

---

## Table of Contents

.....	1
PART ONE: SIR MODEL CONSTANT POPULATION .....	1
PART TWO: SIR MODEL VARIABLE TOTAL POPULATION .....	2
PART THREE: SIR MODEL LOSS OF IMMUNITY .....	4

```
% Pierce Zhang, CMOR220, Fall 2023, Project 6: Infectious Disease Model
% infectious_disease.m
% Script to solve several models to simulate SIR model over time
% Last Modified: 14 October 2023
```

```
% Main driver
function infectious_disease
solvesir;
end
```

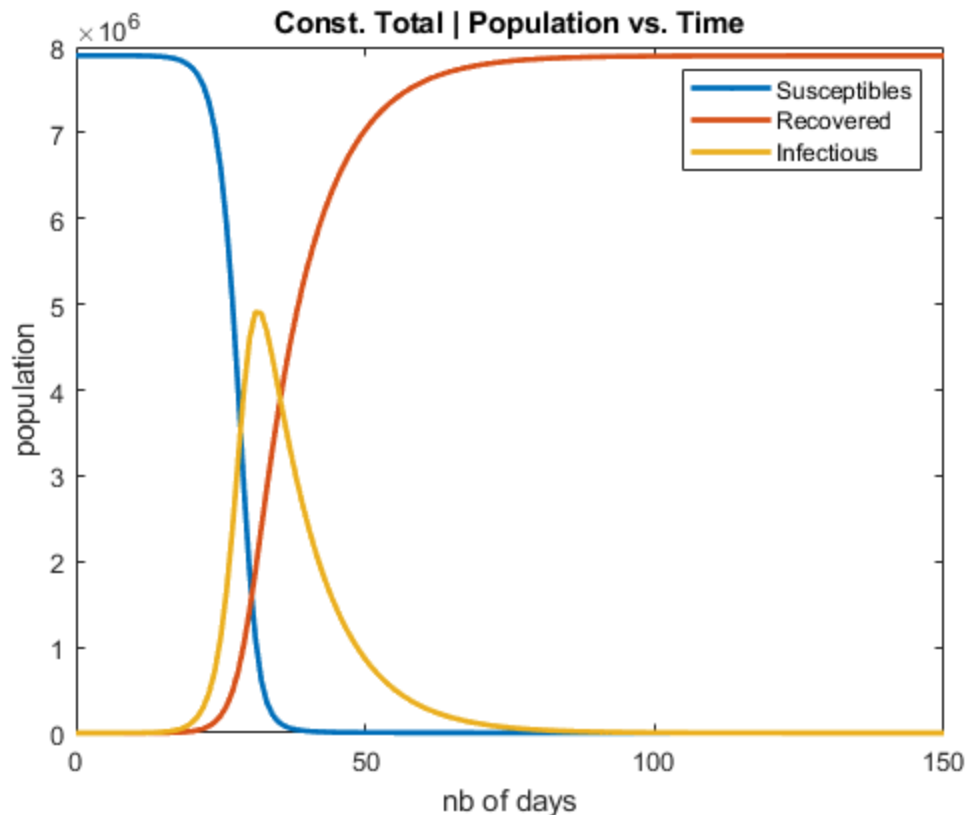
```
% Project driver
function solvesir()
```

## PART ONE: SIR MODEL CONSTANT POPULATION

Initialize parameters

```
alpha = 0.7; beta = 0.1; M = 7.9e6; R0 = 0; I0 = 10; Tfinal = 150;
[Sval, Rval, Ival] = simpleSIR(M, alpha, beta, [M-I0-R0 R0], 150);
% Plot and label
figure(); grid on
plot(0:Tfinal, [Sval; Rval; Ival], "LineWidth",2);
xlabel("nb of days"); ylabel("population");
legend("Susceptibles","Recovered","Infectious");
title("Const. Total | Population vs. Time");
```

```
% QUESTION ANSWERED (PART 1.2): The parameters were altered so that
% alpha increased from 0.5 to 0.7 and beta decreased from 0.33 to 0.1.
% This means that each day, each infected had a greater number of
% contacts, causing susceptible pop. to fall as the infection rate
% would thereby increase. Of course, the infected population would also
% increase in response. This explains the steeper slope => faster rate
% of disease spread. Additionally, in the original model, the
% infectious population recovered before every susceptible could become
% infected; but in the modified model, the infectious spread so quickly
% that not enough could recover in time, so all members of the
% population became infected. Of course, since recovery fraction was
% constant, all infected eventually recovered and no more infected were
% left, labelling everyone as recovered. This explains the different end
% behavior.
clear;
```

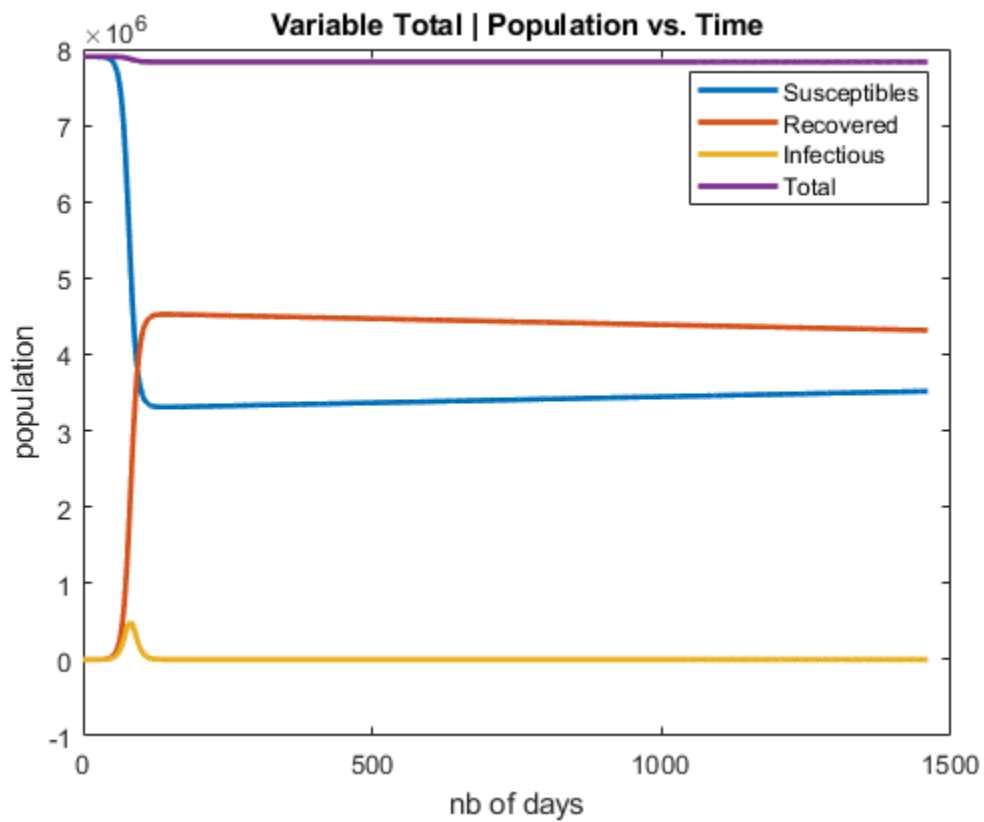
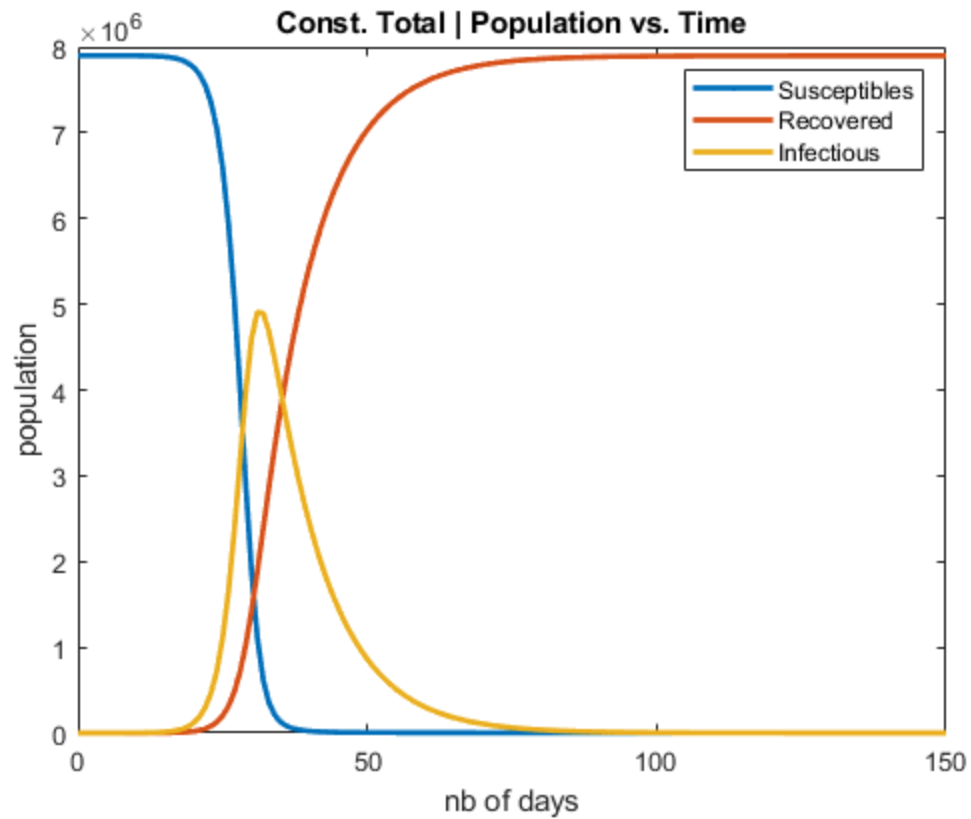


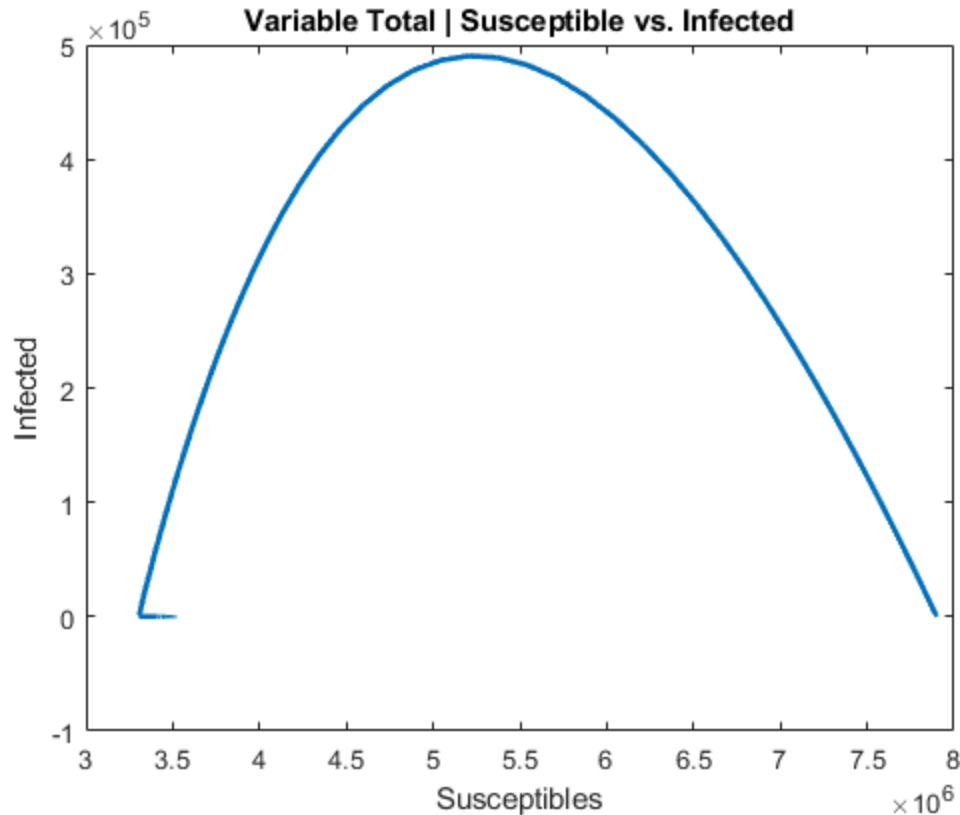
## PART TWO: SIR MODEL VARIABLE TOTAL POPULATION

Initialize parameters

```
alpha = 0.5; beta = 1/3; gamma = 0.01; mu = 1/(76*365); S0 = 7.9e6;
R0 = 0; I0 = 10; Tfinal = 4*365;
[Sval, Rval, Ival, Mval] = variableSIR(alpha, beta, gamma, mu, [S0 R0 I0],
    Tfinal);
% Plot and label
figure(); grid on
plot(0:Tfinal, [Sval; Rval; Ival; Mval], "LineWidth",2);
xlabel("nb of days"); ylabel("population");
legend("Susceptibles","Recovered","Infectious","Total");
title("Variable Total | Population vs. Time");

figure(); grid on
plot(Sval, Ival,"LineWidth",2);
xlabel("Susceptibles"); ylabel("Infected");
title("Variable Total | Susceptible vs. Infected");
clear;
```



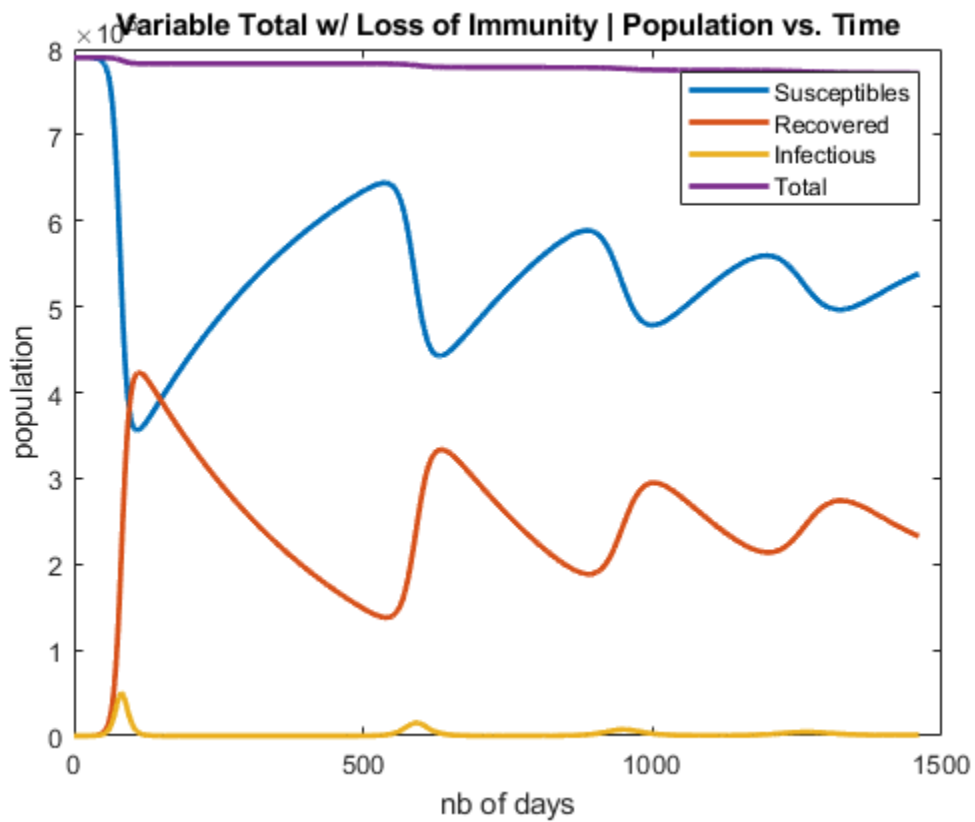
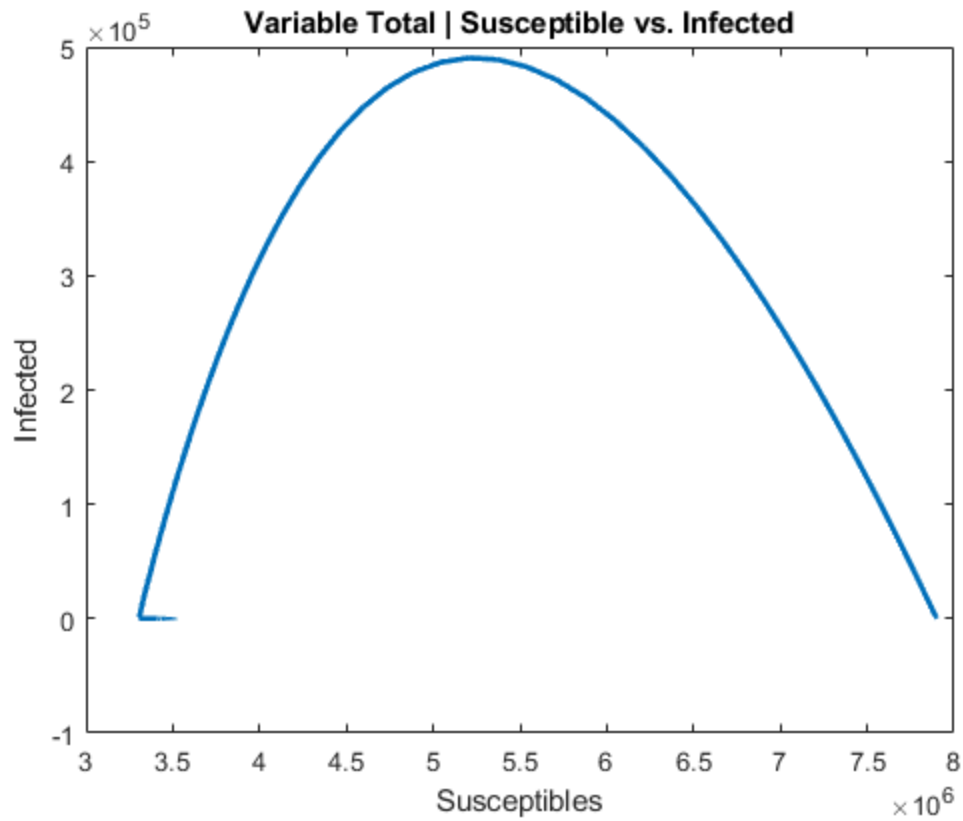


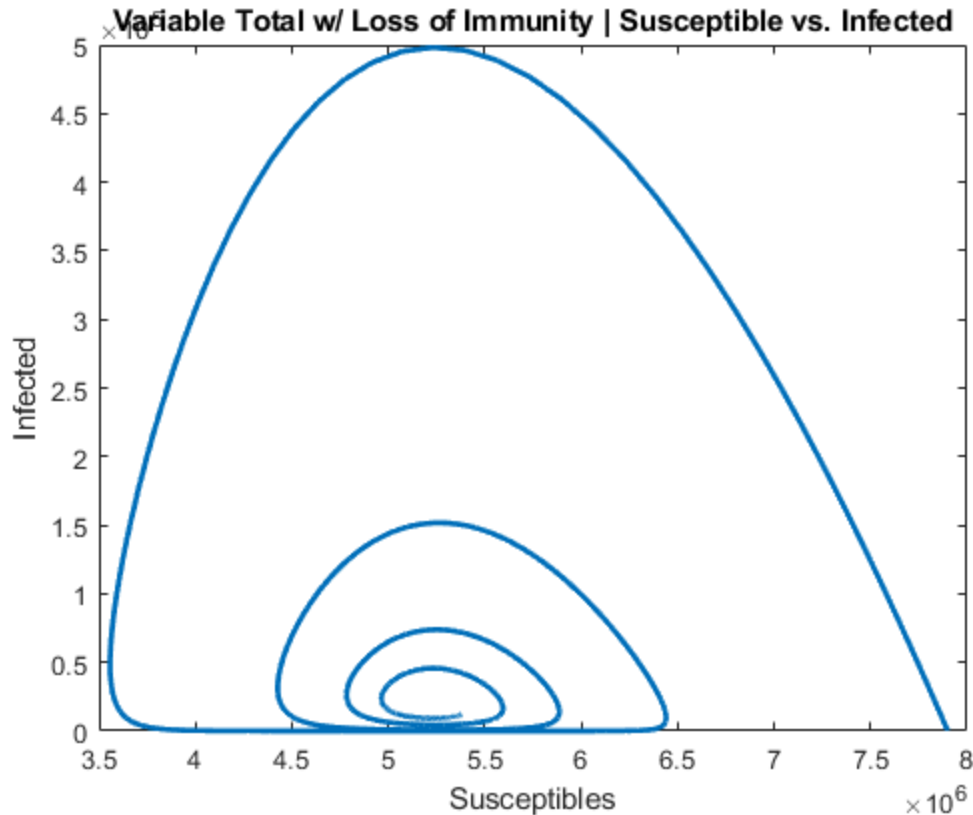
## PART THREE: SIR MODEL LOSS OF IMMUNITY

Initialize parameters

```
alpha = 0.5; beta = 1/3; gamma = 0.01; mu = 1/(76*365); omega = 1/365;
S0 = 7.9e6; R0 = 0; I0 = 10; Tfinal = 4*365;
[Sval, Rval, Ival, Mval] = variableimmSIR(alpha, beta, gamma, mu, omega, [S0
R0 I0], Tfinal);
% Plot and label
figure(); grid on
plot(0:Tfinal, [Sval; Rval; Ival; Mval], "LineWidth",2);
xlabel("nb of days"); ylabel("population");
legend("Susceptibles","Recovered","Infectious","Total");
title("Variable Total w/ Loss of Immunity | Population vs. Time");

figure(); grid on
plot(Sval, Ival,"LineWidth",2);
xlabel("Susceptibles"); ylabel("Infected");
title("Variable Total w/ Loss of Immunity | Susceptible vs. Infected");
```





end

```
% Inputs:
% - M, total population
% - alpha, number of contacts per infected
% - beta, recovery fraction
% - initialval, vector containing S0 R0
% - Tfinal, number of days to run simulation
% Output: [Sval, Rval, Ival] vector containing values of each population
% at each specified time delta (per day)
function [Sval,Rval,Ival] = simpleSIR(M,alpha,beta,initialval,Tfinal)
% initialization of the variables
Sval(1) = initialval(1);
Rval(1) = initialval(2);
Ival(1) = M-Sval(1)-Rval(1);

% loop over the time steps
for i=1:Tfinal
    Sval(i+1) = Sval(i)-((alpha/M)*Sval(i)*Ival(i));
    Rval(i+1) = Rval(i)+(beta*Ival(i));
    Ival(i+1) = M-Sval(i+1)-Rval(i+1);
end
end

% Inputs:
% - alpha, number of contacts per infected
```

---

```

% - beta, recovery fraction
% - gamma, death rate due to infection
% - mu, death rate due to unrelated causes
% - initialval, vector containing S0 R0 I0
% - Tfinal, number of days to run simulation
% Output: [Sval, Rval, Ival, Mval] vector containing values of each
% population at each specified time delta (per day)
function [Sval,Rval,Ival,Mval] =
    variableSIR(alpha,beta,gamma,mu,initialval,Tfinal)
% initialization of the variables
Sval(1) = initialval(1);
Rval(1) = initialval(2);
Ival(1) = initialval(3);
Mval(1) = Sval(1)+Rval(1)+Ival(1);

% loop over the time steps
for i=1:Tfinal
    Sval(i+1) = Sval(i)-((alpha/Mval(i))*Sval(i)*Ival(i)) + mu*Mval(i) -
    mu*Sval(i);
    Rval(i+1) = Rval(i)+(beta*Ival(i)) - mu*Rval(i);
    Ival(i+1) = Mval(i)-Sval(i+1)-Rval(i+1) - (mu + gamma)*Ival(i);
    Mval(i+1) = Sval(i) + Rval(i) + Ival(i);
end
end

% Inputs:
% - alpha, number of contacts per infected
% - beta, recovery fraction
% - gamma, death rate due to infection
% - mu, death rate due to unrelated causes
% - omega, fraction of recovered who move into susceptibles (loss of
% immunity)
% - initialval, vector containing S0 R0 I0
% - Tfinal, number of days to run simulation
% Output: [Sval, Rval, Ival, Mval] vector containing values of each
% population at each specified time delta (per day)
function [Sval,Rval,Ival,Mval] =
    variableimmSIR(alpha,beta,gamma,mu,omega,initialval,Tfinal)
% initialization of the variables
Sval(1) = initialval(1);
Rval(1) = initialval(2);
Ival(1) = initialval(3);
Mval(1) = Sval(1)+Rval(1)+Ival(1);

% loop over the time steps
for i=1:Tfinal
    Sval(i+1) = Sval(i)-((alpha/Mval(i))*Sval(i)*Ival(i)) + mu*Mval(i) -
    mu*Sval(i) + omega*Rval(i);
    Rval(i+1) = Rval(i)+(beta*Ival(i)) - mu*Rval(i) - omega*Rval(i);
    Ival(i+1) = Mval(i)-Sval(i+1)-Rval(i+1) - (mu + gamma)*Ival(i);
    Mval(i+1) = Sval(i) + Rval(i) + Ival(i);
end
end

```

---

---

*Published with MATLAB® R2023a*