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_STLcity = COVID_MO(COVID_MO.name == "St. Louis",:);
covidstlcity_full = double(table2array(COVID_STLcity(1:100,
[3:4])));%./2805473;

coviddata = covidstlcity_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.

```
sirafun= @(x)siroutput(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 1 0 0 0 0 0;
    0 0 1 0 0 0 0;
    1 0 0 0 0 0 0;
    0 0 0 0 0 1 0;
    0 0 0 0 0 0 1;
    0 0 0 1 0 0 0 ];
b = [.16, 0.92, 0.7, 0.85, 0.15, 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   .7   .7  .3]';

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

% Specify some initial parameters for the optimizer to start from
x0 = [.2 0 0 1 0 0 0];

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

Y_fit = siroutput_full(x,t);


% Make some plots that illustrate your findings.

cumlsum = cumsum(Y_fit);

figure();

cumlsumFinal = cumlsum(: , [3,4]).*7; % extracting and scaling model
 data.
hold on;
plot(coviddata./2805473); % make numbers fractional & plot
plot(1:t, cumlsumFinal./2805473); % make numbers fractional & plot
%Plot labling:
legend("i-real", "d-real","i","d");
xlabel("Days");
ylabel("Fraction Population");
title("SIRD Model");
hold off;


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.


x =

    0.0150    0.0100    0.9200    0.2800    0.7000    0.0100    0.0100


sys_sir_base =

  A =
```

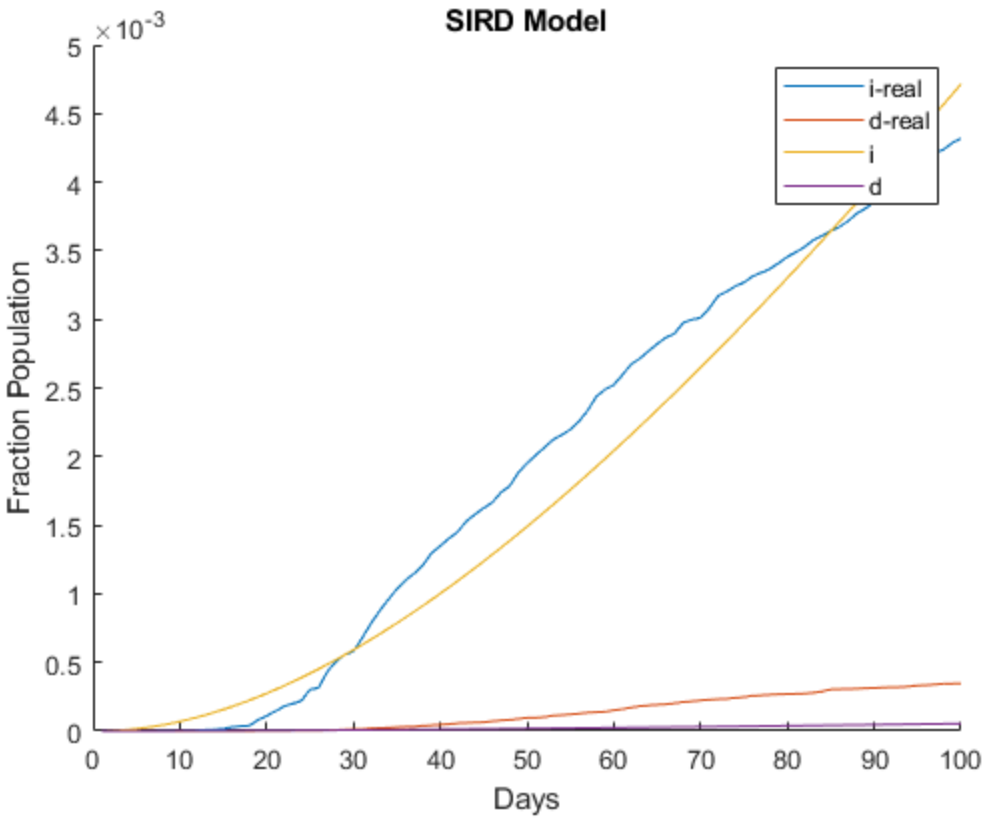
```
          x1      x2      x3      x4
 x1   0.985       0       0       0
 x2   0.015   0.975       0       0
 x3       0    0.92       1       0
 x4       0    0.01       0       1

B =
      u1
 x1    0
 x2    0
 x3    0
 x4    0

C =
      x1  x2  x3  x4
 y1    1   0   0   0
 y2    0   1   0   0
 y3    0   0   1   0
 y4    0   0   0   1

D =
      u1
 y1    0
 y2    0
 y3    0
 y4    0

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



Published with MATLAB® R2021a