ModSim Exercise 3

Jamie Cho

1.

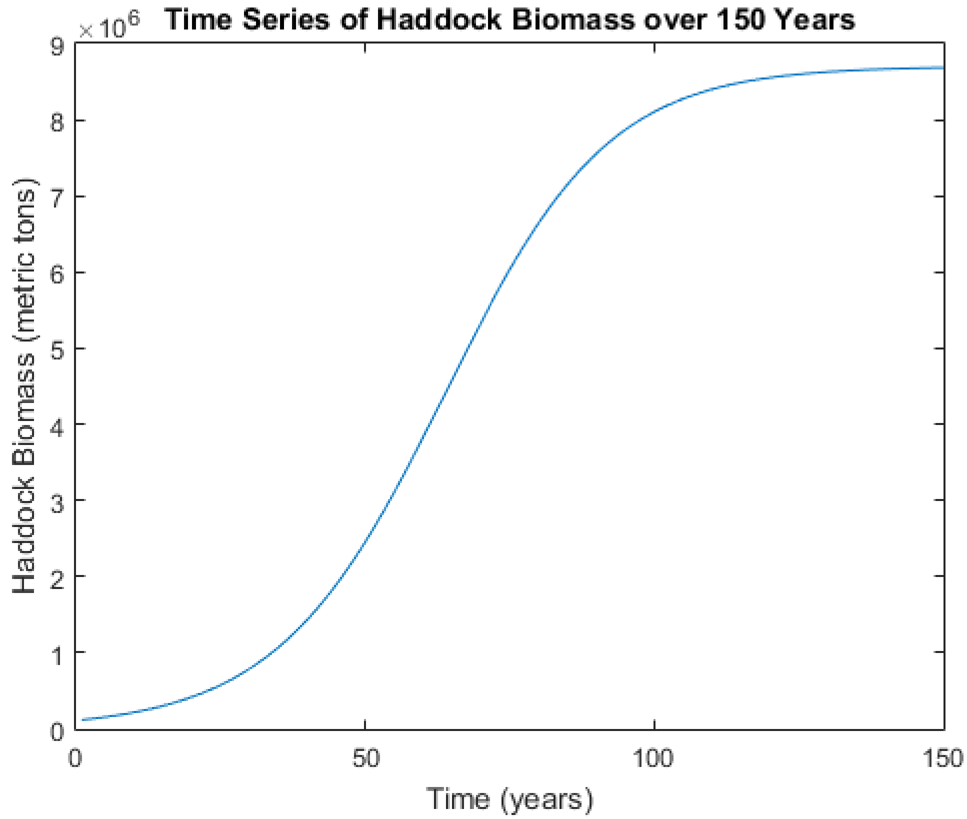
1) haddock_flow.m:

```
%res = haddock_flow(H,K,g)
%computes the flow level from
%current value of Stock, Carrying Capacity, and
%growth rate.
function res = haddock_flow(H,K,g)
    res = g*H*(1 - H/K);
end
```

2) simulate_haddock3.m:

```
%simulate_haddock3()
%simulates haddock stock population
%over the course of 150 years.
function simulate_haddock3()
    haddock = 120000;%metric tons
    cap = 8.7*1000*1000;%metric tons
    growth = 0.07; %per year

    hVec = zeros(150,1);
    for i = 1:150
        hVec(i) = haddock;
        haddock = haddock + haddock_flow(haddock,cap,growth);
    end

    disp(haddock);
    plot(hVec);
    xlabel('Time (years)');
    ylabel('Haddock Biomass (metric tons)');
    title('Time Series of Haddock Biomass over 150 Years');
end
```

3) resultant plot:



2.

1) infection_flow.m:

```
%res = infection_flow(S,P,C,k_c,k_t)
%returns infection flow amount with values:
%S = Susceptible Personnel
%P = Total Population
%C = Contagious Personnel
%k_c = contact rate constant
%k_t = transmissivity constant
function res = infection_flow(S,P,C,k_c,k_t)
    res = round(k_c*k_t*S/P*C);
end
```

2) progression_flow.m:

```
%res = progression_flow(D)
%D = Last Element of Dormant Population
function res = progression_flow(D)
    res = D;
end
```

3) cure_flow_m:

```
%res = cure_flow(C)
```

```
%C = Last Element of Contagious Population
function res = cure_flow(C)
    res = C;
end
```

4) simulate_measles_m:

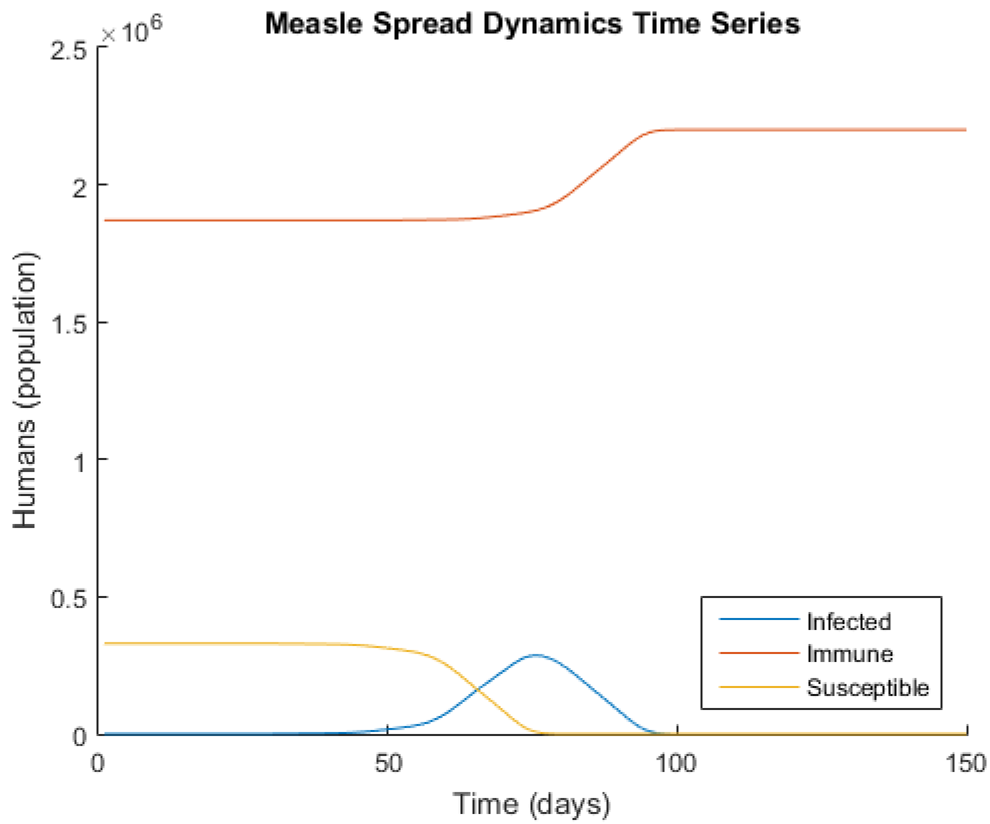
```
%simulate_measles(vaccination, population, contact_rates,
transmissibility)
%simulation of measle spread
%with internal given constants
%and control variables.
%Units:
%vaccination = decimal
%population = 1 person
%contact_rates = person / day
%transmissibility = person / contact

function simulate_measles(vaccination, population, contact_rates, ...
    transmissibility)
    %controls
    %vaccination = 0.85;
    %population = 2200000;
    %contact_rates = 12;
    %transmissibility = 0.9;

    %initialization
    contagious = zeros(8,1);
    dormant = zeros(12,1);
    immune = vaccination * population;
    susceptible = (1 - vaccination) * population;
    contagious(1) = 1;

    %plotting vectors
    vInfected = zeros(150,1);
    vImmune = zeros(150,1);
    vSusceptible = zeros(150,1);

    %simulation
    for time = 1:150
        immune = immune + cure_flow(contagious(end));

        for i = length(contagious)-1:-1:1
            contagious(i+1) = contagious(i);
        end
        contagious(1) = progression_flow(dormant(end));
        for i = length(dormant)-1:-1:1
            dormant(i+1) = dormant(i);
        end
        infection =
infection_flow(susceptible,population,sum(contagious),...
            contact_rates,transmissibility);
        dormant(1) = infection;
        susceptible = susceptible - infection;
```

```
            vInfected(time) = sum(dormant) + sum(contagious);
            vImmune(time) = immune;
            vSusceptible(time) = susceptible;
        end
        hold on
        plot(vInfected);
        plot(vImmune);
        plot(vSusceptible);
        title('Measle Spread Dynamics Time Series');
        xlabel('Time (days)');
        ylabel('Humans (population)');
        legend('show')
        legend('Infected','Immune','Susceptible','Location','southeast');
        hold off
    end
```

5) command line:

```
>> simulate_measles(0.85,2200000,12,0.9);
```

6) resultant plot:



In both scenarios, the plots were equivalent to that of InsightMaker.

- Note: as the implementation of the model was strictly based on the daily propagation from one state to the next – i.e., the "contagious" group will be cured in 8 days – the arrangement in time steps could not be accommodated.

The major difference in the approach of the Solution and my solution was due to the different method by which the implementation of the scenario occurred: I tracked the state-transitions of each group in the population: on the 14th day, the dormant period would end and the infected person would become contagious. This is a different approach from the solution that applied simpler probabilistic mechanism to the flow.