STAT3004 — Project 1

Matthew Low

May 27, 2020

The code snippets for this assignment were done in Python 3.8.

Question 1

Equation (4.1.4) presents the expected numbers in the Greenwood model.

$$\mathbb{E}[X_t \mid X_0 = x_0] = \alpha^t x_0, \quad \mathbb{E}[Y_t \mid X_0 = x_0] = \alpha^{t-1}(1-\alpha)x_0. \tag{4.1.4}$$

1a *Derive these equations.*

We first note that from the text that

$$\mathbb{P}((X,Y)_{t+1} = (x,y)_{t+1} \mid (X,Y)_t = (x,y)_t) = \binom{x_t}{x_{t+1}} \alpha^{x_{t+1}}(1-\alpha)^{x_t - x_{t+1}}.$$

We identify that this probability is the probability mass function of a binomial distribution, with $x_t := n$, $x_{t+1} := k$ and $\alpha := p$. This means that we can utilise the expectation

$$\mathbb{E}[X_{t+1} \mid X_t] = np = \alpha X_t.$$

Note that $\mathbb{E}[X_{t+1} \mid X_t] \equiv \mathbb{E}[X_t \mid X_{t-1}]$. We can apply another expected value to both sides to obtain

$$\mathbb{E}[\mathbb{E}[X_t \mid X_{t-1}]] = \mathbb{E}[X_t] = \alpha \mathbb{E}[X_{t-1}].$$

This can be applied iteratively/recursively to obtain

$$\mathbb{E}[X_t] = \alpha^t \mathbb{E}[X_0] = \alpha^t x_0,$$

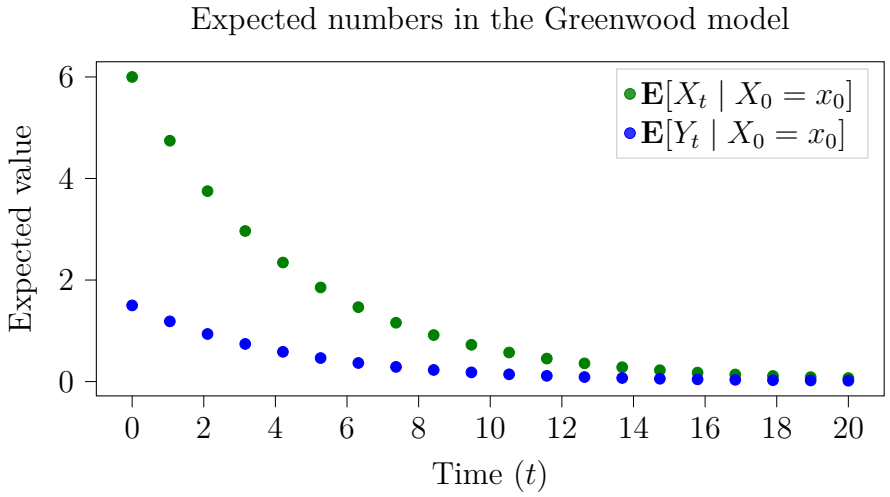
therefore deriving the first part of this equation. Recall that X_t denotes the number of susceptibles at time t, and Y_t denotes the number of infectives at time t. Since $X_t + Y_t = X_{t-1}$, we can rearrange to obtain $Y_t = X_{t-1} - X_t$. This means that we have

$$\begin{aligned}\mathbb{E}[Y_t \mid X_0 = x_0] &= \mathbb{E}[X_{t-1} - X_t \mid X_0 = x_0] = \mathbb{E}[X_{t-1} \mid X_0 = x_0] - \mathbb{E}[X_t \mid X_0 = x_0] \\ &= \alpha^{t-1}x_0 - \alpha^t x_0 = \alpha^{t-1}(1-\alpha)x_0\end{aligned}$$

1b *Assume $x_0 = 6$ and $\alpha = 0.8$. Then plot these expected values for some suitable time horizon.*

```
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
import tikzplotlib

plt.rcParams.update({
    "font.family": "serif",
    "text.usetex": True,
```

```
})

# Set up parameters
alpha = 0.8
x0 = 6

# Set up grid
t = np.linspace(0, 20, 20)

# Plot expected values using step since discrete time values
plt.scatter(t, (alpha ** t) * x0, \
    label='$\mathbf E[X_t \mid X_0 = x_0]$', color='g')
plt.scatter(t, (alpha ** (t-1)) * (1-alpha) * x0, \
    label='$\mathbf E[Y_t \mid X_0 = x_0]$', color='b')

plt.xlabel('Time ($t$)')
plt.ylabel('Expected value')
plt.title("Expected numbers in the Greenwood model")
plt.legend()
tikzplotlib.save("assignments/stat3004-stochastic/tikz/1b.tikz", \
    axis_width='12cm', axis_height='6cm')
```



Question 2

Equation (4.2.1) presents a recursion for the expected number of susceptibles and infected in the Reed-Frost model.

$$\mathbb{E}[(X,Y)_{t+1} \mid (X,Y)_t = (x,y)_t] = (x_t\alpha^{y_t}, x_t(1-\alpha^{y_t})). \tag{4.2.1}$$

2a *Derive these equations.*

We derive the X and Y components of this expectation independent. First, we note that $X_{t+1} \sim \text{Bin}(X_t, \alpha^{Y_t})$. This gives us the conventional binomial expectation

$$\mathbb{E}[X_{t+1} \mid X_t = x_t] = np = \boxed{x_t\alpha^{y_t}}.$$

For the Y component, we first note that $Y_{t+1} = X_t - X_{t+1}$. We can then split up the expectations as follows:

$$\begin{aligned}\mathbb{E}[Y_{t+1} \mid X_t = x_t, Y_t = y_t] &= \mathbb{E}[X_t \mid X_t = x_t] - \mathbb{E}[X_{t+1} \mid X_t = x_t] \\ &= \mathbb{E}[X_t \mid X_t = x_t] - x_t\alpha^{y_t}.\end{aligned}$$

We now prove $\mathbb{E}[X_t \mid X_t = x_t] = x_t$. Using the definition of conditional expectation, we have

$$\mathbb{E}[X_t \mid X_t] = \sum_{x_t} x_t \mathbb{P}(X_t = x_t \mid X_t = x_t) = \sum_{x_t} x_t \frac{\mathbb{P}(X_t = x_t \cap X_t = x_t)}{\mathbb{P}(X_t = x_t)} = \sum_{x_t} x_t \frac{\mathbb{P}(X_t = x_t)}{\mathbb{P}(X_t = x_t)} = x_t.$$

Returning back to $\mathbb{E}[Y_{t+1} \mid X_t x_t, Y_t = y_t]$, we have

$$\mathbb{E}[Y_{t+1} \mid X_t x_t, Y_t = y_t] = \mathbb{E}[X_t \mid X_t = x_t] - x_t \alpha^{y_t} = x_t - x_t \alpha^{y_t} = \boxed{x_t(1 - \alpha^{y_t})}.$$

Combining $\mathbb{E}[X_{t+1} \mid X_t = x_t, Y_t = y_t]$ and $\mathbb{E}[Y_{t+1} \mid X_t = x_t, Y_t = y_t]$ we obtain our desired result

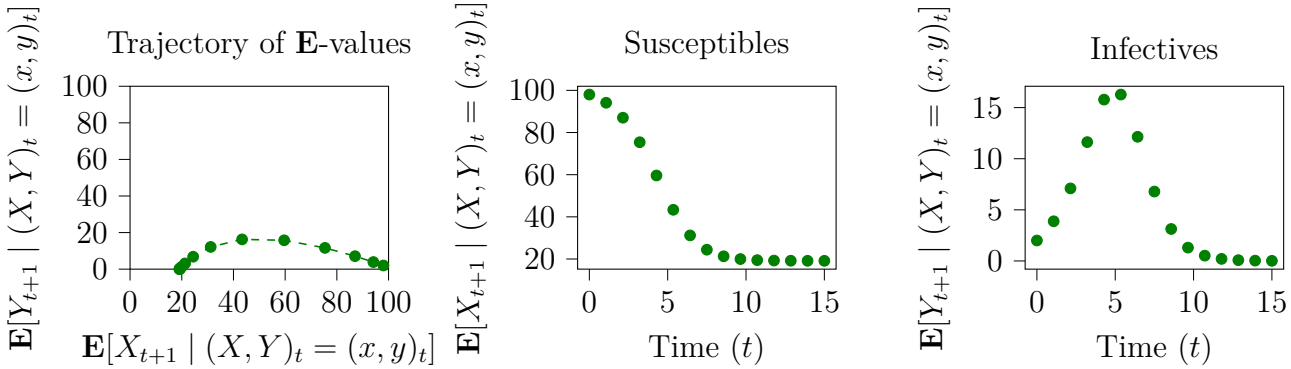
$$\mathbb{E}[(X,Y)_{t+1} \mid (X,Y)_t = (x,y)_t] = (x_t \alpha^{y_t}, x_t(1 - \alpha^{y_t})).$$

2b *Reproduce Figure 4.2 and also plot the trajectory of expected values, jointly on the (X, Y) plane in a similar manner to Figure 10.1 of [SWJ-10] (there, the plot is for a predator-prey model).*

Note From now on, imports and other "boilerplate" Python code will be omitted from snippets to save space.

```
# Set up parameters
alpha = 0.98
N, I = 100, 1

EXt = lambda x, y: x * (alpha ** y)
EYt = lambda x, y: x * (1 - alpha ** y)

coordinates_x = []
coordinates_y = []
# Set starting values
x, y = N, I
t = np.linspace(0, 15, 15)

# Iterate
for i in range(15):
    x, y = (EXt(x, y), EYt(x,y))
    print(x,y)
    coordinates_x.append(x)
    coordinates_y.append(y)

plt.subplot(131)
plt.xlim(0, 100)
plt.ylim(0, 100)
plt.title("Trajectory of $\mathbf E$-values")
plt.scatter(coordinates_x, coordinates_y, color='g')
plt.plot(coordinates_x, coordinates_y, 'g--')
plt.xlabel('$\mathbf E[X_{t+1} \mid (X,Y)_t = (x,y)_t]$')
plt.ylabel('$\mathbf E[Y_{t+1} \mid (X,Y)_t = (x,y)_t]$')

plt.subplot(132)
plt.title('Susceptibles')
plt.scatter(t, coordinates_x, color='g')
plt.xlabel('Time ($t$)')
plt.ylabel('$\mathbf E[X_{t+1} \mid (X,Y)_t = (x,y)_t]$')

plt.subplot(133)
plt.title('Infectives')
plt.scatter(t, coordinates_y, color='g')
plt.xlabel('Time ($t$)')
plt.ylabel('$\mathbf E[Y_{t+1} \mid (X,Y)_t = (x,y)_t]$')
```

```
tikzplotlib.save("assignments/stat3004-stochastic/tikz/2b.tikz", \
    axis_width='5cm', axis_height='4cm')
```



Question 3

Consider the Markov chain $\{X_t, t = 0, 1, 2 \ldots\}$ *defined by* $X_0 = x_0$ *and the transition probability matrix as in equation (4.1.5) for the Greenwood model. The state space is* $\{0, 1, 2, \ldots, x_0\}$.

For reference:

$$\mathbb{E}[X_t \mid X_0 = x_0] = \alpha^t x_0, \quad \mathbb{E}[Y_t \mid X_0 = x_0] = \alpha^{t-1}(1-\alpha)x_0. \tag{4.1.4}$$

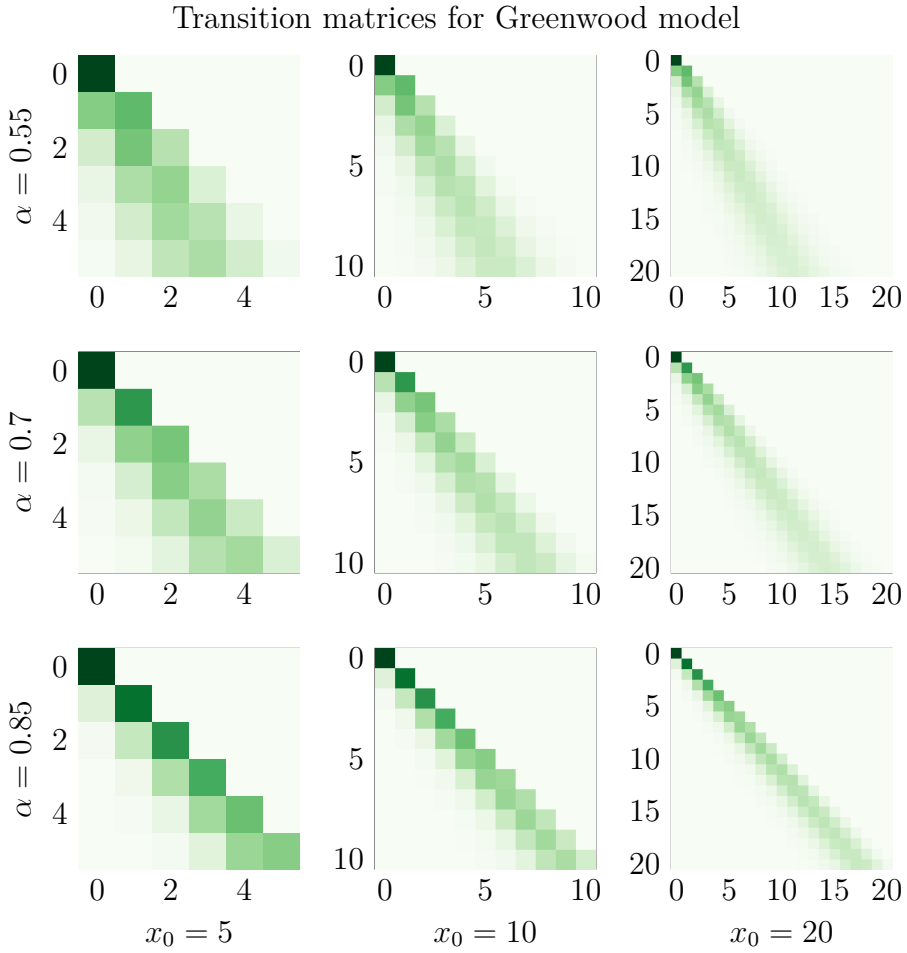
$$P = \begin{pmatrix} 1 & . & . & \ldots & . \\ 1-\alpha & \alpha & . & \ldots & . \\ (1-\alpha)^2 & 2(1-\alpha)\alpha & \alpha^2 & \ldots & . \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (1-a)^{x_0} & x_0(1-\alpha)^{x_0-1}\alpha & \binom{x_0}{2}(1-\alpha)^{x_0-2}\alpha^2 & \ldots & \alpha^{x_0} \end{pmatrix} \tag{4.1.5}$$

3a *Plot a heat-map of this transition probability matrix for* $x_0 = 5$, $x_0 = 10$ *and* $x_0 = 20$ *and some* $\alpha \in (0.5, 0.9)$ *of your choice.*

For this question, I chose the α values $\alpha \in \{0.55, 0.7, 0.85\}$. I did a grid of subplots, with rows representing different α and columns representing different x_0.

```
# Set up parameters
alphas = [0.55, 0.7, 0.85]
x0s = [5, 10, 20]

# First subplot
index = 331
plt.suptitle("Transition matrices for Greenwood model")

# Generate matrices for each specified alpha and size
for alpha_num, alpha in enumerate(alphas):
    for x0_num, x0 in enumerate(x0s):
        # Generate matrix
        matrix = np.zeros((x0 + 1, x0 + 1))
        for i in range(x0 + 1):
            for j in range(x0 + 1):
                if i < j:
                    continue
                elif i == j:
                    matrix[i,j] = alpha ** i
                else:
```

```
                    matrix[i,j] = math.comb(i, j) \
                        * (1 - alpha) ** (i - j) \
                        * (alpha) ** j
            plt.subplot(index)
            # Using green colourmap, increasing from 0 (white) to 1 (green)
            plt.imshow(matrix, cmap='Greens', interpolation='nearest')
            plt.xticks([], [])
            plt.yticks([], [])
            index += 1
            # Some code to print out row and column labels
            if alpha_num == len(alphas) - 1:
                plt.xlabel(f'$x_0 = {x0s[x0_num]}$')
            if x0_num == 0:
                plt.ylabel(f'$\\alpha = {alphas[alpha_num]}$')

tikzplotlib.save("assignments/stat3004-stochastic/tikz/3a.tikz", \
    axis_width='4.5cm', axis_height='4.5cm')
```

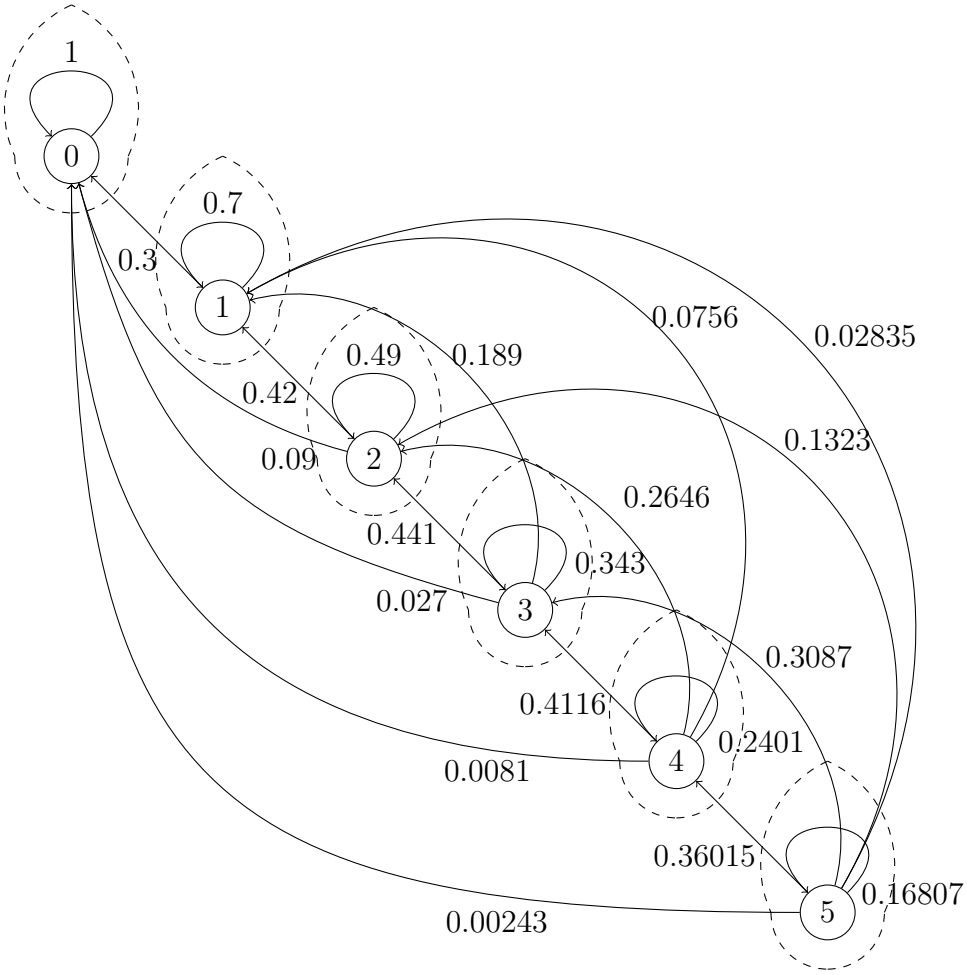


3b *Determine the communicating classes of this Markov chain. How many are there? Which are recurrent? Which are transient?* Observe that for any state in the state space $\{0, 1, 2, \ldots, x_0\}$, it is only ever possible to go down a state, but not up a state. For example, let us draw the probability matrix in the case where $x_0 = 5$ and $\alpha = 0.7$. We can compute the numerical values of this transition matrix

by running a modified version of the above code, to obtain

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0.3 & 0.7 & 0 & 0 & 0 & 0 \\ 0.09 & 0.42 & 0.49 & 0 & 0 & 0 \\ 0.027 & 0.189 & 0.441 & 0.343 & 0 & 0 \\ 0.0081 & 0.0756 & 0.2646 & 0.4116 & 0.2401 & 0 \\ 0.00243 & 0.02835 & 0.1323 & 0.3087 & 0.36015 & 0.16807 \end{pmatrix}.$$

Notice that this is a lower-triangular matrix. Drawing out this Markov chain gives us the following:



We observe that no state i can at any point go to any state j such that $j > i$. Therefore, there is no two-way communication between any states. As a result, the communication classes of this Markov chain are

$$\{0\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}$$

and in general for some x_0, the communicating classes of the Markov chain are

$$\boxed{\{0\}, \{1\}, \{2\}, \ldots, \{x_0\}}$$

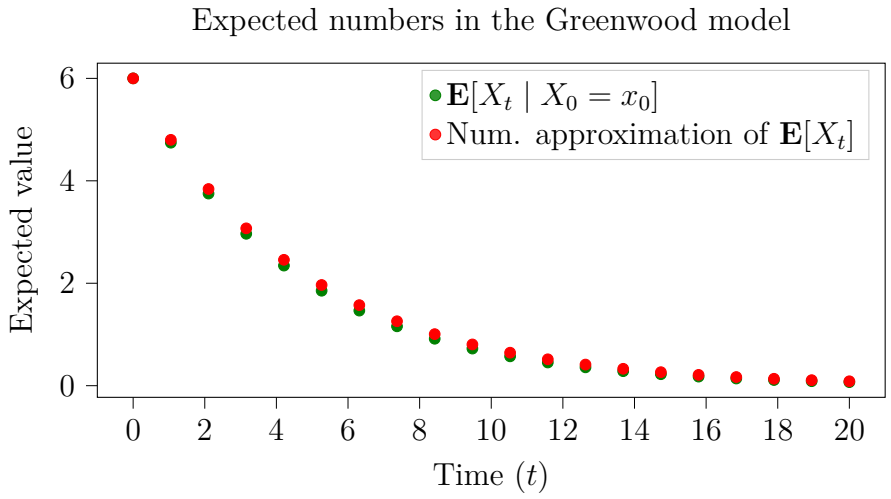
and as a result $\boxed{\text{there are } x_0 + 1 \text{ communicating classes}}$. Recall that a recurrent state is one which you will keep coming back to, and a transient state is one that will eventually be left forever (in less rigorous terms). As previously mentioned, you can only ever go down but never up, and therefore the $\boxed{\text{recurrent class is } \{0\}}$, and the $\boxed{\text{transient classes are } \{1\}, \{2\}, \ldots, \{x_0\}}$.

3c *The part of equation (4.1.4) for X_t, presenting the expected value, can be obtained in a much more cumbersome way to what you did in 1a above. For this, take the power P^t and compute $\mathbf{e}_{x_0+1}^\top P^t \mathbf{v}$, where $\mathbf{e}_{x_0+1}$ is the x_0+1 long unit vector $(0, 0, \ldots, 1)^\top$ and $\mathbf{v}$ is the vector $(0, 1, 2, \ldots, x_0)^\top$. Compute*

this numerically and see that the results numerically agree with those in the plot of 1b. Explain why this holds.

The code to compute this numerically is as follows (with some code from previous plots omitted for the sake of brevity):

```
for i in range(x0 + 1):
    v[i] = i

# Generate numerical approximation eT * P^t * v
numX = np.zeros((20, 1))
for i in range(20):
    numX[i] = e.T @ np.matrix(matrix) ** (i) @ v

# Plot on top of previous plot
plt.scatter(t, numX, color='r',\
    label='Num. approximation of $\mathbf E[X_t]$')
plt.legend()
tikzplotlib.save("assignments/stat3004-stochastic/tikz/3c.tikz", \
    axis_width='12cm', axis_height='6cm')
```



We can explain the accuracy of this numerical approximation by performing the normal matrix multiplication algorithm on our matrices one by one. Recall that for matrix multiplication, we work our way through the rows on the left matrix and then work through the columns of the right matrix, summing each column. In the case of a $1 \times n$ multiplied by an $n \times n$ matrix, it computes the sum of each column, but since our $1 \times n$ matrix is zero everywhere except the last column, it just extracts the last row of the right hand side matrix, giving us

$$\mathbf{e}_{x_0+1}^\top P = \begin{pmatrix} (1-\alpha)^{x_0} & x_0(1-\alpha)^{x_0-1}\alpha & \binom{x_0}{2}(1-\alpha)^{x_0-2}\alpha^2 & \dots & \alpha^{x_0} \end{pmatrix}$$

We now have a new $1 \times n$ matrix, which we will multiply by the $n \times 1$ matrix $\mathbf{v}$ (inner product) to obtain a single value. Since $\mathbf{v}$ is just the ascending integers $\{0, 1, \dots, x_0\}$, we obtain the sum

$$\mathbf{e}_{x_0+1}^\top P\mathbf{v} = \sum_{i=0}^{x_0} i \binom{x_0}{i} (1-\alpha)^{x_0-i}\alpha^i = \alpha x_0.$$

Extending this to the case where we have P^t instead of just P, using a similar recursive argument to question 1a, we obtain

$$\mathbf{e}_{x_0+1}^\top P^t\mathbf{v} = \cdots = \alpha^t x_0.$$

One can also observe that the left multiplication $\mathbf{e}_{x_0+1}^\top P^t$ gives the row vector corresponding to the distribution after t states. Since the values of $\mathbf{v}$ correspond to the states in ascending order, the inner product of $\mathbf{e}_{x_0+1}^\top P^t$ by $\mathbf{v}$ gives the expectation $\mathbb{E}[X_t \mid X_0 = x_0] = \alpha^t x_0$.

3d *Attempt to carry out a similar numerical computation for the expectation of Y_t in equation (4.1.4) and explain your method.* To obtain our method, we use the above result that $\mathbf{e}_{x_0+1}^\top P^t \mathbf{v} = \alpha^t x_0 = \mathbb{E}[X_t \mid X_0 = x_0]$. We can then also recall that

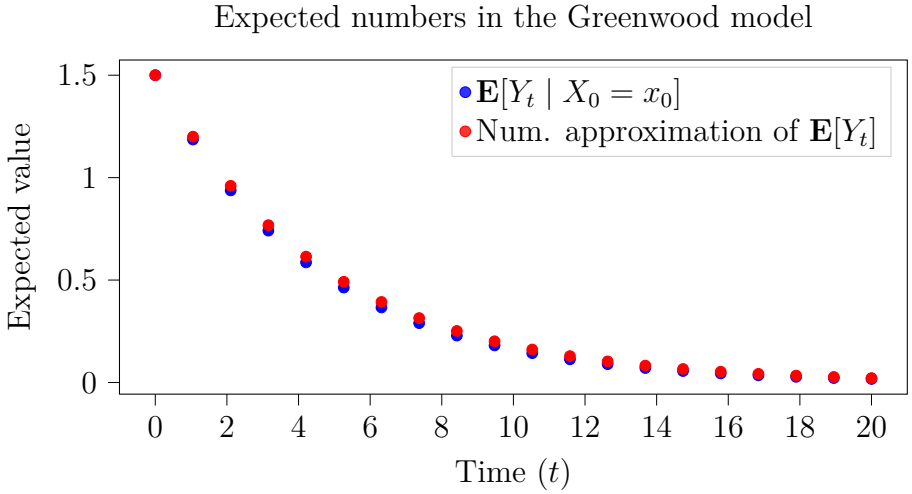
$$\mathbb{E}[Y_t \mid X_0 = x_0] = \mathbb{E}[X_{t-1} \mid X_0 = x_0] - \mathbb{E}[X_t \mid X_0 = x_0]$$

With some algebra, we can obtain the formula

$$\begin{aligned}\mathbb{E}[Y_t \mid X_0 = x_0] &= \mathbf{e}_{x_0+1}^\top P^{t-1}\mathbf{v} - \mathbf{e}_{x_0+1}^\top P^t \mathbf{v} \\ &= \mathbf{e}_{x_0+1}^\top P^{t-1}\mathbf{v} - \mathbf{e}_{x_0+1}^\top P^{t-1} P\mathbf{v} \\ &= \mathbf{e}_{x_0+1}^\top P^{t-1}(\mathbf{v} - P\mathbf{v}) = \mathbf{e}_{x_0+1}^\top P^{t-1}(I - P)\mathbf{v}.\end{aligned}$$

We can implement this numerical approximation below:

```
# Generate numerical approximation eT * P^(t-1) . (1-a) * v
numY = np.zeros((20, 1))
for i in range(20):
    numY[i] = e.T @ np.matrix(matrix) ** (i-1) @ (np.eye(x0+1, x0+1) - np.matrix(
        matrix)) @ v

# Plot on top of previous plot
plt.scatter(t, numY, color='r',\
    label='Num. approximation of $\mathbf E[Y_t]$')
plt.legend()
tikzplotlib.save("assignments/stat3004-stochastic/tikz/3d.tikz", \
    axis_width='12cm', axis_height='6cm')
```



Question 4

Consider now the joint distribution of (W, T) as described in subsection 4.1.1 (dealing with the Greenwood model). Here, T is the first time in which there are no infectives and W is the number of susceptibles that have been infected by that time. That is the random variables T and W describe the "end of the infection". The main aim is to know the probabilities

$$\Gamma(k, n \mid x_0) = \mathbb{P}((W, T) = (k, n) \mid X_0 = x_0, Y_0 > 0),$$

for $k = 0, 1, \ldots, x_0$ and $n = 1, 2, \ldots$. These assume that at onset x_0 family members are sick and there is an infection in the household. For all of the numerical computations in this question, use $x_0 = 6$ and some fixed $\alpha \in (0.7, 0.9)$ of your choice.

Note For question 4 and 5, we choose $x_0 = 6$, $y_0 = 1$, $\alpha = 0.8$.

4a *Explain equation (4.1.6).* For reference, equation 4.1.6 is

$$p_j^t \equiv \mathbb{P}(X_t = j, Y_t > 0) = \sum_{i=j+1}^{x_0-(t-1)} p_i^{t-1} p_{ij} \tag{4.1.6}$$

To elucidate this formula a bit more, we can expand it to

$$p_j^t = p_{j+1}^{t-1} p_{j+1,j} + p_{j+2}^{t-1} p_{j+2,j} + \cdots + p_{x_0-t+1}^{t-1} p_{x_0-t+1,j};$$

that is, the probability of being in state j at time t is the probability of being in state $j+1$ at time $t-1$ times $p_{j+1,j}$. By definition, $p_{ij} = \mathbb{P}(X_{t+1} = j \mid X_t = 1, Y_t > 0)$ so we can interpret $p_{j+1,j}$ as the probability of going down 1 state, $p_{j+2,j}$ as the probability of going down 2 states and so on.

Therefore, the above expression for p_j^t is the probability of going down one state $\times$ the probability of being at state $j+1$, plus the probability of going down two states $\times$ the probability of being at state $j+2$, and so on. In other words, the expression above is the sum of all arrows heading towards state j, recursively. We note that since we have the requirement that $Y_t > 0$, and $Y_t = X_{t-1} - X_t$, we do not include the probability of staying in the same state.

4b *Use the recursive relationship* $\Gamma(k, n \mid x_0) = p_{x_0-k}^{n-1} \alpha^{x_0-k}$ *to (numerically) compute* $\mathbb{P}(W > 4)$. The code for this computation is below:

```
x0 = 6
y0 = 1
alpha = 0.8

# Defined as p_{i, j} in EM-4
def prob(i, j):
    return math.comb(i, j) * (1-alpha)**(i-j) * (alpha**j)

# Defined as p_j^t in EM-4
def p(j, t):
    # print(j, t)
    if t == 0:
        return 1 if x0 == j else 0
    result = 0
    # print("Here")
    for i in range(j+1, x0-(t-1)+1):
        # print("Loop")
        result += p(i, t-1) * prob(i, j)
    return result

# P((W,T) = (k,n) | X0 = i, Y0 > 0) in EM-4
def pwt(k, n, i):
    return p(i-k, n-1) * alpha**(i-k)

# Compute P((W,T) = (k,n)) for all n, then for all k > 4, then sum them all
sum_num = 0
for i in range(1, 100):
    for j in range(5, 7):
        sum_num += pwt(j, i, x0)

print(sum_num)
```

Running this code gives the result $\boxed{\mathbb{P}(W > 4) \approx 0.130298}$.

4c *Compare your numerical result to an estimate obtained by a Monte-Carlo simulation creating* 10^6 *repeated trajectories and using those to estimate* $\mathbb{P}(W > 4)$. The code for this computation is below:

```
infected_nums = []
iterations = 10**6

for iteration in range(iterations):
    x = 6
    alpha = 0.8
    # Set dummy value that isn't 0
    y = -1
    # Number of infected so far
    infected = 0

    # Loop until y = 0 (no infected at current time t)
    while y != 0:
        new_x = np.random.binomial(x, alpha)
        y = x - new_x
        infected += y
        x = new_x

    infected_nums.append(infected)

# Monte-Carlo estimate of P(W>4) using list comprehensions
print(len([i for i in infected_nums if i > 4])/iterations)
```

On one sample run, this Monte Carlo estimate gives us $\boxed{\mathbb{P}(W > 4) \approx 0.13039}$, within 3 decimal places of the numerical computation in 4b above.

4d *Attempt to reproduce the PGF computations in subsection 4.1.1 to then obtain the same numerical result (this item is longer and slightly more challenging).*

We note that to compute this numerical result from the joint PGF, we can compute

$$\Psi_{W,T}(\varphi, 1) \equiv \Psi_W(\varphi) = A'(I - \bar{P}(\varphi))^{-1} Q E.$$

which follows from a property of generating functions:

$$\Psi_{W,T}(\varphi, 1) = \mathbb{E}[\varphi^W 1^T] = \mathbb{E}[\varphi^W] = \Psi_W(\varphi).$$

We also recall that to compute the PMF from the PGF, we apply

$$\mathbb{P}(X = k) = \frac{\Psi^{(k)}(0)}{k!},$$

where $\Psi^{(k)}(0)$ is the k-th derivative of Ψ at 0. We can notice from equation (4.1.14) in [1] that the kth derivatives correspond to the coefficients of φ in expanded form:

$$\begin{aligned} A'(I - \bar{P}(\varphi))^{-1} Q E &= (0,0,0,1)(I - \bar{P}(\varphi))^{-1}(1, \alpha, \alpha^2, \alpha^3)^\top \\ \Psi_W(\varphi) &= \alpha^3 + \varphi(3\alpha^4(1-\alpha)) + \varphi^2(3\alpha^2(1+2\alpha^2)(1-\alpha)^2) \\ &\quad + \varphi^3((1-\alpha)^3(1+3\alpha+3\alpha^2+6\alpha^3)). \end{aligned}$$

To be continued...

Question 5

Consider the Markov chain for the Reed-Frost model with transition probability matrix as in (4.2.2). For all the numerical computations in this question, use $x_0 = 6$, some y_0 of your choice, and some fixed $\alpha \in (0.7, 0.9)$ of your choice (use the same α which you used for the previous question).

5a *What is the state space?* As explained in the textbook [1],

> *P_{ij} records the transition probabilities from $Y_t = i$ to $Y_{t+1} = j$, with values of X_t and X_{t+1} indexing the rows and columns respectively, ranging from 0 to x_0.*

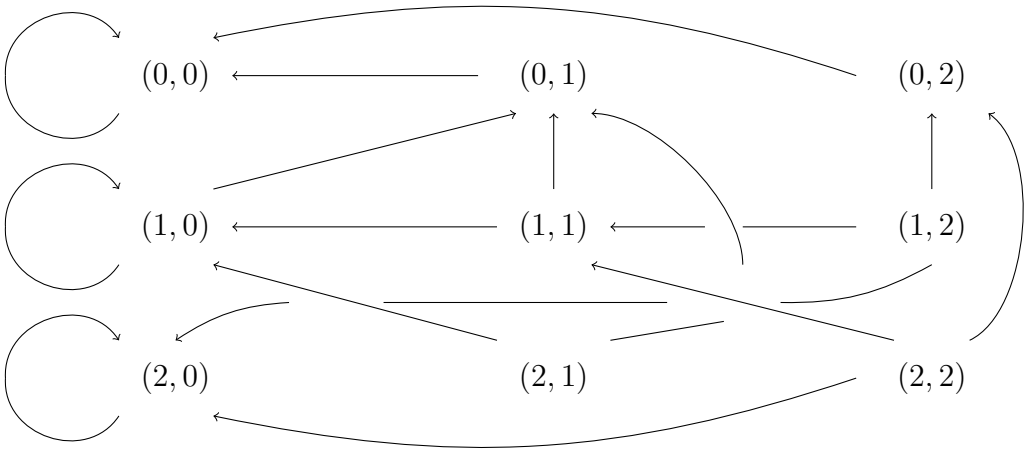
Since the full transition matrix consists of $(x_0 + 1) \times (x_0 + 1)$ block matrices, we have that $x \in \{0, \dots, x_0\} \triangleq X$ and $y \in \{0, \dots, x_0\} = X$ so we can take the state space to be the Cartesian product $\boxed{X \times X = \{0, \dots, x_0\} \times \{0, \dots, x_0\} = \{(0,0), (0,1), \dots, (x_0, x_0 - 1), (x_0, x_0)\}}$, where the first value in each 2-tuple corresponds to the value of x_t and the second corresponds to the value of y_t.

5b *Try to describe the communicating classes in a compact manner. If not possible, constrain to a small fixed x_0.*

To help us understand the communicating classes a bit better, it is helpful to refer to the heat map in section 5c. We note that, as previously mentioned, the transition probability matrix is assembled from $(x_0+1)\times(x_0+1)$ submatrices of size $x_0+1\times x_0+1$. Each row of submatrices corresponds to y_t, and each column of submatrices corresponds to y_{t+1}. Each row of each submatrix corresponds to x_t, and each column of each submatrix corresponds to x_{t+1}. Therefore, when looking at the heatmap, we can make the following interpretation:

- The top left corner of each submatrix in the first column of submatrices is 1. That corresponds to $x_{t+1} = 0$. Under the Reed-Frost model, we can only stay at 0 if we arrive at $x_t = 0$ for some t.
- In the first row of submatrices, we have the identity matrix as a submatrix followed by x_0 submatrices of 0. This indicates that $y_{t+1} = 0$ and $x_{t+1} = x_t$ forever.
- The second row of each row of submatrices (except row 1 of submatrices) has two filled in elements. The leftmost element in this row corresponds to when we remain in the current state of $x_{t+1} = x_t$, whereas the rightmost element in this row corresponds to when we move to $x_{t+1} = 0$.
- For row n of each row of submatrices (except row 1 of submatrices), we have n filled in elements, and moving right along the submatrices indicates how much we are decrementing x_{t+1} from x_t.

Let us for instance examine the communication classes of $x_0 = 1$. We first draw a diagram of the *possible* transitions between states in this case. Note each state below is (x_t, y_t).



We have the following results, which we can visually generalise to any x_0:

- If $y_t = 0$, then we stay in the same state;
- If $x_t = 0$, then we move to $x_{t+1} = y_{t+1} = 0$.
- Otherwise, we can move to any of the states $(x, y) = (0, y)$, $(1, y - 1)$, ... all the way up to $(y, 0)$. This is due to the fact that $y_t = x_{t-1} - x_t$.

However, note that all of these tend towards the top or right hand side of this chain. We either go

to the very left of the chain and recur in the state where $y = 0$, or go to the top of the chain and transition to $(x, y) = (0, 0)$. This means that we have $(x_0 + 1) \times (x_0 + 1) = (x_0 + 1)^2$ communication classes, as there is no two-way communication.

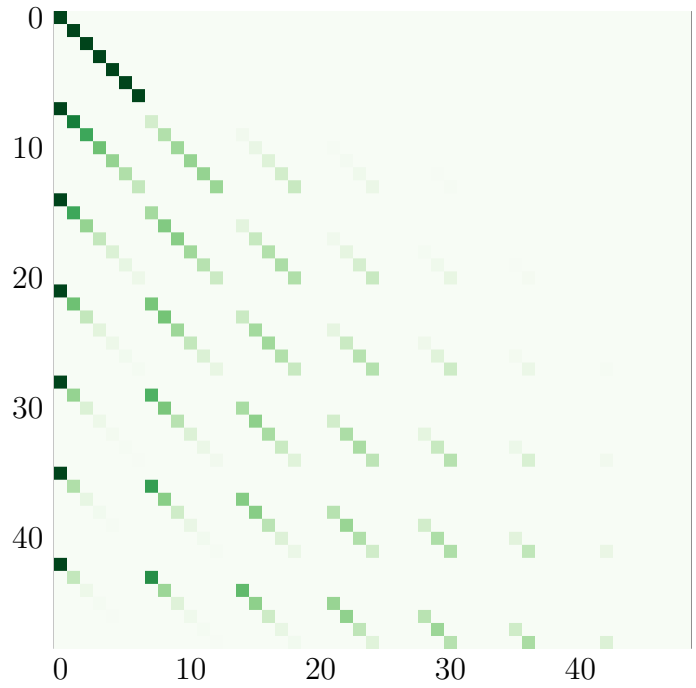
The communication classes are

$$\boxed{\{(0,0)\}, \{(0,1)\}, \ldots, \{(0, x_0 + 1)\}, \ldots, \{(x_0 + 1, x_0 + 1)\}}.$$

The classes on the left, where $y = 0$, are recurrent, i.e. $\boxed{\{(0,0)\}, \{(1,0)\}, \ldots, \{(x_0 + 1, 0)\}}$, and the $\boxed{\text{rest of the classes}}$ are transient. In other words:

$$\{(x, y)\} = \begin{cases} \text{recurrent} & y = 0, \\ \text{transient} & y \neq 0. \end{cases}$$

5c *Plot a heat-map similarly to 3a (you may want to use block-matrices in your software).*



This plot was generated using the following code:

```
# Set up parameters
x0 = 6
y0 = 1
alpha = 0.8

# First subplot
plt.title("Transition matrix for Reed-Frost model, $x_0 = 6, y_0 = 1, \\alpha =
    0.8$")

# Compute the Pij submatrix
def Pij(i, j):
    # Corner cases
    if i == 0 and j == 0:
        return np.eye(x0 + 1, x0 + 1)
    if i == 0:
        return np.zeros((x0 + 1, x0 + 1))
    else:
```

```
        result = np.zeros((x0 + 1, x0 + 1))
        # Construct diagonal
        for k in range(x0 + 1):
            for l in range(x0 + 1):
                if j + l == k:
                    result[k, l] = math.comb(k, l) * (alpha**(i*l)) * ((1-alpha**i
                        )**j)
        return result

P = np.block([[Pij(i,j) for j in range(x0 + 1)] for i in range(x0 + 1)])

plt.imshow(P, cmap='Greens', interpolation='nearest')
plt.xticks([], [])
plt.yticks([], [])

tikzplotlib.save("assignments/stat3004-stochastic/tikz/5c.tikz", \
    axis_width='10cm', axis_height='10cm')
```

5d *Run a Monte-Carlo simulation to obtain an estimate for* $\mathbb{P}(W > 4)$ *similarly to 4c. How does the result compare to 4c? Explain why.* The code for the computation (which is *very* similar to 4c's code snippet) is below:

```
infected_nums = []
iterations = 10**6

for iteration in range(iterations):
    x = 6
    alpha = 0.8
    y = 1
    # Number of infected so far
    infected = 0

    # Loop until y = 0 (no infected at current time t)
    while y != 0:
        new_x = np.random.binomial(x, alpha ** y)
        y = x - new_x
        infected += y
        x = new_x

    infected_nums.append(infected)

# Monte-Carlo estimate of P(W>4) using list comprehensions
print(len([i for i in infected_nums if i > 4])/iterations)
```

Running this Monte Carlo estimate gives us $\boxed{\mathbb{P}(W > 4) \approx 0.256392}$. The reason for the higher probability of large number of infected susceptibles is because in the Reed-Frost model, an individual susceptible at time t is still susceptible at time $t+1$ only if infectious contact is avoided, whereas in the Greenwood model, the cause of infection is not related to the number of infectives. We can also note that we have a probability of α^{Y_k} instead of α for our binomial sample, increasing the value of $\mathbb{P}(W > 4)$.

Question 6

Executive Summary

We construct a model for COVID-19 spread in the case of a motel quarantine system. This motel holds all patients in the case of an infection until it has been removed completely, and releases the patients if no infection is detected.

Upon the application of a modified Reed-Frost stochastic model, we notice an upward trend in infections per person as the capacity of the motel increases. This trend also suggests that quarantining in this motel system results in a higher community transmission rate than what would normally be encountered externally.

Scenario

The local regional town of Regionalville has adopted a motel quarantine system to manage the spread of the new COVID-19 virus. All new arrivals to the town are required to quarantine in the local Regionalville Motel. The motel is constantly filled with new arrivals. All new arrivals have a probability of being infected with COVID-19 of 0.05 (represented in the model by η), and upon arrival all occupants are tested for COVID-19, with results returned the next day. If any patient tests positive, they are removed from the facility and the remaining occupants are held in the motel until no one tests positive. If all patients test negative, they are released from the motel and a new batch of patients is accepted.

Model

The Reed-Frost model considers interactions between susceptibles and infectives that affect the rate of infection, in contrast to other simpler models such as the Greenwood model. We define X_t as the number of susceptibles and Y_t as the number of infectives. An individual susceptible at time t is still susceptible at time $t+1$ only if contact with all Y_t infectives is avoided, which has probability α^{Y_t}. Therefore, we have

$$X_{t+1} \sim \text{Bin}(X_t, \alpha^{Y_t}), \quad p_{(x,y)_t,(x_y)_{t+1}} = \binom{x_t}{x_{t+1}} \alpha^{y_t x_{t+1}} (1-\alpha^{y_t})^{y_{t+1}}.$$

For more details on the Reed-Frost model, refer to [1]. We follow a modified Reed-Frost model with $p = 0.1$ and $\beta = 0.05$. This means that we have a probability that there is no infection due to any single infective of $\alpha = 1 - p\beta = 0.995$. We assume that the motel has a capacity of x_0 people, and there are y_t infective people at any time t. This implies that $x_t + y_t \leq x_0$ for any time t, leading to a number of "impossible states". We also have an incoming rate of infections *per person* of $\eta = 0.05$, implying that for the first day in the hotel, the number of infections is distributed $\text{Bin}(x_0, \eta)$.

In order to test this model, we use a Monte Carlo method, running simulations for capacities $x_0 = \{1, 2, \ldots, 10\}$, and with 10^6 iterations/trajectories per simulation. The algorithm is as follows:

- We first draw from $\text{Bin}(x_0, \eta)$ to determine the number of initial infections.
- If there are 0 infections, we get a new batch.
- If there are infections, we continuously simulate community transmission according to the Reed-Frost model (drawing from $\text{Bin}(X_t, \alpha^{Y_t})$) until there are no more infections. Once there are no more infections, we accept a new batch.

The code to run this model is below:

```
iterations = 10**6
per_day = []
per_person = []
```

```
for x0 in range(1, 11):
    infections_total = 0
    for iteration in range(iterations):
        # Initial parameters
        x = x0
        y = 0
        eta = 0.05
        p = 0.1
        beta = 0.05
        alpha = 1 - p * beta
        # Number of infected so far
        infected = 0
        # Number of infections on "first" day (based on external)
        # Here, "first" refers to a new batch of people, where the infection
        # rate no longer depends on community transmission but external transmission.
        y = np.random.binomial(x, eta)
        infections_total += y
        # Loop until there are no infected
        # Note: if there are no infections, this while loop won't be run at all
        while y > 0:
            # Use Reed-Frost model to calculate infections in motel
            new_x = np.random.binomial(x, alpha ** y)
            y = x - new_x
            infections_total += y
            x = new_x
    # Infections per day, infections per person
    pd = round(infections_total/iterations, 5)
    pp = round(infections_total/(x0 * iterations), 5)
    per_day.append(pd)
    per_person.append(pp)
    print(pd, pp)

# Plot in scatter plot
x0s = np.linspace(1, 10, 10)

plt.title("Expected no. of infections per person under modified Reed-Frost Model (Monte Carlo
    with $10^6$ iterations)")
plt.xlabel('Capacity ($x_0$)')
plt.ylabel('\# infections per person')
plt.scatter(x0s, per_person)

tikzplotlib.save("assignments/stat3004-stochastic/tikz/6.tikz", \
    axis_width='8cm', axis_height='6cm')
```

Assumptions

We have to make a number of simplifying assumptions, some of which may be oversimplifying and unrealistic, and may require some more detailed analysis. These assumptions are detailed below.

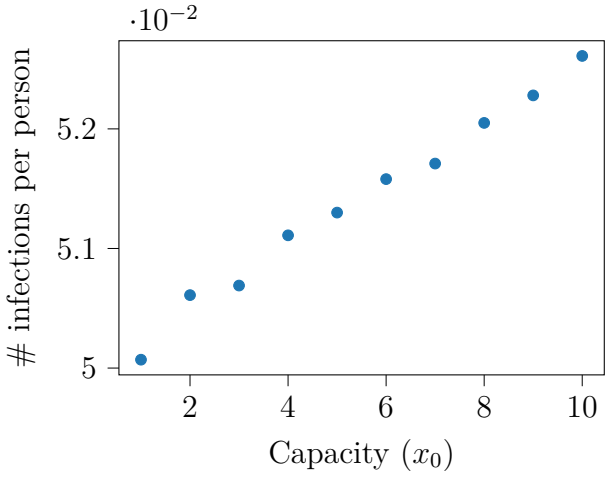
- We assume that there is no immunity to the COVID-19 virus, and that every person has an equal chance of getting the disease. This is an unrealistic assumption. In reality, there are a number of factors that have been established to affect infection rate, including age, sex and medical history. The exact effects that these characteristics have on COVID-19 rates are still under investigation and hence this model requires more work in this area.
- We assume that there is a steady flow of arrivals into the motel. Due to the overwhelming travel restrictions worldwide, the number of new people arriving in the town may ebb and flow as restrictions vary.
- We assume that there are adequate health care facilities that any infected patients are able to be transported to in the event of a positive test. In reality, the health resources of a small regional town may be exhausted relatively quickly.
- We assume that all new arrivals will actually quarantine in the motel instead of heading home. As seen in many real-world situations, this may not be the case.
- We assume that there are enough tests to test everyone in the motel, but as tests are already

in short supply worldwide and are needing to be rationed, this is also unlikely.

Results

We ran our modified Reed-Frost model using a Monte Carlo simulation, with 10^6 iterations, and the capacity of the hotel ranging from $x_0 = \{1, 2, \ldots, 10\}$. Here are the results:

Expected no. of infections per person under modified Reed-Frost Model
(Monte Carlo with 10^6 iterations)



x_0	Per day	Per person
1	0.05007	0.05007
2	0.10121	0.05061
3	0.15208	0.05069
4	0.20445	0.05111
5	0.25650	0.05130
6	0.30951	0.05158
7	0.36195	0.05171
8	0.41641	0.05205
9	0.47052	0.05228
10	0.52609	0.05261

Conclusion

The increase in number of infections per person appears to be linear over capacity, although more work would need to be done to confirm this hypothesis. However, one thing is clear; increasing the capacity of the quarantine motel increases the expected rate of infections per person under this model. We also note that all of the hotel capacities result in a higher expected rate of infections per person than ordinary community transmission with η. Further work needs to be done in this area, with more detailed models, to verify these hypotheses.

References

[1] Daryl Daley and Joseph Gani. *Epidemic modelling: an introduction*. first. Cambridge studies in mathematical biology. Cambridge University Press, 1999.