MAE 384 Group Project

Team: Sicko Code

Contributors: Hayden Redmond, Ottavia Formicone, Brady Bauer, Ava Masche

Part 1: Modeling disease spread using SIR model

Susceptible individuals : $\frac{\mathrm{dS}(t)}{\mathrm{dt}} = -\frac{\beta}{N}S(t)I(t)$

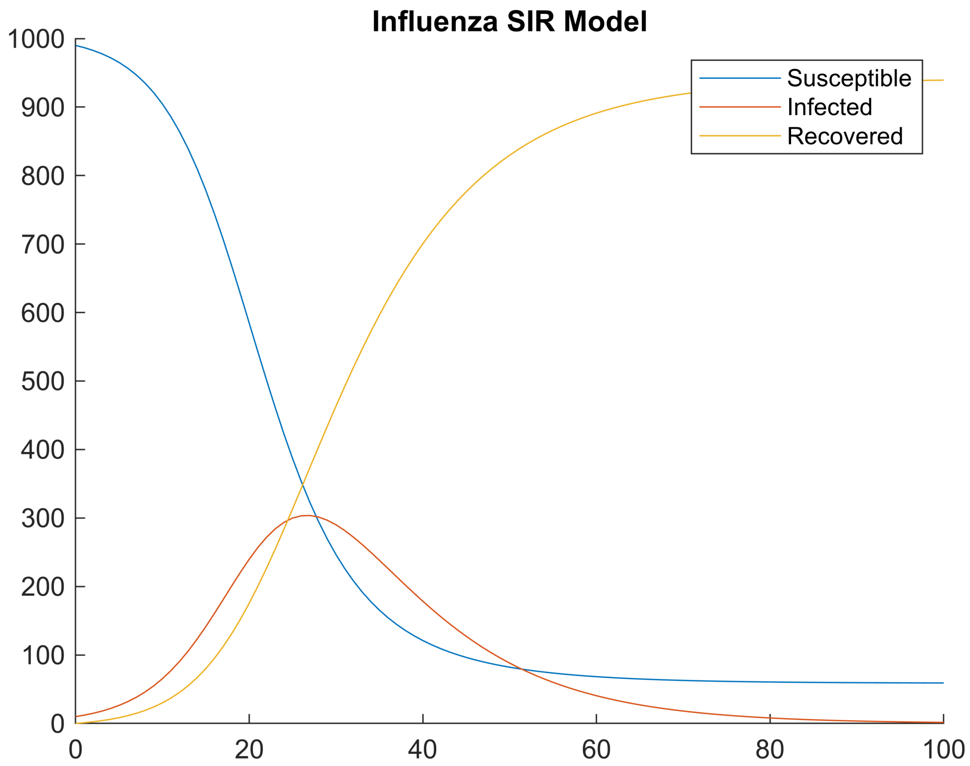
Infected individuals : $\frac{\mathrm{dI}(t)}{\mathrm{dt}} = \frac{\beta}{N}S(t)I(t) - \gamma I(t)$

Recovered individuals : $\frac{\mathrm{dR}(t)}{\mathrm{dt}} = \gamma I(t)$

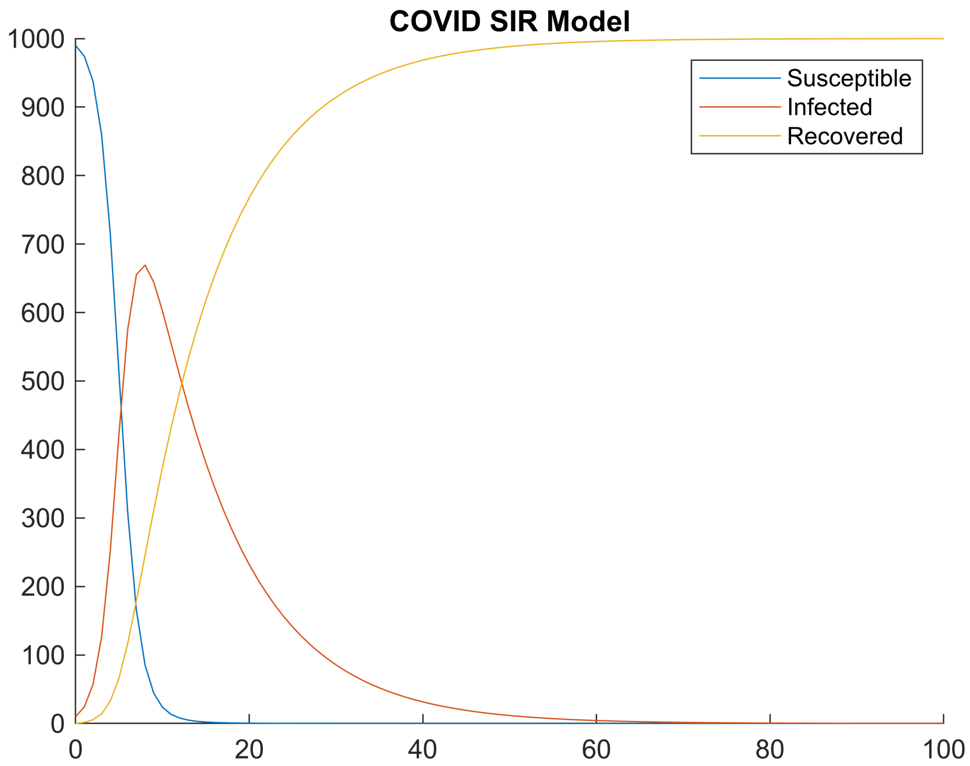
Total population : $N = S(t) + I(t) + R(t)$

β = transmission rate $\quad$ γ = recovery rate

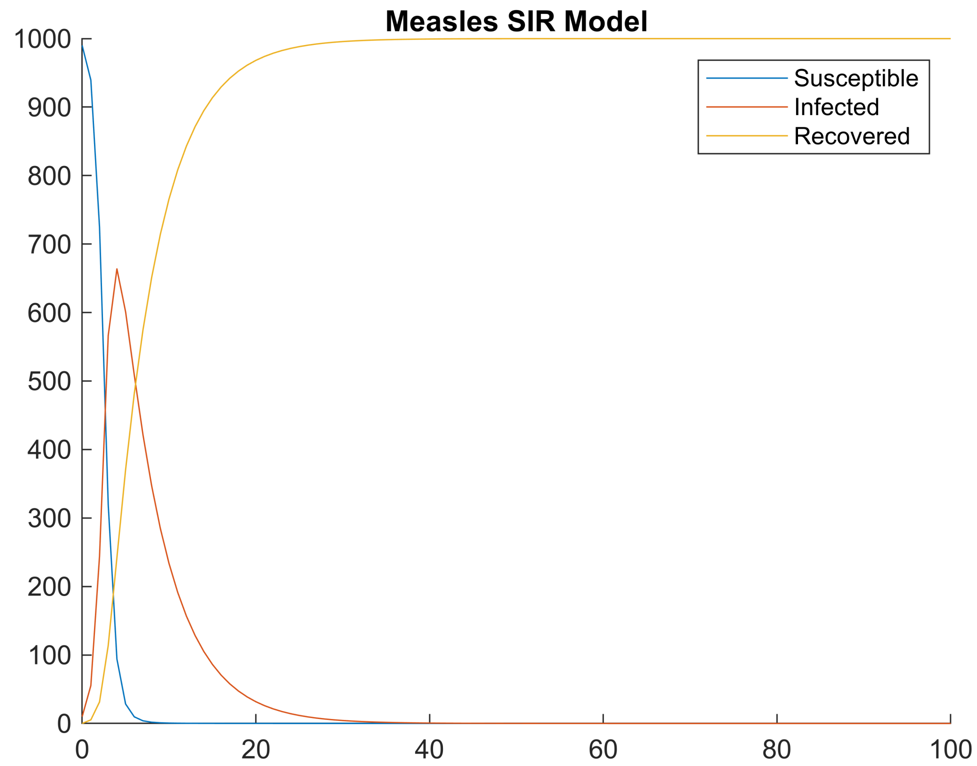
```
clear; clc; close all
Influenza_Tran_Rate = 0.3;
Influenza_Recov_Rate = 0.1;

COVID_19_Tran_Rate = 1;
COVID_19_Recov_Rate = 0.1;

Measles_Tran_Rate = 2;
Measles_Recov_Rate = 0.2;

[Flu_S,Flu_I,Flu_R] = SIR_Model(990,10,0,100,1,Influenza_Tran_Rate,Influenza_Recov_Rate);
[COVID_S,COVID_I,COVID_R] = SIR_Model(990,10,0,100,1,COVID_19_Tran_Rate,COVID_19_Recov_Rate);
[Measles_S,Measles_I,Measles_R] = SIR_Model(990,10,0,100,1,Measles_Tran_Rate,Measles_Recov_Rate);

Time_Step_1Day = 0:1:100;

figure(1)
hold on
title('Influenza SIR Model')
plot(Time_Step_1Day,Flu_S)
plot(Time_Step_1Day,Flu_I)
plot(Time_Step_1Day,Flu_R)
legend('Susceptible','Infected','Recovered')
xlim([0 100])
ylim([0 1000])
hold off
```




```
figure(2)
hold on
title('COVID SIR Model')
plot(Time_Step_1Day,COVID_S)
plot(Time_Step_1Day,COVID_I)
plot(Time_Step_1Day,COVID_R)
legend('Susceptible','Infected','Recovered')
xlim([0 100])
ylim([0 1000])
hold off
```




```
figure(3)
hold on
title('Measles SIR Model')
plot(Time_Step_1Day,Measles_S)
plot(Time_Step_1Day,Measles_I)
plot(Time_Step_1Day,Measles_R)
legend('Susceptible','Infected','Recovered')
xlim([0 100])
ylim([0 1000])
hold off
```



Discussion:

The transmission and recovery rates have a significant impact on the results of the simulation. For higher transmission rates, it's visibly clear that the susceptible population decreases faster and the infected population increases faster when compared to lower transmission rates. A similar but opposite case is true for the recovery rates. For a higher recovery rate, the infected population decreases faster and the recovered population increases faster when compared to lower recovery rates. The results found make sense intuitively because the align with my previous discussion on how transmission and recovery rates affect each type of population (susceptible, infected, and recovered).

Part 2: Interpolation

```
% Part a and b: Parameters
Influenza_Tran_Rate= 0.3;
Influenza_Recov_Rate= 0.1;
T = 100;
h_fine = 1; % Fine time step
h_coarse = 2; % Coarse time step

% Initial conditions
S0 = 990;
I0 = 10;
R0 = 0;
y0 = [S0; I0; R0];

% Define SIR model
sir_model = @(t, y) [-Influenza_Tran_Rate* y(1) * y(2); ...
                     Influenza_Tran_Rate* y(1) * y(2) - Influenza_Recov_Rate* y(2); ...
                     Influenza_Recov_Rate* y(2)];
```

```
% Solve using fine step to get reference values
[t_fine, y_fine] = ode45(sir_model, 0:h_fine:T, y0);

% Solve using coarse step
[t_coarse, y_coarse] = ode45(sir_model, 0:h_coarse:T, y0);

% Interpolation for odd days
odd_days = 1:2:T; % Odd days
S_linear = interp1(t_coarse, y_coarse(:, 1), odd_days, 'linear');
I_linear = interp1(t_coarse, y_coarse(:, 2), odd_days, 'linear');
R_linear = interp1(t_coarse, y_coarse(:, 3), odd_days, 'linear');

S_quadratic = interp1(t_coarse, y_coarse(:, 1), odd_days, 'pchip'); % Quadratic
I_quadratic = interp1(t_coarse, y_coarse(:, 2), odd_days, 'pchip');
R_quadratic = interp1(t_coarse, y_coarse(:, 3), odd_days, 'pchip');

% Reference values from fine time step
S_fine = interp1(t_fine, y_fine(:, 1), odd_days);
I_fine = interp1(t_fine, y_fine(:, 2), odd_days);
R_fine = interp1(t_fine, y_fine(:, 3), odd_days);

% Compute L2 errors for each variable
N_int = length(odd_days); % Number of interpolated points
E_L2_S_linear = sqrt(sum((S_linear - S_fine).^2) / N_int);
E_L2_I_linear = sqrt(sum((I_linear - I_fine).^2) / N_int);
E_L2_R_linear = sqrt(sum((R_linear - R_fine).^2) / N_int);

E_L2_S_quadratic = sqrt(sum((S_quadratic - S_fine).^2) / N_int);
E_L2_I_quadratic = sqrt(sum((I_quadratic - I_fine).^2) / N_int);
E_L2_R_quadratic = sqrt(sum((R_quadratic - R_fine).^2) / N_int);

% Create a table of errors
error_table = table(["S(t)"; "I(t)"; "R(t)"], ...
    [E_L2_S_linear; E_L2_I_linear; E_L2_R_linear], ...
    [E_L2_S_quadratic; E_L2_I_quadratic; E_L2_R_quadratic], ...
    'VariableNames', {'Quantity', 'Linear_Interpolation', 'Quadratic_Interpolation'});

disp(error_table);
```

Quantity	Linear_Interpolation	Quadratic_Interpolation
"S(t)"	70.004	43.752
"I(t)"	69.477	46.768
"R(t)"	1.0583	0.049976

Part 3: Least-Squares

```
[ln_I, Initial_Infected_est, Tran_Rate_est, k_est, t_data, X, theta] = Linear_Least_Squares(990,10,0,30,1,Influenza_Tran_Rate,Influenza_Recov_Rate);
type Linear_Least_Squares.m
```

```
% % MATLAB Code for Part III: Least Squares
function [ln_I, Initial_Infected_est, Tran_Rate_est, k_est, t_data, X, theta] = Linear_Least_Squares(Initial_Susceptible,Initial_Infected,Initial_Recovered,Sim_Time,Time_Step,Tran_Rate,Recov_Rate)




%% Constants and Initial Conditions
N = 1000;                  % Total population
% Initial_Susceptible = 990;                  % Initial susceptible population
% Initial_Infected = 10;                   % Initial infected population
% Initial_Recovered = 0;                    % Initial recovered population
% Tran_Rate = 0.3;           % Transmission rate
% Recov_Rate = 0.1;               % Recovery rate
% Time_Step = 1;                      % Time step (days)
% Sim_Time = 30;                     % Total time (days)
t = 0:Time_Step:Sim_Time;                 % Time vector
num_steps = length(t);

%% Runge-Kutta 4th-order Method for SIR Model (Non-linear)
S = zeros(1, num_steps);   % Susceptible population
I = zeros(1, num_steps);   % Infected population
R = zeros(1, num_steps);   % Recovered population

% Initial conditions
S(1) = Initial_Susceptible;
I(1) = Initial_Infected;
R(1) = Initial_Recovered;

% RK4 Loop
for n = 1:num_steps-1
    % Calculate derivatives
    dS1 = -Tran_Rate * S(n) * I(n) / N;
    dI1 = Tran_Rate * S(n) * I(n) / N - Recov_Rate * I(n);
    dR1 = Recov_Rate * I(n);

    dS2 = -Tran_Rate * (S(n) + 0.5 * Time_Step * dS1) * (I(n) + 0.5 * Time_Step * dI1) / N;
    dI2 = Tran_Rate * (S(n) + 0.5 * Time_Step * dS1) * (I(n) + 0.5 * Time_Step * dI1) / N - Recov_Rate * (I(n) + 0.5 * Time_Step * dI1);
    dR2 = Recov_Rate * (I(n) + 0.5 * Time_Step * dI1);

    dS3 = -Tran_Rate * (S(n) + 0.5 * Time_Step * dS2) * (I(n) + 0.5 * Time_Step * dI2) / N;
    dI3 = Tran_Rate * (S(n) + 0.5 * Time_Step * dS2) * (I(n) + 0.5 * Time_Step * dI2) / N - Recov_Rate * (I(n) + 0.5 * Time_Step * dI2);
    dR3 = Recov_Rate * (I(n) + 0.5 * Time_Step * dI2);

    dS4 = -Tran_Rate * (S(n) + Time_Step * dS3) * (I(n) + Time_Step * dI3) / N;
    dI4 = Tran_Rate * (S(n) + Time_Step * dS3) * (I(n) + Time_Step * dI3) / N - Recov_Rate * (I(n) + Time_Step * dI3);
    dR4 = Recov_Rate * (I(n) + Time_Step * dI3);

    % Update values
    S(n+1) = S(n) + Time_Step * (dS1 + 2*dS2 + 2*dS3 + dS4) / 6;
    I(n+1) = I(n) + Time_Step * (dI1 + 2*dI2 + 2*dI3 + dI4) / 6;
    R(n+1) = R(n) + Time_Step * (dR1 + 2*dR2 + 2*dR3 + dR4) / 6;
end

%% Linear Least Squares
% Transform to log scale
ln_I = log(I(2:end)); % Exclude I(0) as log(0) is undefined
t_data = t(2:end);    % Corresponding time values

% Perform least squares fitting
X = [ones(length(t_data), 1), t_data']; % Design matrix [1, t]
theta = X \ ln_I';                    % Solve linear least squares
ln_Initial_Infected_est = theta(1);                  % Intercept: ln(I0)
k_est = theta(2);                     % Slope: k
```

```
% Back-calculate parameters
Initial_Infected_est = exp(ln_Initial_Infected_est);               % Estimated I(0)
Tran_Rate_est = (N * (k_est + Recov_Rate)) / Initial_Susceptible; % Estimated Tran_Rate

%% Results
% disp('Estimated Parameters:');
% fprintf('I(0) (Estimated): %.2f\n', Initial_Infected_est);
% fprintf('k (Estimated): %.4f\n', k_est);
% fprintf('Tran_Rate (Estimated): %.4f\n', Tran_Rate_est);

%% Plot Results
% figure;
% scatter(t_data, ln_I, 'co', 'DisplayName', 'True ln(I(t))');
% hold on;
% plot(t_data, X * theta, 'g-', 'LineWidth', 1.5, 'DisplayName', 'Fitted Line');
% xlabel('Time (days)');
% ylabel('ln(I(t))');
% title('Linear Least Squares Fit for ln(I(t))');
% legend;
% grid on;
end
```

```
disp('Estimated Parameters:');
```

```
Estimated Parameters:
```

```
fprintf('I(0) (Estimated): %.2f\n', Initial_Infected_est);
```

```
I(0) (Estimated): 17.91
```

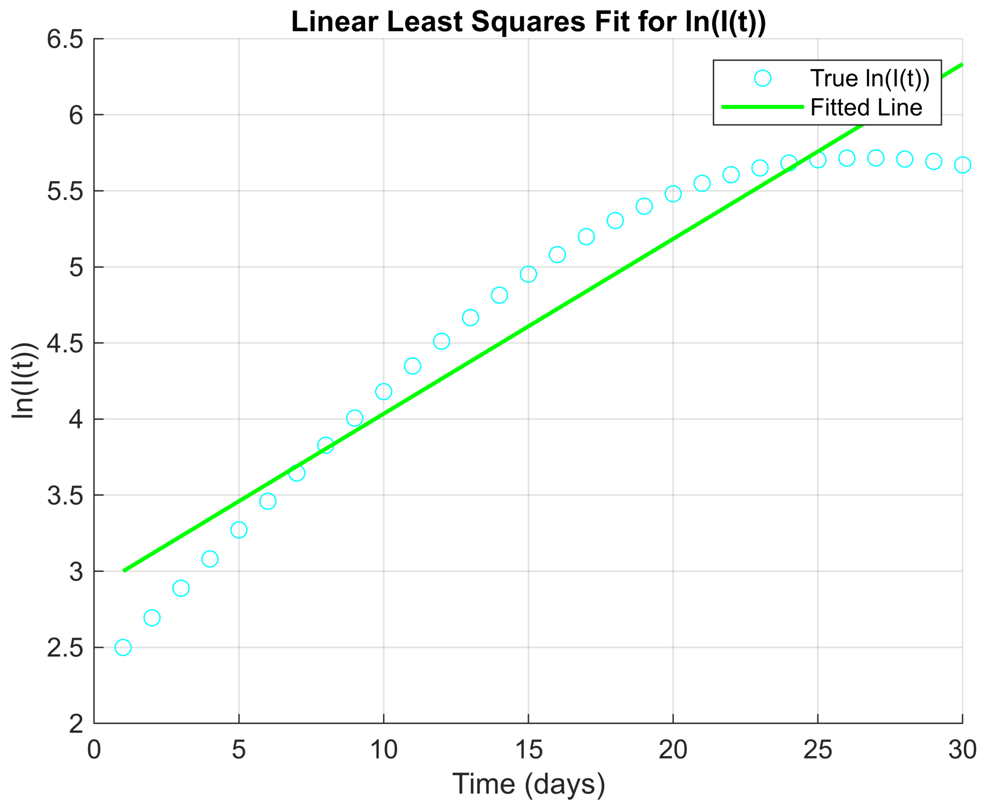
```
fprintf('k (Estimated): %.4f\n', k_est);
```

```
k (Estimated): 0.1149
```

```
fprintf('Tran_Rate (Estimated): %.4f\n', Tran_Rate_est);
```

```
Tran_Rate (Estimated): 0.2171
```

```
% Plot Results
figure(10);
scatter(t_data, ln_I, 'co', 'DisplayName', 'True ln(I(t))');
hold on;
plot(t_data, X * theta, 'g-', 'LineWidth', 1.5, 'DisplayName', 'Fitted Line');
xlabel('Time (days)');
ylabel('ln(I(t))');
title('Linear Least Squares Fit for ln(I(t))');
legend;
grid on;
hold off;
```




```
[ln_I_10, Initial_Infected_est_10, Tran_Rate_est_10, k_est_10, t_data_10, X_10, theta_10] =
Linear_Least_Squares(990,10,0,10,1,Influenza_Tran_Rate,Influenza_Recov_Rate);
disp('Estimated Parameters:');
```

```
Estimated Parameters:
```

```
fprintf('I(0) (Estimated): %.2f\n', Initial_Infected_est);
```

```
I(0) (Estimated): 17.91
```

```
fprintf('k (Estimated): %.4f\n', k_est);
```

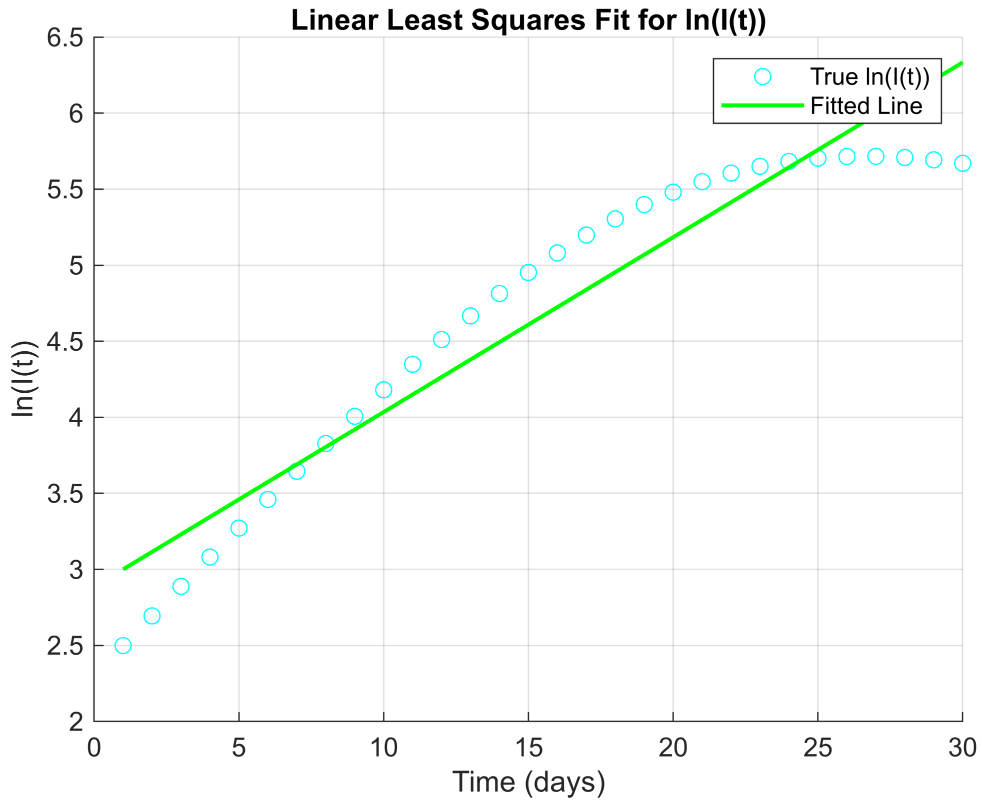
```
k (Estimated): 0.1149
```

```
fprintf('Tran_Rate (Estimated): %.4f\n', Tran_Rate_est);
```

```
Tran_Rate (Estimated): 0.2171
```

```
% Plot Results
figure(12);
scatter(t_data, ln_I, 'co', 'DisplayName', 'True ln(I(t))');
hold on;
plot(t_data, X * theta, 'g-', 'LineWidth', 1.5, 'DisplayName', 'Fitted Line');
xlabel('Time (days)');
ylabel('ln(I(t))');
title('Linear Least Squares Fit for ln(I(t))');
```

```
legend;
grid on;
hold off;
```



Discussion

By using 30 days we were able to get a more comprehensive view of the epidemic which resulted in a more accurate and reliable parameter estimate. While 10 days may be sufficient for a rough approximation, having a longer time period is more accurate.

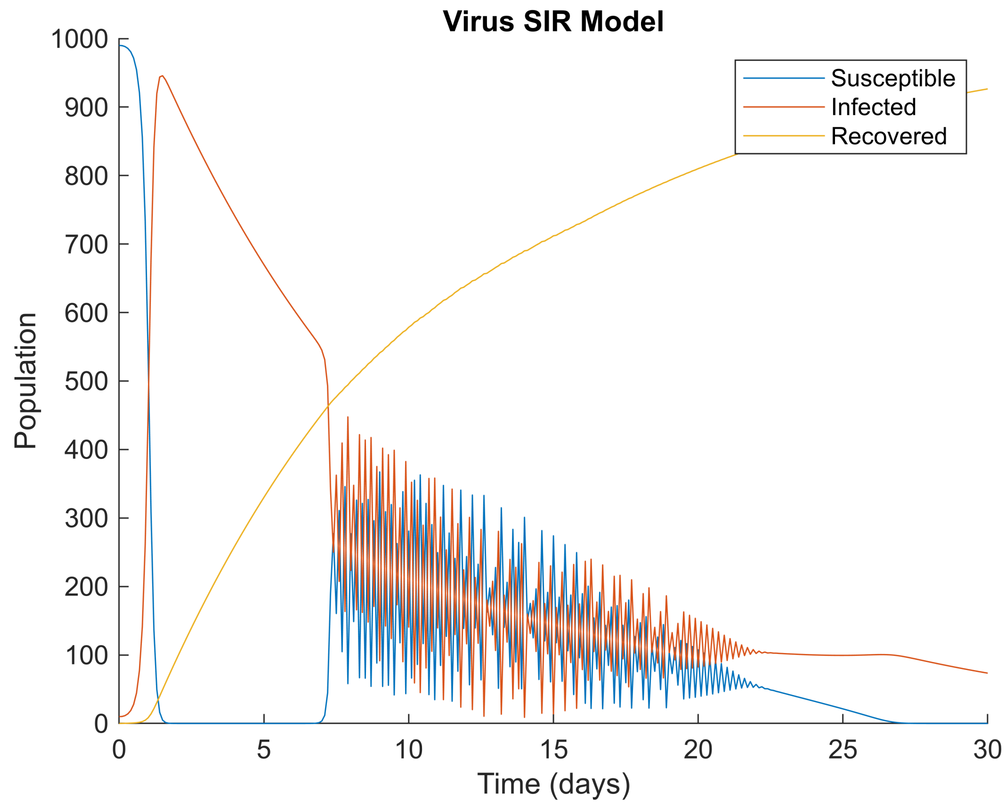
Part 4: Fourier Analysis

```
clear; clc; close all


type Periodic_SIR_Model.m
```

```
function [S,I,R,Time] = Periodic_SIR_Model(Initial_Susceptible,Initial_Infected,Initial_Recovered,Sim_Time,Time_Step,Recov_Rate,Initial_Transmission_Rate,Amplitude,Angular_Frequency)


Time = 0:Time_Step:Sim_Time;
Tran_Rate = Initial_Transmission_Rate.*(1 + Amplitude .* Angular_Frequency .* Time);


  Initial_Counts = [Initial_Susceptible,Initial_Infected,Initial_Recovered];
```

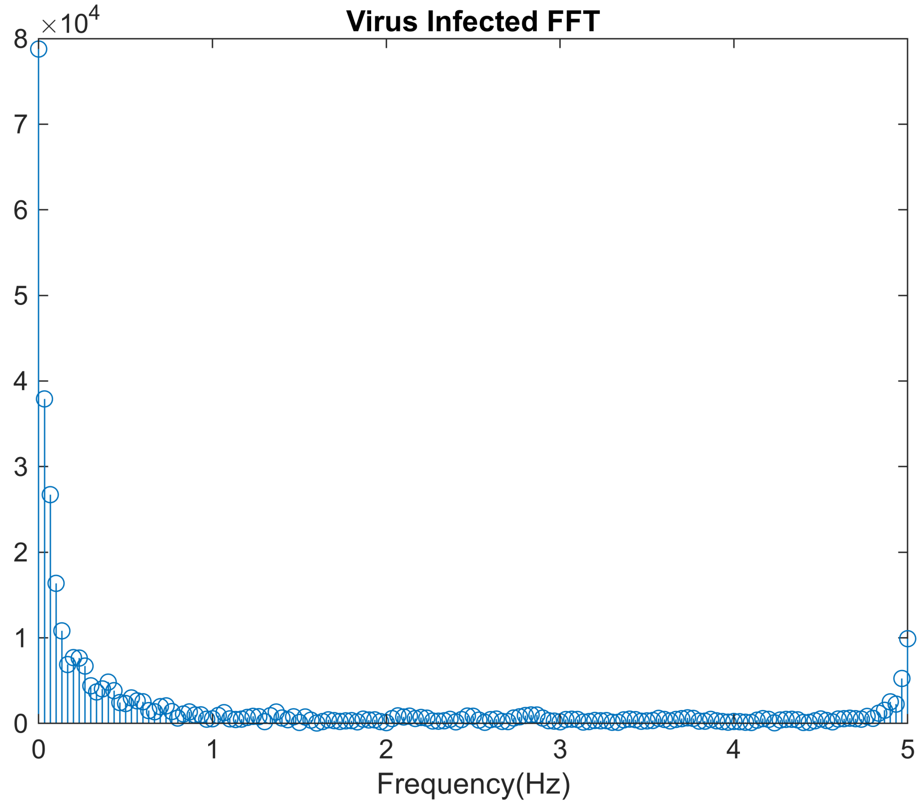
```
    N = sum(Initial_Counts);

    Solution = zeros(length(Time),3);
    Solution(1,:) = Initial_Counts;

    SIR_System = @(Time,Solution_Array,Tran_Rate) [
        -(Tran_Rate / N) * Solution_Array(1) * Solution_Array(2);
        (Tran_Rate / N) * Solution_Array(1) * Solution_Array(2) - Recov_Rate * Solution_Array(2);
        Recov_Rate * Solution_Array(2);
    ];

    for i = 1:(length(Time) - 1)
        Solution_Transposed = Solution(i,:)';

        k1 = SIR_System(Time(i),Solution_Transposed,Tran_Rate(i)) * Time_Step;
        k2 = SIR_System(Time(i) + (Time_Step / 2), Solution_Transposed + (k1 / 2),Tran_Rate(i)) * Time_Step;
        k3 = SIR_System(Time(i) + (Time_Step / 2), Solution_Transposed + (k2 / 2),Tran_Rate(i)) * Time_Step;
        k4 = SIR_System(Time(i) + Time_Step, Solution_Transposed + k3,Tran_Rate(i)) * Time_Step;

        Solution(i+1,:) = (Solution_Transposed + (k1 + 2*k2 + 2*k3 + k4) / 6)';
    end
    S = Solution(:,1);
    I = Solution(:,2);
    R = Solution(:,3);
end
```

```
Recov_Rate = 0.1;
Initial_Tran_Rate = 0.3;
Amplitude = 5;
Angular_Frequency = 2*pi;
Duration = 30;
timeStep = 0.1;


[Virus_S,Virus_I,Virus_R,Time] = Periodic_SIR_Model(990,10,0,Duration,timeStep,Recov_Rate,Initial_Tran_Rate,Amplitude,Angular_Frequency);



figure
hold on
title('Virus SIR Model')
plot(Time,Virus_S)
plot(Time,Virus_I)
plot(Time,Virus_R)
legend('Susceptible','Infected','Recovered')
xlabel('Time (days)');
ylabel('Population');
xlim([0 30])
ylim([0 1000])
hold off
```



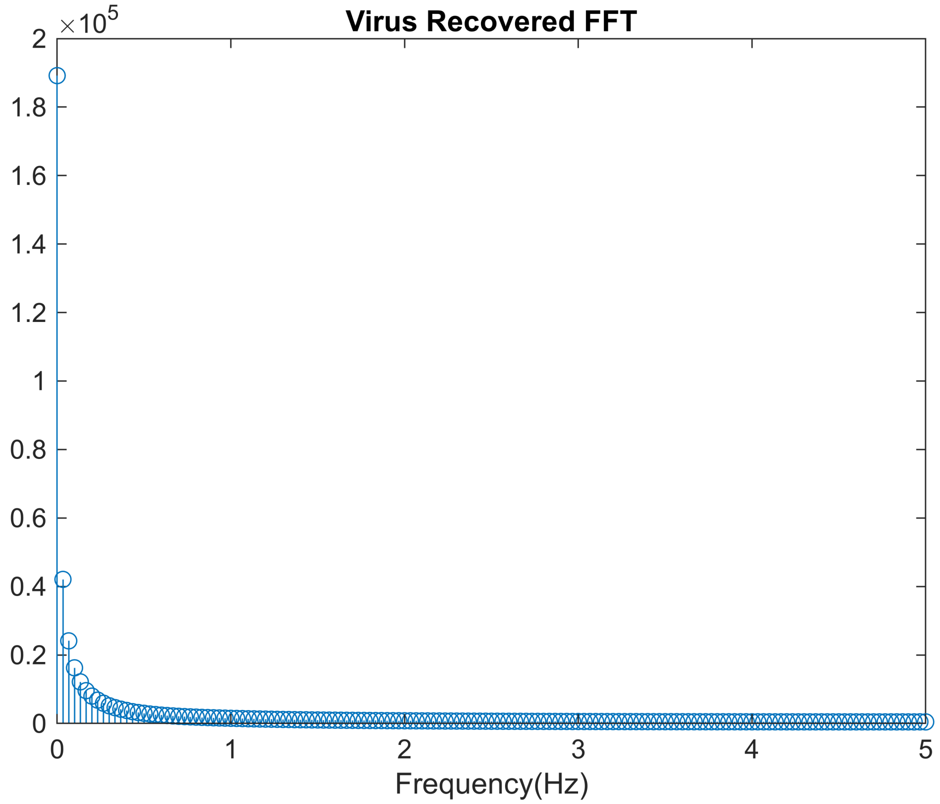
In the above figure of the SIR model for the given virus with a periodic transmission rate, fluctuations can mostly be seem in the infected and susceptible trend lines. This behavior starts to take to effect between day seven and day 22 and has a decreasing envelope through the oscillatory phase.

```
Virus_I_FFT = fft(Virus_I);
Virus_R_FFT = fft(Virus_R);
Virus_S_FFT = fft(Virus_S);

freq = 1/Duration*(0:(Duration/(timeStep*2)));



figure
stem(freq, abs(Virus_I_FFT(1:(Duration/(timeStep*2))+1)));
title('Virus Infected FFT')
xlabel('Frequency(Hz)');
```

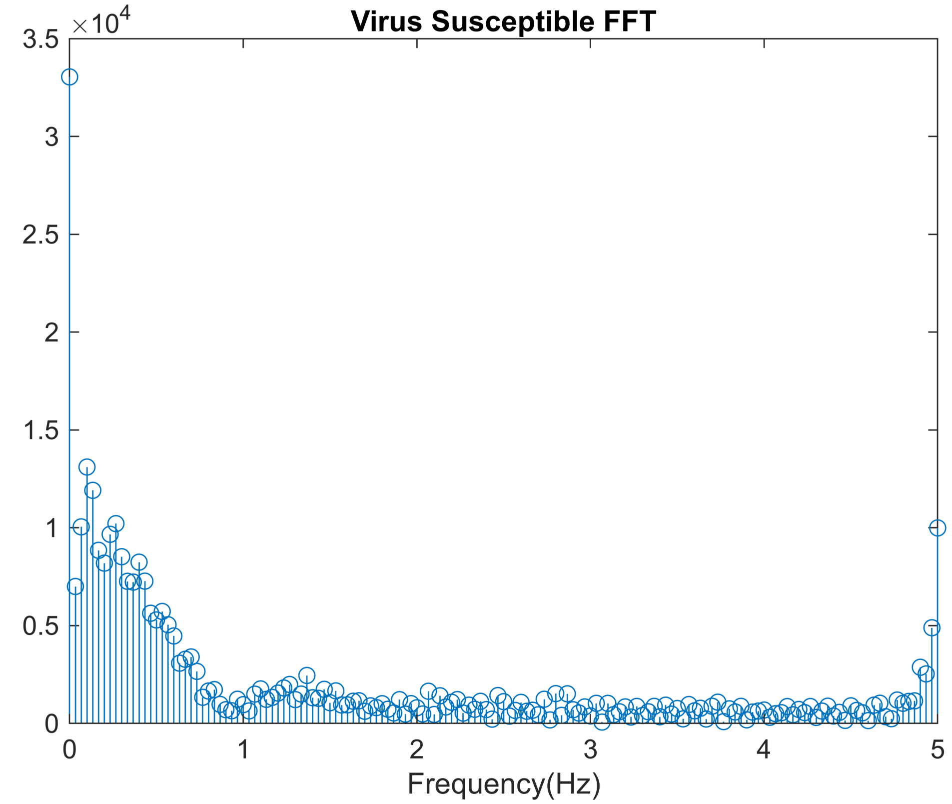


In the above figure showing the frequency spectrum of the infected cases there seems to be a frequency peak around 5Hz. This could make physical sense given that it is a frequency value that would be feasible with the given system.

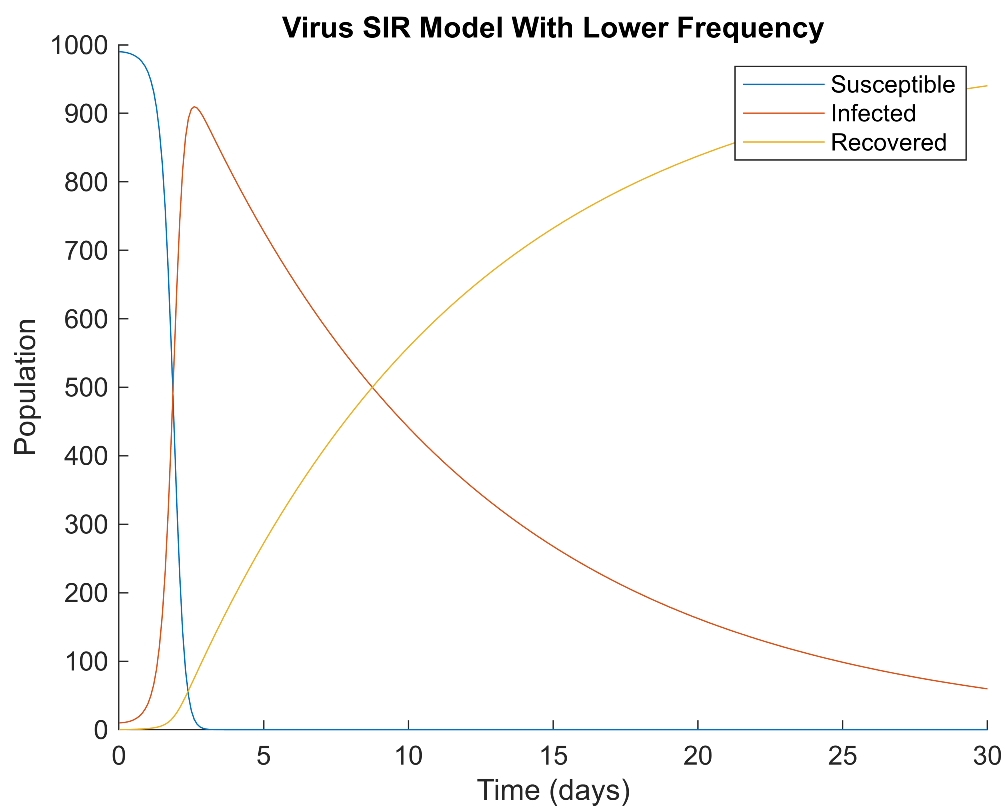
```
figure
stem(freq, abs(Virus_R_FFT(1:(Duration/(timeStep*2))+1)));
title('Virus Recovered FFT')
xlabel('Frequency(Hz)');
```



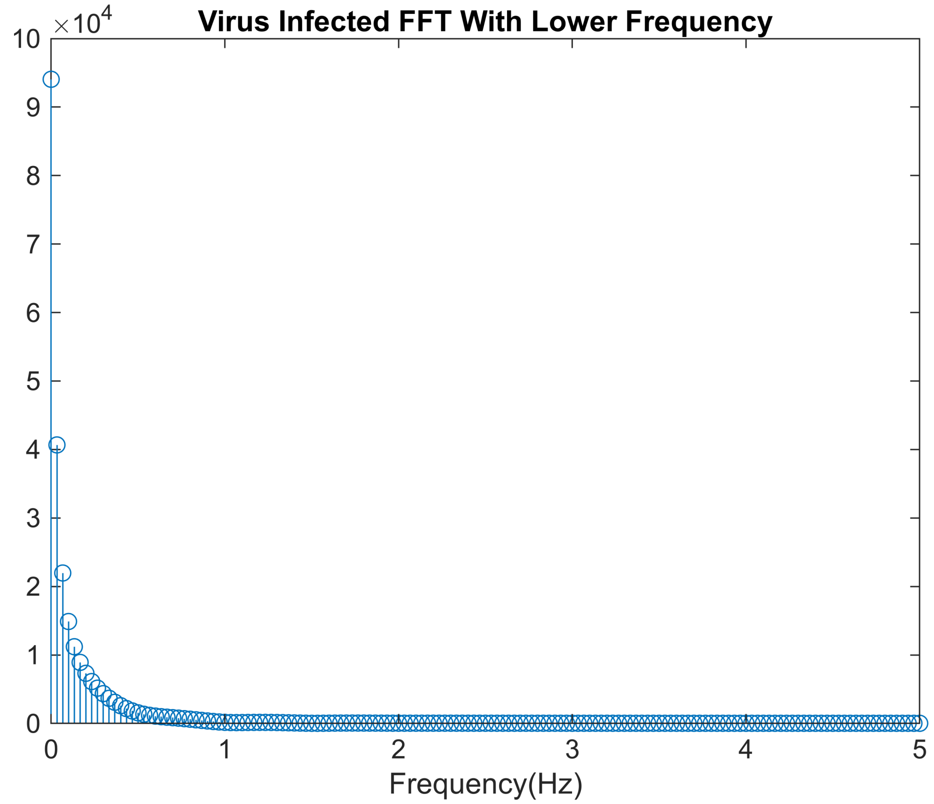

```
figure
stem(freq, abs(Virus_S_FFT(1:(Duration/(timeStep*2))+1)));
title('Virus Susceptible FFT')
xlabel('Frequency(Hz)');
```



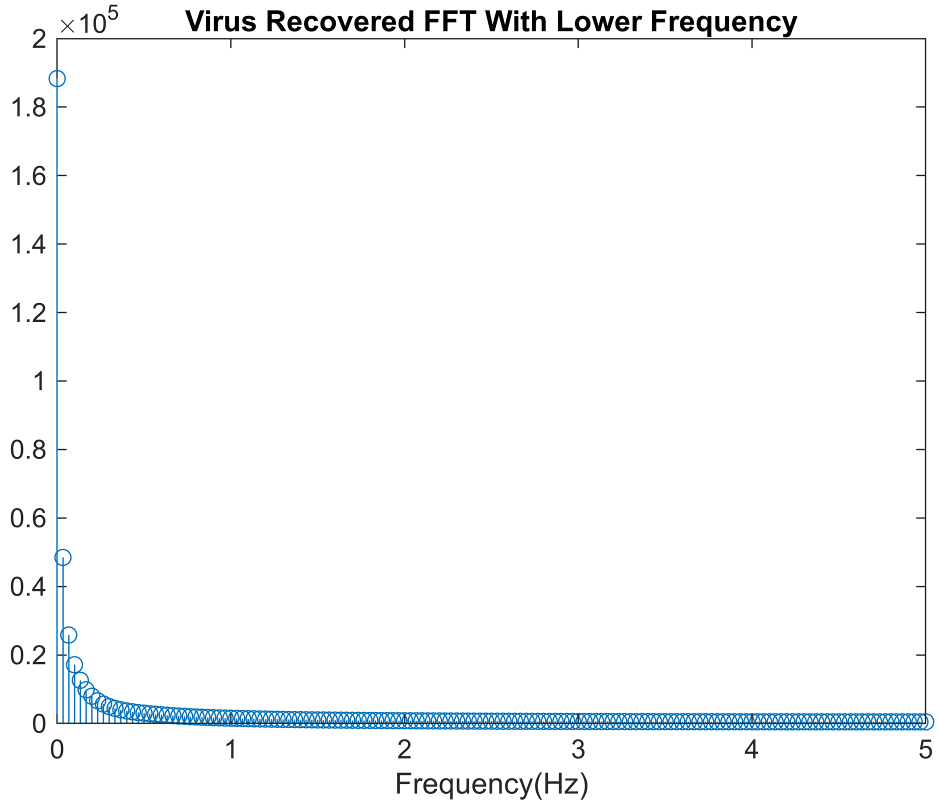

```
Angular_Frequency = 2*pi*100/365;


[Virus_S,Virus_I,Virus_R,Time] = Periodic_SIR_Model(990,10,0,Duration,timeStep,Recov_Rate,Initial_Tran_Rate,Amplitude,Angular_Frequency);

figure
hold on
title('Virus SIR Model With Lower Frequency')
plot(Time,Virus_S)
plot(Time,Virus_I)
plot(Time,Virus_R)
legend('Susceptible','Infected','Recovered')
xlabel('Time (days)');
ylabel('Population');
xlim([0 30])
ylim([0 1000])
hold off
```



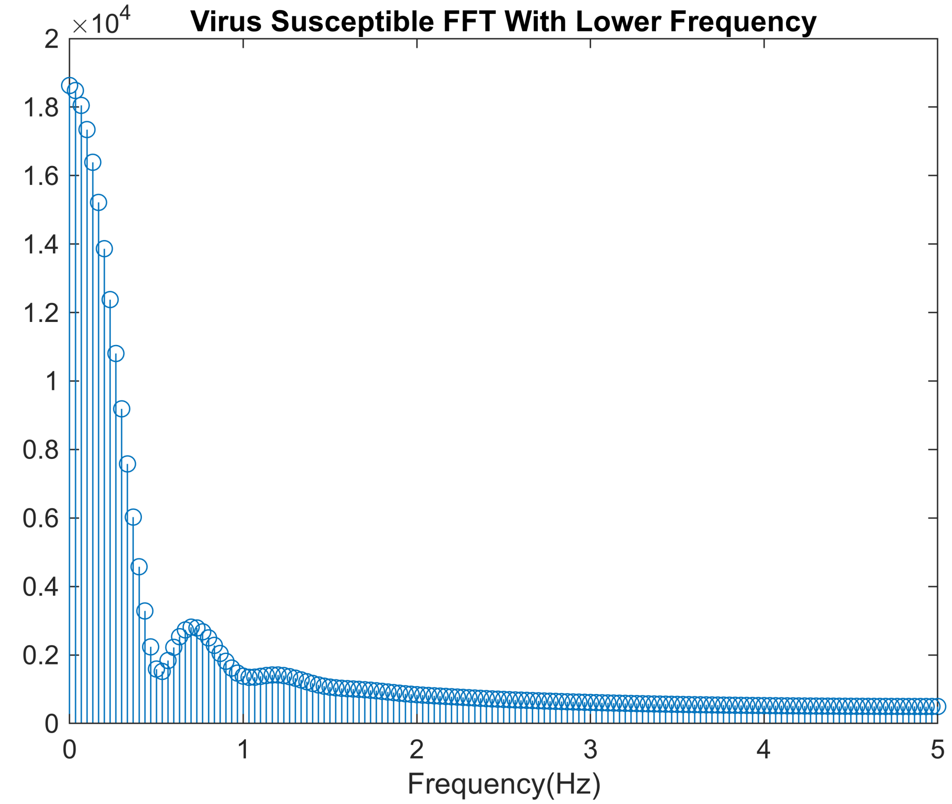

```
Virus_I_FFT = fft(Virus_I);
Virus_R_FFT = fft(Virus_R);
Virus_S_FFT = fft(Virus_S);

freq = 1/Duration*(0:(Duration/(timeStep*2)));



figure
stem(freq, abs(Virus_I_FFT(1:(Duration/(timeStep*2))+1)));
title('Virus Infected FFT With Lower Frequency')
xlabel('Frequency(Hz)');
```




```
figure
stem(freq, abs(Virus_R_FFT(1:(Duration/(timeStep*2))+1)));
title('Virus Recovered FFT With Lower Frequency')
xlabel('Frequency(Hz)');
```




```
figure
stem(freq, abs(Virus_S_FFT(1:(Duration/(timeStep*2))+1)));
title('Virus Susceptible FFT With Lower Frequency')
xlabel('Frequency(Hz)');
```



The change in the peak frequency seems to shift to lower values with a lower transmission rate fluctuation frequency. This seems like it would be a reasonable finding given that changes in the fluctuation of the transmission rate would result in diffrent effects on the system as a whole with the same general trend, ie the reduction in transmission rate frequency would decrease the overall fundamental frequencies that build up the individual signals.