

---

## Table of Contents

Load covid data into tables .....	1
Set up rate and initial condition constraints .....	1
Set up some fixed constraints .....	2
Set up upper and lower bound constraints .....	2
Run model and plot the results for STL city .....	2
Run model and plot the results for STL county .....	3
Run model and plot the results for STL metropolitan area .....	5
Run model and plot the results for KC metropolitan area .....	7
Run model and plot the results for remaining MO .....	8

## Load covid data into tables

```
clear;
load('COVIDdata.mat');

% STL city
covidstlcity_full = double(table2array(COVID_STLcity(:,[5:6])));

% STL county
covidstlcnty_full = double(table2array(COVID_STLcnty(:,[5:6])));

% STL metropolitan area
covidstlmetro_full = double(table2array(COVID_STLmetro(:,[5:6])));
% covidstlmetro_1 = double(table2array(COVID_STLmetro(1:118,[5:6])));
% covidstlmetro_2 = double(table2array(COVID_STLmetro(119:217,
[5:6])));
% covidstlmetro_3 = double(table2array(COVID_STLmetro(218:240,
[5:6])));

% KC metropolitan area
covidkcmetro_full = double(table2array(COVID_KCmetro(:,[5:6])));
% covidkcmetro_1 = double(table2array(COVID_KCmetro(1:110,[5:6])));
% covidkcmetro_2 = double(table2array(COVID_KCmetro(111:240,[5:6])));

% Remaining Missouri
covidremainmo_full = double(table2array(COVID_remainMO(:,[5:6])));
% covidremainmo_1 = double(table2array(COVID_remainMO(1:100,[5:6])));
% covidremainmo_2 = double(table2array(COVID_remainMO(101:180,
[5:6])));
% covidremainmo_3 = double(table2array(COVID_remainMO(181:235,
[5:6])));
```

## 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  $Af \cdot x = bf$  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

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

```
% Initial parameters for the optimizer to start from  
x0 = [0.003, 0.004, 0.66, 1, 0, 0, 0];
```

## Run model and plot the results for STL city

STL city data

```
coviddata = covidstlcity_full;  
t = length(coviddata(:, 1));  
pop = (STLcityPop)*100000;
```

```
% Set sirafun as the function siroutput with t and coviddata specified  
sirafun = @(x) siroutput(x, t, (coviddata/pop));
```

```
% Optimize model parameters to fit the data and run the model  
x = fmincon(sirafun, x0, A, b, Af, bf, lb, ub);
```

```
% Model results  
Y_fit = siroutput_full(x, t)*pop;  
casesModel = (pop - Y_fit(:, 1));  
deathsModel = Y_fit(:, 4);
```

```
% Plot the model results
```

```
figure;  
hold on
```

```
plot(casesModel, 'linewidth', 2);  
plot(deathsModel, 'linewidth', 2);  
plot(coviddata, ':', 'linewidth', 2);
```

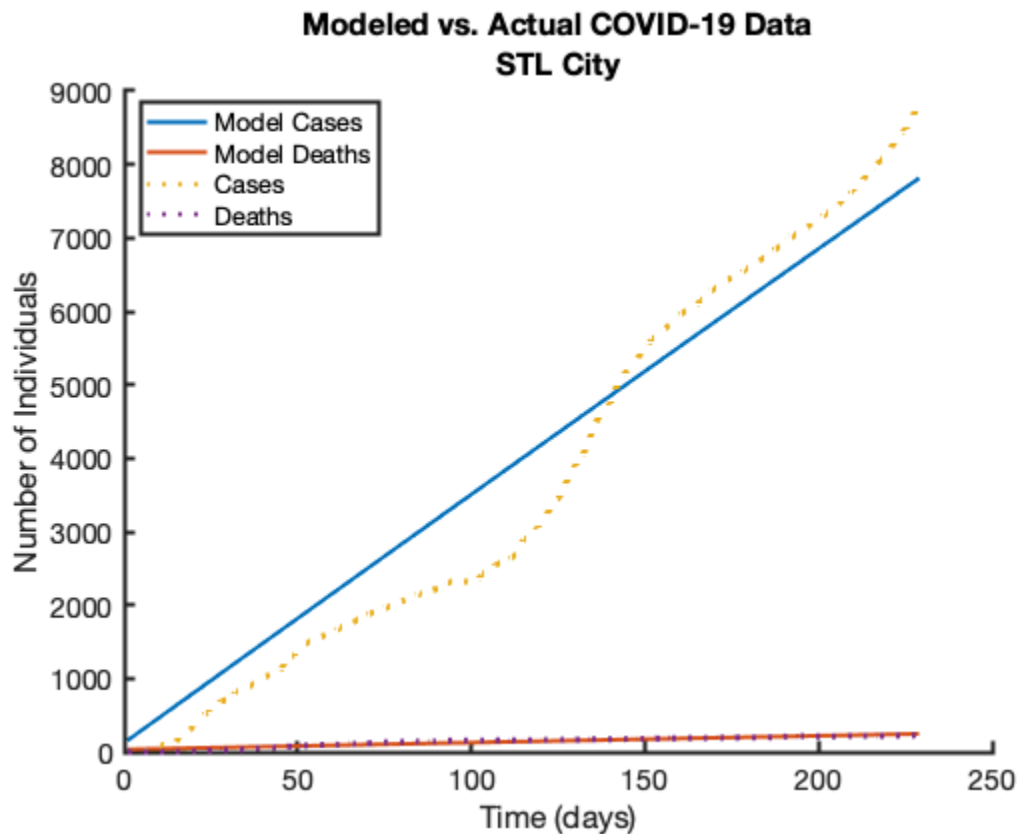
```
set(gca, 'linewidth', 2);  
set(gca, 'fontsize', 14);  
title(["Modeled vs. Actual COVID-19 Data"; "STL City"]);  
xlabel("Time (days)");  
ylabel("Number of Individuals");  
legend("Model Cases", "Model Deaths", "Cases", "Deaths", ...  
      'location', 'northwest');
```

---

```
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.*



## Run model and plot the results for STL county

STL county data

```
coviddata = covidstlcnty_full;  
t = length(coviddata(:, 1));  
pop = (STLcntyPop)*100000;  
  
% Set sirafun as the function siroutput with t and coviddata specified  
sirafun= @(x)siroutput(x,t,(coviddata/pop));  
  
% Optimize model parameters to fit the data and run the model  
x = fmincon(sirafun,x0,A,b,Af,bf,lb,ub);  
  
% Model results  
Y_fit = siroutput_full(x,t)*pop;
```

---

```
casesModel = (pop - Y_fit(:, 1));
deathsModel = Y_fit(:, 4);

% Plot the model results
figure;
hold on

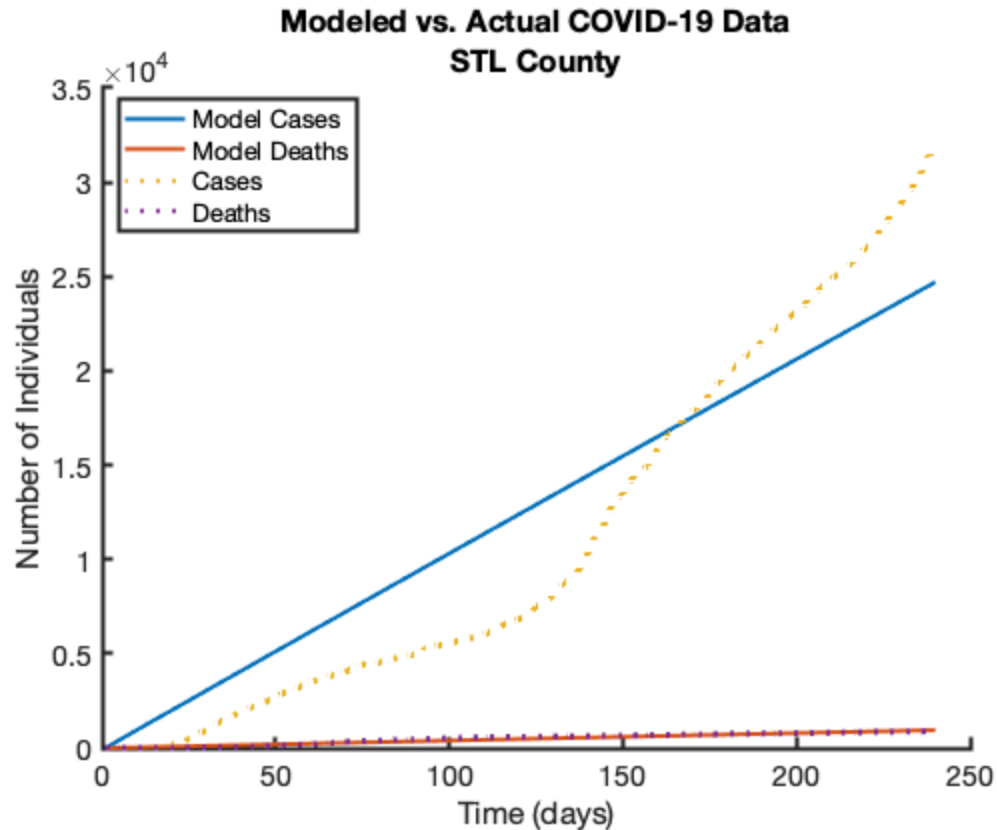
plot(casesModel, 'linewidth', 2);
plot(deathsModel, 'linewidth', 2);
plot(coviddata, ':', 'linewidth', 2);

set(gca, 'linewidth', 2);
set(gca, 'fontsize', 14);
title(["Modeled vs. Actual COVID-19 Data"; "STL County"]);
xlabel("Time (days)");
ylabel("Number of Individuals");
legend("Model Cases", "Model Deaths", "Cases", "Deaths", ...
       'location', 'northwest');

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.*



## Run model and plot the results for STL metropolitan area

STL metropolitan area data

```
coviddata = covidstlmetro_full;
t = length(coviddata(:, 1));
pop = (STLmetroPop)*100000;

% Set sirafun as the function siroutput with t and coviddata specified
sirafun= @(x)siroutput(x,t,(coviddata/pop));

% Optimize model parameters to fit the data and run the model
x = fmincon(sirafun,x0,A,b,Af,bf,lb,ub);

% Model results
Y_fit = siroutput_full(x,t)*pop;
casesModel = (pop - Y_fit(:, 1));
deathsModel = Y_fit(:, 4);

% Plot the model results
figure;
hold on

plot(casesModel, 'linewidth', 2);
```

---

```

plot(deathsModel, 'linewidth', 2);
plot(coviddata, ':', 'linewidth', 2);

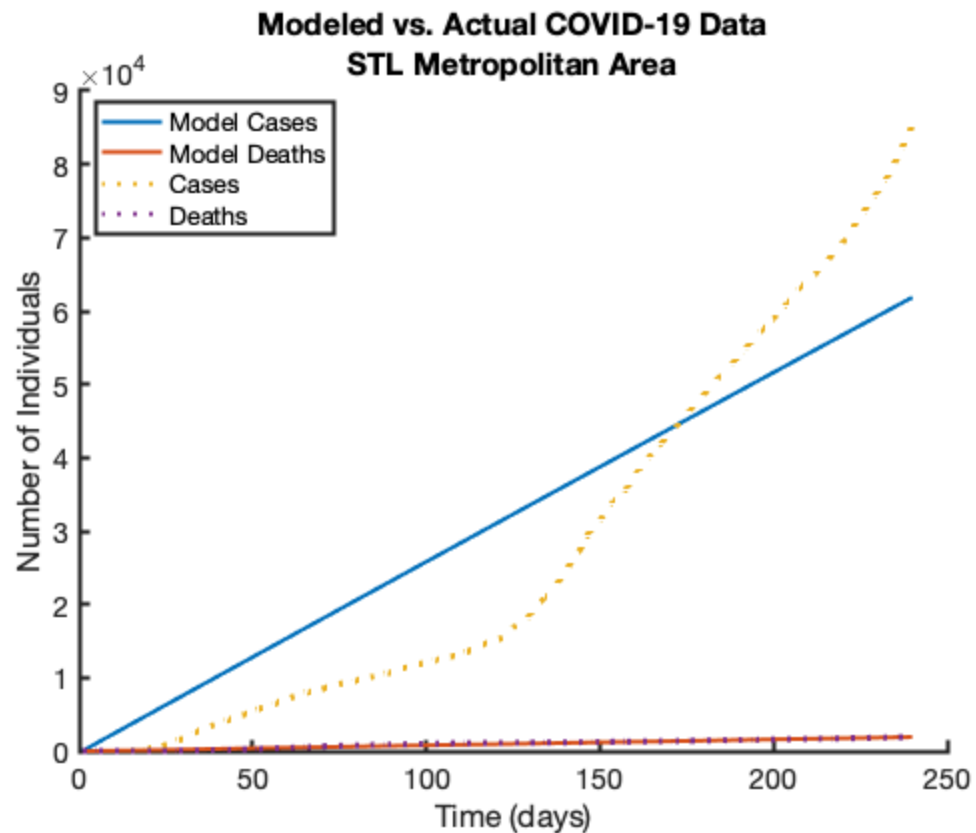
set(gca, 'linewidth', 2);
set(gca, 'fontsize', 14);
title(["Modeled vs. Actual COVID-19 Data"; "STL Metropolitan Area"]);
xlabel("Time (days)");
ylabel("Number of Individuals");
legend("Model Cases", "Model Deaths", "Cases", "Deaths",...
       'location', 'northwest');

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.*



---

# Run model and plot the results for KC metropolitan area

STL metropolitan area data

```
coviddata = covidkcmetro_full;
t = length(coviddata(:, 1));
pop = (KCmetroPop)*100000;

% Set sirafun as the function siroutput with t and coviddata specified
sirafun= @(x)siroutput(x,t,(coviddata/pop));

% Optimize model parameters to fit the data and run the model
x = fmincon(sirafun,x0,A,b,Af,bf,lb,ub);

% Model results
Y_fit = siroutput_full(x,t)*pop;
casesModel = (pop - Y_fit(:, 1));
deathsModel = Y_fit(:, 4);

% Plot the model results
figure;
hold on

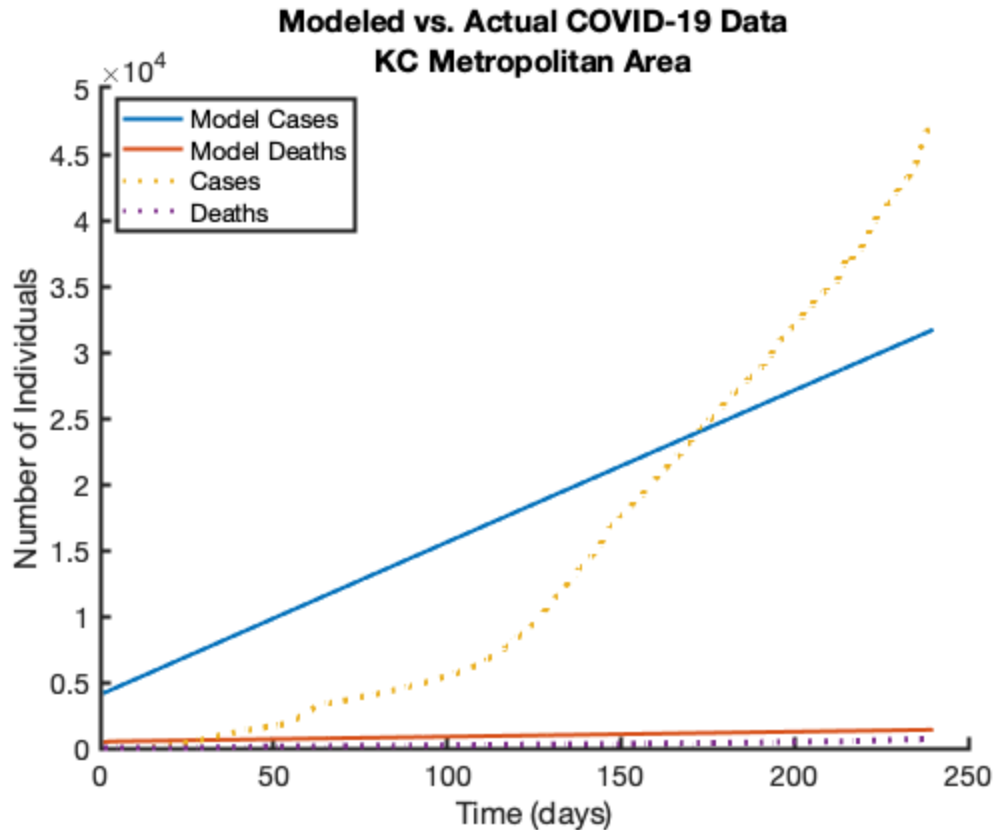
plot(casesModel, 'linewidth', 2);
plot(deathsModel, 'linewidth', 2);
plot(coviddata, ':', 'linewidth', 2);

set(gca, 'linewidth', 2);
set(gca, 'fontsize', 14);
title(["Modeled vs. Actual COVID-19 Data"; "KC Metropolitan Area"]);
xlabel("Time (days)");
ylabel("Number of Individuals");
legend("Model Cases", "Model Deaths", "Cases", "Deaths",...
       'location', 'northwest');

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.*



## Run model and plot the results for remaining MO

Remaining MO data

```
coviddata = covidremainmo_full;
t = length(coviddata(:, 1));
pop = (remainMOPop)*100000;

% Set sirafun as the function siroutput with t and coviddata specified
sirafun = @(x)siroutput(x,t,(coviddata/pop));

% Optimize model parameters to fit the data and run the model
x = fmincon(sirafun,x0,A,b,Af,bf,lb,ub);

% Model results
Y_fit = siroutput_full(x,t)*pop;
casesModel = (pop - Y_fit(:, 1));
deathsModel = Y_fit(:, 4);

% Plot the model results
figure;
hold on

plot(casesModel, 'linewidth', 2);
```



---

```

plot(deathsModel, 'linewidth', 2);
plot(coviddata, ':', 'linewidth', 2);

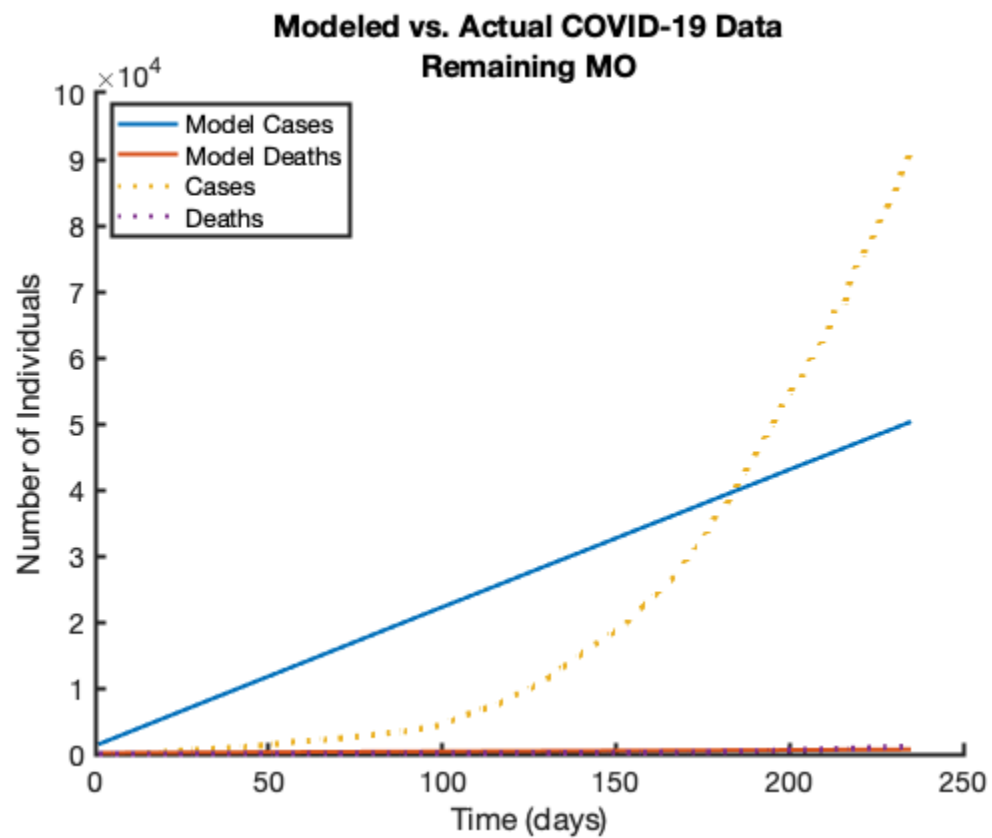
set(gca, 'linewidth', 2);
set(gca, 'fontsize', 14);
title(["Modeled vs. Actual COVID-19 Data"; "Remaining MO"]);
xlabel("Time (days)");
ylabel("Number of Individuals");
legend("Model Cases", "Model Deaths", "Cases", "Deaths",...
       'location', 'northwest');

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.*



*Published with MATLAB® R2020a*