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
```

```
load('COVIDdata.mat');
COVID_springfield = COVID_MO(COVID_MO.name == "Springfield",:);
covidspringfield_full = double(table2array(COVID_springfield(301:589,
[3:4]))); %./2805473;

coviddata = covidspringfield_full;
t = length(coviddata);
```

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.

```
sirafun2= @(x)sliroutput(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 = [1 1 1 1 0 0 0 0 0;
     0 1 1 0 0 0 0 0 0];
b = [1, 0.5];
```

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 0 1 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 .7  .7  .3];
```

```
lb = [.015 0.01 .3 .3 .3 .1 .01 .01 .01];

% Specify some initial parameters for the optimizer to start from
x0 = [.4 .03 .4 .2 .6 .4 0 0 0];

x = fmincon(sirafun2,x0,A,b,Af,bf,lb,ub)

Y_fit = sliroutput_full(x,t);

% Make some plots that illustrate your findings.

cumlsum = cumsum(Y_fit);

figure();
cumlsumFinal = cumlsum(: , [4,5]).*7;
hold on;
plot(coviddata./475220);
split = ((cumlsumFinal./2805473)*100)/3; % splitting the data to apply
 a manual fit to the data.
plot(1:t, (split(:,1) + .075));
plot(1:t, split(:,2));

%Plot labling:
legend("i-real", "d-real","i","d");
xlabel("Days");
ylabel("Fraction Population");
title("Springfield SLIRD Model");
hold off
```

```
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.


x =

  Columns 1 through 7

    0.0150    0.0100    0.4900    0.3000    0.3000    0.1000    0.0100

  Columns 8 through 9

    0.5800    0.0100


sys_sir_base =

  A =
              x1         x2         x3         x4         x5
   x1      0.385        0.4          0          0          0
   x2        0.3        0.6        0.5          0          0
```

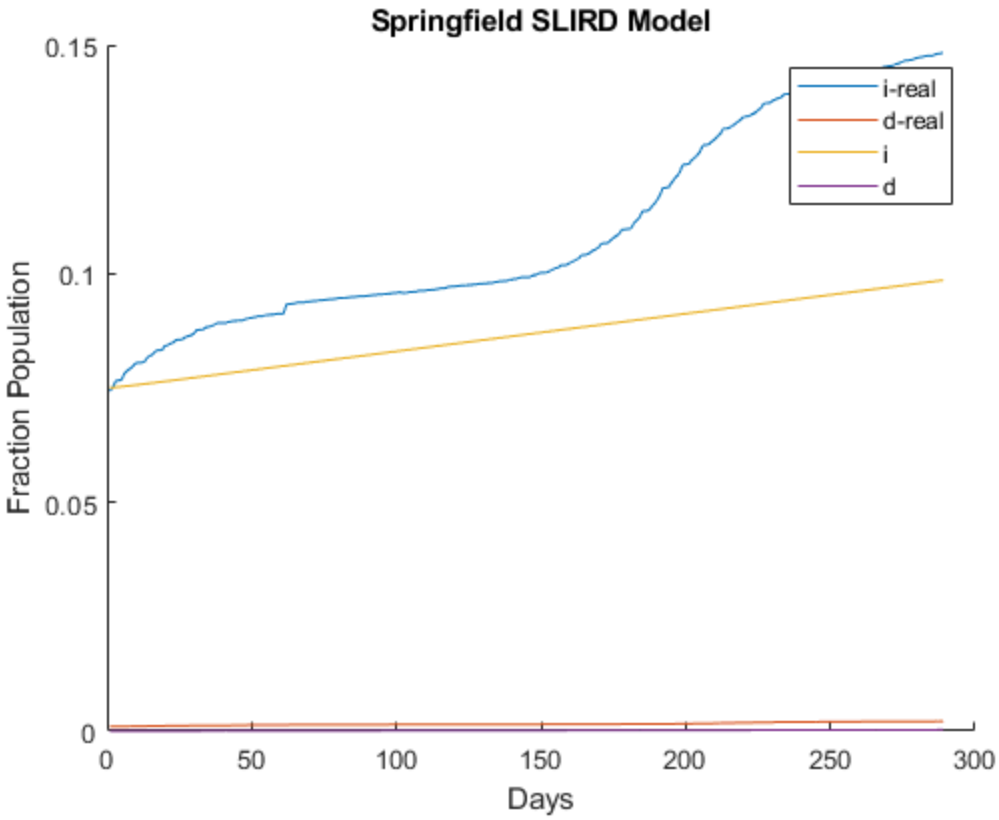
```
 x3      0.015           0  4.553e-07          0          0
 x4        0.3           0       0.49          1          0
 x5          0           0       0.01          0          1

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.
```



Published with MATLAB® R2021a