STAT3004: Project 1
Epidemic Modelling

Alistair Falconer

All codes were done in Julia and are provided in the appendices at the end of this document. I'll also upload them to Github on 30/4 at: `https://github.com/alistairfalconer/STAT3004-Project-1`

Question 1

We're given the transition probabilities as:

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

We note that X_{t+1} can take on discrete values from 0 to x_t, assuming there is an infection present, $y_t > 0$, we can find the conditional expectation as:

$$\begin{aligned}
\mathbb{E}\left[X_{t+1} \mid X_t = x_t\right] &= \sum_{k=1}^{x_t} k \mathbb{P}\left[X_{t+1} = k \mid X_t = x_t\right] && (2)\\
&= \sum_{k=0}^{x_t} k \binom{x_t}{k} \alpha^k (1-\alpha)^{x_t - k} && (3)\\
&= \sum_{k=1}^{x_t} k \frac{x_t!}{k!(x_t-k)!} \alpha^k (1-\alpha)^{x_t-k} && (4)\\
&= x_t \alpha \sum_{k=1}^{x_t} \frac{(x_t-1)!}{(k-1)!((x_t-1)-(k-1))!} \alpha^{k-1} (1-\alpha)^{(x_t-1)-(k-1)} && (5)\\
&= x_t \alpha \sum_{k=0}^{x_t-1} \frac{(x_t-1)!}{k!((x_t-1)-k)!} \alpha^{k} (1-\alpha)^{(x_t-1)-k} && (6)\\
&= x_t \alpha (\alpha + (1-\alpha))^{x_t - 1} = x_t \alpha && (7)
\end{aligned}$$

Where the final expression comes from substituting the binomial formula. This gives us a recursive relation on the expectation, which we can use to evaluate:

$$\begin{aligned}
\mathbb{E}\left[X_{t+1} \mid X_0 = x_0\right] &= \mathbb{E}\left[\mathbb{E}\left[X_{t+1} \mid X_t\right] \mid X_0 = x_0\right] && (8)\\
&= \mathbb{E}\left[\alpha X_t \mid X_0 = x_0\right] && (9)\\
&= \alpha \mathbb{E}\left[X_t \mid X_0 = x_0\right] && (10)
\end{aligned}$$

The general form follows from this:

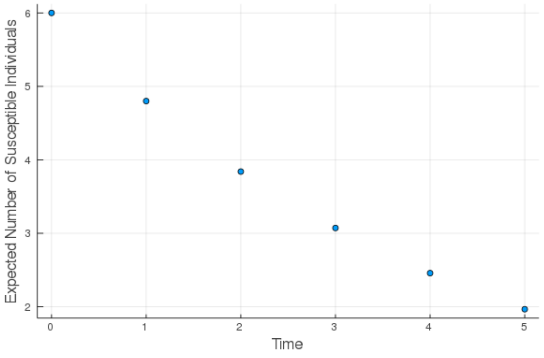
$$\mathbb{E}\left[X_1 \mid X_0 = x_0\right] = \alpha x_0, \qquad \mathbb{E}\left[X_2 \mid X_0 = x_0\right] = \alpha^2 x_0, \qquad \ldots, \qquad \mathbb{E}\left[X_t \mid X_0 = x_0\right] = \alpha^t x_0 \tag{11}$$

It's simpler to find the expected number of infectives from this value, from linearity of expectation:

$$\begin{aligned}
\mathbb{E}\left[Y_t \mid X_0 = x_0\right] &= \mathbb{E}\left[X_{t-1} - X_t \mid X_0 = x_0\right] && (12)\\
&= \mathbb{E}\left[X_{t-1} \mid X_0 = x_0\right] - \mathbb{E}\left[X_t \mid X_0 = x_0\right] && (13)\\
&= \alpha^{t-1} x_0 - \alpha^t x_0 = \alpha^{t-1}(1-\alpha) x_0 && (14)
\end{aligned}$$

b) A numerical implementation of the above relation was performed, code in Appendix A:

Figure 1: Evolution of expected number of susceptible individuals over time under the Greenwood model with $x_0 = 6$, $\alpha = 0.8$.



Question 2

a) We start with the transition probabilities of the Reed-Frost model:

$$\mathbb{P}\left[(X,Y)_{t+1} = (x,y)_{t+1} \mid (X,Y)_t = (x,y)_t\right] = \binom{x_t}{x_{t+1}} \alpha^{y_t x_{t+1}} (1-\alpha^{y_t})^{y_{t+1}} \tag{15}$$

Following the derivation 1a up to equation 7 with the substitution $\alpha \to \alpha^{y_t}$ will give the desired result for the number of susceptibles:

$$\mathbb{E}\left[X_{t+1} \mid X_t = x_t, Y_t = y_t\right] = \alpha^{y_t} x_t \tag{16}$$

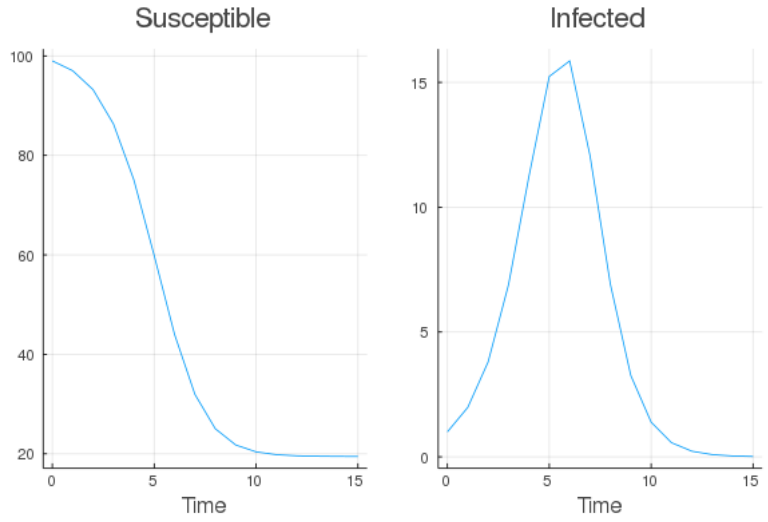
Note that we can no longer turn this into a general form because we now have different transition probabilities associated with different values of Y_t, so the state space becomes much more complicated. We can find the expectation of the number of infectives again by using the relation $Y_{t+1} = X_t - X_{t+1}$:

$$\mathbb{E}\left[Y_{t+1} \mid X_t = x_t, Y_t = y_t\right] = \mathbb{E}\left[X_t \mid X_t = x_t, Y_t = y_t\right] - \mathbb{E}\left[X_{t+1} \mid X_t = x_t, Y_t = y_t\right] \tag{17}$$

$$= x_t - \alpha^{y_t} x_t = x_t(1-\alpha^{y_t}) \tag{18}$$

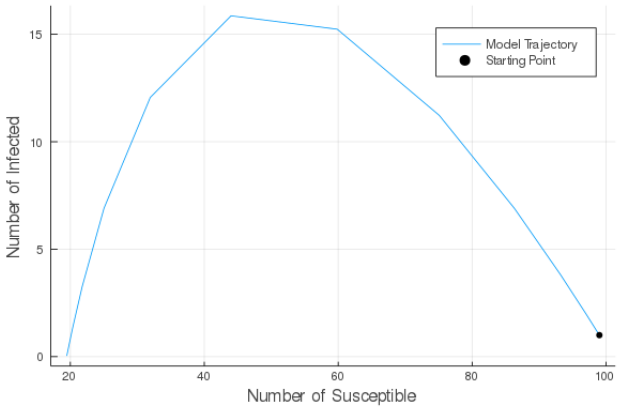
b) Code in Appendix B:

Figure 2: Evolution of expected number of susceptibles and infectives over time under the Reed-Frost model with $x_0 = 99$, $y_0 = 1$ and $\alpha = 0.98$.



The trajectory of expected values is shown in figure 3, the code is in appendix C:

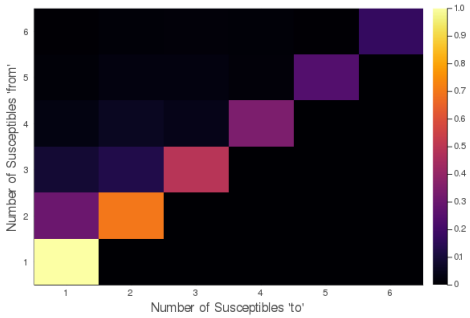
Figure 3: Joint trajectory of expected number of susceptibles and infectives over time under the Reed-Frost model with $x_0 = 99$, $y_0 = 1$ and $\alpha = 0.98$.



Question 3

a) Shown here is the heat-map of the matrix for $x_0 = 5$, $\alpha = 0.7$, examples with $x_0 = 10$ and $x_0 = 20$ can be found in appendix D

Figure 4: Heatmap of probability transition matrix with $x_0 = 5$, $\alpha = 0.7$, note that the y axis is reversed from how a matrix would typically be written.



b) We see that $\mathbb{P}[0 \to 0] = 1$, which means that $x = 0$ is a recurrent state, it also means that this state doesn't communicate with any other, and therefore $\{0\}$ is it's own recurrent class. For all other states we see that there is some probability of moving to a lower state as more people become infected, and no probability of moving to a higher state. Note that if we ever have $x_{t+1} = x_t$ then it's implied that $y_{t+1} = 0$, and therefore the infection has ended. In this sense the model is only Markovian for as long as the infection lasts, as once the virus ends the transition matrix should be replaced with the identity.

As all states only commuicate in a single direction, we conclude that each state makes up its own communicating class, and all but $\{0\}$ are transient, as each $x_t > 0$ has some positive probability of transitioning to a state $x_{t+1} < x_t$ such that there is 0 probability of getting back to x_t. Note that the matrix they've presented hides the fact that if we ever have a stable transition, $X_{t+1} = X_t$, then we'd essentially have a recurrent state as the infection would have ended. It would be possible to construct a transition matrix with $2(x_0 + 1)$ states. For each possible number of susceptibles $x = 1, ..., x_0 + 1$ we'd have two states, $(x, 1)$ and $(x, 0)$, representing that whether or not any infectives are present. In this case each state $(x, 0)$ would be recurrent and each state $(x, 1)$ transient, but that's not what the textbook did.

Therefore there is one recurrent class, $\{0\}$ and x_0 transient classes, $\{1\}, ..., \{x_0\}$.

c) The definition of expectation on a discrete variable:

$$\mathbb{E}\left[X_t \mid X_0 = x_0\right] = \sum_{k \in \{X_t\}_{\omega \in \Omega}} \left(k \mathbb{P}\left[X_t = k \mid X_0 = x_0\right]\right) \tag{19}$$

Where $\{X_t\}_{\omega \in \Omega}$ indicates the set of possible values that X_t can take. We know that from the properties of the model that X_t can only take integer values $0 \leq X_t \leq x_0$, and given that we know from the properties of a Markov process that $\mathbb{P}\left[X_t = k \mid X_0 = x_0\right] = P^t_{x_0,k}$, we can say:

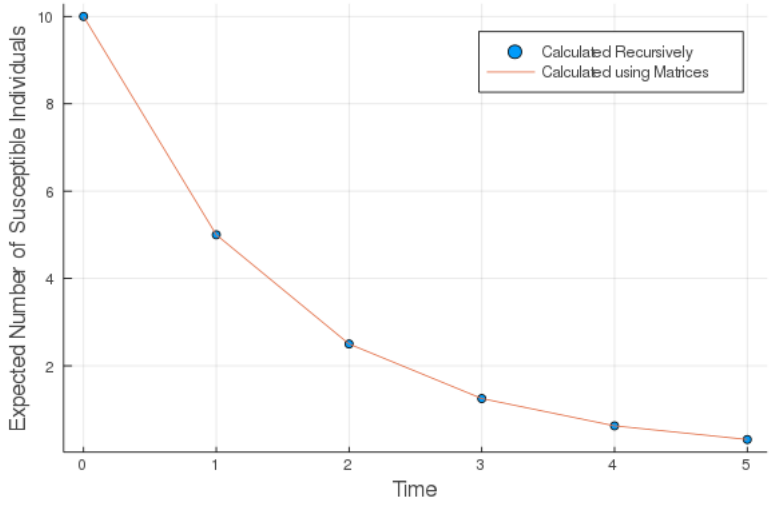
$$\mathbb{E}\left[X_t \mid X_0 = x_0\right] = \sum_{k=0}^{x_0} \left(k P^t_{x_0,k}\right) \tag{20}$$

$$= \sum_{k=0}^{x_0} \left[\begin{pmatrix} 0 & 0 & \dots & 1 \end{pmatrix} P^t\right]_k \cdot k \tag{21}$$

$$= \begin{pmatrix} 0 & 0 & \dots & 1 \end{pmatrix} P^t \begin{pmatrix} 0 \\ 1 \\ \vdots \\ x_0 \end{pmatrix} \tag{22}$$

The row vector introduced at equation 21 has length $x_0 + 1$, all elements but the last are 0. As seen in figure 5 the results match those obtained by using the recursive method on the expectations.

Figure 5: Evolution of expected number of suscepibles over time with $x_0 = 10$, $\alpha = 0.5$, this was calculated using both the recursive relation on the expectations and using the matrix methid detailed above.



d) In part c) we used the fact that transition probabilities can be expressed as a function of (x_t, x_{t+1}), it isn't possible to do this with (y_t, y_{t+1}) so solving the expected number of infectives with a matrix method is far more complicated than the number of susceptibles. We use a modified formula 4.1.2 from EM:

$$p_{(a,b),(k,\ell)} = \mathbb{P}\left[(X,Y)_{t+1} = (k,\ell) \mid (X,Y)_t = (a,b)\right] = \begin{cases} \binom{a}{k} \alpha^a (1-\alpha)^k & k + \ell = a \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

We need to keep track of the values X_t, so we use a dual state space (x_t, y_t), with the following $(x_0+1)^2 \times (x_0+1)^2$ transition matrix:

$$P = \begin{pmatrix} p_{(0,0),(0,0)} & p_{(0,0),(0,1)} & \cdots & p_{(0,0),(0,x_0)} & p_{(0,0),(1,0)} & \cdots & p_{(0,0),(x_0,x_0)} \\ p_{(0,1),(0,0)} & p_{(0,1),(0,1)} & \cdots & p_{(0,1),(0,x_0)} & p_{(0,1),(1,0)} & \cdots & p_{(0,1),(x_0,x_0)} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ p_{(0,x_0),(0,0)} & p_{(0,x_0),(0,1)} & \cdots & p_{(0,x_0),(0,x_0)} & p_{(0,x_0),(1,0)} & \cdots & p_{(0,x_0),(x_0,x_0)} \\ p_{(1,0),(0,0)} & p_{(1,0),(0,1)} & \cdots & p_{(1,0),(0,x_0)} & p_{(1,0),(1,0)} & \cdots & p_{(1,0),(x_0,x_0)} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ p_{(x_0,x_0),(0,0)} & p_{(x_0,x_0),(0,1)} & \cdots & p_{(x_0,x_0),(0,x_0)} & p_{(x_0,x_0),(1,0)} & \cdots & p_{(x_0,x_0),(x_0,x_0)} \end{pmatrix} \tag{24}$$

This (very sparse) matrix is a Markov transition matrix, knowing that Y_t can take integer values from 0 to x_0 we can then evaluate the expectations as:

$$\mathbb{E}\left[Y_t \mid X_0 = x_0\right] = \sum_{k=0}^{x_0} k\mathbb{P}\left[Y_t = k \mid X_0 = x_0\right] \tag{25}$$

$$= \sum_{k=0}^{x_0} \left[\begin{pmatrix} 0 & 0 & \ldots & 1 & 0 & \ldots & 0 \end{pmatrix} P^t\right]_k \cdot k \tag{26}$$

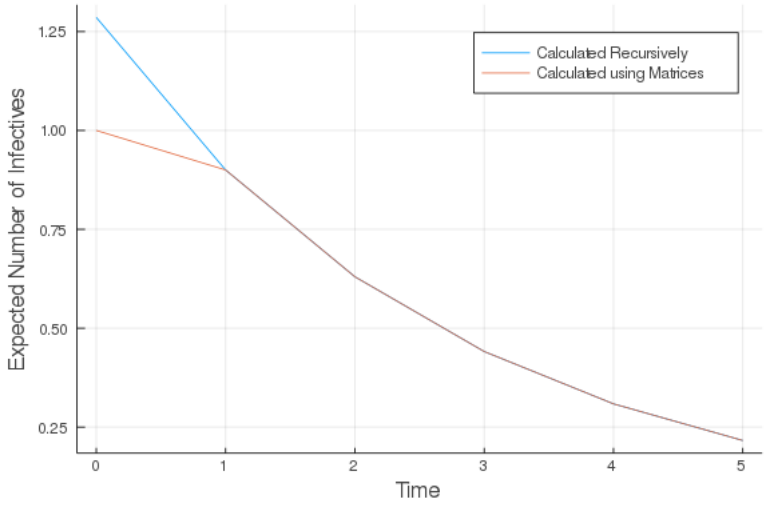
$$= \begin{pmatrix} 0 & 0 & \ldots & 1 & 0 & \ldots & 0 \end{pmatrix} P^t \vec{v} \tag{27}$$

Where the row matrix introduced in equation 26 has length $(x_0+1)^2$ and has a 1 in the position corresponding to $(x_0, 1)$, representing the assumption that $X_0 = x_0$, $Y_0 = 1$. $\vec{v}$ is a column vector that represents the number of infectives in each state, so it counts out $0 : x_0$ a total of (x_0+1) times.

$$\vec{v} = \begin{pmatrix} 0, 1, 2, ..., x_0, 0, 1, 2, ...x_0, ...x_0 \end{pmatrix}^T \tag{28}$$

The code for this implementation is given in appendix F, the plot of results is shown in figure 6

Figure 6: Expected number of infectives over time calculated using both methods. Note that they mismatch at $t = 0$ because this value is given by the initialising assumption on Y_0 rather than being calculated within the model. The parameters used are $x_0 = 6$, $\alpha = 0.8$



Question 4

a) We have:

$$p_j^t = \mathbb{P}\left[X_t = j, Y_t > 0\right] \qquad p_{ij} = \mathbb{P}\left[X_{t+1} = j \mid X_t = i, Y_t > 0\right] \tag{29}$$

Each possible state $x_{t-1} = i$ has some probability of transitioning to $x_t = j$, so we can find the probability p_j^t by summing through the probabilities of all possible previous states p_i^{t-1} weighted by the probability of the transition $i \to j$.

$$p_j^t = \sum_{i=0}^{x_0} p_i^{t-1} p_{ij} \tag{30}$$

We note the assumption $Y_t > 0$ implies that $x_t < x_{t-1}$, or otherwise $p_{i,j} = 0$ for $i < j + 1$. Also as there has been at least an infection in every time-step so far, $p_i^{t-1} = 0$ for $i > x_0 - (t-1)$, we can therefore remove terms from the summation for:

$$p_j^t = \sum_{i=j+1}^{x_0-(t-1)} p_i^{t-1} p_{ij} \tag{31}$$

b) We have that T is the earliest time without infectives and W is the total number of infections, where $\Gamma(k, n \mid x_0)$ gives the probability of $W = k$ and $T = n$ given infection starts with x_0 susceptibles. For the infection to continue there must be at least one infection per time step, so T can take values $n = 1, ..., x_0$, where x_0 is the earliest time where we're guaranteed to have run out of susceptibles or infectives. Similarly, the total number of infections cannot exceed the original number of susceptibles, so $k = 0, ..., x_0$.

$$\mathbb{P}\left[W > 4 \mid x_0\right] = \sum_{k=5}^{x_0} \sum_{n=1}^{x_0} \Gamma(k, n \mid x_0) \tag{32}$$

A numerical calculation based on the formulae given in the textbook and the problem description is provided in appendix G, giving a 12.9% chance of more than 4 infections with $x_0 = 6$, $\alpha = 0.8$.

c) The Monte Carlo simulation of the Greenwood model is given in appendix H, giving a 13.0% chance of more than 4 infections with $x_0 = 6$ and $\alpha = 0.8$ after 10^6 simulations.

d) Following EM-4, we partition the transition matrix P as:

$$P = \bar{P} + Q \tag{33}$$

Where Q is a diagonal matrix $Q = \text{diag}(1, \alpha, ..., \alpha^{x_0})$ and $\bar{P}$ only has non-zero entries below the diagonal. We observe that the transitions represented by the probabilities in $\bar{P}$ are always down transitions, with a change in the number of susceptibles and/or infectives. The transitions represented in Q however are all stable, there was no change in the number of susceptibles or infectives, and therefore the infection is over. If the first time with no infections is T, then this means there must have been $T - 1$ down-transitions followed by a single stable transition as the last round of infectives are removed from the model. Mathematically this gives:

$$\mathbb{P}\left[T = t, X_t = k\right] = (A' \bar{P}^{t-1} Q)_k \tag{34}$$

Where A' encodes the initial probability distribution, typically this is the $x_0 + 1$-unit vector with the final element a 1. In the textbook they're interested in T so they use:

$$\mathbb{P}\left[T = t\right] = \sum_{k \in X_t} \mathbb{P}\left[T = t, X_t = k\right] = A' \bar{P}^{t-1} Q E \tag{35}$$

With E being a x_0+1-column vector full of 1's, here I've used the shorthand $k \in X_t$ where X_t is a discrete-valued random variable to indicate that k iterates through the possible values of X_t. We're interested in $W = (x_0 - X_T)$, which we can find through X_T, so we do the calculations:

$$\mathbb{P}[X_T = x] = \sum_{t \in T} \mathbb{P}[T = t, X_t = x] \tag{36}$$

$$= \sum_{t \in T} (A' \bar{P}^{t-1} Q)_x \tag{37}$$

$$= \sum_{t \in T} (A' \bar{P}^{t-1} Q E_x) \tag{38}$$

Where E_x is a x_0+1-column vector where the only non-zero element is a 1 in the element corresponding to x. We can simplify this by noting that all matrices are constant in time:

$$\mathbb{P}[X_T = x] = A' \sum_{t \in T} (\bar{P}^{t-1}) Q E_x \tag{39}$$

$$= A'(1 - \bar{P})^{-1} Q E_x \tag{40}$$

The simplification comes from noting that T can take values $1, ..., x_0 + 1$, and that $\bar{P}^{x_0+1} = \bar{0}$, so:

$$\sum_{t=1}^{x_0+1} (\bar{P}^{t-1})(1 - \bar{P}) = (1 - \bar{P}) + (\bar{P} - \bar{P}^2) + ... + (\bar{P}^t + \bar{P}^{t+1}) \tag{41}$$

$$= 1 + \bar{P}^{t+1} = 1 \tag{42}$$

As all internal terms cancel from the expansion. This means $\sum_{t=1}^{x_0+1} (\bar{P}^{t-1}) = (1 - \bar{P})^{-1}$. We then calculate the probability generating function as:

$$\mathcal{G}_{X_T}(z) = \mathbb{E}_{X_T}(z^{X_T}) = \sum_{x \in X_T} (z^x \mathbb{P}[X_T = x]) \tag{43}$$

$$= \sum_{x \in X_T} z^x A'(1 - \bar{P})^{-1} Q E_x \tag{44}$$

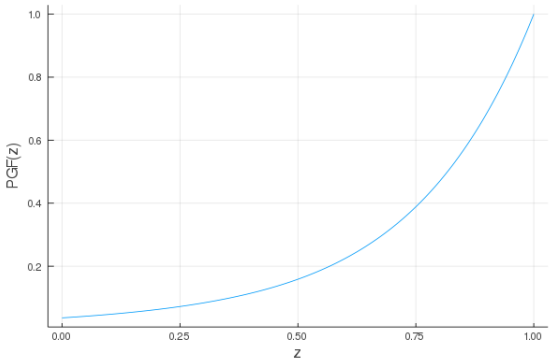
$$= A'(1 - \bar{P})^{-1} Q E(z) \tag{45}$$

Where $E(z)$ is a column vector with elements $E_i = z^i$. Then from the properties of the probability generating function:

$$\mathbb{P}[W > 4] = \sum_{x=0}^{x_0-5} \mathbb{P}[X_t = x] = \sum_{x=0}^{x_0-5} \frac{1}{x!} \mathcal{G}^{(x)}_{X_T}(z) \tag{46}$$

A numerical implementation is given in appendix I, giving a 13.0% chance of greater than 4 infections with $x_0 = 6$ and $\alpha = 0.8$. This is very consistent with the values obtained in parts b),c) which is nice.

Figure 7: The probability generating function $\mathcal{G}_{X_t}(z)$ calculated above.



Question 5

a) The state space is actually the same as that used in question 3d, the set of all combinations of possible values $(X, Y)_t$, so:

$$S = \{(x, y) : x = 0, ..., x_0, y = 0, ..., x_0\} \tag{47}$$

Note that it would be technically possible to have $y_0 > x_0$ which would require a larger state-space to describe it, but we discount this possibility as this wouldn't be a realistic application of the model.

b) We note that any state with $y = 0$ is recurrent, as no more infections can happen, so :

$$\mathbb{P}\left[(X, Y)_{t+1} = (x, 0) \mid (X, Y)_t = (x, 0)\right] = 1 \tag{48}$$

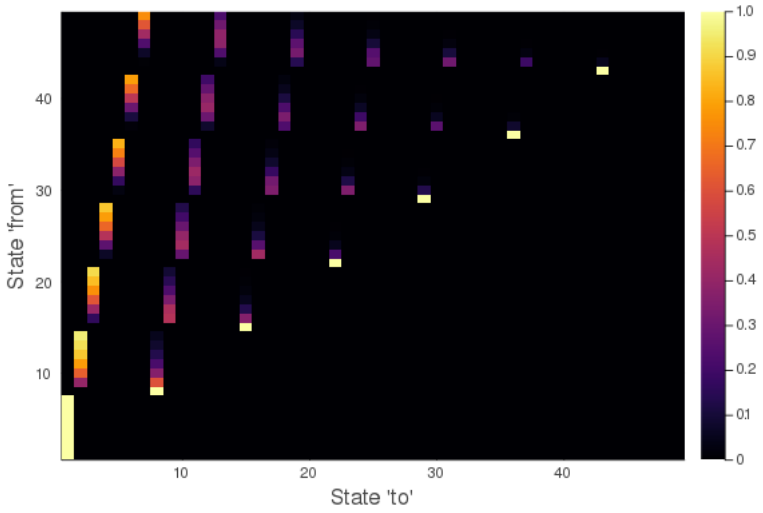
We also know that the number of susceptibles can't increase, so any communicating class must consist of tuples with the same number of susceptible individuals. Given that the number of susceptibles is constant, the number of infectives can't increase, this implies that any communicating class must also have a constant number of infectives. To summarise:

$$x_1 < x_2 \Rightarrow (x_1, y_1) \not\to (x_2, y_2) \tag{49}$$
$$y_1 < y_2 \Rightarrow (x, y_1) \not\to (x, y_2) \tag{50}$$

Together these mean that each $\{(x, y)\}$ forms its own communicating class.

c) The transition matrix was constructed numerically using the code in appendix J, the heatmap is shown in figure 8

Figure 8: A heatmap of the transition probabilities under the Reed-Frost model with $x_0 = 6$, $\alpha = 0.6$, similar to the matrix in equation 24, note that the y axis is reversed, this corresponds to a lower-triangular matrix. Each column on the graph corresponds to states with a given number of susceptibles and a variable number of infectives, the bright points along the identity are transitions of the form $(x, 0) \to (x, 0)$



d) The Monte Carlo simulation of the Reed-Frost model is given in appendix K, giving a 25.6% chance of more than 4 infections with $x_0 = 6$ and $\alpha = 0.8$ after 10^6 simulations. This is a significantly higher chance than obtained under the Greenwood model because the Reed-Frost model is more pessimistic. A key assumption in Greenwood is that the probability of infection is independent of the current number of infected, whereas in the Reed-Frost model the chance of new infections increases with existing infections. This means that the Reed-Frost model has at best an equal rate of infection as Greenwood, and worse for any $y_t > 1$, so we should expect a higher number of total infections.

We can compare this to the result of c) by noting that it's possible to extract $\mathbb{P}[W > 4]$ from the transition matrix as:

$$\mathbb{P}[W > 4] = A' P^T E \tag{51}$$

Where A' is a row vector corresponding to the initial state, E is a vector with 1's corresponding to every column in P that represent a number of susceptibles fewer than $x_0 - 4$, and T is an integer sufficiently high enough that $P^T = P^{T+1}$. In this case we have $x_0 = 6$, so the system must reach a stationary state by $T = 7$, we have that:

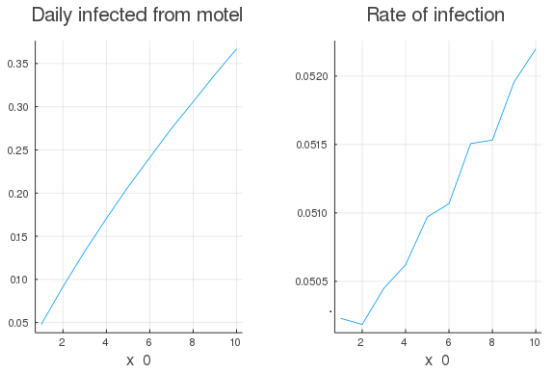
$$A' P^7 E = 25.6\% \tag{52}$$

Which matches the value obtained in the Monte Carlo simulation. Cool.

Executive Summary

We notice that both the expected number of infectives leaving the motel each day and being sent into the towns health system and the expected rate of infection among arrivals to the motel increase with x_0 as shown in figure 9. More precise results are given in figure 10.

Figure 9: Spread of virus within new arrivals with respect to the number of arrivals accepted into the motel in each generation. Increasing the size of the generations will increase both the number of infectives introduced into the towns health system from this motel each day and the total ratio of arrivals we expect to become infected in the motel system.



The Scenario

A small town recieves more than x_0 new arrivals per day from a large external population, of which a fraction η are infected with COVID-19 at any time. The town government has a policy where all new arrivals are isolated in a motel and tested for the disease, the tests take a day to give results. If all occupants of the motel test negative then they are released and a new round of x_0 newcomers are accepted. If any tests come back positive, those individuals are removed and placed into individual isolation, all others are kept in the motel and tested again for the possibility that they were infected while waiting for results. No new occupants are accepted until the motel has been emptied.

Assumptions

- Implicit in the Reed-Frost model is the assumption that each susceptible person is equally likely to become infected, in reality different behaviours would lead to different infection probabilities
- We have assumed that there will be no new arrivals to the motel if there is a chance that one of the current occupants is infected
- We have assumed that the rate of infection outside of the town is a constant, η, whereas in an active infection you expect this rate to change substantially with time
- We have assumed that the test for Covid-19 is 100% accurate, reliable and available

The Model

A Monte Carlo simulation was used to numerically determine the rate of infection among arrivals to the motel, with the spread of infection within the motel modelled using the Reed-Frost model. We retrieve both Y, the expected number of infective people leaving the motel each day, and Z, the expected rate of infection among those processed through the motel. If we assume that all arrivals to the town are being processed in identical motels, Z will be the same as the average rate of daily introduction of infections to the towns. That is if there are $x > x_0$ total arrivals each day, we expect that there will be xZ infectives leaving the motel system each day, because the average number of people released from the motels in a day must be equal to the average rate of arrivals to the motel in the high time limit.

Data: N trials, η chance of existing infection, x_0 arrivals to motel per day, p chance of contact between individuals in motel, β chance of contact between infected and susceptible resulting in new infection
Result: Y; expected number of infectives leaving the motel each day, Z; expected rate of infection among new arrivals

$\alpha \leftarrow 1 - p\beta$;
$\tilde{Y} \leftarrow \{\}$;
for $n \leftarrow 1$ **to** N **do**
  $y_1 \leftarrow$Result of x_0 Bernoulli trials with success rate η;
  $x_1 \leftarrow (x_0 - y_1)$;
  $i \leftarrow 1$;
  while *true* **do**
    if $y_i = 0$ **then**
      break
    end
    $x_{i+1} \leftarrow$ Result of x_i Bernoulli trials with success rate α^{y_i};
    $y_{i+1} \leftarrow x_i - x_{i+1}$;
    $i \leftarrow i + 1$;
  end
  $\tilde{Y} \leftarrow \tilde{Y} \cup (\{y_j\}_{j=1,\dots,i})$;
  $Z_n \leftarrow \frac{1}{x_0} \sum_{j=1}^{i} (y_j)$
end
$Y \leftarrow mean(\tilde{Y})$;
$Z \leftarrow mean(\{Z_n\}_{n=1,\dots,N})$

Algorithm 1: Monte Carlo Simulation of Reed-Frost model of infections within the motel

Results

Both rates Y and Z depend on the number of arrivals to the motel in each day, x_0, are given in figure 10. This was calculated using algorithm 1 and the parameters:

$$N = 10^6 \qquad \eta = 0.05 \qquad p = 0.1 \qquad \beta = 0.05$$

Figure 10: The growth of parameters Y and Z with x_0.

x_0	Y	Z	x_0	Y	Z
1	0.0478	0.0502	6	0.2410	0.0511
2	0.0914	0.0502	7	0.2747	0.0515
3	0.1323	0.0504	8	0.3056	0.0515
4	0.1704	0.0506	9	0.3371	0.0520
5	0.2070	0.0510	10	0.3669	0.0522

As would be expected the motel provides an additional vector for the disease to spread, so increasing x_0 will increase the spread of the disease among arrivals. Note that this model fails to capture several aspects of the infection, for instance human-surface contact has been identified as a substantial factor in the spread of Covid-19, it's unclear if this can be properly described by only modelling human-human contacts. Future investigations may find use in investigating an infection rate independent of number of infectives. Additionally further context is needed to evaluate the specific pressures that will be placed on the towns health system, the model only remains valid as long as the government is able to keep effectively isolating known infectives, performing tests, etc.

A Code for Q1b

```
#Plot infected number of susceptibles over time under the Greenwood model,
# using recursive relation
function Q1(x0::Int,α::Float64;T=5::Int)
    xVals = Float64[x0]
    for t = 1:T
        push!(xVals,α*xVals[t])
    end

    scatter(0:T,xVals,label=:none)
    #title!("Greenwood Model")
    xlabel!("Time")
    ylabel!("Expected Number of Susceptible Individuals")
end
```

B Code for Q2b Seperate Trajectories of Expectation Values

```
    #Plot expected number of susceptibles and infectives over time under the
    #Reed-Frost model using the recursive relation on the expectation
function Q2()
    α = 0.98
    x0 = 99.0
    y0 = 1.0
    T = 15
    xVals = [x0]
    yVals = [y0]
    for t = 1:T
        push!(xVals,α^yVals[t]*xVals[t])
        push!(yVals,(1-α^yVals[t])*xVals[t])
    end

    plot(0:T,[xVals,yVals],layout=(1,2),title=["Susceptible" "Infected"],xlabel =
["Time" "Time"],label=:none)
end
```

C Code for Q2b Joint Trajectory of Expectation Value

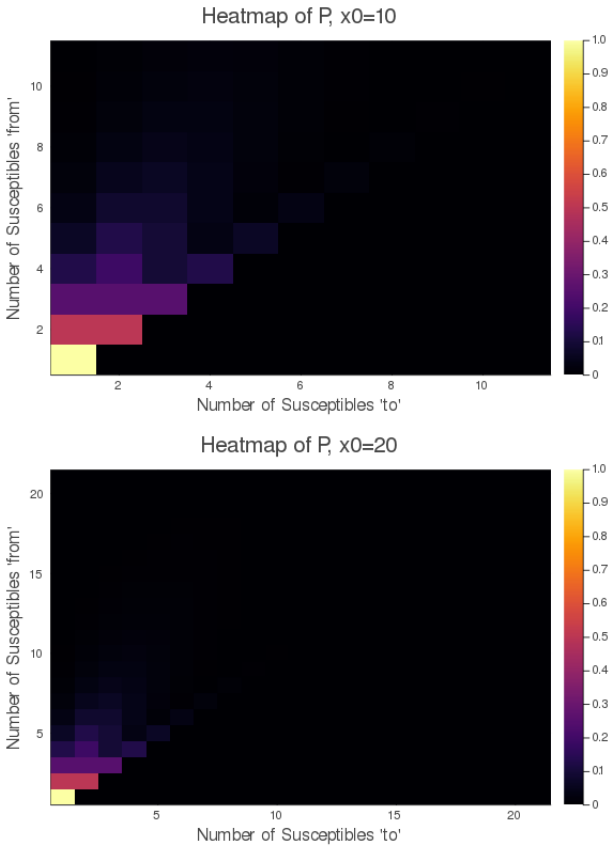
```
    #plot joint trajectory of expectation  of susceptibles and infectives
function Q2other()
    α = 0.98
    x0 = 99.0
    y0 = 1.0
    T = 15
    next(x,y) = [x*α^y,x*(1-α^y)]# update formula
    traj = [[] for _ in 1:T]
    traj[1] = [x0,y0]
    for t in 2:T
        traj[t] = next(traj[t-1]...)
    end
    plot(first.(traj),last.(traj),label="Model Trajectory")
    scatter!([traj[1][1]],[traj[1][2]],c=:black,label="Starting Point")
    xlabel!("Number of Susceptible")
    ylabel!("Number of Infected")
end
```

D Figures and code for Q3a

```
    #Generate heatmap of transition probability matrix under Greenwood model
function Q3a(x0,α)
```

```
    P = zeros(x0+1,x0+1)

    for i in 1:x0+1
        P[i,i] = α^(i-1)
        for j in 1:i-1
            P[i,j] = binomial(i-1,j-1)*(1-α)^(i-1)*α^(j-1)
        end
    end
    heatmap(P)
    #title!("Heatmap of P, x_0=$x0,α=$α")
    ylabel!("Number of Susceptibles 'from'")
    xlabel!("Number of Susceptibles 'to'")
end
```



E Code for 3c

```
    # Use numerical solution to analytical matrix method to evaluate expected
    #number of susceptibles over time under Greenwood model
function Q3c(x0,α;T=5::Int)
    P = zeros(x0+1,x0+1)

    for i in 1:x0+1
        P[i,i] = α^(i-1)
        for j in 1:i-1
            P[i,j] = binomial(i-1,j-1)*(1-α)^(i-j)*α^(j-1)
        end
    end
    e = zeros(1,x0+1)
    e[x0+1] = 1

    v = [x for x in 0:x0]

    vals = [(e*(P^t)*v)[1] for t = 0:T]
```

```
    plot(Q1(x0,α,T=T),label="Calculated Recursively")
    plot!(0:T,vals,label="Calculated using Matrices")
    #title!("Greenwood Model, x_0=$x0, alpha=$α")
end
```

F Code for 3d

```
    # Use numerical solution to analytical matrix method to evaluate expected
    # number of infectives over time under Greenwood model
function Q3d(x0::Int,xt::Int,α::Float64;T=5::Int,y0=1.0::Float64)
    P = zeros((x0+1)^2,(x0+1)^2)
    States = Dict(i => ((i-1)÷(x0+1),(i-1)%(x0+1)) for i in 1:(x0+1)^2)
    #get table of conversions between matrix index and state
    for i in 1:(x0+1)^2
        for j in 1:(x0+1)^2
            xnow,ynow = States[i]
            xnext,ynext = States[j]
            if xnext + ynext == xnow
                P[i,j] = binomial(xnow,xnext)*α^xnext*(1-α)^ynext
            end
        end
    end
    Indices = Dict(value => key for (key,value) in States) #give state to return index
    e = zeros(1,(x0+1)^2)
    e[Indices[x0,y0]] = 1
    v = [(i-1)%(x0+1) for i in 1:(x0+1)^2]
    vals = [(e*(P^t)*v)[1] for t = 0:T]
    plot(0:T,[α^(t-1)*(1-α)*xt for t in 0:T],label="Calculated Recursively")
    plot!(0:T,vals,label="Calculated using Matrices")
    ylabel!("Expected Number of Infectives")
    xlabel!("Time")
    #title!("Greenwood Model, x_0=$x0, alpha=$α")
end
```

G Code for 4b

```
    # defined in
p1(i,j,α) = binomial(i,j)*(1-α)^(i-j)*α^j #p_{ij} in EM-4

_p2 = Dict()
function p2(j,t,x0,α) #equation 4.1.6 - p_j^t in EM-4
    #println(j,t)
    if t == 0
        return Float64(j==x0)
    end
    if (j,t,x0,α) in keys(_p2)
        return _p2[(j,t)]
    end
    vals=[0.0]
    #[println(vals,p2(i,t-1)*p1(i,j) for i in (j+1):(x0-(t-1)))]
    [push!(vals,p2(i,t-1,x0,α)*p1(i,j,α)) for i in (j+1):(x0-(t-1))]
    res = sum(vals)
    #res = sum([p2(i,t-1)*p1(i,j) for i in (j+1):(x0-(t-1))])
    _p2[(j,t,x0,α)] = res
    return res
end

Γ(k,n,x0,α) = p2(x0-k,n-1,x0,α)*α^(x0-k) #Γ given in problem statement

#Use Γ function to determine probability of more than 4 new infections
function Q4b(x0,α)
    PWG4 = sum([Γ(k,n,x0,α) for k in 5:x0 for n in 1:x0])
    println("Numerical calculations give a $(round(100*PWG4,digits=2))% chance
    of more than 4 total infections")
end
```

H Code for 4c

```
    #Use Monte Carlo simulations to determine probability of more than 4 new infections
function Q4c(x0,α;N=1e6)
    WList = []
    TList = []
    for i in 1:N
        xList = [x0]
        yList = [1]
        while true
            x,y = Infected(xList[end],α)
            push!(xList,x)
            push!(yList,y)
            if x*y == 0
                break
            end
        end
        push!(WList,sum(yList)-1)
        push!(TList,length(xList)-1)
    end
    nWG4 = count(W->W>4,WList)
    println("Under Monte Carlo Simulation with $N tests, $(100*nWG4/N)% had more
    than 4 total infections")
end
```

I Code for 4d

```
    #Find probability of more than 4 infections using numerical solutions to an
    #analytical matrix method
function Q4d(x0::Int,α::Float64;dz=0.001::Float64)
    zVals = 0:dz:1
    GVals = zero(zVals)
    A = zeros(x0+1)
    A[end] = 1#Initial condition

    Pbar = zeros(x0+1,x0+1)
    Q = zeros(x0+1,x0+1)
    for i in 1:x0+1
        Q[i,i] = α^(i-1)
        for j in 1:i-1
            Pbar[i,j] = binomial(i-1,j-1)*(1-α)^(i-j)*α^(j-1)
        end
    end
    ident = Matrix{Float64}(I,x0+1,x0+1)
    for j in 1:length(zVals)
        z = zVals[j]
        E = [z^i for i in 0:x0]
        GVals[j] = A'*inv(ident-Pbar)*Q*E
    end
    PWg4 = 0.0
    for x in 0:x0-5
        PWg4 += 1/(factorial(x)) * NewtonDiff0(GVals,dz,x)
    end
    println("The probability that W>=4 is $PWg4")
    plot(zVals,GVals,legend=:none)
    xlabel!("z")
    ylabel!("PGF(z)")
end

#Newtons formula for numerical derivatives
function NewtonDiff0(xList,dt::Float64,n::Int)
    return 1/(dt^n)*sum([(-1)^(k+n)*binomial(n,k)*xList[1+k] for k in 0:n])
end
```

J Code for 5c

```
    #Heatmap of transition probability matrix of Reed-Frost model
```

```
function Q5c(x0::Int,α::Float64;T=5::Int)
    P = zeros((x0+1)^2,(x0+1)^2)
    States = Dict(i => ((i-1)÷(x0+1),(i-1)%(x0+1)) for i in 1:(x0+1)^2)
    #get table of conversions between matrix index and state
    for i in 1:(x0+1)^2
        for j in 1:(x0+1)^2
            xnow,ynow = States[i]
            xnext,ynext = States[j]
            if xnext + ynext == xnow
                P[i,j] = binomial(xnow,xnext)*α^(ynow*xnext)*(1-α^ynow)^ynext
            end
        end
    end
    heatmap(P)
    #title!("Heatmap of Reed-Frost Transition Matrix, x_0=$x0,α=$α")
    ylabel!("State 'from'")
    xlabel!("State 'to'")
end
```

K Code for 5d

```
    #Monte Carlo simulation of Reed-Frost model to determine probability of more
    #than 4 total new infections
function Q5d(x0::Int,α::Float64;N=1e6::Float64)
    WList = []
    TList = []
    for i in 1:N
        xList = [x0]
        yList = [1]
        while true
            x,y = Infected(xList[end],α^yList[end])
            push!(xList,x)
            push!(yList,y)
            if x*y == 0
                break
            end
        end
        push!(WList,sum(yList))
        push!(TList,length(xList)-1)
    end
    nWG4 = count(W->W>4,WList)
    println("Under Monte Carlo Simulation with $N tests, $(100*nWG4/N)% had more
    than 4 total infections")
end
```

L Code for 6

```
    #For N people each with a chance α of avoiding infection, return tuple of
    #(uninfected,infected)
function Infected(N::Int,α::Float64)
    x = sum(rand(N).<α)
    y = N-x
    return x::Int,y::Int
end

# Runs a generation of arrivals to the motel, accepting x0 arrivals each with a
#chance η of already being infected, within the motel there is a chance p of coming into
#contact with each person, and β of that contact being sufficient to communicate the infection
function Motel(x0::Int;η=0.05::Float64,p=0.1::Float64,β=0.05::Float64)
    x0,y0 = Infected(x0,1-η)
    xList = [x0]
    yList = [y0]
    while true
        if (yList[end]==0)
            return (xList,yList)
        end
        #Reed-Frost
        α = 1-p*β
        (x,y) = Infected(xList[end],α^(yList[end]))
        push!(xList,x)
```

```
        push!(yList,y)
    end
    return xList::Array{Int,1},yList::Array{Int,1}
end

#Run many instances of motel generations
function Q6(x0;N=1e6,η=0.05,p=0.1,β=0.05,Loud=true)
    sumList = []
    bigyList =[]
    for i = 1:N
        xList,yList = Motel(x0,η=η,p=p,β=β)
        push!(sumList,sum(yList)/x0)
        push!(bigyList,yList...)
    end
    Y = Statistics.mean(bigyList)
    Z = Statistics.mean(sumList)
    if Loud
        println("The expected rate of infections from $N trials is $Z per person,
         as opposed to a raw rate of η=$η. We expect $Y infectives to leave the motel per day" )
    end
    return Y,Z
end

#run Q6(.) for provided values of x0 and plot results
function Q6graph(x0List::Array;N=1e6,η=0.05,p=0.1,β=0.05)
    YList = []
    ZList = []
    for x0 in x0List
        Y,Z = Q6(x0;N=1e6,η=0.05,p=0.1,β=0.05,Loud=false)
        push!(YList,Y)
        push!(ZList,Z)
    end
    plot(x0List,[YList,ZList],layout=(1,2),title=["Daily infected from motel"
    "Rate of infection"],xlabel = ["x_0" "x_0"],label=:none)
end
```