Table of Contents


```
data = load("COVIDdata.mat");
COVID_STLmetro = deal(data.COVID_STLmetro);
STLmetroPopFixed=27.3714*100000;

COVID_MO_array=table2array(COVID_STLmetro(:,5:6));
COVID_MO_proportion=COVID_MO_array/STLmetroPopFixed;

coviddata = COVID_MO_proportion; % TO SPECIFY
t = length(coviddata(:,1)); % TO SPECIFY

% The following line creates an 'anonymous' function that will return the cost
 (i.e., the model fitting error) given a set
% of parameters.  There are some technical reasons for setting this up in this
 way.
% Feel free to peruse the MATLAB help at
% https://www.mathworks.com/help/optim/ug/fmincon.html
% and see the sectiono on 'passing extra arguments'
% Basically, 'sirafun' is being set as the function siroutput (which you
% will be designing) but with t and coviddata specified.
sirafun= @(x)siroutput_part2(x,t,coviddata);
```

set up rate and initial condition constraints

Set A and b to impose a parameter inequality constraint of the form A*x < b Note that this is imposed element-wise If you don't want such a constraint, keep these matrices empty.

```
A = [0 0 0 .99 .99 .99 .99];
b = [1];
```

set up some fixed constraints

Set Af and bf to impose a parameter constraint of the form Af*x = bf Hint: For example, the sum of the initial conditions should be constrained If you don't want such a constraint, keep these matrices empty.

```
Af = [0 0 0 1 1 1 1];
bf = [1];
```

set up upper and lower bound constraints

Set upper and lower bounds on the parameters lb < x < ub here, the inequality is imposed element-wise If you don't want such a constraint, keep these matrices empty.

```
ub = [1 1 1 1 1 1 1]';
lb = [0 0 0 0 0 0 0]';

% Specify some initial parameters for the optimizer to start from
x0 = [0 0 0 1 0 0 0];

% This is the key line that tries to opimize your model parameters in order to
% fit the data
% note tath you
x = fmincon(sirafun,x0,A,b,Af,bf,lb,ub);

x_new_4_1 = x;
x_new_4_1(1)= 0.20*x(1);

x_new_4_2 = x;
x_new_4_2(2)= 0.04*x(2);
x_new_4_2(1)= 0.20*x(1);

%plot(Y);
%legend('S','I','R','D');
%xlabel('Time')

Y_fit = siroutput_full(x,t);
Y_fit_4_A = siroutput_full(x_new_4_1,365);

x_new_4_2(4) = Y_fit_4_A(365,1);
x_new_4_2(5) = Y_fit_4_A(365,2);
x_new_4_2(6) = Y_fit_4_A(365,3);
x_new_4_2(7) = Y_fit_4_A(365,4);

Y_fit_4_B = siroutput_full(x_new_4_2,t-365);

Y_fit_new_4_2=cat(1,Y_fit_4_A,Y_fit_4_B);
```

```
Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the value of the optimality tolerance,
and constraints are satisfied to within the value of the constraint tolerance.
```

Model each segment

```
%Segments to focus on:
%0-100, 101-250, 251-350, 351-500, 500-650, 651-700, 701-end

%Model section for segment 1
segment_1 = coviddata(1:100,:);
fun1= @(x1)siroutput(x1,100,segment_1);
x1 = fmincon(fun1,x0,A,b,Af,bf,lb,ub);
Y_fit_1 = siroutput_full(x1, 100);
```

```
%model section for segment 2 using new parameters
segment_2 = coviddata(101:250, :);
fun2= @(x2)siroutput(x2,150,segment_2);
params2 = [x1(1), x1(2), x1(3), Y_fit_1(100,1), Y_fit_1(100, 2), Y_fit_1(100,
 3), Y_fit_1(100, 4)];
x2 = fmincon(fun2, params2, A, b, Af, bf, lb, ub);
Y_fit_2 = siroutput_full(x2, 150);

segment_3 = coviddata(251:350,:);
fun3= @(x)siroutput(x,100,segment_3);
params3 = [x2(1) x2(2) x2(3) Y_fit_2(150, 1) Y_fit_2(150, 2) Y_fit_2(150, 3)
 Y_fit_2(150, 4)];
x3 = fmincon(fun3, params3, A, b, Af, bf, lb, ub);
Y_fit_3 = siroutput_full(x3, 100);

segment_4 = coviddata(351:500,:);
fun4= @(x)siroutput(x,150,segment_4);
params4 = [x3(1) x3(2) x3(3) Y_fit_3(100, 1) Y_fit_3(100, 2) Y_fit_3(100, 3)
 Y_fit_3(100, 4)];
x4 = fmincon(fun4, params4, A, b, Af, bf, lb, ub);
Y_fit_4 = siroutput_full(x4, 150);

segment_5 = coviddata(501:650,:);
fun5= @(x)siroutput(x,150,segment_5);
params5 = [x4(1) x4(2) x4(3) Y_fit_4(150, 1) Y_fit_4(150, 2) Y_fit_4(150, 3)
 Y_fit_4(150, 4)];
x5 = fmincon(fun5, params5, A, b, Af, bf, lb, ub);
Y_fit_5 = siroutput_full(x5, 150);

segment_6 = coviddata(651:700,:);
fun6= @(x)siroutput(x,50,segment_6);
params6 = [x5(1) x5(2) x5(3) Y_fit_5(150, 1) Y_fit_5(150, 2) Y_fit_5(150, 3)
 Y_fit_5(150, 4)];
x6 = fmincon(fun6, params6, A, b, Af, bf, lb, ub);
Y_fit_6 = siroutput_full(x6, 50);

segment_7 = coviddata(701:798,:);
fun7= @(x)siroutput(x,t-700,segment_7);
params7 = [x6(1) x6(2) x6(3) Y_fit_6(50, 1) Y_fit_6(50, 2) Y_fit_6(50, 3)
 Y_fit_6(50, 4)];
x7 = fmincon(fun7, params7, A, b, Af, bf, lb, ub);
Y_fit_7 = siroutput_full(x7, t-700);

Segmented_Fit = cat(1, Y_fit_1, Y_fit_2, Y_fit_3, Y_fit_4, Y_fit_5, Y_fit_6,
 Y_fit_7);
```

Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the value of the step size tolerance and constraints are
satisfied to within the value of the constraint tolerance.

```
Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the value of the step size tolerance and constraints are
satisfied to within the value of the constraint tolerance.


Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the value of the step size tolerance and constraints are
satisfied to within the value of the constraint tolerance.


Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the value of the step size tolerance and constraints are
satisfied to within the value of the constraint tolerance.


Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the value of the step size tolerance and constraints are
satisfied to within the value of the constraint tolerance.


Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the value of the step size tolerance and constraints are
satisfied to within the value of the constraint tolerance.


Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the value of the step size tolerance and constraints are
satisfied to within the value of the constraint tolerance.
```

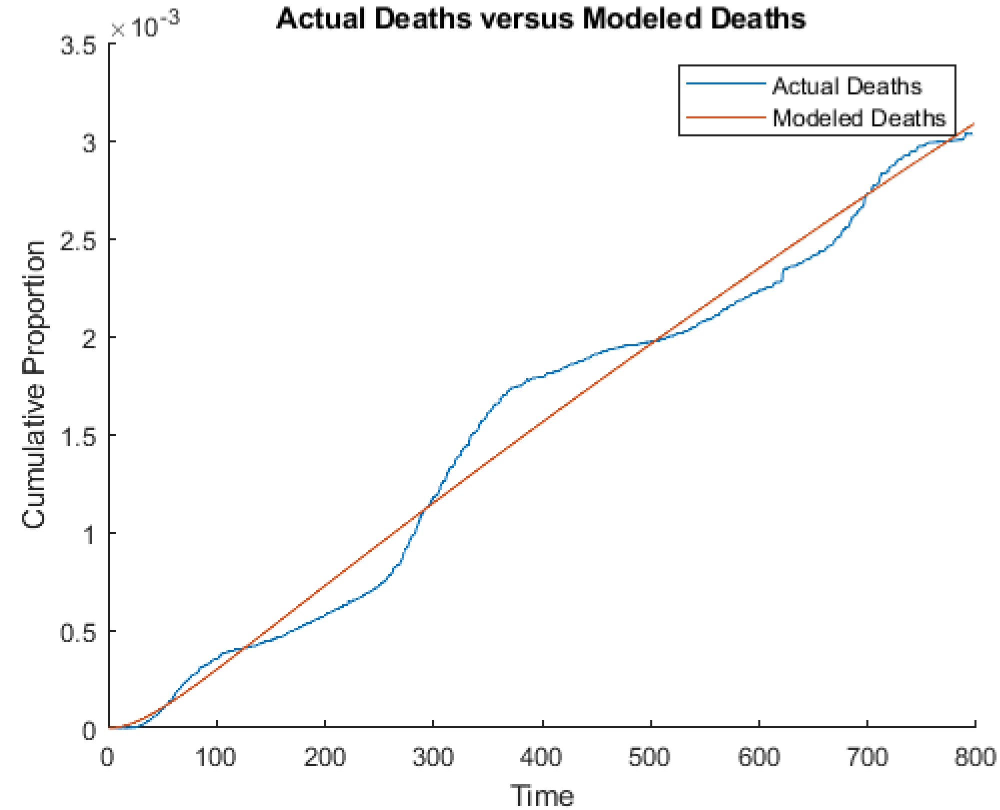
SEGMENT MODIFICATION OF MODDED DATA

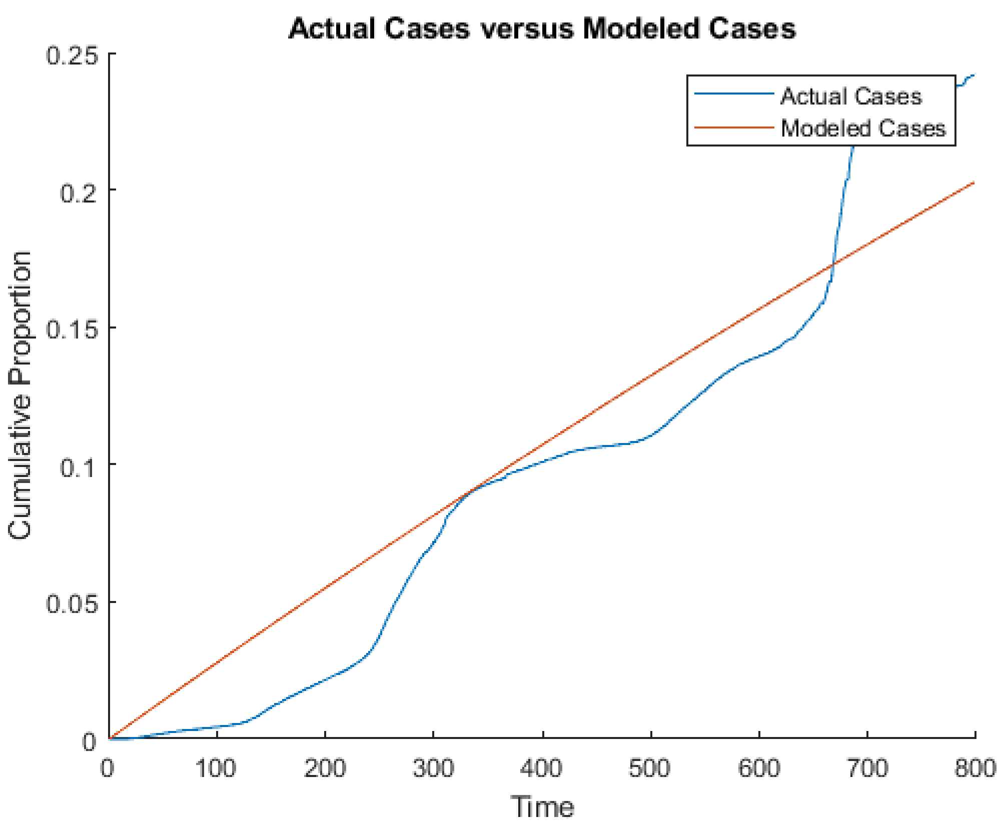
```
%Result will be Y_fit_new and should look identical to Segmented_Fit at the
%end. As of right now that is not the case, but its a work in progress

%Adjust first segment of pandemic
x_new_5_1 = x1;
%Adjust infection rate
x_new_5_1(1)= x1(1)*.75;
%Adjust fatality rate
x_new_5_1(2) = x1(2)*.75;
```

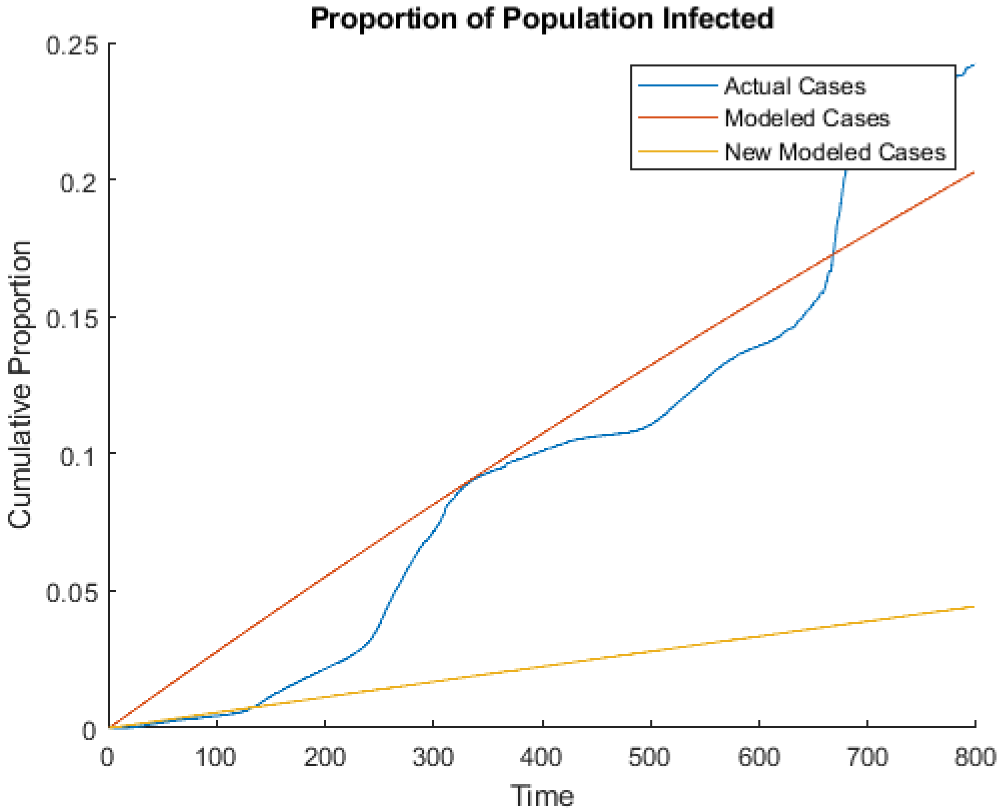
```
%adjust second segment of pandemic
x_new_5_2 = x2;
%Adjust Infection Rate
x_new_5_2(1)= x2(1)*.75;
%Adjust fatalities
x_new_5_2(2)= x2(2)*.75;

%adjust 3rd segment of pandemic
x_new_5_3 = x3;
%Adjust Infection Rate
x_new_5_3(1)= x3(1)*.75;
%Adjust fatalities
x_new_5_3(2)= x3(2)*.75;

%adjust 4th segment of pandemic
x_new_5_4 = x4;
%Adjust Infection Rate
x_new_5_4(1)= x4(1)*.75;
%Adjust fatalities
x_new_5_4(2)= x4(2)*.75;

%adjust 5th segment of pandemic
x_new_5_5 = x5;
%Adjust Infection Rate
x_new_5_5(1)= x5(1)*.75;
%Adjust fatalities
x_new_5_5(2)= x5(2)*.75;

%adjust 6th segment of pandemic
x_new_5_6 = x6;
%Adjust Infection Rate
x_new_5_6(1)= x6(1)*.75;
%Adjust fatalities
x_new_5_6(2)= x6(2)*.75;

%adjust 7th segment of pandemic
x_new_5_7 = x7;
%Adjust Infection Rate
x_new_5_7(1)= x7(1)*.75;
%Adjust fatalities
x_new_5_7(2)= x7(2)*.75;

%plot(Y);
%legend('S','I','R','D');
%xlabel('Time')

%Segment out portions of data
Y_fit_5_A = siroutput_full(x_new_5_1,100);
x_new_5_2(4) = Y_fit_5_A(100,1);
x_new_5_2(5) = Y_fit_5_A(100,2);
x_new_5_2(6) = Y_fit_5_A(100,3);
x_new_5_2(7) = Y_fit_5_A(100,4);
```

```
Y_fit_5_B = siroutput_full(x_new_5_2,150);
x_new_5_3(4) = Y_fit_5_B(150,1);
x_new_5_3(5) = Y_fit_5_B(150,2);
x_new_5_3(6) = Y_fit_5_B(150,3);
x_new_5_3(7) = Y_fit_5_B(150,4);

Y_fit_5_C = siroutput_full(x_new_5_3,100);
x_new_5_4(4) = Y_fit_5_C(100,1);
x_new_5_4(5) = Y_fit_5_C(100,2);
x_new_5_4(6) = Y_fit_5_C(100,3);
x_new_5_4(7) = Y_fit_5_C(100,4);

Y_fit_5_D = siroutput_full(x_new_5_4,150);
x_new_5_5(4) = Y_fit_5_D(150,1);
x_new_5_5(5) = Y_fit_5_D(150,2);
x_new_5_5(6) = Y_fit_5_D(150,3);
x_new_5_5(7) = Y_fit_5_D(150,4);

Y_fit_5_E = siroutput_full(x_new_5_5,150);
x_new_5_6(4) = Y_fit_5_E(150,1);
x_new_5_6(5) = Y_fit_5_E(150,2);
x_new_5_6(6) = Y_fit_5_E(150,3);
x_new_5_6(7) = Y_fit_5_E(150,4);

Y_fit_5_F = siroutput_full(x_new_5_6,50);
x_new_5_7(4) = Y_fit_5_F(50,1);
x_new_5_7(5) = Y_fit_5_F(50,2);
x_new_5_7(6) = Y_fit_5_F(50,3);
x_new_5_7(7) = Y_fit_5_F(50,4);

Y_fit_5_G = siroutput_full(x_new_5_7,t-700);

Y_fit_new_5_2 = cat(1, Y_fit_5_A, Y_fit_5_B, Y_fit_5_C, Y_fit_5_D, Y_fit_5_E,
 Y_fit_5_F, Y_fit_5_G);

%Y_fit_B = siroutput_full(x_new_2,t-365);

%Y_fit_new_4_2=cat(1,Y_fit_A,Y_fit_B);
figure(1);
hold on
plot(COVID_MO_proportion(:,2));
plot(Y_fit(:,4));
legend('Actual Deaths','Modeled Deaths');
xlabel('Time')
ylabel('Cumulative Proportion')
title('Actual Deaths versus Modeled Deaths')

hold off

figure(2);
hold on
plot(COVID_MO_proportion(:,1));
plot(1-Y_fit(:,1));
legend('Actual Cases','Modeled Cases');
```

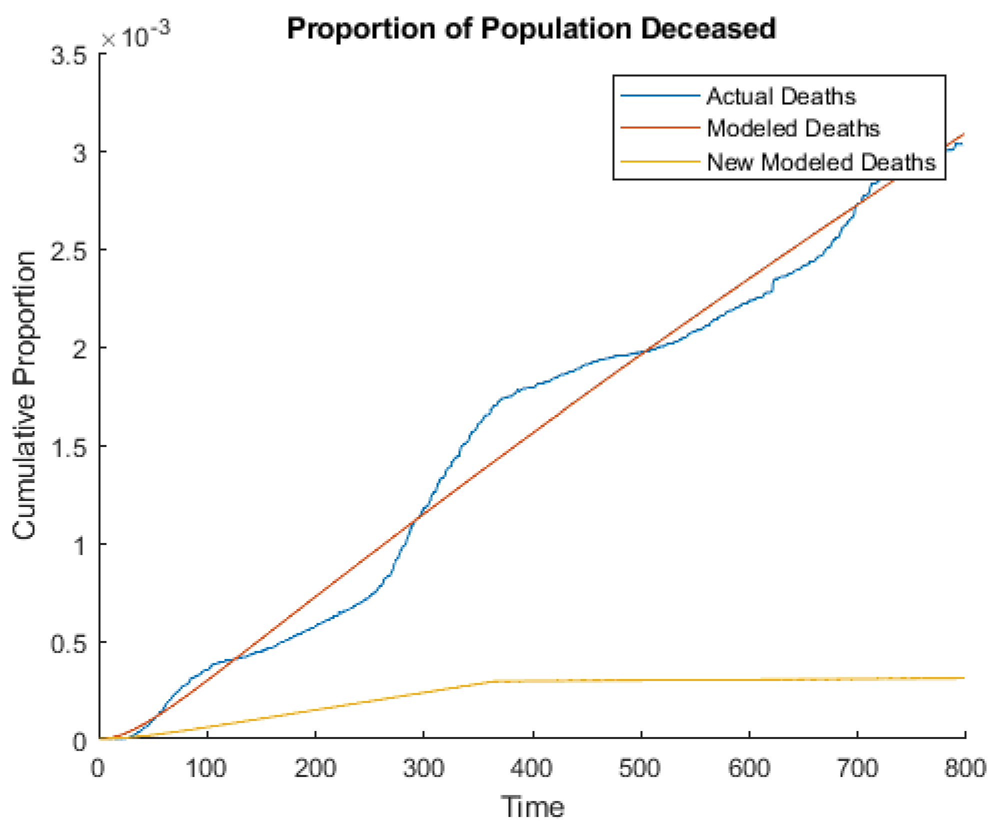
```
xlabel('Time')
ylabel('Cumulative Proportion')
title('Actual Cases versus Modeled Cases')
hold off

% Make some plots that illustrate your findings.
% TO ADD
figure(3);
hold on
plot(COVID_MO_proportion(:,1));
plot(1-Y_fit(:,1));
plot(1-Y_fit_new_4_2(:,1));
legend('Actual Cases','Modeled Cases', 'New Modeled Cases');
xlabel('Time')
ylabel('Cumulative Proportion')
title('Proportion of Population Infected')
hold off

figure(4);
hold on
plot(COVID_MO_proportion(:,2));
plot(Y_fit(:,4));
plot(Y_fit_new_4_2(:,4));
legend('Actual Deaths','Modeled Deaths', 'New Modeled Deaths');
xlabel('Time')
ylabel('Cumulative Proportion')
title('Proportion of Population Deceased')
hold off

figure(5);
hold on
plot(COVID_MO_proportion(:,1));
plot(1-Y_fit(:,1));
plot(1-Y_fit_new_5_2(:,1));
plot(1 - Segmented_Fit(:,1));
legend('Actual Cases','Modeled Cases', 'New Modeled Cases', 'Modified Model
 Cases');
xlabel('Time')
ylabel('Cumulative Proportion')
title('Cumulative Proportion of Population Infected')
hold off

figure(6);
hold on
plot(COVID_MO_proportion(:,2));
plot(Y_fit(:,4));
plot(Y_fit_new_5_2(:,4));
plot(Segmented_Fit(:,4));
legend('Actual Deaths','Modeled Deaths', 'New Modeled Deaths', 'Modified
 Modeled Deaths');
xlabel('Time')
ylabel('Cumulative Proportion')
title('Cumulative Proportion of Population Deceased')
hold off
```


Actual Deaths versus Modeled Deaths
$\times 10^{-3}$
Actual Deaths
Modeled Deaths
Cumulative Proportion
Time
0
0.5
1
1.5
2
2.5
3
3.5
0
100
200
300
400
500
600
700
800

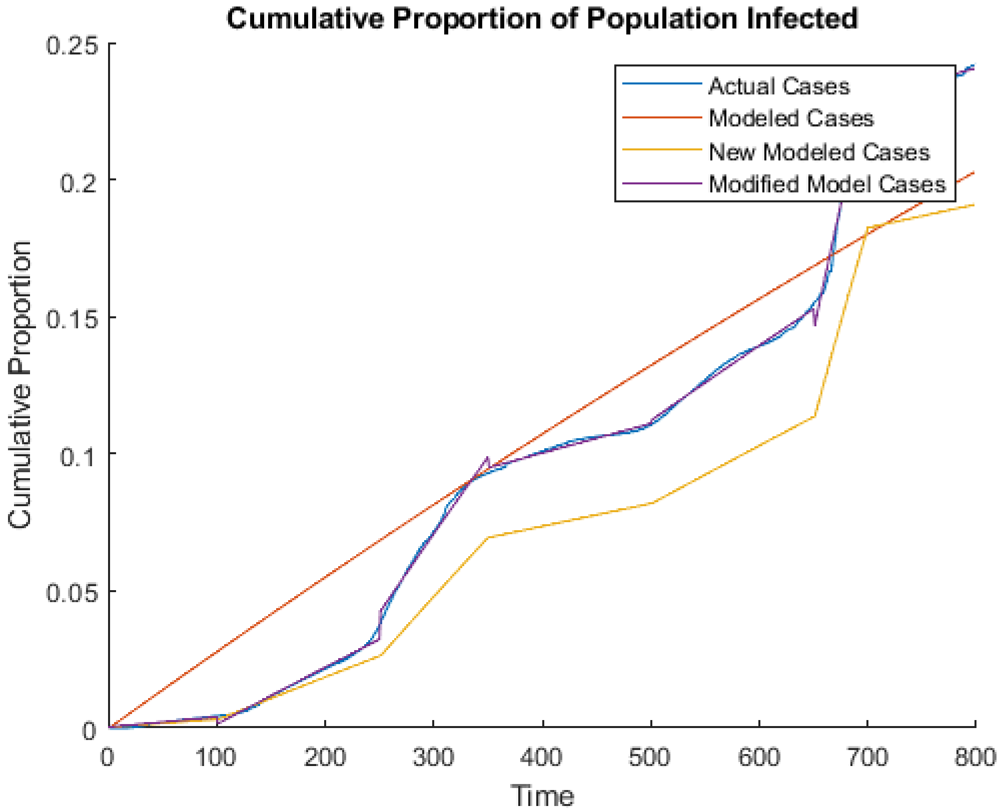

Actual Cases versus Modeled Cases
Actual Cases
Modeled Cases
Cumulative Proportion
Time
0
0.05
0.1
0.15
0.2
0.25
0
100
200
300
400
500
600
700
800

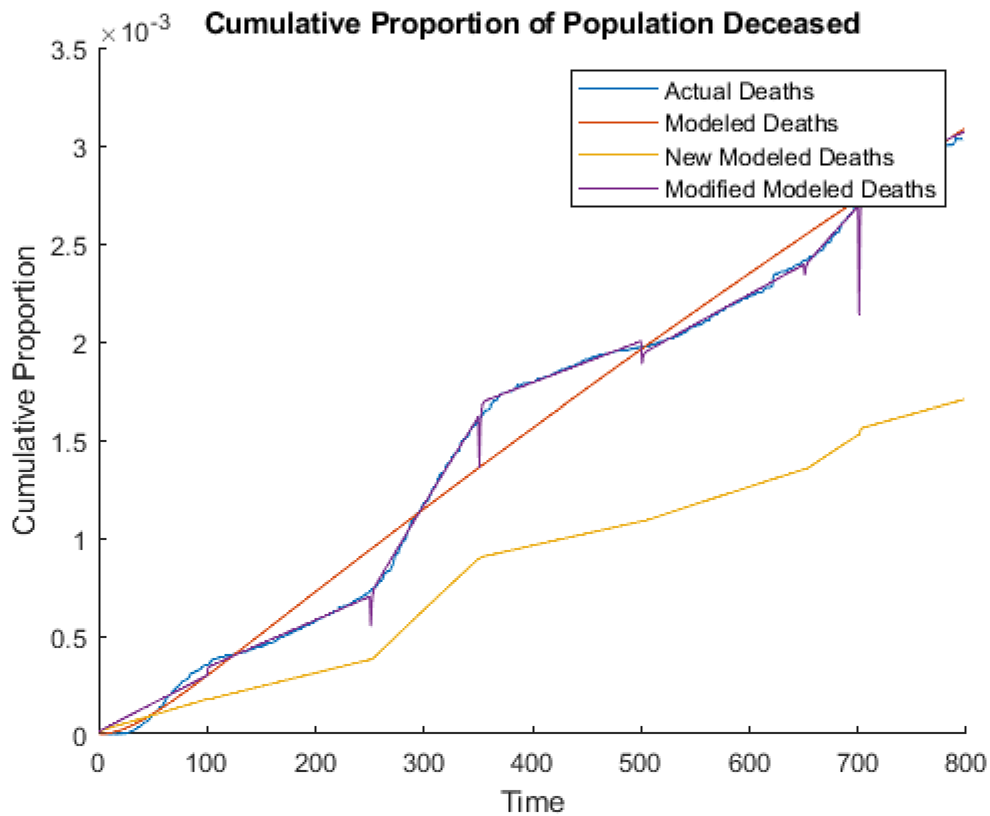


Proportion of Population Infected
Actual Cases
Modeled Cases
New Modeled Cases
Cumulative Proportion
0.25
0.2
0.15
0.1
0.05
0
0
100
200
300
400
500
600
700
800
Time



Proportion of Population Deceased
× 10^-3
Actual Deaths
Modeled Deaths
New Modeled Deaths
Cumulative Proportion
3.5
3
2.5
2
1.5
1
0.5
0
0
100
200
300
400
500
600
700
800
Time


Cumulative Proportion of Population Infected
Actual Cases
Modeled Cases
New Modeled Cases
Modified Model Cases
Cumulative Proportion
Time
0.25
0.2
0.15
0.1
0.05
0
100
200
300
400
500
600
700
800


Cumulative Proportion of Population Deceased
$\times 10^{-3}$
Actual Deaths
Modeled Deaths
New Modeled Deaths
Modified Modeled Deaths
Cumulative Proportion
Time
3.5
3
2.5
2
1.5
1
0.5
0
100
200
300
400
500
600
700
800

Published with MATLAB® R2022a