Section 3

Table of Contents

Discussion

The estimates of I(0) and beta impove as the time frame is shortened. I(0) true was 10, I(0) on the 30-day period was 17.9, I(0) on the 10-day period of 10.23.

Beta true was 0.3, Beta estimate on the 30-day period was 0.21, Beta estimate on the 10-day period was 0.0.29.

This is because the original equation is parabolic and the smaller the sample of data you look at makes the function appear to be more linear, thus making a linear regression estimate more and more accurate.

```
for j = 1:3
    %j = 1 = Q1 - True Data Set (30 day model)
    %j = 2 = Q2 - Estimate via Linear Least Sq (30 day model)
    %j = 3 = Q3 - Estimate via Linear Least Sq (10 day model)

    if j == 1
        % Parameter sets
        h = 1;
        S0 = 990;
        I0 = 10;
        R0 = 0;
        T = 30;
        time_vec = 0:h:T;

        steps = length(time_vec);

        beta_true = 0.3;
        gam_true = 0.1;

        % Create blank figure for this step
%          figure(j);

        % True data set
        [S, I, R] = runge_kutta(S0, I0, R0, beta_true, gam_true, h, steps);
        plot(time_vec, S, 'b', time_vec, I, 'r', time_vec, R, 'g');
        title('True Data Set I(t)');
        xlabel('Time');
        ylabel('Population');
        legend('Susceptible', 'Infected', 'Recovered');
        grid on;
        hold off
```

```
elseif j == 2
    % Parameter sets
    h = 1;
    S0 = 990;
    I0 = 10;
    R0 = 0;
    T = 30;
    time_vec = 1:h:T;

    steps = length(time_vec);

    beta_true = 0.3;
    gam_true = 0.1;
    N = 30;

    % Create blank figure for this step
    figure(j);

    % Linear Regression calcs
    x_input = time_vec.';
    y_output = log(I(2:31));
    x2 = x_input.^2;
    xy = x_input .* y_output;
    x_avg = mean(x_input);
    y_avg = mean(y_output);

    a1 = (N*sum(xy)) - (sum(x_input))*(sum(y_output));
    a1 = a1 / (N*sum(x2) - (sum(x_input)^2));

    a0 = y_avg - (a1*x_avg);

    %Graphing to double check visually
    y_graph = a0 + (time_vec.*a1);
    plot(x_input, y_graph, 'r');
    title('Linear Least Sq Regression, 30-day period');
    xlabel('Time');
    ylabel('Population');
    grid on;
    hold on
    scatter(x_input,y_output)
    legend('Estimated Regression Model', 'True data, I(t) function');


    %Finding I0 and beta
    % I0 calcs
    I0_est_30 = exp(a0);

    % Beta calc
    k = a1;
    N = 1000;

    beta_est_30 = (N*(k+gam_true))/S0;
    elseif j == 3
        % Parameter sets
        h = 1;
        S0 = 990;
        I0 = 10;
        R0 = 0;
        T = 10;
        time_vec = 1:h:T;

        steps = length(time_vec);

        beta_true = 0.3;
        gam_true = 0.1;
        N = 10;

        % Create blank figure for this step
        figure(j);

        % Linear Regression calcs
        x_input = time_vec.';
        y_output = log(I(2:11));
        x2 = x_input.^2;
        xy = x_input .* y_output;
        x_avg = mean(x_input);
        y_avg = mean(y_output);

        a1 = (N*sum(xy)) - (sum(x_input))*(sum(y_output));
        a1 = a1 / (N*sum(x2) - (sum(x_input)^2));

        a0 = y_avg - (a1*x_avg);

        %Graphing to double check visually
        y_graph = a0 + (time_vec.*a1);
        plot(x_input, y_graph, 'r');
        title('Linear Least Sq Regression, 10-day period');
        xlabel('Time');
        ylabel('Population');
        grid on;
        hold on
        scatter(x_input,y_output)
        legend('Estimated Regression Model', 'True data, I(t) function')

        %Finding I0 and beta
        % I0 calcs
        I0_est_10 = exp(a0);

        % Beta calc
        k = a1;
        N = 1000;

        beta_est_10 = (N*(k+gam_true))/S0;

    end
end
```

Function library

```
% Runge-Kutta
function [S, I, R] = runge_kutta(S0, I0, R0, beta, gamma, h, steps)
    S = zeros(steps, 1);
    I = zeros(steps, 1);
    R = zeros(steps, 1);

    % initial conditions
    S(1) = S0;
    I(1) = I0;
    R(1) = R0;

    % Total pop
    N = S0 + I0 + R0;

    for t = 1:steps-1
        % k1
        dS1 = -beta * S(t) * I(t) / N;
        dI1 = beta * S(t) * I(t) / N - gamma * I(t);
        dR1 = gamma * I(t);

        % k2
        dS2 = -beta * (S(t) + h*dS1/2) * (I(t) + h*dI1/2) / N;
        dI2 = beta * (S(t) + h*dS1/2) * (I(t) + h*dI1/2) / N - gamma * (I(t) +
 h*dI1/2);
        dR2 = gamma * (I(t) + h*dI1/2);

        % k3
        dS3 = -beta * (S(t) + h*dS2/2) * (I(t) + h*dI2/2) / N;
        dI3 = beta * (S(t) + h*dS2/2) * (I(t) + h*dI2/2) / N - gamma * (I(t) +
 h*dI2/2);
        dR3 = gamma * (I(t) + h*dI2/2);

        % k4
        dS4 = -beta * (S(t) + h*dS3) * (I(t) + h*dI3) / N;
        dI4 = beta * (S(t) + h*dS3) * (I(t) + h*dI3) / N - gamma * (I(t) +
 h*dI3);
        dR4 = gamma * (I(t) + h*dI3);

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

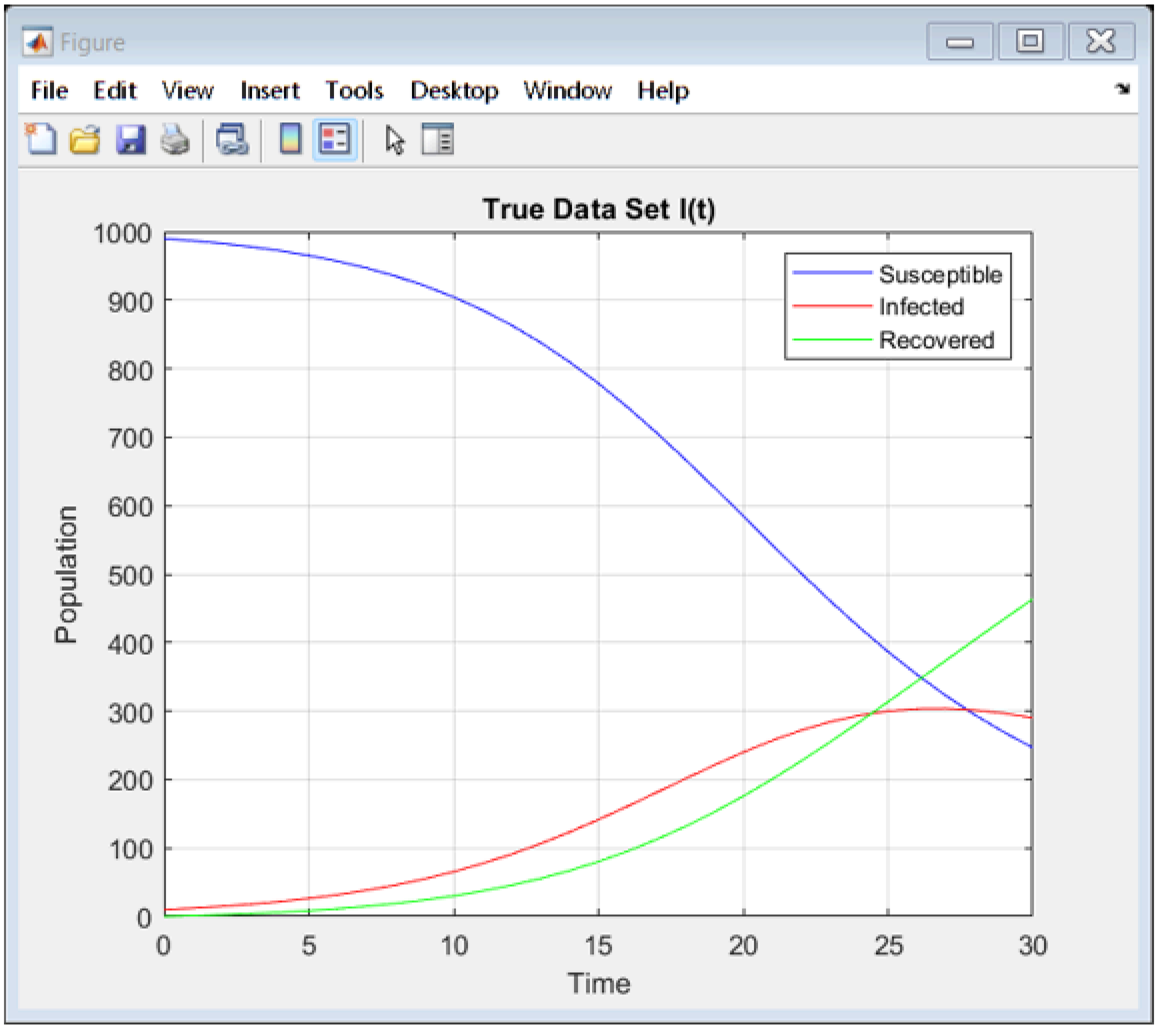


Figure
File
Edit
View
Insert
Tools
Desktop
Window
Help
True Data Set I(t)
Population
Time
0
100
200
300
400
500
600
700
800
900
1000
0
5
10
15
20
25
30
Susceptible
Infected
Recovered

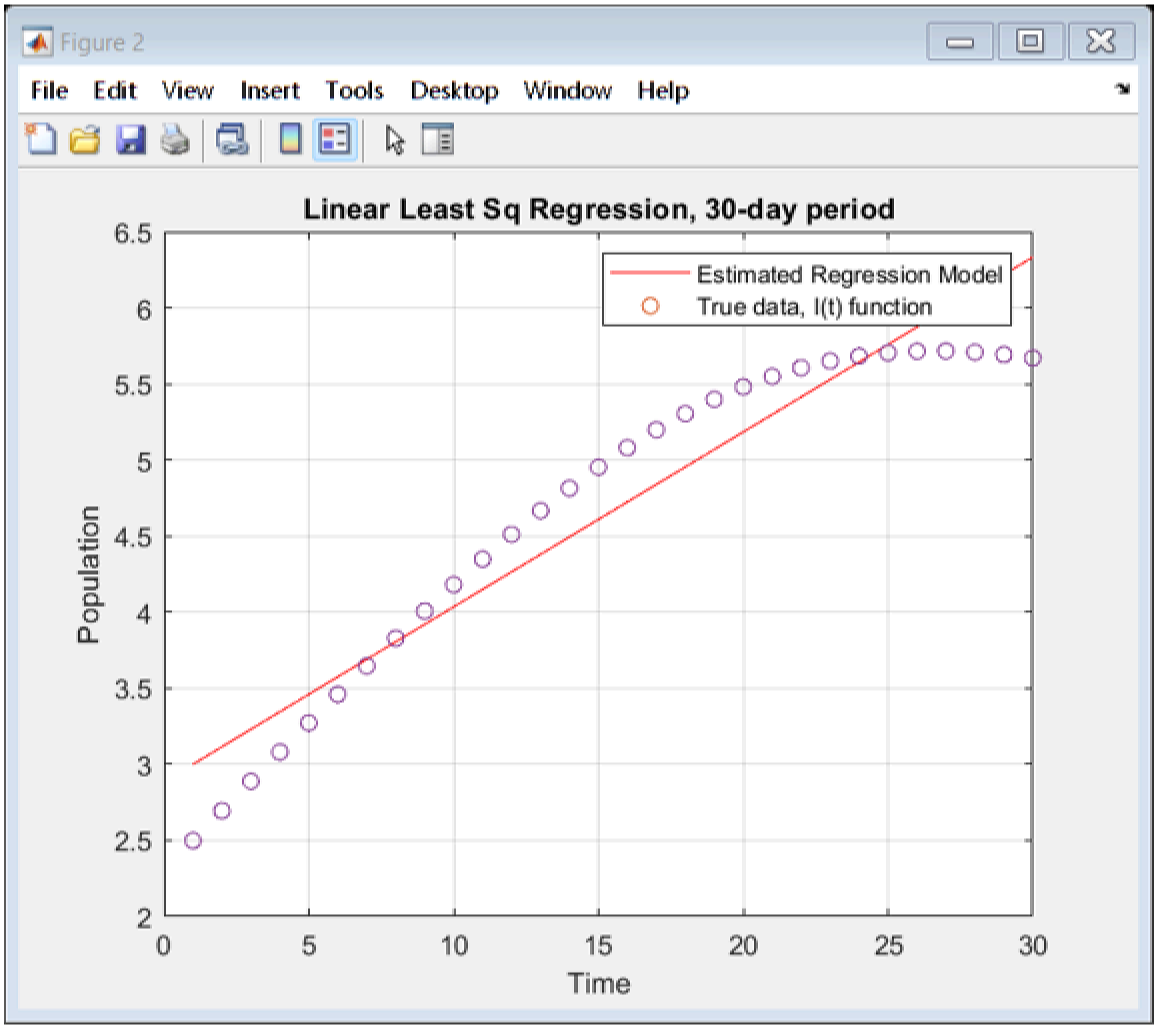
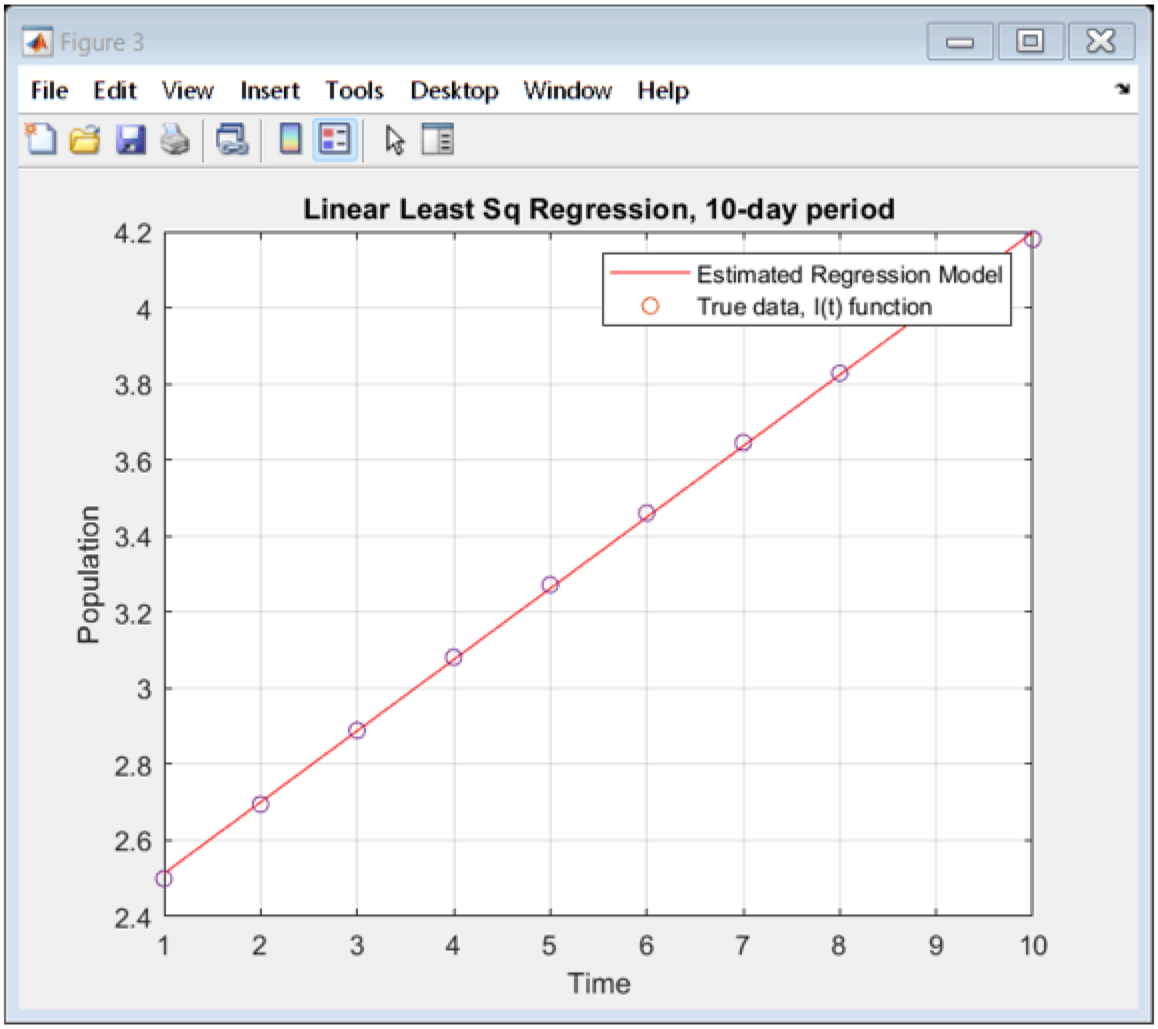


Figure 2
File Edit View Insert Tools Desktop Window Help
Linear Least Sq Regression, 30-day period
Estimated Regression Model
True data, I(t) function
Population
Time
6.5
6
5.5
5
4.5
4
3.5
3
2.5
2
0
5
10
15
20
25
30



Published with MATLAB® R2023a