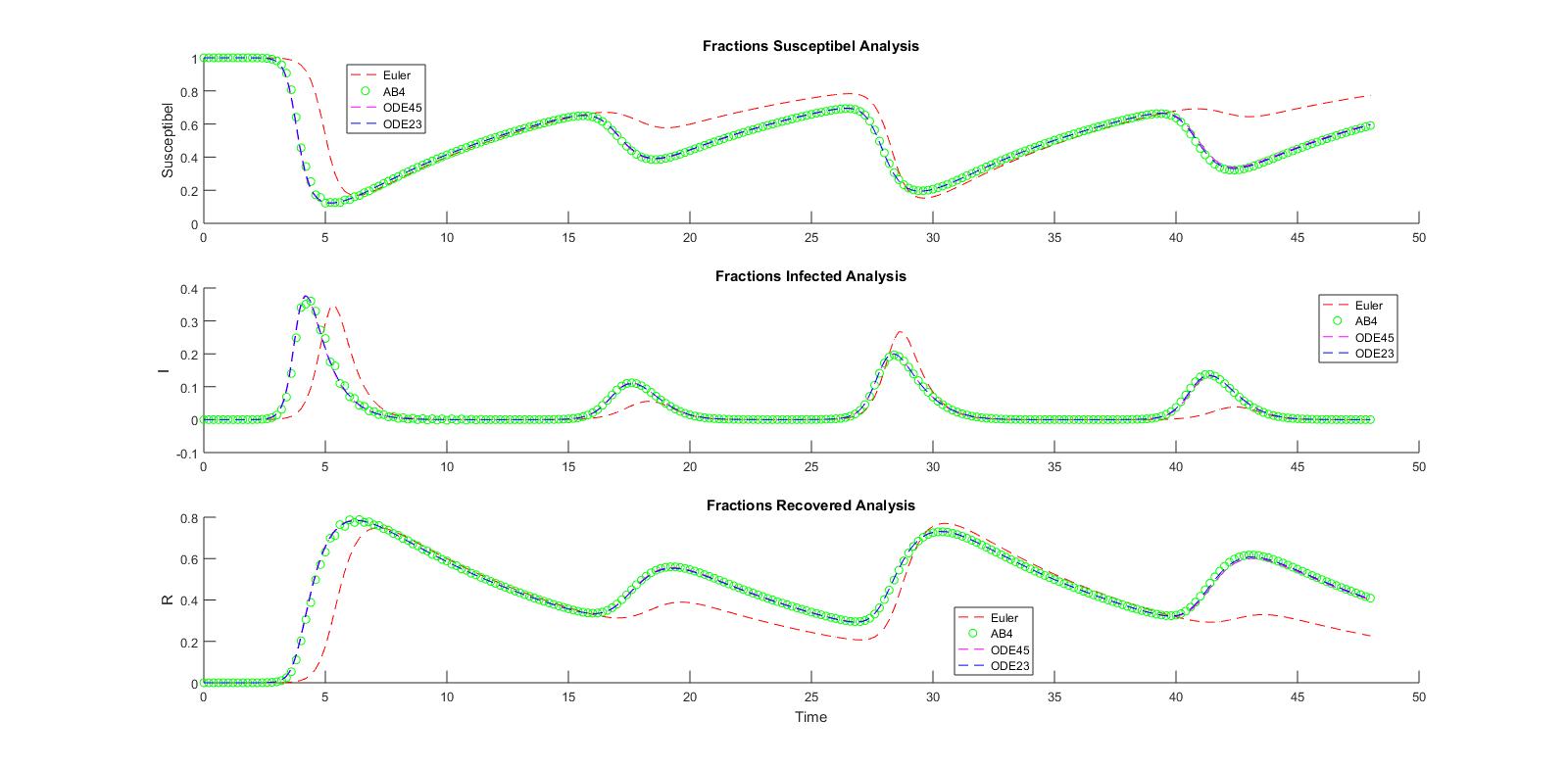
Professor Stephanie Bostwick

18 Mar, 2018

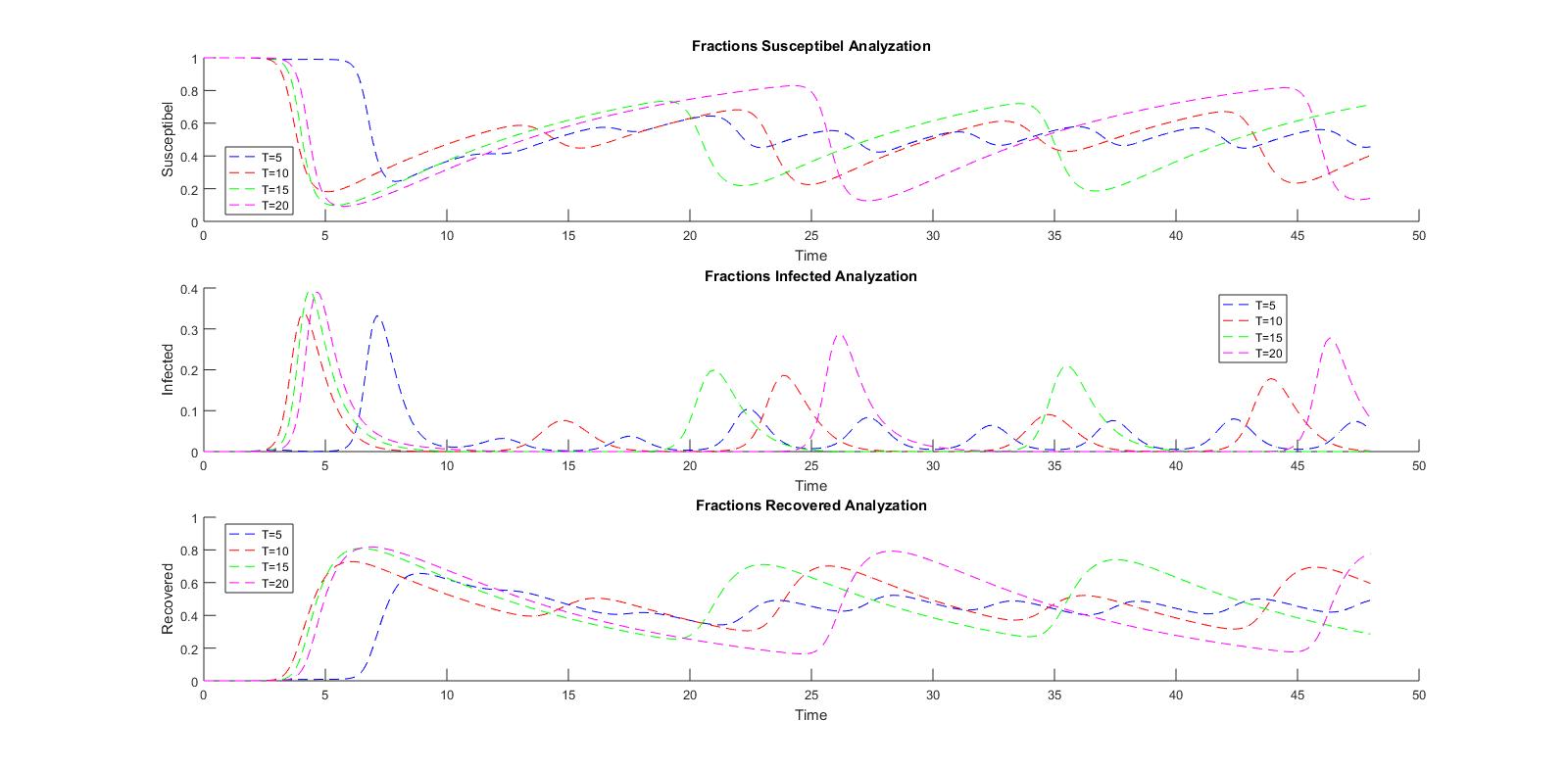
**Project 4**

**Resulting graphs:**

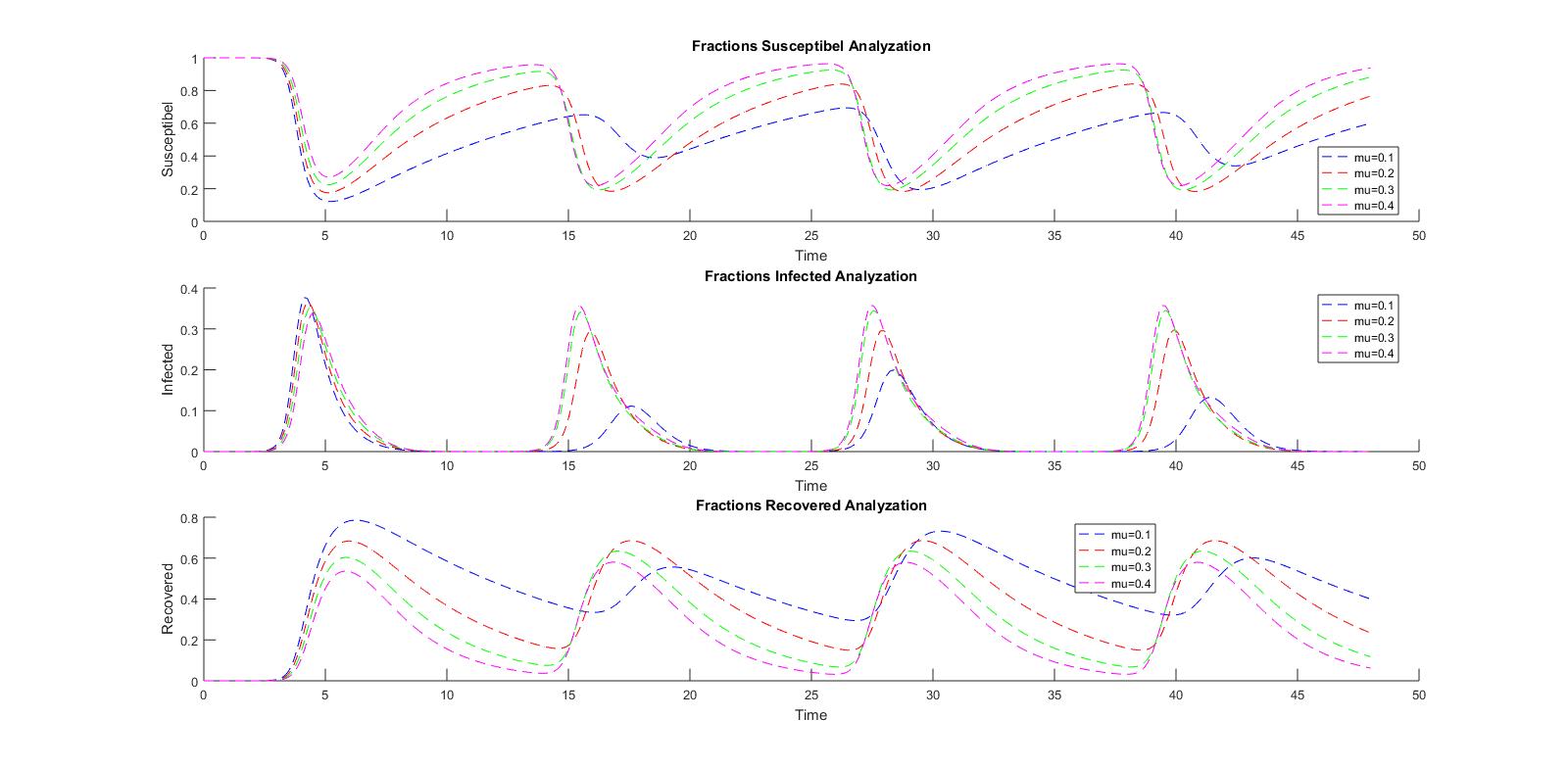
**1. Initial values given**

****

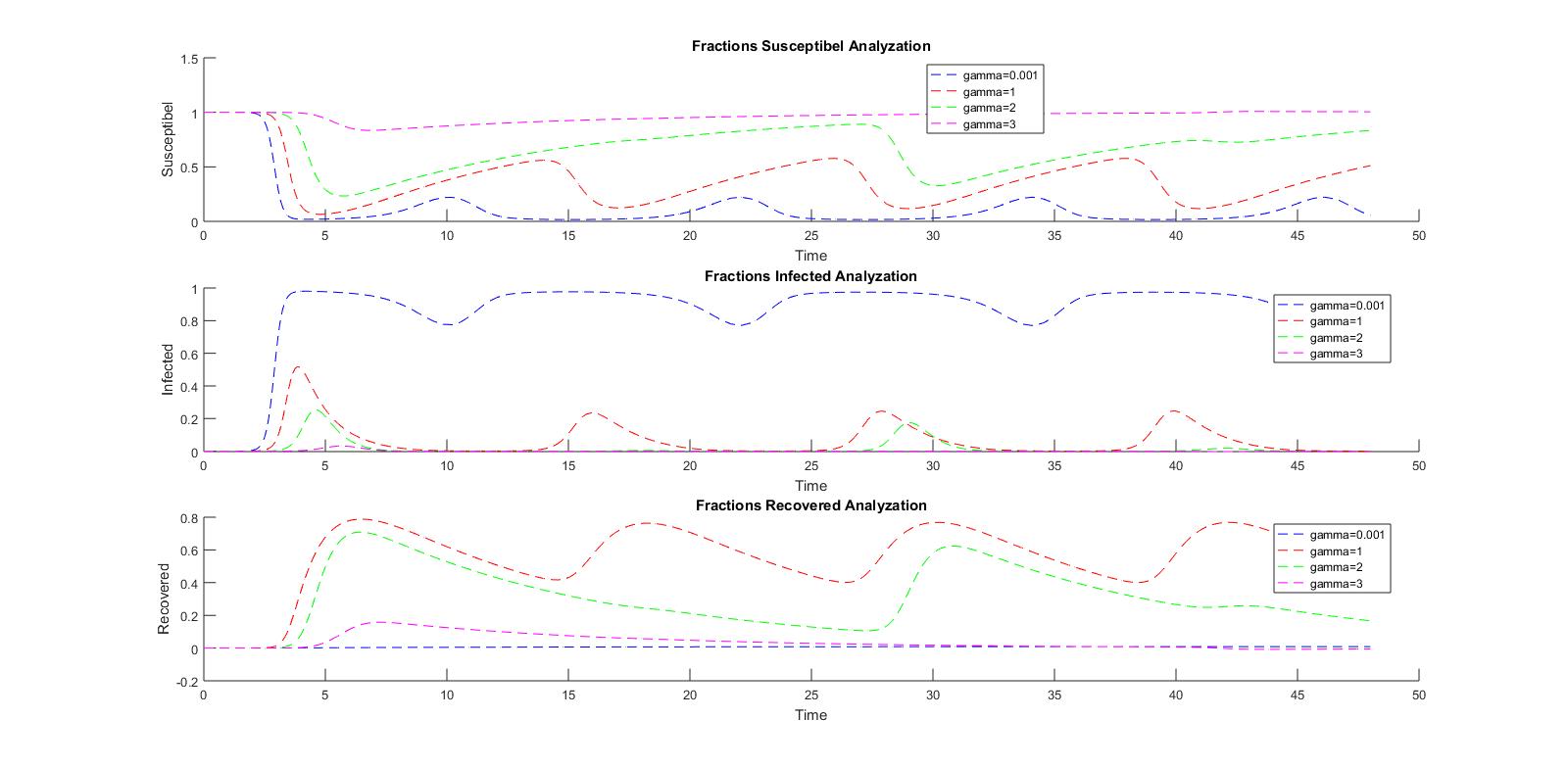
**2. Value of T changes**

****

**3. Value of mu changes**

****

**4. Value of gamma changes**

****

**Discussion:**

Firstly, we used the Euler method as it was the ODE solving method that we learned during the course and unfortunately it did not give us a reasonable set of answers. On our second try, we used Adam-Bashforth-Moulton using our self-written function, we used it because of its flexibility, and to visualize our data by choosing to solve the problem by the safest method. It ran pretty well as a result. We then tried 2 built-in functions which are ode23 and ode45 as they can provide very accurate data and they could change the interval to give better integrating results. So on the non-varied problem, we settled on the ode23 method as it provides really better results than ABM but used less time (tested by tic toc command) and resources than ode45. However, after trying the problem where some of the parameters vary, ode23 does not work as correctly due to oscillation problem which output errors.

We are confident with our results because we tried multiple methods and most of them resulted the same graphs, which means similar results are created. Because of that, we had an idea about what the results look like. However, we are not completely confident with our answers as we do not have an official way to check our results.

The most challenging aspect of this project is the process of selecting the most effective ODE solver. We then had to vary the parameters in the initial function and solve the ODE. However, because we used the built-in ode45, the t interval varied to fit each case. Therefore, the use of a for loop was not suitable and we had to manually create a variable for each case and solve each of them to maximize the efficiency of the program. However, the code is repetitive.

The most interesting aspect of this project is how the effectiveness and accuracy of each ODE methods really varied depending on the conditions and inputs of the problem. By trials and errors, we really had a chance to test which can be used and which cannot.

**Needed functions:**

**Adam-Bashforth**

function [t,y] = odeAB4sys(dydt,tspan,y0,h,varargin)

% fourth-order Runge-Kutta for a system of ODEs

% [t,y] = odeAB4sys(dydt,tspan,y0,h,p1,p2,...): integrates a

% system of ODEs with classical fourth-order Runge-Kutta

% inputs

% dydt = name of the function that evaluates the ODEs

% tspan = [ti, tf]; initial and final times with output

% generated at interval of h, or

% y0 = initial values of dependent variables (row vector)

% h = step size

% p1,p2,... = additional parameters used by dydt

% outputs

%

% t = vector of independent variable

% y = vector of solution for dependent variables

if nargin<4,error('at least 4 input arguments required'),end

t0 = tspan(1);tf = tspan(2);

t = (t0:h:tf)'; n = length(t);

% if necessary, add an additional value of t

% so that range goes from t = ti to tf

if t(n)<tf

t(n+1) = tf;

n = n+1;

end

%initial conditions

n\_eqn = length(y0);

y = ones(n,n\_eqn);

y(1,:) = y0;

% first three step with 4th Order Runge-Kutta

for i = 1:3

k1 = dydt(t(i),y(i,:),varargin{:})';

ymid2 = y(i,:) + k1\*h/2;

k2 = dydt(t(i)+h/2,ymid2,varargin{:})';

ymid3 = y(i,:) + k2\*h/2;

k3 = dydt(t(i)+h/2,ymid3,varargin{:})';

yend = y(i,:) + k3\*h;

k4 = dydt(t(i)+h,yend,varargin{:})';

phi = (k1+2\*k2+2\*k3+k4)/6;

y(i+1,:) = y(i,:) + phi\*h;

end

%4th Order Adams-Bashforth

for i=4:n-1

k1=55\*dydt(t(i),y(i,:),varargin{:})';

k2=-59\*dydt(t(i-1),y(i-1,:),varargin{:})';

k3=37\*dydt(t(i-2),y(i-2,:),varargin{:})';

k4=-9\*dydt(t(i-3),y(i-3,:),varargin{:})';

y(i+1,:)=y(i,:)+(h/24)\*(k1+k2+k3+k4);

end

**Euler**

function [t,y] = eulersys(dydt,tspan,y0,h,varargin)

% Euler method for a system of ODEs

% [t,y] = eulersys(dydt,tspan,y0,h,p1,p2,...):

%

% inputs

% dydt = name of function that defines ODE system (dydt = f(t,y)).

% Returns a column vector of derivative values.

% tspan = [t0, tf] initial and final values of independent variable

% y0 = row vector of initial values of solution

% h = step size

% p1,p2,... = additional parameters used by dydt

%

% outputs

% t = vector of independent variable

% y = matrix of solutions for y\_i(t). Each column has one y solution.

if nargin<4,error('at least 4 input arguments required'),end

t0 = tspan(1);tf = tspan(2);

t = (t0:h:tf)'; n = length(t);

% if necessary, add an additional value of t

% so that range goes from t = ti to tf

n\_eqn = length(y0);

if t(n)<tf

t(n+1) = tf;

n = n+1;

end

%initial conditions

y = ones(n,n\_eqn);

y(1,:)=y0;

for i = 1:n-1

dydt\_i = dydt(t(i),y(i,:),varargin{:});

y(i+1,:) = y(i,:) + dydt\_i'\*h;

end

**Main m-file:**

%% Project 4

clc

clear

close

%% Constants

S0=0.999999;

I0=1.0e-6;

R0=0;

T=12;

beta0=3.0;

mu=0.1;

gamma=1.5;

tspan=[0 48];

h=0.2;

es=1.0e-6;

y0=[0.999999 1.0e-6 0];

dydt= @(t,y,T,gamma,mu)[mu\*(1-y(1))-beta0\*(1+sin(2\*pi\*t/T))\*y(2)\*y(1)

beta0\*(1+sin(2\*pi\*t/T))\*y(1)\*y(2)-(gamma+mu)\*y(2)

gamma\*y(2)-mu\*y(3)];

%% Euler Method

[t,y\_euler] = eulersys(dydt,tspan,y0,h,T,gamma,mu);

S\_euler=y\_euler(:,1);

I\_euler=y\_euler(:,2);

R\_euler=y\_euler(:,3);

%% Adams-Bashforth-Moulton

tic

[t,y\_AB4] = odeAB4sys(dydt,tspan,y0,h,T,gamma,mu);

S\_AB4=y\_AB4(:,1);

I\_AB4=y\_AB4(:,2);

R\_AB4=y\_AB4(:,3);

toc

%% ODE45

tic

[t\_ODE45,y\_ODE45]=ode45(dydt,tspan,y0,[],T,gamma,mu);

S\_ODE45=y\_ODE45(:,1);

I\_ODE45=y\_ODE45(:,2);

R\_ODE45=y\_ODE45(:,3);

toc

%% ODE23

tic

[t\_ODE23,y\_ODE23]=ode23(dydt,tspan,y0,[],T,gamma,mu);

S\_ODE23=y\_ODE23(:,1);

I\_ODE23=y\_ODE23(:,2);

R\_ODE23=y\_ODE23(:,3);

toc

%% Vary in T

T\_vary=5:5:20;

[t\_Tvary1,y\_Tvary1]=ode45(dydt,tspan,y0,h,T\_vary(1),gamma,mu);

S\_Tvary1=y\_Tvary1(:,1);

I\_Tvary1=y\_Tvary1(:,2);

R\_Tvary1=y\_Tvary1(:,3);

[t\_Tvary2,y\_Tvary2]=ode45(dydt,tspan,y0,h,T\_vary(2),gamma,mu);

S\_Tvary2=y\_Tvary2(:,1);

I\_Tvary2=y\_Tvary2(:,2);

R\_Tvary2=y\_Tvary2(:,3);

[t\_Tvary3,y\_Tvary3]=ode45(dydt,tspan,y0,h,T\_vary(3),gamma,mu);

S\_Tvary3=y\_Tvary3(:,1);

I\_Tvary3=y\_Tvary3(:,2);

R\_Tvary3=y\_Tvary3(:,3);

[t\_Tvary4,y\_Tvary4]=ode45(dydt,tspan,y0,h,T\_vary(4),gamma,mu);

S\_Tvary4=y\_Tvary4(:,1);

I\_Tvary4=y\_Tvary4(:,2);

R\_Tvary4=y\_Tvary4(:,3);

%% Vary in recovery rate

gamma\_vary=[0.001 1 2 3];

[t\_gammavary1,y\_gammavary1]=ode45(dydt,tspan,y0,h,T,gamma\_vary(1),mu);

S\_gammavary1=y\_gammavary1(:,1);

I\_gammavary1=y\_gammavary1(:,2);

R\_gammavary1=y\_gammavary1(:,3);

[t\_gammavary2,y\_gammavary2]=ode45(dydt,tspan,y0,h,T,gamma\_vary(2),mu);

S\_gammavary2=y\_gammavary2(:,1);

I\_gammavary2=y\_gammavary2(:,2);

R\_gammavary2=y\_gammavary2(:,3);

[t\_gammavary3,y\_gammavary3]=ode45(dydt,tspan,y0,h,T,gamma\_vary(3),mu);

S\_gammavary3=y\_gammavary3(:,1);

I\_gammavary3=y\_gammavary3(:,2);

R\_gammavary3=y\_gammavary3(:,3);

[t\_gammavary4,y\_gammavary4]=ode45(dydt,tspan,y0,h,T,gamma\_vary(4),mu);

S\_gammavary4=y\_gammavary4(:,1);

I\_gammavary4=y\_gammavary4(:,2);

R\_gammavary4=y\_gammavary4(:,3);

%% Vary in Birth death rate

mu\_vary=[0.1 0.2 0.3 0.4];

[t\_muvary1,y\_muvary1]=ode45(dydt,tspan,y0,h,T,gamma,mu\_vary(1));

S\_muvary1=y\_muvary1(:,1);

I\_muvary1=y\_muvary1(:,2);

R\_muvary1=y\_muvary1(:,3);

[t\_muvary2,y\_muvary2]=ode45(dydt,tspan,y0,h,T,gamma,mu\_vary(2));

S\_muvary2=y\_muvary2(:,1);

I\_muvary2=y\_muvary2(:,2);

R\_muvary2=y\_muvary2(:,3);

[t\_muvary3,y\_muvary3]=ode45(dydt,tspan,y0,h,T,gamma,mu\_vary(3));

S\_muvary3=y\_muvary3(:,1);

I\_muvary3=y\_muvary3(:,2);

R\_muvary3=y\_muvary3(:,3);

[t\_muvary4,y\_muvary4]=ode45(dydt,tspan,y0,h,T,gamma,mu\_vary(4));

S\_muvary4=y\_muvary4(:,1);

I\_muvary4=y\_muvary4(:,2);

R\_muvary4=y\_muvary4(:,3);

%% Plot graph

figure(1)

subplot(311)

hold on

plot(t,S\_euler,'r--');

plot(t,S\_AB4,'og');

plot(t\_ODE45,S\_ODE45,'m--');

plot(t\_ODE23,S\_ODE23,'b--');

title('Fractions Susceptibel Analysis');

ylabel('Susceptibel');

legend('Euler','AB4','ODE45','ODE23','location','best');

hold off

subplot(312)

hold on

plot(t,I\_euler,'r--');

plot(t,I\_AB4,'og');

plot(t\_ODE45,I\_ODE45,'m--');

plot(t\_ODE23,I\_ODE23,'b--');

title('Fractions Infected Analysis');

ylabel('I');

legend('Euler','AB4','ODE45','ODE23','location','best');

hold off

subplot(313)

hold on

plot(t,R\_euler,'r--');

plot(t,R\_AB4,'og');

plot(t\_ODE45,R\_ODE45,'m--');

plot(t\_ODE23,R\_ODE23,'b--');

title('Fractions Recovered Analysis');

ylabel('R');

legend('Euler','AB4','ODE45','ODE23','location','best');

xlabel('Time');

hold off

%% Plot functions that vary in T

figure(2)

subplot(311)

hold on

plot(t\_Tvary1,S\_Tvary1,'b--');

plot(t\_Tvary2,S\_Tvary2,'r--');

plot(t\_Tvary3,S\_Tvary3,'g--');

plot(t\_Tvary4,S\_Tvary4,'m--');

title('Fractions Susceptibel Analyzation');

ylabel('Susceptibel');

xlabel('Time');

legend('T=5','T=10','T=15','T=20','location','best');

hold off

subplot(312)

hold on

plot(t\_Tvary1,I\_Tvary1,'b--');

plot(t\_Tvary2,I\_Tvary2,'r--');

plot(t\_Tvary3,I\_Tvary3,'g--');

plot(t\_Tvary4,I\_Tvary4,'m--');

title('Fractions Infected Analyzation');

ylabel('Infected');

xlabel('Time');

legend('T=5','T=10','T=15','T=20','location','best');

hold off

subplot(313)

hold on

plot(t\_Tvary1,R\_Tvary1,'b--');

plot(t\_Tvary2,R\_Tvary2,'r--');

plot(t\_Tvary3,R\_Tvary3,'g--');

plot(t\_Tvary4,R\_Tvary4,'m--');

title('Fractions Recovered Analyzation');

ylabel('Recovered');

xlabel('Time');

legend('T=5','T=10','T=15','T=20','location','best');

hold off

%% Plot functions that vary in recovery rate

figure(3)

subplot(311)

hold on

plot(t\_gammavary1,S\_gammavary1,'b--');

plot(t\_gammavary2,S\_gammavary2,'r--');

plot(t\_gammavary3,S\_gammavary3,'g--');

plot(t\_gammavary4,S\_gammavary4,'m--');

title('Fractions Susceptibel Analyzation');

ylabel('Susceptibel');

xlabel('Time');

legend('gamma=0.001','gamma=1','gamma=2','gamma=3','location','best');

hold off

subplot(312)

hold on

plot(t\_gammavary1,I\_gammavary1,'b--');

plot(t\_gammavary2,I\_gammavary2,'r--');

plot(t\_gammavary3,I\_gammavary3,'g--');

plot(t\_gammavary4,I\_gammavary4,'m--');

title('Fractions Infected Analyzation');

ylabel('Infected');

xlabel('Time');

legend('gamma=0.001','gamma=1','gamma=2','gamma=3','location','best');

hold off

subplot(313)

hold on

plot(t\_gammavary1,R\_gammavary1,'b--');

plot(t\_gammavary2,R\_gammavary2,'r--');

plot(t\_gammavary3,R\_gammavary3,'g--');

plot(t\_gammavary4,R\_gammavary4,'m--');

title('Fractions Recovered Analyzation');

ylabel('Recovered');

xlabel('Time');

legend('gamma=0.001','gamma=1','gamma=2','gamma=3','location','best');

hold off

%Plot vary in birth and death rate

figure(4)

subplot(311)

hold on

plot(t\_muvary1,S\_muvary1,'b--');

plot(t\_muvary2,S\_muvary2,'r--');

plot(t\_muvary3,S\_muvary3,'g--');

plot(t\_muvary4,S\_muvary4,'m--');

title('Fractions Susceptibel Analyzation');

ylabel('Susceptibel');

xlabel('Time');

legend('mu=0.1','mu=0.2','mu=0.3','mu=0.4','location','best');

hold off

subplot(312)

hold on

plot(t\_muvary1,I\_muvary1,'b--');

plot(t\_muvary2,I\_muvary2,'r--');

plot(t\_muvary3,I\_muvary3,'g--');

plot(t\_muvary4,I\_muvary4,'m--');

title('Fractions Infected Analyzation');

ylabel('Infected');

xlabel('Time');

legend('mu=0.1','mu=0.2','mu=0.3','mu=0.4','location','best');

hold off

subplot(313)

hold on

plot(t\_muvary1,R\_muvary1,'b--');

plot(t\_muvary2,R\_muvary2,'r--');

plot(t\_muvary3,R\_muvary3,'g--');

plot(t\_muvary4,R\_muvary4,'m--');

title('Fractions Recovered Analyzation');

ylabel('Recovered');

xlabel('Time');

legend('mu=0.1','mu=0.2','mu=0.3','mu=0.4','location','best');

hold off