Table of Contents


```
% Here is an example that reads in infection and fatalities from STL
 City
% and loads them into a new matrix covidstlcity_full
% In addition to this, you have other matrices for the other two
 regions in question

%covidstlcity_full = double(table2array(COVID_STLcity(:,
[5:6])))./300000;

stl = COVID_MO(string(COVID_MO.name) == 'St. Louis', :);
springfield = COVID_MO(string(COVID_MO.name) == 'Springfield', :);
jefferon = COVID_MO(string(COVID_MO.name) == 'Jefferson City', :);
```

Dates of interest for STL

```
stl_period1 =
 stl(isbetween(stl.date,datetime('2020-07-01'),datetime('2020-10-31')), :);
stl_period2 =
 stl(isbetween(stl.date,datetime('2020-10-31'),datetime('2021-02-03')), :);
stl_period3 =
 stl(isbetween(stl.date,datetime('2021-07-14'),datetime('2021-10-06')), :);

stl_period1_array = double(table2array(stl_period1(:,
[3:4])))./2805473;
stl_period2_array = double(table2array(stl_period2(:,
[3:4])))./2805473;
stl_period3_array = double(table2array(stl_period3(:,
[3:4])))./2805473;
```

Dates of interest for springfield

```
spring_period1 =
 springfield(isbetween(springfield.date,datetime('2020-11-02'),datetime('2021-01-3
spring_period2 =
 springfield(isbetween(springfield.date,datetime('2021-06-08'),datetime('2021-09-1
```

```
spring_period1_array = double(table2array(spring_period1(:,
[3:4]))./475220;
spring_period2_array = double(table2array(spring_period2(:,
[3:4]))./475220;
```

Dates of interest for Jefferson

```
jeff_period1 =
 jefferon(isbetween(jefferon.date,datetime('2020-11-02'),datetime('2021-01-31')),
jeff_period2 =
 jefferon(isbetween(jefferon.date,datetime('2021-06-08'),datetime('2021-09-15')),

jeff_period1_array = double(table2array(jeff_period1(:,
[3:4]))./150198;
jeff_period2_array = double(table2array(jeff_period2(:,
[3:4]))./150198;
```

Data for the entire range

```
stl_all = double(table2array(stl(:,[3:4])))./2805473;
springfield_all = double(table2array(springfield(:,[3:4])))./475220;
jefferson_all = double(table2array(jefferon(:,[3:4])))./150198;
```

Select the Data to be fit

```
coviddata = jefferson_all ;
period = jefferon;
t = height(coviddata);
dates =  table2array(period(:,1));

sirafun= @(x)sirloutput(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 = [];
b = [];
```

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.

```
%form of x = [new_infections, continued infections, fatalities,
 recovery with immunity, initial S, intial I, initial R, initial D]
Af = [1 1 1 1 1 1 1 0 0 0 0 0; 0 0 0 0 0 0 0 1 1 1 1 1];
bf = [1; 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 = [0.30, .995, 0.1, 1, 0.5, 0.5, 1, 1, 0.7, 1, 0.2, 0.1]';
lb = [0, 0, 0, 0, 0, 0, 0,.90, 0, 0, 0, 0]';

% Specify some initial parameters for the optimizer to start from
% form of x = [new_infections, continued infections, fatalities,
 recovery with immunity, initial S, intial I, initial R, initial D]

x0 = [0.05,0.85,0.01,0.1,0,0,0.04,1,0,0,0,0];

% 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);

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

Y_fit = sirloutput_full(x,t);
```

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.

sys_sir_base =

A =

	x1	x2	x3	x4	x5
x1	0.955	0.9562	0.3877	0	0
x2	0.04351	0.04379	0	0	0
x3	0.001508	0	0.4611	0	0
x4	0	0	0.04447	1	0
x5	0	0	0.0179	0	1

B =

	u1
x1	0
x2	0
x3	0
x4	0
x5	0

C =

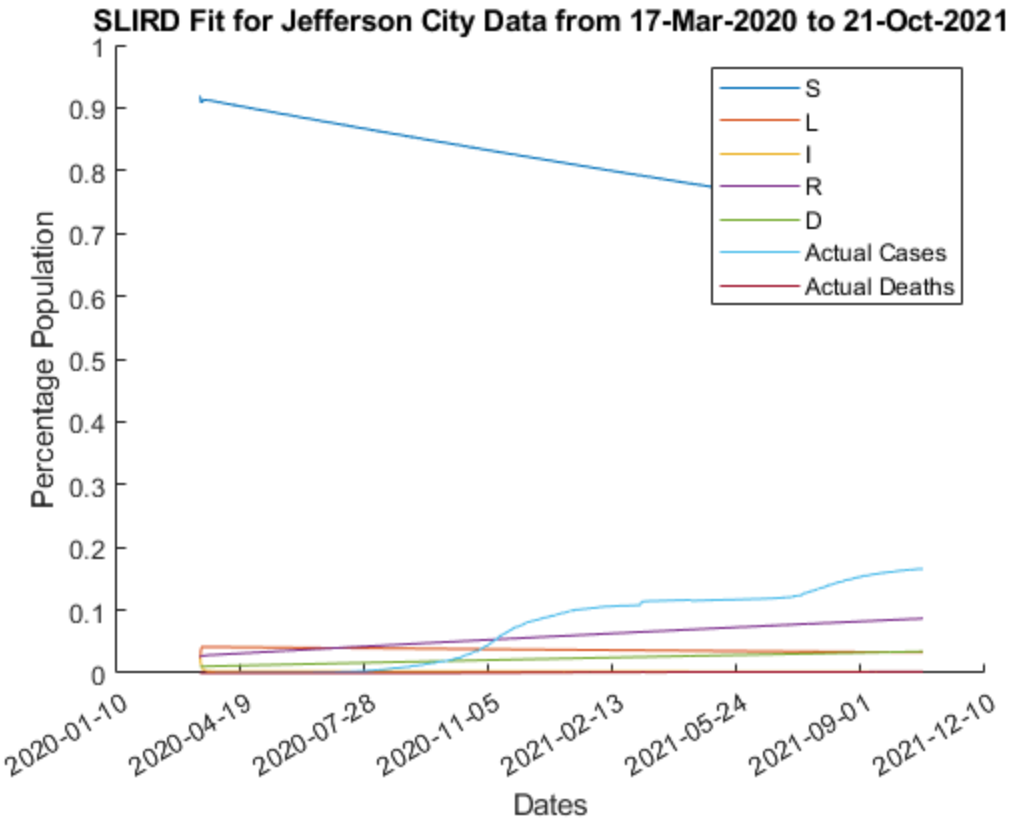
```
       x1  x2  x3  x4  x5
  y1    1   0   0   0   0
  y2    0   1   0   0   0
  y3    0   0   1   0   0
  y4    0   0   0   1   0
  y5    0   0   0   0   1

  D =
       u1
  y1    0
  y2    0
  y3    0
  y4    0
  y5    0

Sample time: 1 seconds
Discrete-time state-space model.
```

Plots were created here

```
% Test Plot over all st louis data
figure();
hold on;
plot(datenum(dates),Y_fit);
plot(datenum(dates),coviddata);
datetick('x', 'yyyy-mm-dd','keepticks');
hold off;
legend('S','L','I','R','D','Actual Cases', 'Actual Deaths');
xlabel('Dates')
ylabel('Percentage Population')
title("SLIRD Fit for Jefferson City Data from " + datestr(dates(1))
 + " to " + datestr(dates(length(dates))))
```



Published with MATLAB® R2021a