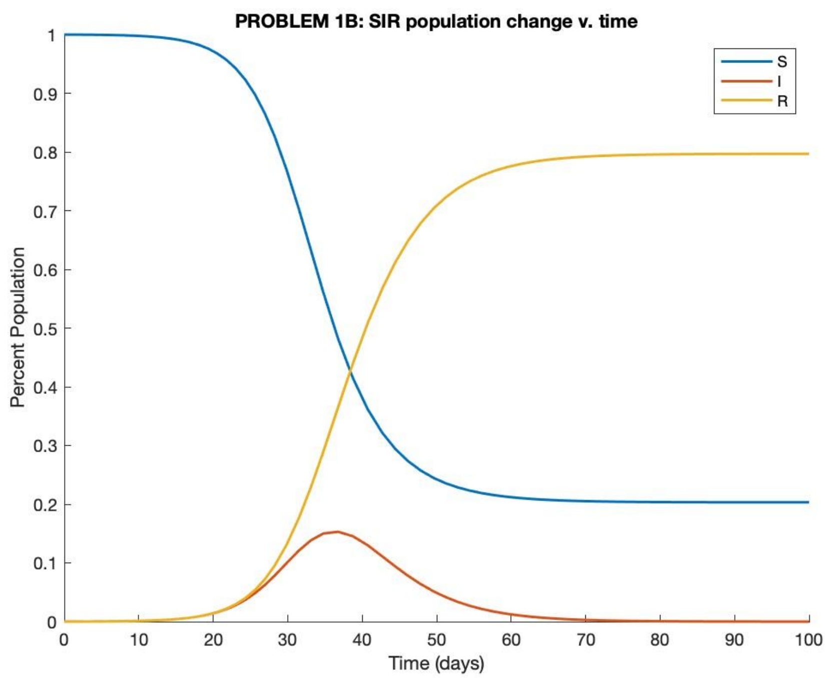
Robert Heeter
BIOE 446
20 October 2023

Lab 8: SIR Models for Infectious Disease Dynamics

ANSWERS

1. **Simulating SIR model (DEMO)**
   a. dS/dt = -0.5e-4
      dI/dt = 0.25e-4
      dR/dt = 0.25e-4

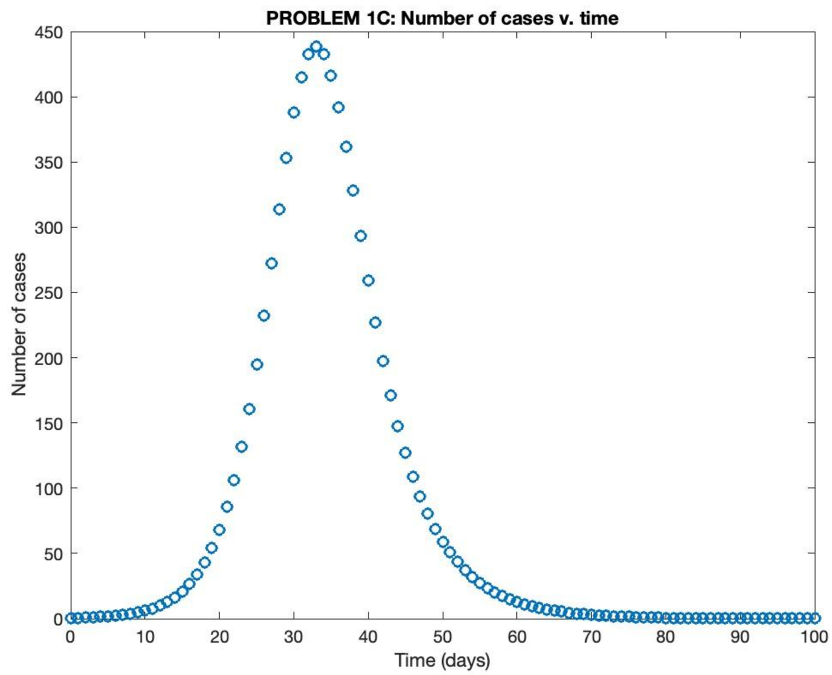
   b. R = 0.25
      R0 = 2



Figure

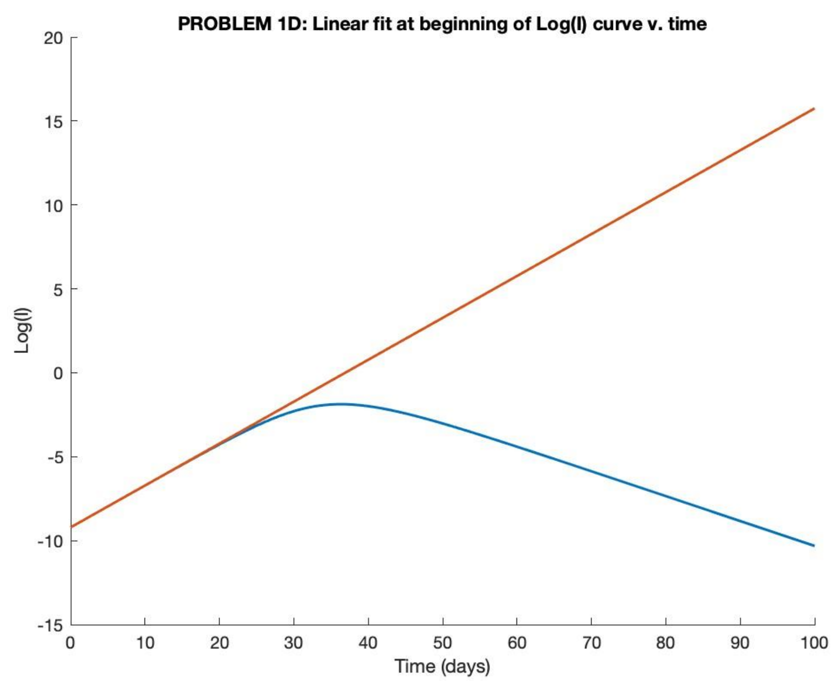
The simulation results and r suggest that an outbreak will occur with the given parameters; r is positive and the curve for infected people I(t) increases to indicate an outbreak.

   c. Figure



d. r_est = 0.2496

Figure



The estimated value of r_est = 0.2496 from the simulated fraction of infectious individuals I(t) is very close to the calculated value of r = 0.25.

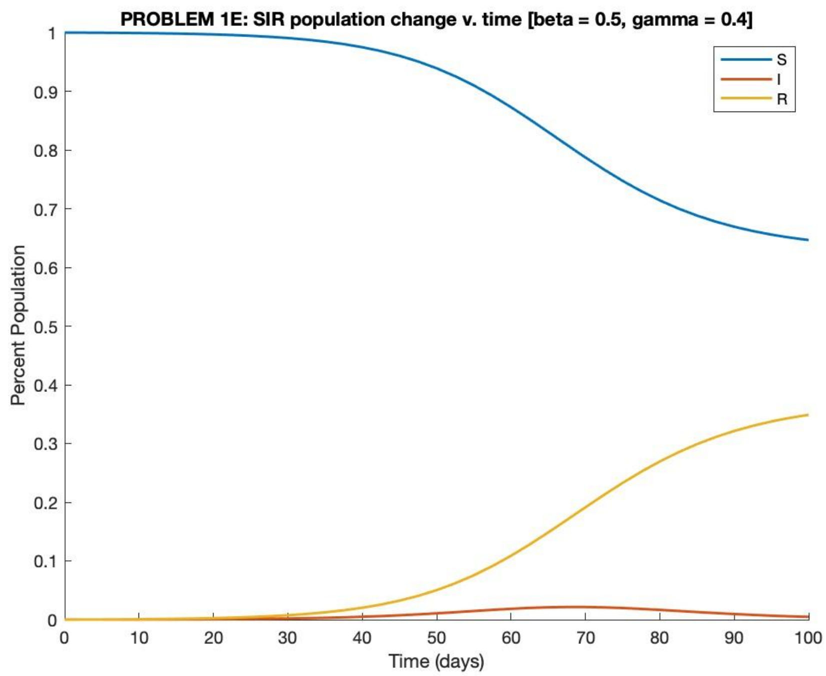
e.

i. For beta = 0.5, gamma = 0.4:

dS/dt = -0.5e-4
dI/dt = 0.1e-4
dR/dt = 0.4e-4

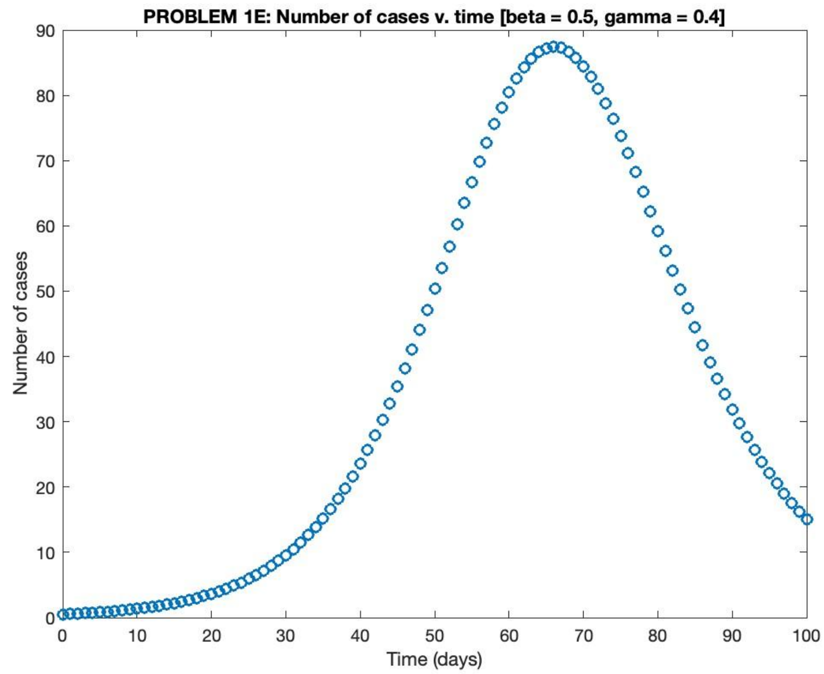
R = 0.1
R0 = 1.25

Figure



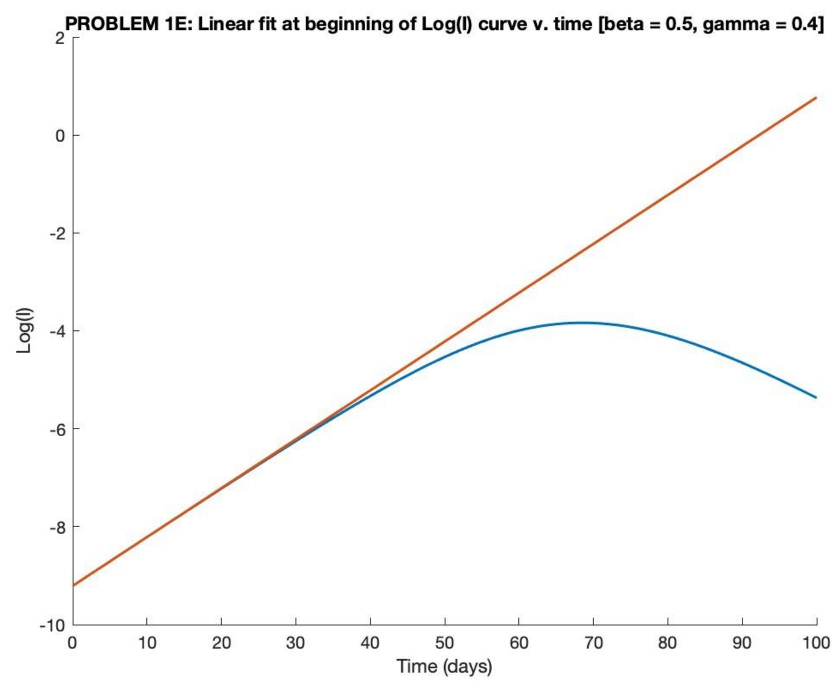
The simulation results and r suggest that a smaller outbreak will occur with the given parameters; r is positive and the curve for infected people I(t) increases to indicate an outbreak.

Figure



r_est = 0.0998

Figure



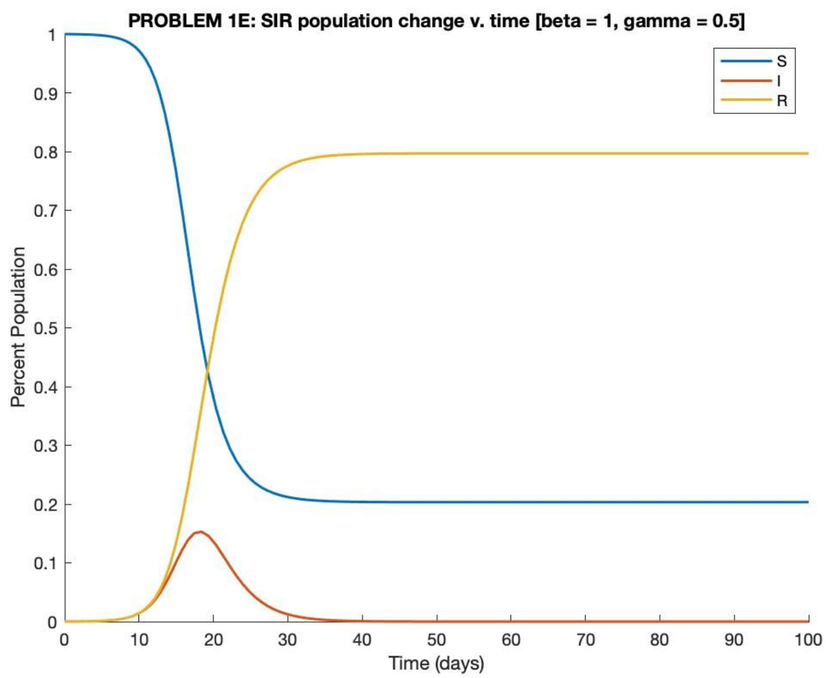
The estimated value of r_est = 0.0998 *from the simulated fraction of infectious individuals I(t) is very close to the calculated value of r = 0.1.*

ii. For beta = 1, gamma = 0.5:

dS/dt = -1e-4
dI/dt = 0.5e-4
dR/dt = 0.5e-4

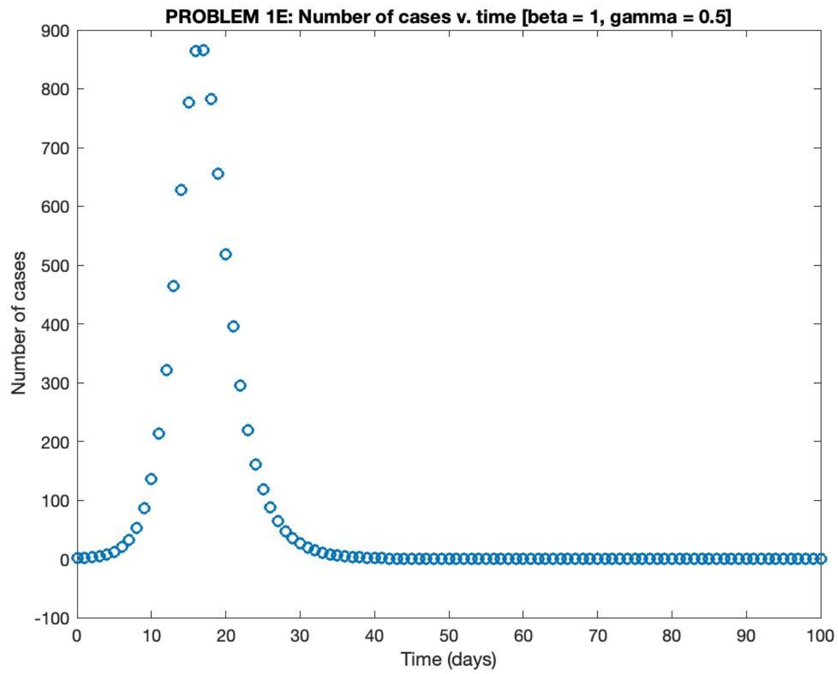
R = 0.5
R0 = 2

Figure



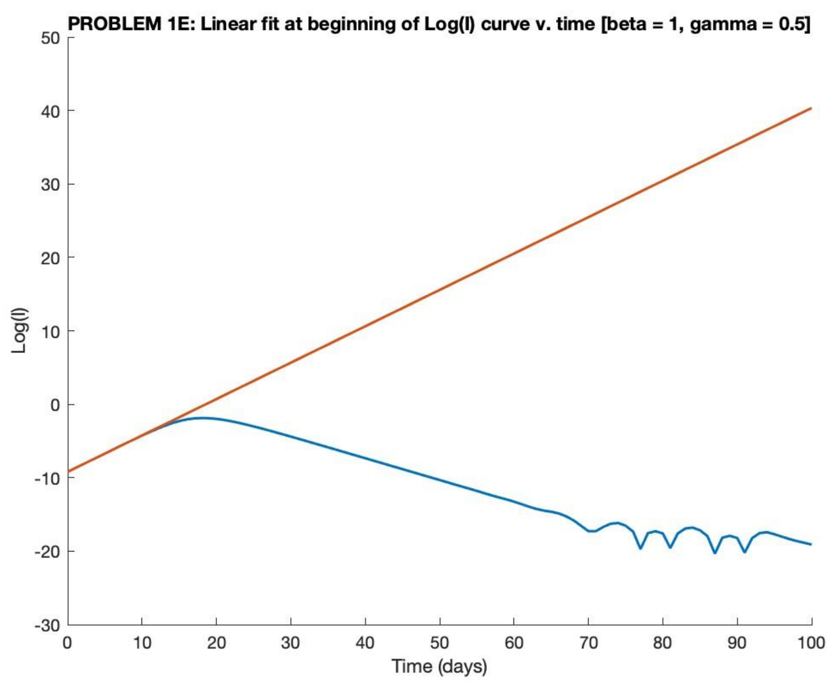
The simulation results and r suggest that an outbreak will occur with the given parameters; r is positive and the curve for infected people I(t) increases to indicate an outbreak.

Figure



r_est = 0.4954

Figure



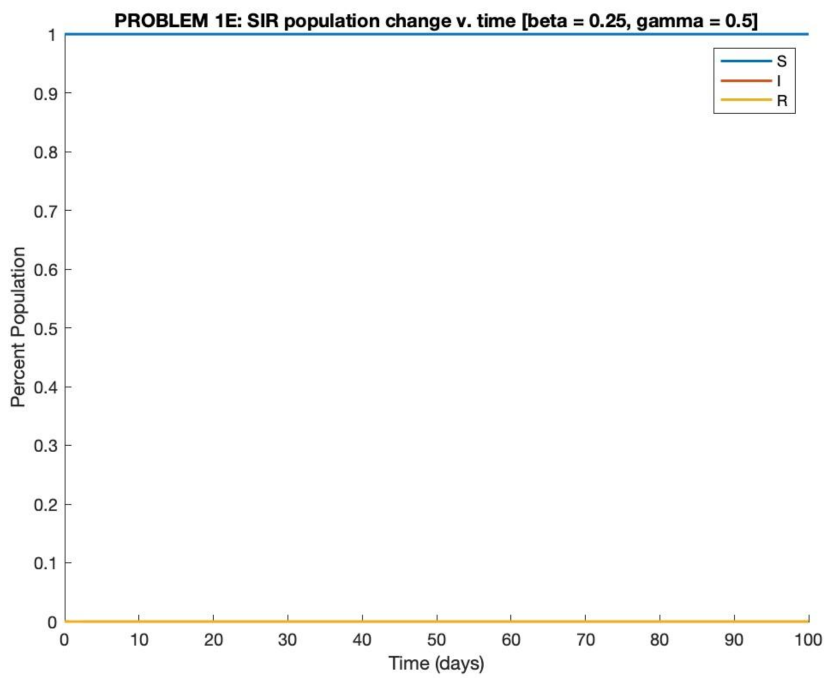
The estimated value of r_est = 0.4954 *from the simulated fraction of infectious individuals I(t) is very close to the calculated value of r = 0.5.*

iii. For beta = 0.25, gamma = 0.5:

dS/dt = -0.25e-4
dI/dt = -0.25e-4
dR/dt = 0.5e-4

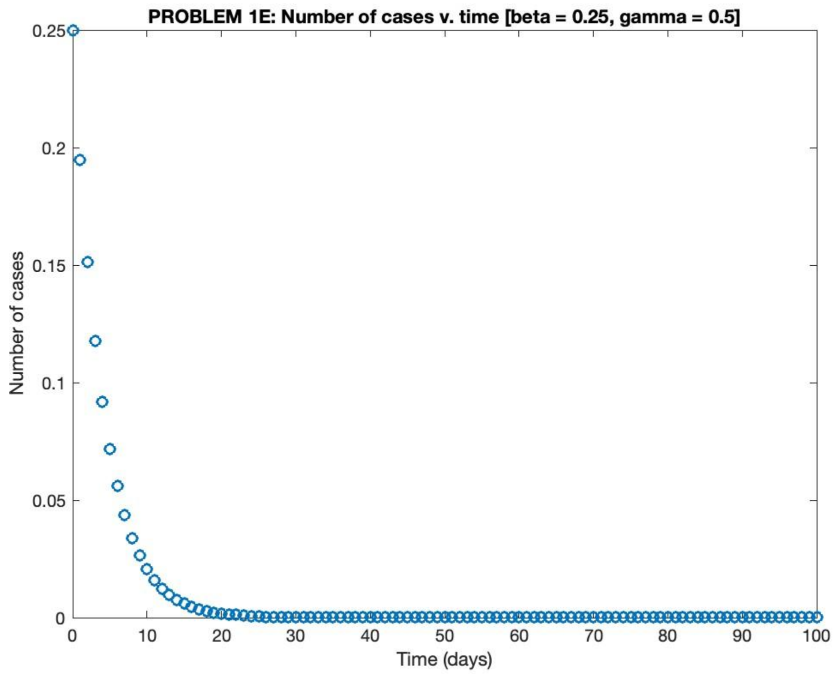
R = -0.25
R0 = 0.5

Figure



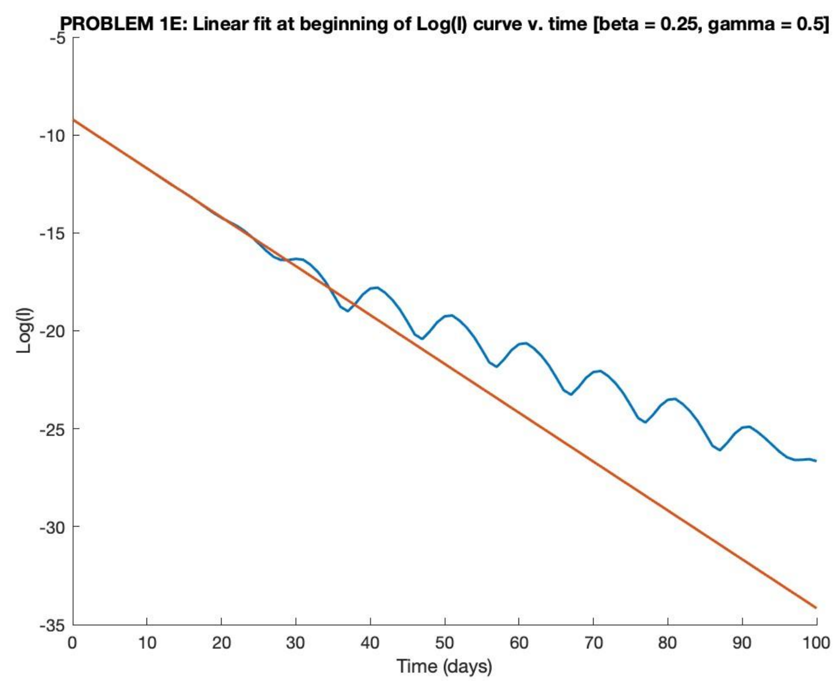
The simulation results and r suggest that an outbreak will not occur with the given parameters; r is negative and the curve for infected people I(t) does not increase.

Figure




```
r_est = -0.2495
```

Figure



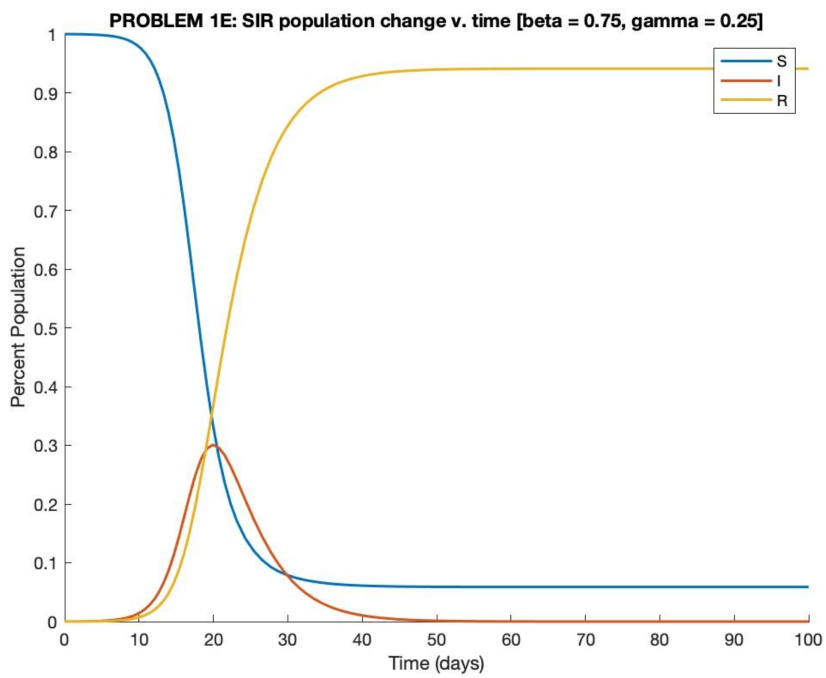
The estimated value of r_est = -0.2495 *from the simulated fraction of infectious individuals I(t) is very close to the calculated value of r = -0.25.*

iv. For beta = 0.75, gamma = 0.25:

dS/dt = -0.75e-4
dI/dt = 0.5e-4
dR/dt = 0.25e-4

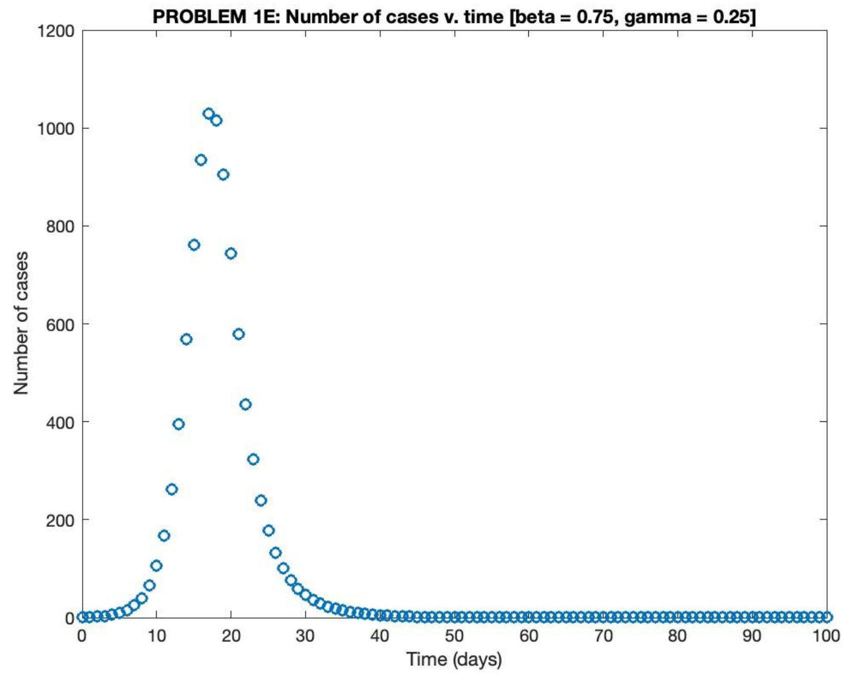
R = 0.5
R0 = 3

Figure



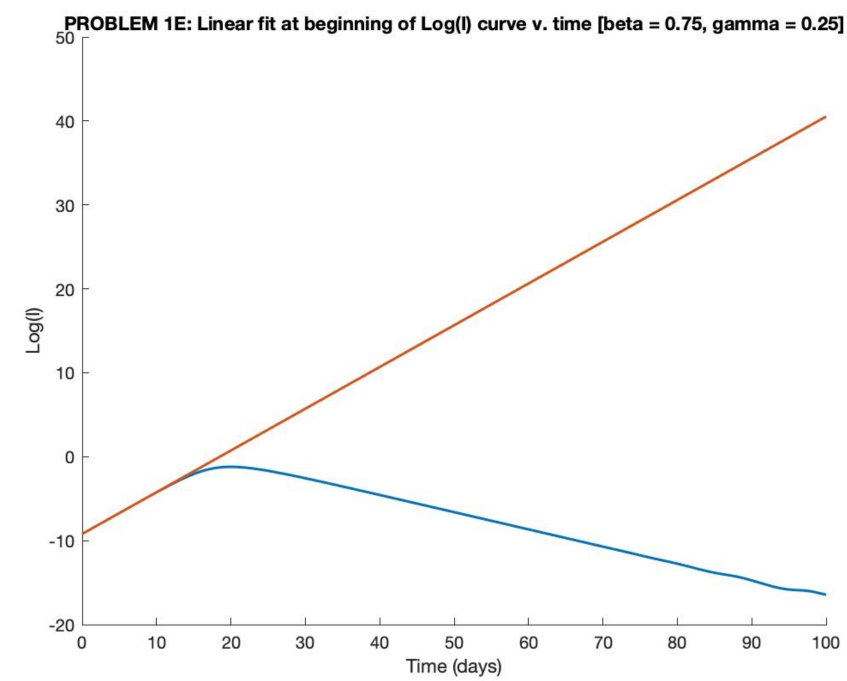
The simulation results and r suggest that an outbreak will occur with the given parameters; r is positive and the curve for infected people I(t) increases to indicate an outbreak.

Figure



r_est = 0.4974

Figure

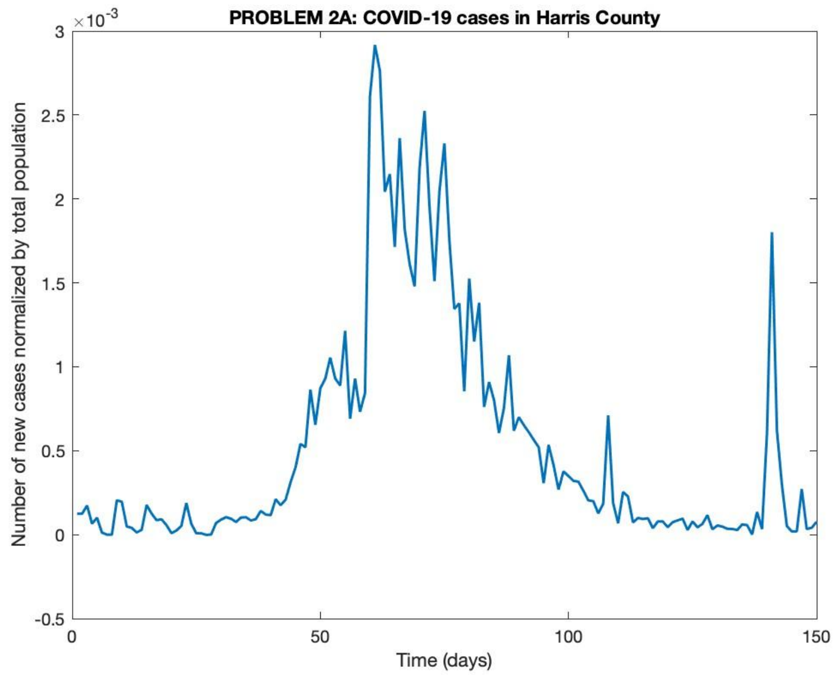


The estimated value of r_est = 0.4974 *from the simulated fraction of infectious individuals I(t) is very close to the calculated value of r = 0.5.*

The values of beta and gamma affect whether or not an outbreak will occur; as beta increases, the infection rate increases; as gamma increases, the recovery rate increases and infection rate decreases.

2. Estimating COVID-19 infections with an SIR model

a. Figure



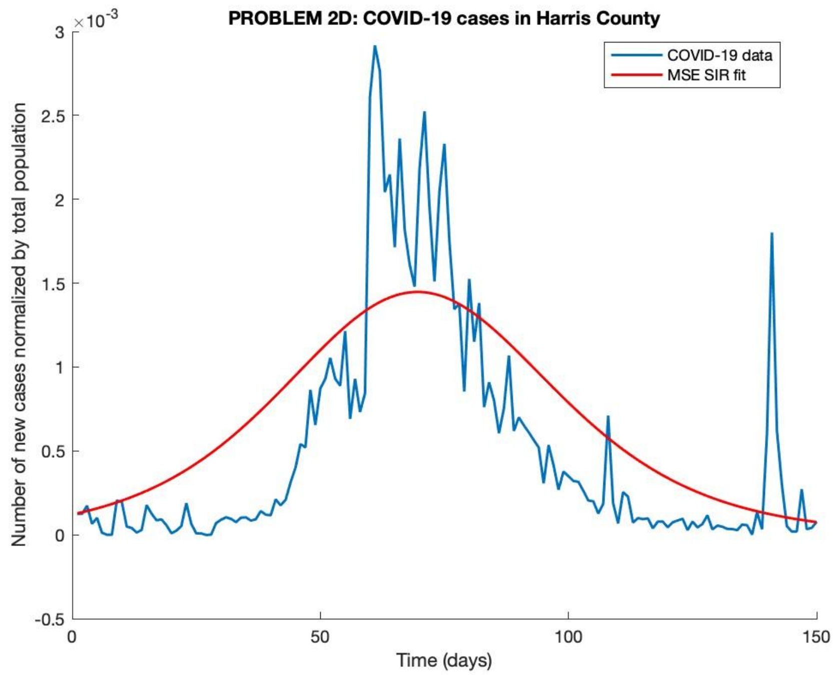
b. See MATLAB code for objective function.

c. beta_fit = 1.0191
gamma_fit = 0.9661

r_est = 0.0530
R0_est = 1.0549

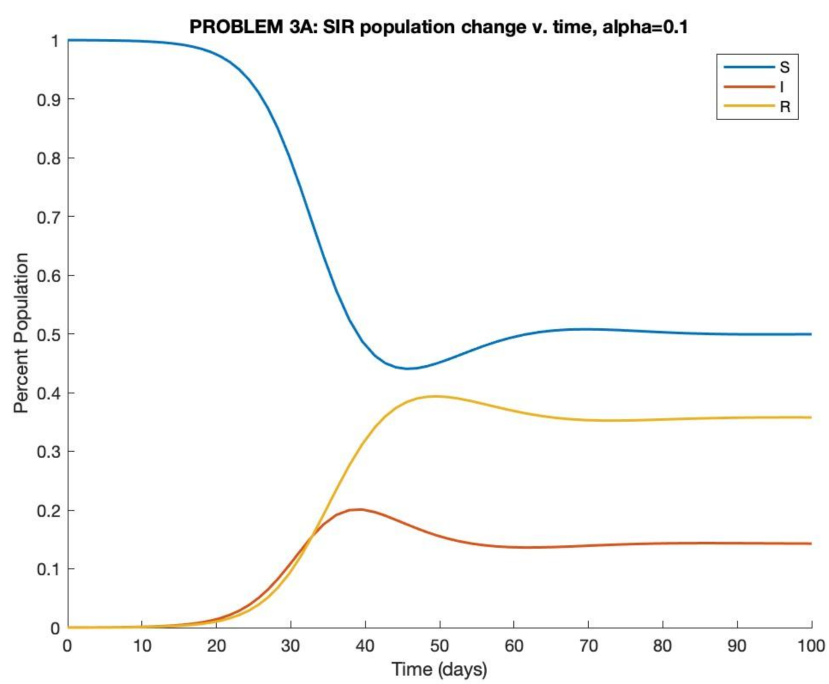
R_est = 0.0530 suggests that an outbreak will occur with the given data; r is positive, which indicates an outbreak will happen.

d. Figure



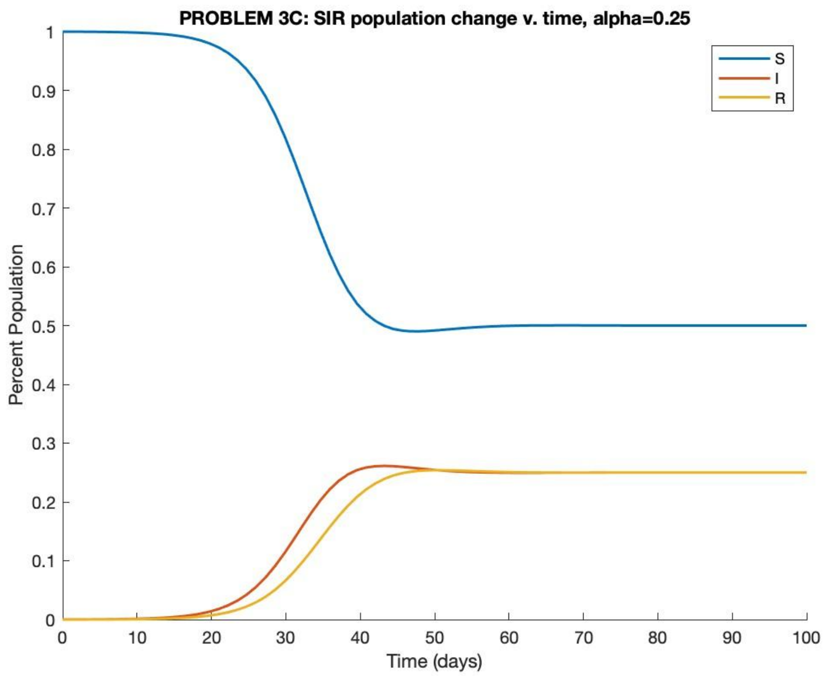
The SIR model fits the data somewhat well, but it does not account for large peaks in the data between days 50 and 100, and around day 140.

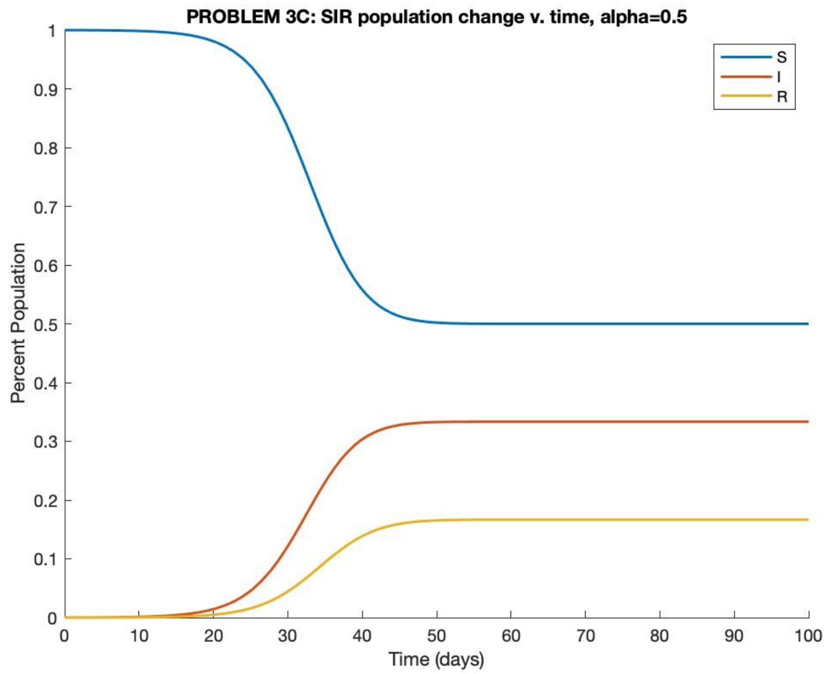
3. **Simulating the SIR model with waning immunity**
    a. Figure



b. *With waning immunity, the population reaches an equilibrium where there is a constant ratio of susceptible, infected, and recovered individuals; the number of infected individuals does not ever reach zero as in the model without waning immunity. The initial outbreak results in a peak in infected people, which stabilizes to a steady-state. The number of susceptible individuals is higher than without waning immunity as recovered people become susceptible.*

c. Figures

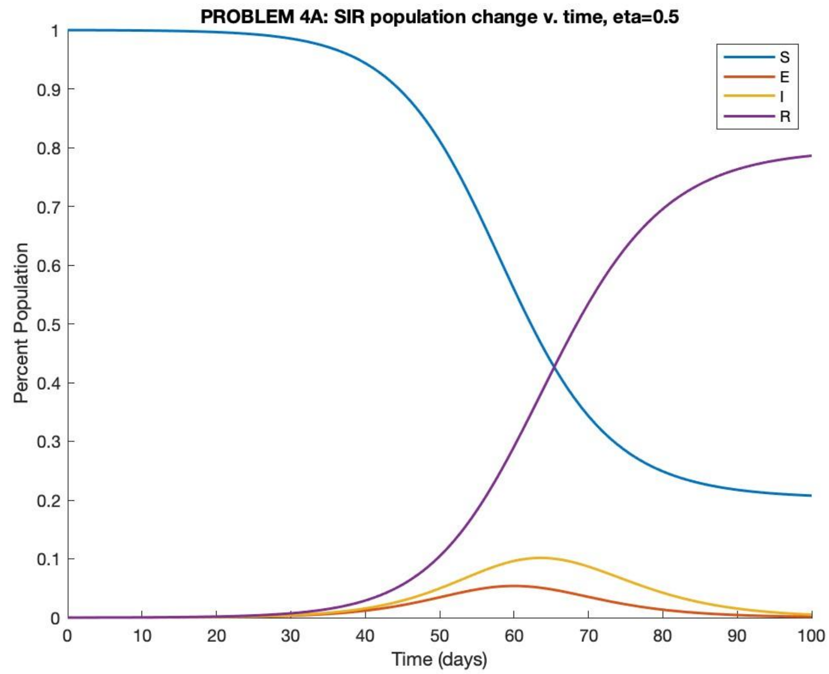




d. *With a higher rate of immunity loss, the steady-state infected proportion of the population is greater and the steady-state recovered proportion of the population is smaller. The steady-state susceptible proportion of the population remains fixed at about 0.5. As individuals lose immunity faster and become susceptible, the number of infected individuals should increase.*

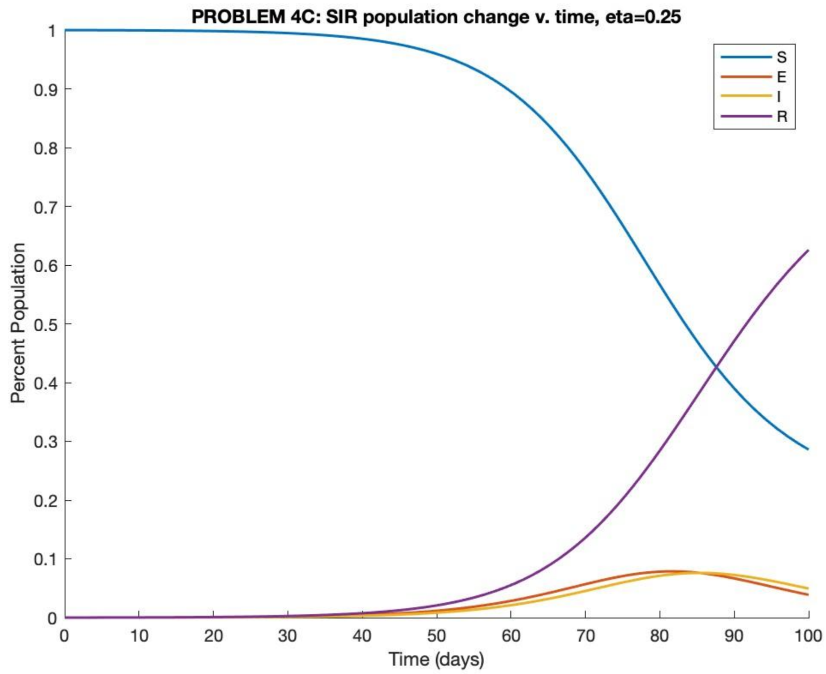
4. Adding an exposed compartment to the SIR model

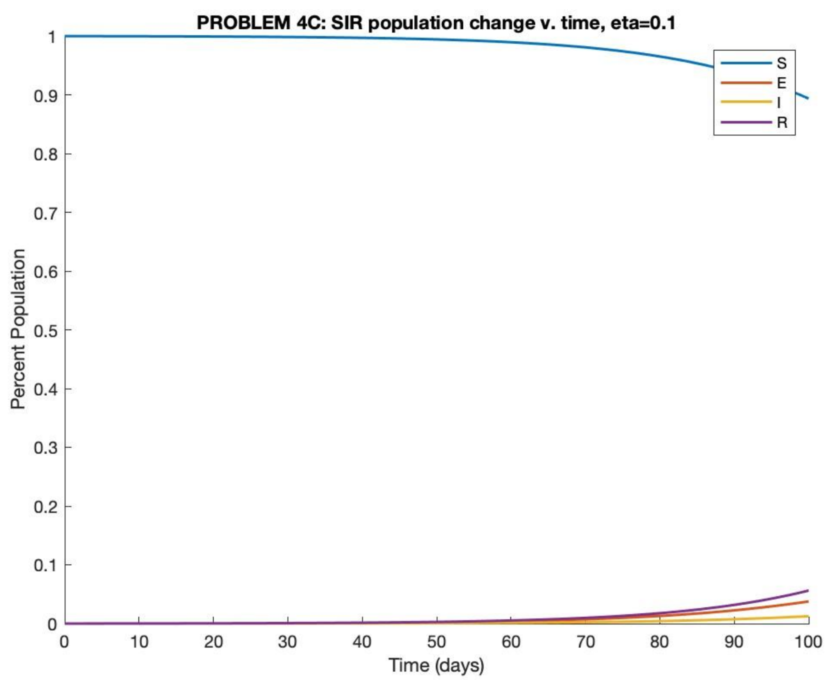
a. Figure



b. *Adding a latency period causes the peak in infections to come after the peak in exposures. With latency, the number of infected individuals is lower since individuals must first go through the exposure delay; this causes the peak in infections to come much later than in the model without a latency period. The number of susceptible and recovered people change more gradually as a result.*

c. Figures





d. *Increasing the latency period (decreasing latency rate eta) delays the peak and maximum number of infections since there is a long exposure period, during which infected individuals can recover. This effectively “flattens the curve.” The number of susceptible people decreases more gradually, and the number of recovered people increases more gradually.*

CODE

```
close all
clc

%% Problem 1: Simulating SIR model (DEMO)
disp('PROBLEM 1')
clear

%% Problem 1, Part A
N = 10000;
I0 = 1/N;
S0 = 1 - I0;
R0 = 0;
initial_states_a = [S0, I0, R0];
beta = 0.5;
gamma = 0.25;
params = [beta, gamma];

SIR = sir_model(0, initial_states_a, params)

%% Problem 1, Part B
tspan_b = [0 100];
N = 10000;
Ii = 1/N;
Si = 1 - I0;
Ri = 0;
initial_states_b = [Si, Ii, Ri];
beta = 0.5;
gamma = 0.25;
params = [beta, gamma];

[t, y] = ode45(@(t, n) sir_model(t, n, params), tspan_b, initial_states_b);

figure(1)
hold on
plot(t, y, LineWidth=1.5)
xlabel('Time (days)')
ylabel('Percent Population')
title('PROBLEM 1B: SIR population change v. time')
legend('S', 'I', 'R')

r = beta - gamma
Ro = beta/gamma

%% Problem 1, Part C
tspan_c = 0:1:100;
initial_states_c = initial_states_b;
[t, y] = ode45(@(t, n) sir_model(t, n, params), tspan_c, initial_states_c);
S = y(:, 1);
I = y(:, 2);
new_daily_cases = N*beta.*S.*I;
```

```
figure(2)
plot(t, new_daily_cases, 'o', LineWidth=1.5)
xlabel('Time (days)')
ylabel('Number of cases')
title('PROBLEM 1C: Number of cases v. time')

%% Problem 1, Part D
log_I = log(I);
p = polyfit(t(t<=10), log_I(t<=10), 1);

figure(3)
hold on
plot(t, log_I, LineWidth=1.5)
plot(t, p(2)+p(1)*t, LineWidth=1.5)
xlabel('Time (days)')
ylabel('Log(I)')
title('PROBLEM 1D: Linear fit at beginning of Log(I) curve v. time')

r_est = p(1)

%% Problem 1, Part E
betas = [0.5, 1, 0.25, 0.75];
gammas = [0.4, 0.5, 0.5, 0.25];

for i = 1:4
    beta = betas(i);
    gamma = gammas(i);
    params = [beta, gamma];

    fprintf("beta = %f; gamma = %f\n", beta, gamma)

    SIR = sir_model(0, initial_states_a, params)

    [t, y] = ode45(@(t, n) sir_model(t, n, params), tspan_b, initial_states_b);

    figure((i*3)+1)
    hold on
    plot(t, y, LineWidth=1.5)
    xlabel('Time (days)')
    ylabel('Percent Population')
    title("PROBLEM 1E: SIR population change v. time [beta = " + beta + ", gamma = "
+ gamma + "]")
    legend('S', 'I', 'R')

    r = beta - gamma
    Ro = beta/gamma

    tspan_c = 0:1:100;
    [t, y] = ode45(@(t, n) sir_model(t, n, params), tspan_c, initial_states_c);
    S = y(:, 1);
```

```
    I = y(:, 2);
    new_daily_cases = N*beta.*S.*I;

    figure((i*3)+2)
    plot(t, new_daily_cases, 'o', LineWidth=1.5)
    xlabel('Time (days)')
    ylabel('Number of cases')
    title("PROBLEM 1E: Number of cases v. time [beta = " + beta + ", gamma = " +
gamma + "]")

    log_I = log(I);
    p = polyfit(t(t<=10), log_I(t<=10), 1);

    figure((i*3)+3)
    hold on
    plot(t, log_I, LineWidth=1.5)
    plot(t, p(2)+p(1)*t, LineWidth=1.5)
    xlabel('Time (days)')
    ylabel('Log(I)')
    title("PROBLEM 1E: Linear fit at beginning of Log(I) curve v. time [beta = " +
beta + ", gamma = " + gamma + "]")

    r_est = p(1)
end


%% Problem 2: Estimating COVID-19 infections with an SIR model
disp('PROBLEM 2')
clear

%% Problem 2, Part A
covid_19_data = load("covid_19_data.mat");
data = covid_19_data.data;

figure(16)
plot(data, LineWidth=1.5)
xlabel('Time (days)')
ylabel('Number of new cases normalized by total population')
title('PROBLEM 2A: COVID-19 cases in Harris County')

%% Problem 2, Part B
% See function below.

%% Problem 2, Part C
new_daily_cases = data(1:100);

I0 = new_daily_cases(1);
R0 = 0;
S0 = 1 - I0;

initial_states = [S0, I0, R0];
```

```
tspan = 1:1:100;

p0 = [0.1, 0.1];
p_fit = fminsearch(@(p) objective_sir(new_daily_cases, p, initial_states, tspan),
p0);

beta_fit = p_fit(1)
gamma_fit = p_fit(2)

r_est = beta_fit - gamma_fit
Ro_est = beta_fit/gamma_fit

%% Problem 2, Part D
tspan = 1:1:150;
params = [beta_fit, gamma_fit];

[t, y] = ode45(@(t, n) sir_model(t, n, params), tspan, initial_states);
S = y(:, 1);
I = y(:, 2);

new_daily_cases_pred = beta_fit.*S.*I;

figure(17)
hold on
plot(data, LineWidth=1.5, DisplayName='COVID-19 data')
plot(new_daily_cases_pred, '-r', LineWidth=1.5, DisplayName='MSE SIR fit')
xlabel('Time (days)')
ylabel('Number of new cases normalized by total population')
title('PROBLEM 2D: COVID-19 cases in Harris County')
legend(location='best')


%% Problem 3: Simulating the SIR model with waning immunity
disp('PROBLEM 3')
clear

%% Problem 3, Part A
N = 10000;
I0 = 1/N;
S0 = 1 - I0;
R0 = 0;
initial_states = [S0, I0, R0];
beta = 0.5;
gamma = 0.25;
alpha = 0.1;
params = [beta, gamma, alpha];
tspan = [0 100];

[t, y] = ode45(@(t, n) sir_model_2(t, n, params), tspan, initial_states);

figure(18)
```

```
hold on
plot(t, y, LineWidth=1.5)
xlabel('Time (days)')
ylabel('Percent Population')
title('PROBLEM 3A: SIR population change v. time, alpha=0.1')
legend('S', 'I', 'R')

%% Problem 3, Part B
% No MATALB code required.

%% Problem 3, Part C
alpha = 0.25;
params = [beta, gamma, alpha];
[t, y] = ode45(@(t, n) sir_model_2(t, n, params), tspan, initial_states);

figure(19)
hold on
plot(t, y, LineWidth=1.5)
xlabel('Time (days)')
ylabel('Percent Population')
title('PROBLEM 3C: SIR population change v. time, alpha=0.25')
legend('S', 'I', 'R')

alpha = 0.5;
params = [beta, gamma, alpha];
[t, y] = ode45(@(t, n) sir_model_2(t, n, params), tspan, initial_states);

figure(20)
hold on
plot(t, y, LineWidth=1.5)
xlabel('Time (days)')
ylabel('Percent Population')
title('PROBLEM 3C: SIR population change v. time, alpha=0.5')
legend('S', 'I', 'R')

%% Problem 3, Part D
% No MATLAB code required.


%% Problem 4: Adding an exposed compartment to the SIR model
disp('PROBLEM 4')
clear

%% Problem 4, Part A
N = 10000;
I0 = 1/N;
S0 = 1 - I0;
E0 = 0;
R0 = 0;
initial_states = [S0, E0, I0, R0];
beta = 0.5;
```

```
gamma = 0.25;
eta = 0.5;
params = [beta, gamma, eta];
tspan = [0 100];

[t, y] = ode45(@(t, n) sir_model_3(t, n, params), tspan, initial_states);

figure(21)
hold on
plot(t, y, LineWidth=1.5)
xlabel('Time (days)')
ylabel('Percent Population')
title('PROBLEM 4A: SIR population change v. time, eta=0.5')
legend('S', 'E', 'I', 'R')

%% Problem 4, Part B
% No MATLAB code required.

%% Problem 4, Part C
eta = 0.25;
params = [beta, gamma, eta];
[t, y] = ode45(@(t, n) sir_model_3(t, n, params), tspan, initial_states);

figure(22)
hold on
plot(t, y, LineWidth=1.5)
xlabel('Time (days)')
ylabel('Percent Population')
title('PROBLEM 4C: SIR population change v. time, eta=0.25')
legend('S', 'E', 'I', 'R')

eta = 0.1;
params = [beta, gamma, eta];
[t, y] = ode45(@(t, n) sir_model_3(t, n, params), tspan, initial_states);

figure(23)
hold on
plot(t, y, LineWidth=1.5)
xlabel('Time (days)')
ylabel('Percent Population')
title('PROBLEM 4C: SIR population change v. time, eta=0.1')
legend('S', 'E', 'I', 'R')

%% Problem 4, Part D
% No MATLAB code required.


%% Functions

% Problem 1
function output = sir_model(t, n, p)
```

```matlab
    S = n(1);
    I = n(2);
    R = n(3);

    beta = p(1);
    gamma = p(2);

    dSdt = -beta*S*I;
    dIdt = beta*S*I - gamma*I;
    dRdt = gamma*I;

    output = [dSdt; dIdt; dRdt];
end

% Problem 2
function mse = objective_sir(new_daily_cases, p, initial_states, tspan)
    beta = p(1);
    gamma = p(2);
    params = [beta, gamma];

    [t, y] = ode45(@(t, n) sir_model(t, n, params), tspan, initial_states);
    S = y(:, 1);
    I = y(:, 2);

    new_daily_cases_pred = beta.*S.*I;
    mse = sum((new_daily_cases - new_daily_cases_pred').^2);
end

% Problem 3
function output = sir_model_2(t, n, p)
    S = n(1);
    I = n(2);
    R = n(3);

    beta = p(1);
    gamma = p(2);
    alpha = p(3);

    dSdt = -beta*S*I + alpha*R;
    dIdt = beta*S*I - gamma*I;
    dRdt = gamma*I - alpha*R;

    output = [dSdt; dIdt; dRdt];
end

% Problem 4
function output = sir_model_3(t, n, p)
    S = n(1);
    E = n(2);
    I = n(3);
    R = n(4);
```

```
    beta = p(1);
    gamma = p(2);
    eta = p(3);

    dSdt = -beta*S*I;
    dEdt = beta*S*I - eta*E;
    dIdt = eta*E - gamma*I;
    dRdt = gamma*I;

    output = [dSdt; dEdt; dIdt; dRdt];
end
```