BENG 249 Project Report

Evan Collins, Ellie Gabriel, Anmei Little

1 Introduction

We sought to model the COVID-19 pandemic using MATLAB. We were interested in a real-time simulation of people moving around as well as graphs, displaying the progress of infection in the population. We chose to make the geographic area proportionally resemble 1/10th the size of Yale's campus and population (consisting of students and faculty). Some assumptions that we made in this model include: 1) The total population is constant (i.e. no entry, exit, births, or deaths) 2) Recovered individuals are immune to reinfection 3) Individuals who are designated as infected and asymptomatic cannot become symptomatic over time 4) People randomly walk around campus and do not congregate in particular areas 6) After 14 days, infected individuals either recover or die 7) All individuals are healthy at the start and have equal chances of being infected. Our model incorporates six groups of individuals: Uninfected (U), Infected Symptomatic (IS), Hospitalized infected symptomatic (H), Infected Asymptomatic (IA), Recovered (R), and Dead (D).

2 Background

The simulation has axis limits from 0 to 100 in both the x and y directions. The units for x and y are meters, such that the simulation takes place in a 10,000 square meter area. Yale's campus is approximately 640,000 square meters and contains approximately 10,500 people. Since we simplified the area to be 10,000 square meters, 64% of the actual campus size, we used a total population of 165 people (64% of 10,500 ~ 165). We also added a hospital area where only people who have been IS for delayIS number of days enter before recovering or dying.

```
% Creating rectangular boundary
figure;
A = 100^2;
w = sqrt(A);
h = w;
rectangle('Position', [0 0 w h]);
axis equal;
close all;
```

To be consistent with units, we used a critical distance in meters as well. The critical distance we used was 3 meters. Within this distance, there is a chance of infection equal to the infection rate. The infection rate we begin the simulation with is 0.112 ((Bi, Wu, et.al, 2020)), but this rate is decreased as we "implement mitigation strategies," such as requiring people to wear masks, social distance, check their temperatures, etc.. The time scale that the simulation loops over is in hours.

```
tot_time = 1680;%70 days
t = 0:1:tot_time; %time scale for 100 days = 2400 hours
dt = 1;
dim = 2; %x and y dimensions for 2D migration
d_crit_sq = 9;
```

```
tot_pop = 165;
D = 0.50; %diffusion coefficient
infectionrate = .112; %actual statistic on the chance of becoming infected
when in proximity of an infected individual
delayIS = 120; %5 day delay in IS quarantining
delayIA = 336; %IA quarantining --> delay decreases
```

Based on a study in March 2020 by a group at Kyoto University, the percentage of infected people who are asymptomatic is 30.25%, while the percentage of infected people who are symptomatic is 69.75%. We used these values when determining which group to assign newly infected people.

```
%interactions during each timepoint between U and IS
      currentidentity = identitymat(i+1, :); %takes all the group identities
      [IS] = find(currentidentity == 2); %Indices of individuals who are IS
      [U] = find(currentidentity == 1); %Indices of individuals who are U
      if(isempty(IS))
      IS = 0;
      else
      for j2 = 1:length(IS) %looping through all the IS individuals
            e = 1; %initialize e
            removeIndices = zeros(1, tot_pop);
            for j4 = 1:length(U) %looping through all U individuals
               if (posmat(i+1, IS(j2),1) - posmat(i+1, U(j4), 1))^2 +
               (posmat(i+1, IS(j2), 2)- posmat(i+1, U(j4), 2))^2 <= d_crit_sq
                 %results of interaction because distance is within 3 meters
                  if rand(1) < infectionrate %U will become infected
                  if rand(1) < 0.6975 %U will become IS
                        identitymat(i+1,U(j4)) = 2;
                     else %U will become IA
                        identitymat(i+1,U(j4)) = 3;
                  end
                  removeIndices(e) = j4; %keep track of who changed from U
                  e = e + 1;
                  recoverymat(i+1, U(j4)) = 1; %newly infected individuals get
                                               1 day of infection
                  end
            end %end the if statement
            end %end the for j4 loop
      [U] = remove_funx2(U,removeIndices); %remove U from list
      end %end the for j2 loop
      end %ends the if else statement
```

```
**separate code chunk for interactions between IA and U not shown here.
```

Based on World-o-Meters statistics on April 20th, 2020, the mortality rate of infected symptomatic people is 0.36 and the recovery rate of infected symptomatic people is 0.64. These values are reflected in our code.

```
%for individuals who have been infected for 14 days--> either recover or die
      if recoverymat(i, j) >= 336 %14 days
            if identitymat(i, j) == 2 %if person was IS
                  if rand(1) < 0.64 %chance of recovering
                  identitymat(i+1, j) = 4; %they recover and become R group
                  else
                  identitymat(i+1, j) = 5; %they die and become D group
                  aliveCount = aliveCount - 1; %calculating total alive
                  deadCount = deadCount + 1; %calculating total dead
                  end
            else %if person was IA
                  identitymat(i+1,j) = 4; %all recover and become R group
            end
            recoverymat(i+1, j) = 0; %return recovery count to 0
        end
  end
  alivevec(i+1) = aliveCount;
  deadvec(i+1) = deadCount;
```

3 Code

3. 1 People and Movement

People in our simulation fall into 1 of 6 categories: Uninfected (U), Infected Asymptomatic (IA), Infected Symptomatic (IS), Hospitalized infected symptomatic (H), Recovered (R), or Dead (D). At the start, the population is all uninfected except for one person who is infected asymptomatic. Asymptomatic people are "carriers" and are the most dangerous. They walk around the simulation, spreading the disease, but they themselves never die from the infection. People who are IA become immobile after delayIA number of days, remaining infectious but infecting less people as they "self isolate" to recovery. People who are infected and show symptoms, IS, move around for a few days (delayIS), spreading the disease, before being sent to the hospital. After 14 days from initial infection, IS people either die and join the D group or recover and join the R group. People in the recovered group are immune to reinfection and move around freely. Dead people are removed from the simulation. We keep track of identities with the identitymat matrix, where numbers 1, 2, 2.5, 3, 4, 5 code for the 6 categories of people, respectively.

```
identitymat(1, :) = 1; %everyone is uninfected at first time point
identitymat(1, 1:3) = 3; %except three person, patient zeros, who are infected
asymptomatic (IA)
recoverymat(1, 1:3) = 1; %patient zeroes start out infected for one day

%random starting positions at first time point for all individuals
for j  = 1:tot_pop
     posmat(1, j, 1) = 100.*rand(1); %2 random numbers between 0 and 100
     posmat(1, j, 2) = 100.*rand(1);
end
```

```
for i = 1:length(t)-1
    aliveCount = alivevec(i); %number of previously alive people
    deadCount = deadvec(i); %number of previously dead people
    for j = 1:tot_pop
        %making general rule that individuals stay in the same group
        identitymat(i+1, j) = identitymat(i, j);

        %movement during each timepoint
        for k = 1:dim
          posmat(i+1, j , k) = posmat(i, j, k) + (2*D*dt)^0.5*randn(1);
            if (recoverymat(i, j) >= delayIS) && ((identitymat(i, j) == 2) ||
               (identitymat(i, j) == 2.5))  %if a certain individual j at a
                certain time point i is IS and needs to go to hospital
                posmat(i+1, j , 1) = (120-100)*rand(1)+100; %isolation in
                                %hospital x limits are 100 to 120
                posmat(i+1, j , 2) = 100*rand(1); %isolation in hospital y
                                %limits are 0 to 100
                identitymat(i+1,j) = 2.5; %assigned to "Hospitalized" group
            elseif (recoverymat(i, j) >= delayIA) && (identitymat(i, j) == 3)
                %IA stop moving after delayIA hours
                posmat(i+1, j , k) = posmat(i, j, k); %no movement
            elseif (identitymat(i, j) == 4) && (posmat(i+1, j, 1) <= 0 ||
                posmat(i+1, j, 1) >= 100 || posmat(i+1, j, 2) <= 0 ||
                posmat(i+1, j, 2) >= 100) %recovered return to general pop
                posmat(i+1, j, 1) = 100.*rand(1);
                posmat(i+1, j, 2) = 100.*rand(1);
            elseif posmat(i+1, j, 1) <= 0 || posmat(i+1, j, 1) >= 100 ||
                posmat(i+1, j, 2) <= 0 || posmat(i+1, j, 2) >= 100
                                                  %rectangular boundary
                posmat(i+1, j , k) = posmat(i, j, k); %no movement because at
                                                     % boundary
            else
                continue
            end
        end

        %generalizing that all currently infected ppl will stay infected
        if recoverymat(i, j) > 0
             recoverymat(i+1, j) = recoverymat(i,j) + 1; %increasing time
       end
        %for individuals who have been infected for 14 days (on average) -->
         either recover or die
        if recoverymat(i, j) >= 336 %14 days; all IS are already in hospital
            if identitymat(i, j) == 2.5 %if person is IS
                if rand(1) < 0.64 %chance of recovering
                    identitymat(i+1, j) = 4; %they recover and become R group
                else
                    identitymat(i+1, j) = 5; %they die and become D group
                    aliveCount = aliveCount - 1; %calculating total alive
                    deadCount = deadCount + 1; %calculating total dead
                end
            else %if person was IA
                identitymat(i+1,j) = 4; %all IA recover and become R group
            end
            recoverymat(i+1, j) = 0; %return recovery count to 0 for the
        end
    end %end for loop over population
```

3. 2 Infections

The code loops around for each time point. At each time point, we loop through a list of infected people, calculating their distance to all of the uninfected people. If they are within the critical distance, a random number is generated to determine if an infection occurs, and if so, another random number determines if the person will be symptomatic or asymptomatic. This uninfected person joins a new group (IS or IA). Then, the distances around the next infected person are calculated. This continues until we find all the U people close to the infected people, and the new groups are assigned. One challenge faced while creating this model was making sure that the effect of one infected person on an uninfected person did not override the effect of another. Essentially, if an uninfected person becomes infected by the first person in the list of infectors, that uninfected person is automatically switched to its new group before the next loop. In this way, the distance between that person and the next infector in the list is not calculated. We wrote a remove_funx2 (see bottom of code) to remove the newly infected people from the U list in between loops over the infected people. This is a variation of the remove_funx written in class, but it removes a list of indices. The most challenging part of this code was ensuring that the indices of the list did not shift as some were removed; this would occur if one index was removed at a time, which is why we needed an entire new function.

```
function [new_list] = remove_funx2(list,ind)
% this function removes multiple elements from a list

new_list = zeros(1, length(list)-length(ind)); % initializes list with zeros
%ind is a list
j = 1;
n = 0;
for i = 1:length(list)
      if (i <= n) %to not have repeats, go to next iteration of i
         continue
      end
        for k = 1:length(ind)
           if i == ind(k)
              i = i + 1;
              n = i;
           end
        end
      if i > length(list)
         break
      else
       new_list(j) = list(i);
       j = j + 1;
      end
end %ends the for loop
end %ends the function
```

3.3 Animations and Plot

We decided that the best format for our animations was one with subplots; this way, the viewer can see both the simulation of Yale's population moving about along with the plot of population proportions simultaneously. Additionally, we plotted a static line graph of the raw number of individuals who are alive versus individuals who are dead.

3.3.1 Animations (See Figure 1)

```
%population proportion plot
prop_U = zeros(1, length(t));
prop_I = zeros(1, length(t));
prop_R = zeros(1, length(t));
prop_H = zeros(1, length(t));
prop_H = zeros(1, length(t));
finalprop_D = deadvec(tot_time)/tot_pop;

figure('units','normalized','outerposition',[0 0 1 1]); %makes full screen
subplot(2,1,1);
h = animatedline('Color','g');
h2 = animatedline('Color','r');
h3 = animatedline('Color','y');
h4 = animatedline('Color','c');
axis([0,1000,0,1])
xlabel('Time (Hours)');
ylabel('Percentage of Individuals');
title('Population Proportions');
Lgnd = legend('Uninfected', 'Infected', 'Recovered',
              'Hospitalized','Color', '#BEBEBE');
Lgnd.Position(1) = 0.01;
Lgnd.Position(2) = 0.5;
set(gcf, 'Position', [500 0 500 500]);
set(gca, 'color', 'k');

subplot(2,1,2);
```

```
dim = [.82 .2 .3 .3];
str = '  Hospital';
annotation('textbox',dim,'String',str,'FitBoxToText','on');

for s = 1:10:length(t)
      hold off;
      for p = 1:tot_pop
      rectangle('Position', [100 0 .1 100], 'EdgeColor', '#F0F0F0',
                'LineWidth', 5);

            if identitymat(s, p) == 1 %U individuals
                  plot(posmat(s, p, 1), posmat(s, p, 2), 'o', 'color', 'g');
                  hold on;

            elseif identitymat(s, p) == 2 %IS individuals
                  plot(posmat(s, p, 1), posmat(s, p, 2), 'o','color', 'c');
                  hold on;

            elseif identitymat(s, p) == 2.5 %IS hospitalized individuals
                  plot(posmat(s, p, 1), posmat(s, p, 2), 'o','color', 'y');
                  hold on;

            elseif identitymat(s, p) == 3 %IA individuals
                   plot(posmat(s, p, 1), posmat(s, p, 2), 'o', 'color', 'w');
                  hold on;

            elseif identitymat(s, p) == 4 %R individuals
                  plot(posmat(s, p, 1), posmat(s, p, 2), 'o', 'color', 'b');
               hold on;
            end
      end
      set(gca, 'color', 'k');
      xlabel('X axis of microcosm of Yale campus in meters');
      ylabel('Y axis of microcosm of Yale campus in meters');
      xlim([0 120]);
      ylim([0 100]);

      % Generate dummy info for plot handles
         p = zeros(5,1);
         p(1) = plot(1680, 'o', 'color', 'g');
         p(2) = plot(1680, 'o','color', 'c');
         p(3) = plot(1680, 'o','color', 'y');
         p(4) = plot(1680, 'o', 'color', 'w');
         p(5) = plot(1680, 'o', 'color', 'b');
         Lgnd = legend(p, 'Uninfected', 'Infected Symptomatic',
             'Hospitalized', 'Infected Asymptomatic', 'Recovered', 'Color',
             '#BEBEBE');
         Lgnd.Position(1) = 0.01;
         Lgnd.Position(2) = 0.25;
         title(s); %changes and is in hours
         pause(.1)

      currentidentity = identitymat(s, :);
      [U] = find(currentidentity == 1); %Indices of people who are U
      [IS] = find(currentidentity == 2); %Indices of people who are IS
      [IS_H] = find(currentidentity == 2.5); %Indices of people who are H
      [IA] = find(currentidentity == 3); %Indices of people who are IA
      [R] = find(currentidentity == 4); %Indices of people who are R
      sum_U = length(U); %number of U people at time s
      sum_I = length(IS) + length(IA); %number of Infected I people at time s
      sum_R = length(R); %number of R people at time s
      sum_H = length(IS_H); %number of Infected hospitalized people at time s
      prop_U(s) = sum_U/tot_pop; %proportion of U in the population at time s
      prop_I(s) = sum_I/tot_pop; %proportion of I in the population at time s
      prop_R(s) = sum_R/tot_pop; %proportion of R in the population at time s
      prop_H(s) = sum_H/tot_pop;
      addpoints(h,s, prop_U(s))
      addpoints(h2,s, prop_I(s))
      addpoints(h3,s, prop_R(s))
      addpoints(h4, s, prop_H(s))
      drawnow update
      pause(.1)

end
```

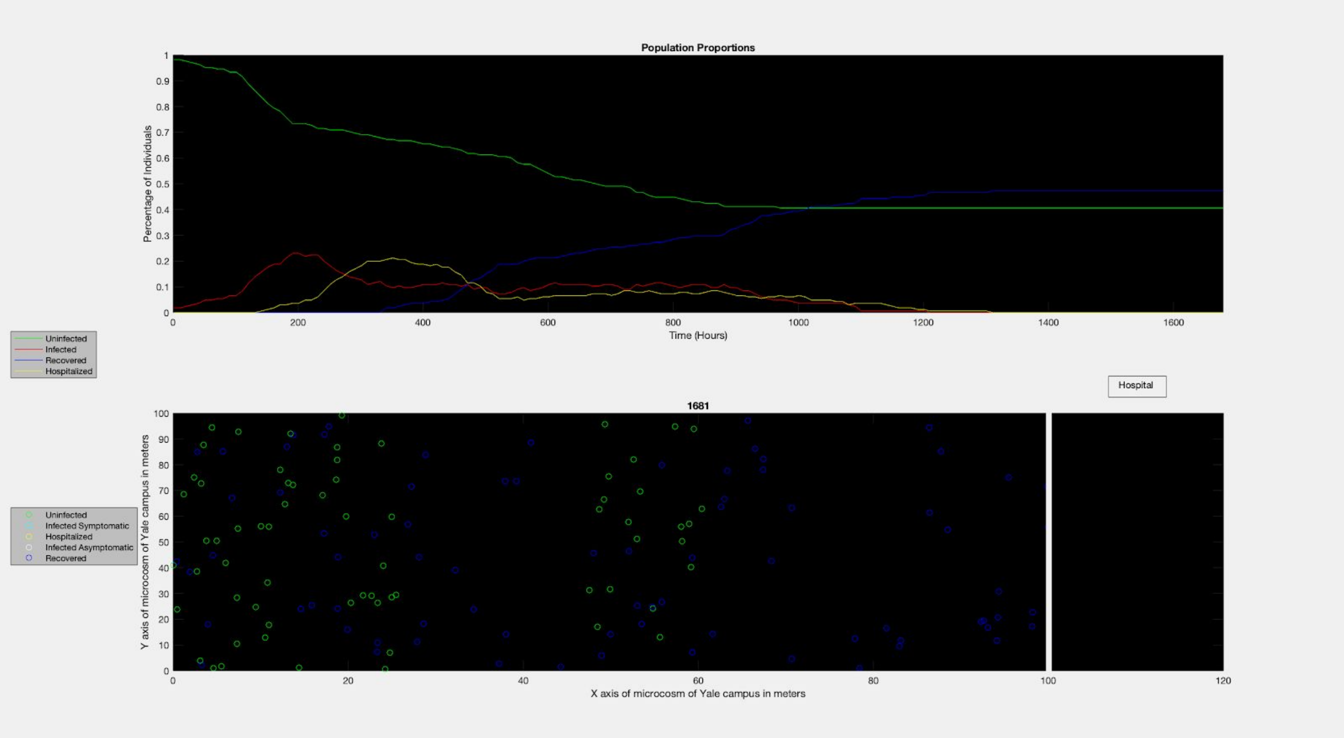


Figure 1. Animation of population proportions and real-time movement of individuals for one simulation.

3.3.2 Line Graph of Alive Versus Dead (See Figure 2)

```
%line plot of alive and dead individuals over time
Figure;
set(gca, 'color', 'k');
set(gcf, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]); % Enlarge figure
to full screen
hold all;
plot(1:dt:length(t), alivevec, 'g', 'LineWidth', 2); %plots number of people
alive over time (decreases)
plot(1:dt:length(t), deadvec, 'w', 'LineWidth', 2); %plots number of people
dead over time (increases)
legend('Number of Alive Individuals', 'Number of Dead Individuals',
'Location', 'northoutside', 'Color', '#BEBEBE');
title('Alive and Dead Counts Over Time')
xlabel('Time (Hours)')
ylabel('Number of Alive or Dead (Individuals)')
txt = num2str(deadvec(length(t)));
text(1000, deadvec(length(t)) + 10, txt);
txt2 = num2str(alivevec(length(t)));
text(1000, alivevec(length(t)) - 10, txt2);
axis([0,1000,0,tot_pop])
```

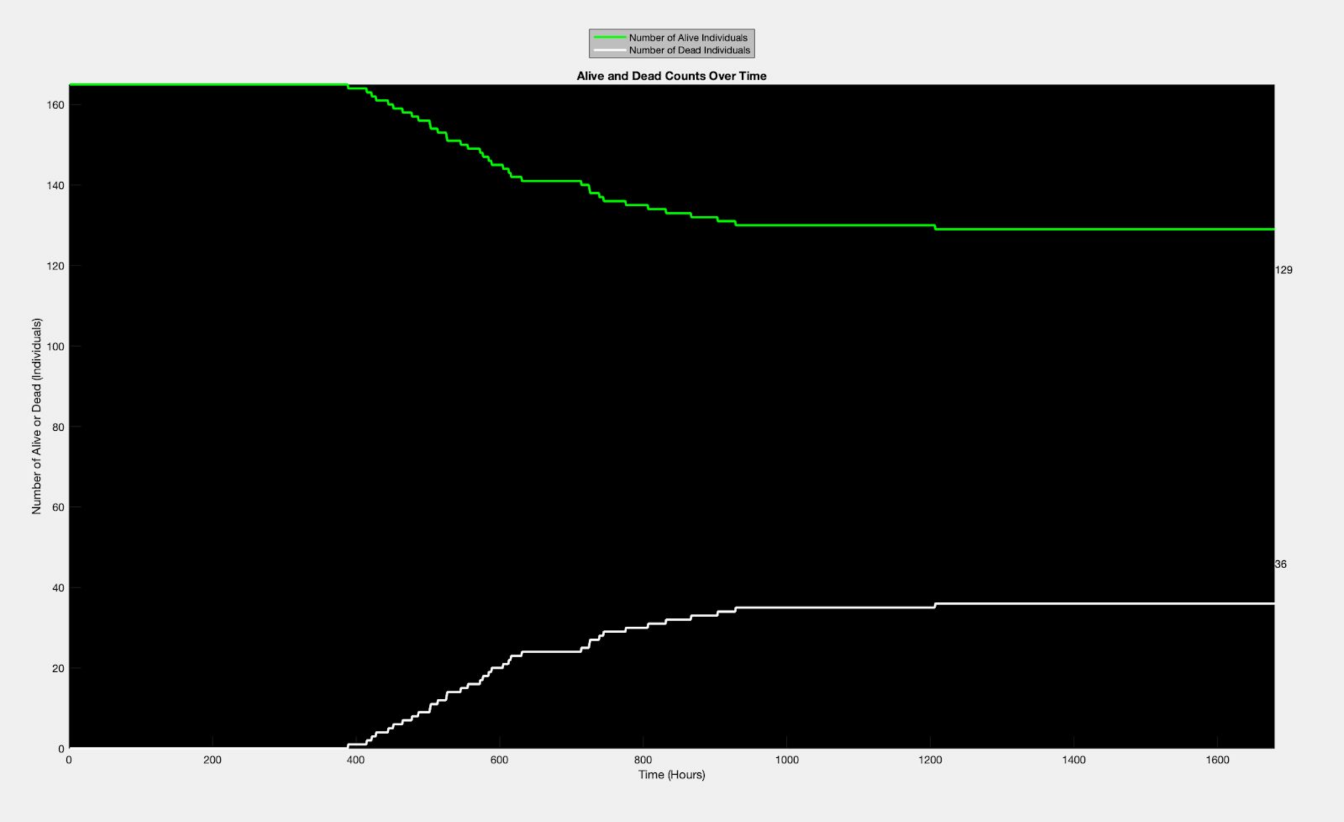


Figure 2. Alive versus dead raw counts over time for one simulation

3.4 Multi Func and Script

After ensuring that the code was running correctly, we created a function out of the original script (see all sections above for code). We then called this function multiple times in a new multi function. This allows for the calculation of group proportion averages over time, both in the original model and in the models after implementing mitigation strategies, that can be plotted.

```
function[averageU, averageI, averageR, averageH, finalU, finalI, finalR, 
finalH, finalD] = Covid_model_multi_func(tot_pop, tot_time, D, infectionrate, 
delayIS, delayIA, num_sims)

for j = 1:num_sims
    [t, prop_U(:, j), prop_I(:, j), prop_R(:, j), prop_H(:,j), finalprop_D(1, 
j)] = Covid_model_func(tot_pop, tot_time, D, infectionrate, delayIS, delayIA, 
1);
    close all;
end

averageU = zeros(1, tot_time);
averageI = zeros(1, tot_time);
averageR = zeros(1, tot_time);
averageH = zeros(1, tot_time);
```

```
for i = 1:length(t) %getting average proportions at each timepoint
    averageU(i) = mean(prop_U(i,:));
    averageI(i) = mean(prop_I(i,:));
    averageR(i) = mean(prop_R(i,:));
    averageH(i) = mean(prop_H(i,:));
end

%determining average proportions at the end of the time period (for pie chart)
finalU = averageU(tot_time);
finalI = averageI(tot_time);
finalR = averageR(tot_time);
finalH = averageH(tot_time);
finalD = mean(finalprop_D(1,:));

end
```

The script calls the original function once and then calls the multi function for 50 simulations. We only wanted the animation to appear once (not 50 times), so we added an input variable to the function called FLAG. When FLAG is one, the function does not run the animation, but when FLAG is zero, the function runs the animation. Therefore, in the script, when the function is called once, it is called with FLAG equal to zero, but when the multi function is called, FLAG is one as defined within the multi function.

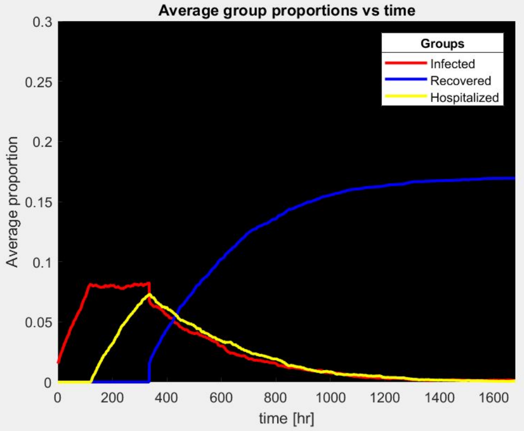
```
clear all
tot_time = 1680;%70 days
t = 0:1:tot_time; %time scale for 100 days = 2400 hours
tot_pop = 165;
D = 0.50; %diffusion coefficient
infectionrate = .112; %actual statistic on the chance of becoming infected
when in proximity of an infected individual
delayIS = 120; %5 day delay in IS quarantining
delayIA = 336; %IA quarantining --> delay decreases
num_sims = 50;

Covid_model_func(tot_pop, tot_time, D, infectionrate, delayIS, delayIA, 0);
%%does the animation

[averageU, averageI, averageR, averageH, finalU, finalI, finalR, finalH,
finalD] = Covid_model_multi_func(tot_pop, tot_time, D, infectionrate, delayIS,
delayIA, num_sims); %does not do the animation
```

3.4.1 Line Graph of Average Group Proportions Versus Time

The average proportions from 50 simulations, calculated from the multi func, were plotted over time.

```
figure;
```

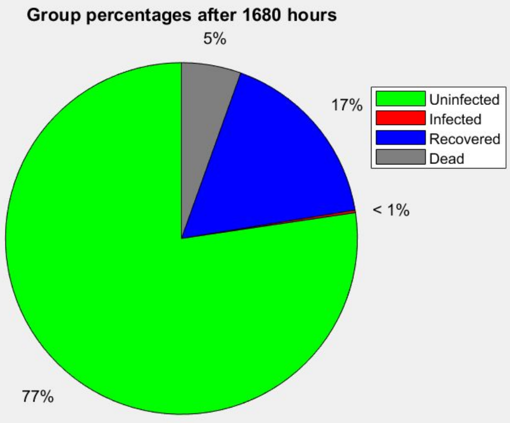
```
hold all;
%plotting average proportions of infected, recovered, and hospitalized
%individuals over time
plot(t, averageI, 'color', 'r', 'LineWidth', 2);
plot(t, averageR, 'color', 'b', 'LineWidth', 2);
plot(t, averageH, 'color', 'y', 'LineWidth', 2);
set(gca, 'color', 'k')
xlim([0 tot_time]);
ylim([0 0.3]);
title('Average group proportions vs time');
ylabel('Average proportion');
xlabel('time [hr]');
l = legend('show');
l.Title.String = 'Groups';
legend('Infected', 'Recovered', 'Hospitalized');
l.Color = 'white'; %legend background color
```



3.4.2 Pie Chart of Final Group Proportions

The average group proportions calculated at the very last time point from 50 simulations were displayed in a pie chart for easy comparison.

```
%pie chart for average percentages of each group at end of time period
figure;
x = [finalU, finalI + finalH, finalR, finalD];
labels = {'Uninfected', 'Infected', 'Recovered', 'Dead'};
pie(x)
legend(labels)
title(['Group percentages after ', num2str(tot_time), ' hours'])
colormap([0 1 0; 1 0 0; 0 0 1; .5 .5 .5])
```



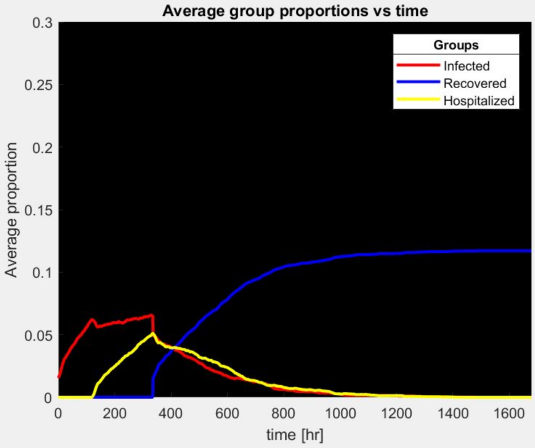
3.5 Mitigation Strategies

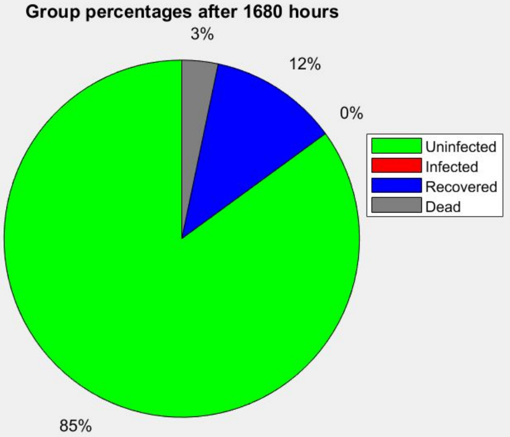
Mitigation modeling is an important facet of this COVID-19 epidemiological project. Evaluating different strategies to curb the spread of infection allows our model to extend beyond simply observing different outcomes; it presents the potential to inform public health policy in the effort to “flatten the curve”.
Our model deploys three different strategies, as described below.

3.5.1 Decreasing Diffusion Rate due to Social Distancing

The last mitigation strategy explored is to model social distancing by decreasing diffusion rates. The hypothesized effect of enforcing social distancing is a decrease in the diffusion rate. The diffusion rate coefficient is decreased from 0.5 to 0.25, reflecting a 75% reduction in movement.

```
%D = 0.25;
```

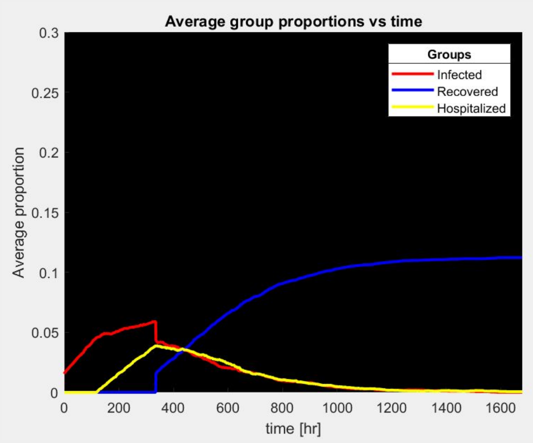


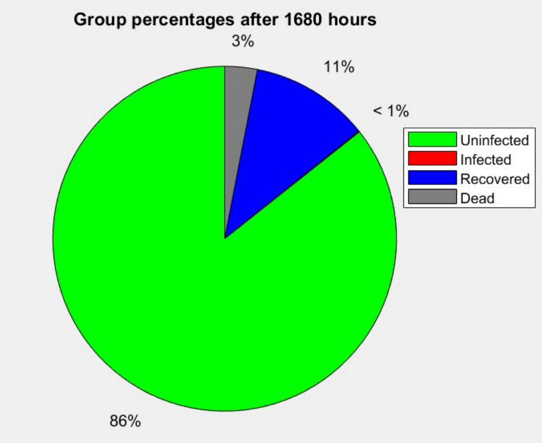


3.5.2 Decreasing the Infection Rate Because People are Wearing Personal Protective Equipment (PPE)

The third mitigation strategy explored is to model more widespread use of personal protective equipment, such as masks. The hypothesized effect of widespread PPE is a decrease in the infection rate. This mitigation strategy is manifested in the variable “infectionrate”; it is decreased from 0.112 to 0.04. The amended 4% infection rate reflecting the efficacy of face masks is referenced from Leung, et. al. (*Nat Med,* 2020). This figure assumes everyone would wear masks.

```
%infectionrate = 0.04;
```

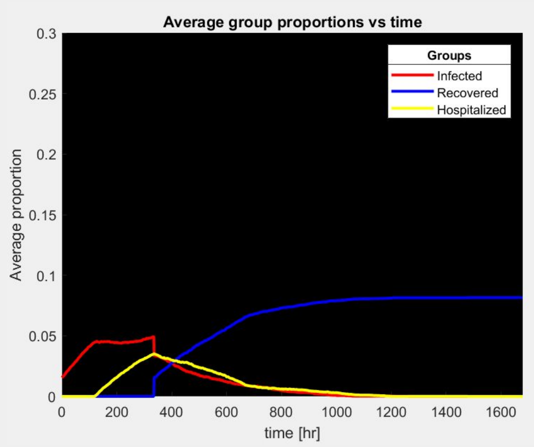


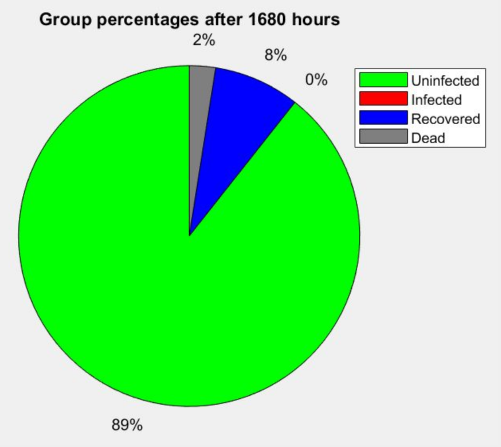


3.5.2.1 Wearing PPE *in addition to* Social Distancing

Both the diffusion rate was decreased by 75% and infection rate is decreased to 4%.

```
%D = 0.25;
%infectionrate = 0.04;
```

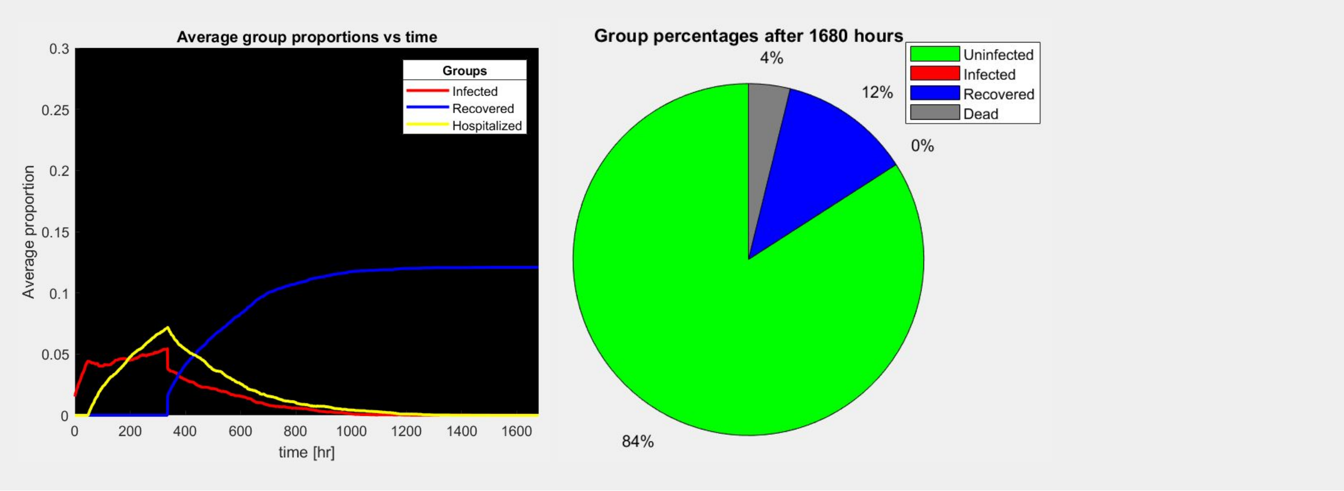




3.5.3 Hospitalizing Infected Symptomatic Individuals after 48 Hours as Opposed to 120 Hours

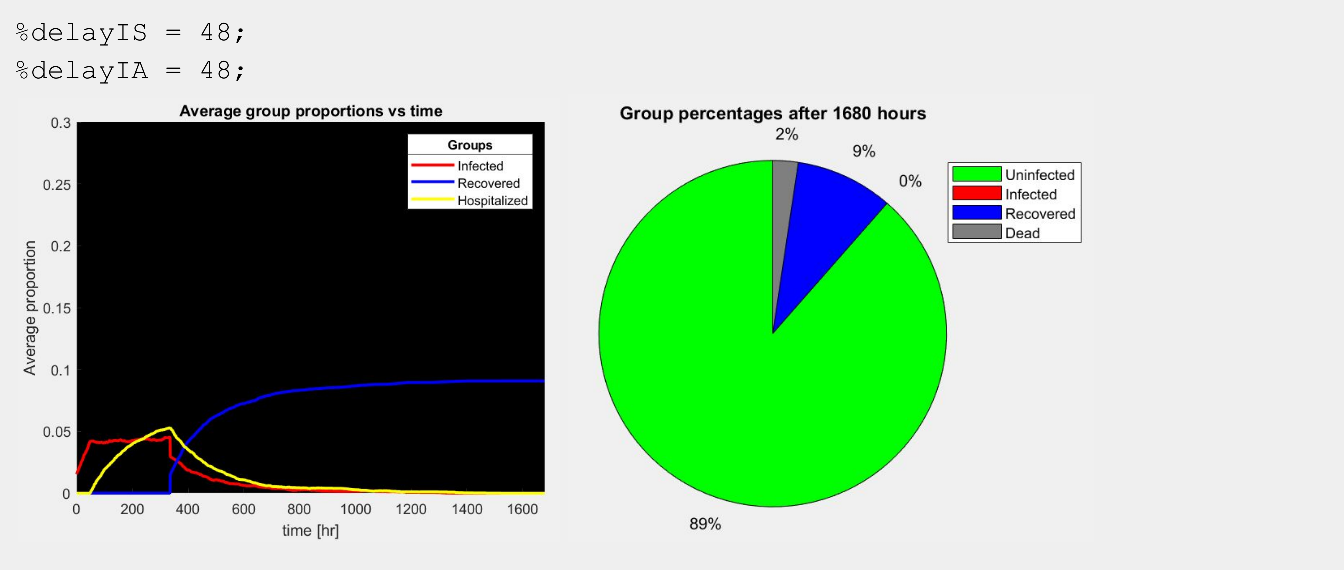
The first mitigation strategy explored is to model more widespread use of accessible population-scale testing. The hypothesized effect of widespread accessible testing is that infected symptomatic individuals would be diagnosed earlier along the course of the disease; thus, informing them to self-isolate even earlier after the initial onset of symptoms. This mitigation strategy is manifested in the variable “self_isolation_delay”; it is decreased from 120 hours to 48 hours.

```
%delayIS = 48;
```



3.5.4 Immobilizing Infected Asymptomatic Individuals after 48 Hours *in addition* to Hospitalizing Symptomatic Individuals after 48 hours

The second mitigation strategy explored is to model more widespread use of *proactive* population-scale testing. Not only would infected *symptomatic* individuals be diagnosed earlier, but an additional hypothesized effect of widespread accessible testing would be that infected *asymptomatic* individuals would be diagnosed earlier along the course of the disease; thus, informing infected asymptomatic individuals to immobilize and infected symptomatic individuals to hospitalize. This mitigation strategy is manifested in the variables "delayIA" (i.e. immobilization of infected asymptomatic individuals) and "delayIS" (i.e. hospitalization of infected symptomatic individuals). Both of these variables are set to 48 hours. (*Note: Without this mitigation strategy applied, the variable "delayIA" was set to 336 hours, which reflects the time at which all IA individuals will become recovered; thus, there is no IA immobilization in the control condition.*)

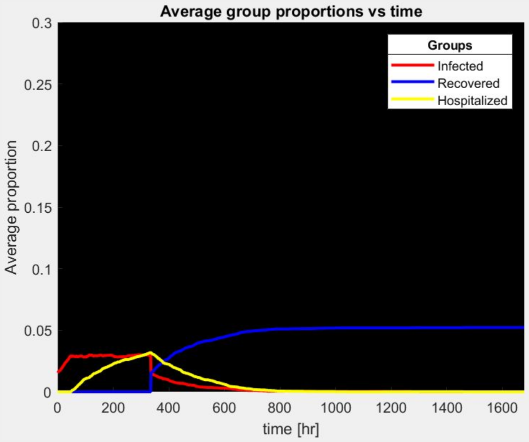


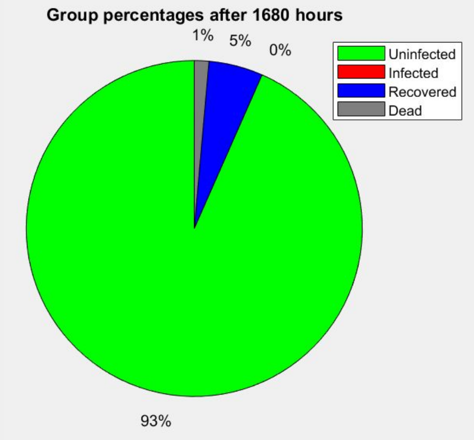
```
%delayIS = 48;
%delayIA = 48;
```

3.5.4 All Strategies

Finally, all four mitigations strategies (social distancing, PPE, general and proactive testing) were implemented.

```
%D = 0.25;
%infectionrate = 0.04;
%delayIS = 48;
%delayIA = 48;
```





4. Results

Without any mitigation strategies employed, our model has the following starting parameters:

tot_pop = 165; tot_time = 1680; D = 0.5; infectionrate = 0.112; delayIS = 120; delayIA = 336.

The resulting mortality rate averaged over 50 runs is 5.48%. The application of each mitigation strategy results in a reduction in mortality rate. The results are shown below.

Mitigation Strategy	Final Percentage Dead
None	5.48%
Moderate Social Distancing	3.29%
Wearing Masks	3.06%
Social Distancing + Masks	2.48%
General Testing	3.84%
General + Proactive Testing	2.36%
All Strategies At Once	1.39%

In the original model with no mitigation strategies implemented, the percentage of uninfected individuals at the end of the time period was 77%. Additionally, the maximum hospitalized percentage at one time point was 7.29% at t = 336 hours. In contrast, when all mitigation strategies were observed, this percentage increased to 93% uninfected. The maximum hospitalized percentage at one time point for this simulation was 3.0% at t = 337 hours.

5 Conclusion

This model predicts the spread of COVID-19 amongst a population of 165 individuals confined to a 10,000 sq m, reflecting 1/10th the size of Yale's campus and population. The model incorporates independent variables for total population (tot_pop), total time (tot_time), diffusion coefficient (D), infection rate (infectionrate), hospitalization delay for infected symptomatic individuals (delayIS), and immobilization delay for infected asymptomatic individuals (delayIA). The variables can be easily changed to reflect other modeling contexts.

Prior to the application of any mitigation strategies, the results demonstrate widespread propagation of the virus, yielding a mortality rate of 5.48%, comparable to the current US mortality rate at 5.85% and less than the peak mortality rate of Italy at 7.2% prior to March 17.

The application of each mitigation strategy - (1) decreasing diffusion rate, (2) decreasing infection rate, (3) hospitalizing infected symptomatic individuals after shorter delay, (4) hospitalizing infected symptomatic individuals after shorter delay *and* immobilizing infected asymptomatic individuals - result in a reduction in mortality rate. Each of these mitigation strategies reflect a societal change in public health services and/or policies. The application of all four mitigation strategies together results in an even smaller mortality rate of 1.39%.

The application of all mitigation strategies together also results in a more "flattened" curve of hospitalized individuals. The reduction in the total number of individuals in the hospital at any given time is important to ensure health services are not overwhelmed with patients. Overcrowded hospitals can present the danger for nosocomial infection and inadequate care; thus, effective mitigation strategies are not only important for reducing mortality rate but also for reducing hospitalizations. It should be noted that employing the (3) & (4) mitigation strategies individually increases the hospitalization peak. Thus, although the overall mortality rate decreases as a result of these mitigation strategies encouraging more proactive hospitalizations, the ramifications of more crowded hospitals must be considered if widespread accessible population-scale testing was implemented.

Possible extensions to this model include incorporating patient outcomes as distributions of time as opposed to the fixed time point of 14 days. Reinfection could also be explored. Considering that this model reflects a college campus, another extension to consider is creating regions of attraction (i.e. to attract the movement of particles towards it) during set time points during the day; this would reflect the common places for interaction on a college campus, such as classrooms, libraries, and dining halls.

We would like to thank Professor Michael Mak for teaching us MATLAB during this semester. This model would not be possible without his help.

References

Bi, Q., Wu, Y., Mei, S., Ye, C., Zou, X., Zhang, Z., Liu, X., Wei, L., Truelove, S. A., Zhang, T., Gao, W., Cheng, C., Tang,
X., Wu, X., Wu, Y., Sun, B., Huang, S., Sun, Y., Zhang, J., … Feng, T. (2020). Epidemiology and transmission of COVID-19 in 391 cases and 1286 of their close contacts in Shenzhen, China: A retrospective cohort study. *The Lancet Infectious Diseases*, *0*(0). https://doi.org/10.1016/S1473-3099(20)30287-5

Crist, C. (n.d.). CDC: COVID-19 Can Spread 13 Feet, Travel on Shoes. WebMD. Retrieved May 4, 2020, from https://www.webmd.com/lung/news/20200414/cdc-covid-19-can-spread-13-feet-travel-on-shoes

Field Briefing: Diamond Princess COVID-19 Cases, 20 Feb Update. (n.d.). Retrieved May 4, 2020, from https://www.niid.go.jp/niid/en/2019-ncov-e/9417-covid-dp-fe-02.html

Health, C. for D. and R. (2020). N95 Respirators and Surgical Masks (Face Masks). FDA. https://www.fda.gov/medical-devices/personal-protective-equipment-infection-control/n95-respirators-and-surgical-masks-face-masks

Leung, N.H.L., Chu, D.K.W., Shiu, E.Y.C. et al. Respiratory virus shedding in exhaled breath and efficacy of face masks. *Nat Med* (2020). https://doi.org/10.1038/s41591-020-0843-2

Mizumoto, K., et. al. Estimating the asymptomatic proportion of coronavirus disease 2019 (COVID-19) cases on board the Diamond Princess cruise ship, Yokohama, Japan, 2020. Retrieved May 4, 2020, from https://www.eurosurveillance.org/content/10.2807/1560-7917.ES.2020.25.10.2000180

Mql_sars-cov-2_-cleared-for-public-release_2020_03_25.pdf. (n.d.). Retrieved May 4, 2020, from https://www.dhs.gov/sites/default/files/publications/mql_sars-cov-2_-cleared-for-public-release_2020_03_25.pdf

Onder, G., Rezza, G., Brusaferro, S. Case-fatality rate and characteristics of patients dying in relation to COVID-19 in Italy. *Jama*. Published online. (2020).

Roser, M., Ritchie, H., Ortiz-Ospina, E., & Hasell, J. (2020). Coronavirus Pandemic (COVID-19). Our World in Data. https://ourworldindata.org/coronavirus

United States Coronavirus: 1,194,434 Cases and 68,999 Deaths - Worldometer. (n.d.). Retrieved May 4, 2020, from https://www.worldometers.info/coronavirus/country/us/

Verity, R., Okell, L. C., Dorigatti, I., Winskill, P., Whittaker, C., Imai, N., Cuomo-Dannenburg, G., Thompson, H., Walker, P. G. T., Fu, H., Dighe, A., Griffin, J. T., Baguelin, M., Bhatia, S., Boonyasiri, A., Cori, A., Cucunubá, Z., FitzJohn, R., Gaythorpe, K., … Ferguson, N. M. (2020). Estimates of the severity of coronavirus disease 2019: A model-based analysis. *The Lancet Infectious Diseases*, 0(0). https://doi.org/10.1016/S1473-3099(20)30243-7

What is the efficacy of standard face masks compared to respirator masks in preventing COVID-type respiratory illnesses in primary care staff? (n.d.). CEBM. Retrieved May 4, 2020, from https://www.cebm.net/covid-19/what-is-the-efficacy-of-standard-face-masks-compared-to-respirator-masks-in-preventing-covid-type-respiratory-illnesses-in-primary-care-staff/

Yale Facts. (2015, August 3). Yale University. https://www.yale.edu/about-yale/yale-facts