Solutions to MATH3104 Practical 2 – Avoiding the Collapse of Wildebeest

Q1.

K=3000000;
R=1.1323;
n=zeros(1,100);
n(1)=250000;
for t=1:99
    n(t+1)=(n(t)*R*K)/(K+(R-1)*n(t));
end
figure(1);
plot(1:100,n);
ylabel('Population Size');
xlabel('Time');

change and rerun:
n(1)=5000000;

These graphs show that with R=1.1323, then the populations will reach the carrying capacity of K=3,000,000 regardless of whether the population starts above or below this population size. When the population starts below the carrying capacity (top graph), the population grows as births exceed deaths. The net production of individuals is greatest (and positive) at low population levels and this decreases to zero (births balance deaths) as the carrying capacity is reached and there are increased density dependent effects. By contrast, when the population starts above the carrying capacity (bottom graph), the population shrinks as deaths exceed births. The net production is the least (largest negative) at low population levels and this increases to zero (births balance deaths) as the carrying capacity is reached and there are increased density dependent effects (deaths drop off and births increase as the population decreases).
Q2. A harvest rate of around 0.04 to 0.08 is sustainable. Above this rate the population crashes to the lower threshold level. Below a harvesting rate of 0.04 the population will continue to grow and fluctuate. I show two model runs below. There are different answers because of the stochasticity in choosing the rainfall level for each year.

Q3. The easiest way to simulate the illegal hunting of lions (and thus the increase in wildebeest survival) is by increasing $R$. Increase the value of $R$ by say 30% by changing the relevant line of the MATLAB code to:

\[ R = 1.1323 \times 1.3; \]
All harvest rates up to 0.2 are now sustainable.

Q4. To determine the effect of decreasing rainfall to 30% of its current level, we change the value of K in the MATLAB code to be:

\[ K = 20748 * \text{rain} * 0.3; \]

No harvesting would be prudent under this scenario as the simulations even at very low harvest rates often touch the lower threshold we set for Wildebeest. In fact the population is virtually not viable under these persistent drought conditions.

Q5. It would be best to extend the programme by performing a Monte Carlo simulation – i.e. by re-running the simulations a large number of times. This could be performed easily by placing the main body of the current code in a loop that runs the simulations 1,000 times. The variable to plot could then be the fraction of the 1,000 simulations that crashed to the lower threshold for each of the harvest levels.
Matlab Code

```matlab
rain=[100, 38, 100, 104, 167, 167, 165, 79, 91, 77, 134, 192, 235, 159, ...
    211, 257, 204, 300, 187, 84, 99, 163, 97, 228, 208, 83, 44, 112, 191, ...
    202, 137, 150, 158, 20];
R=1.1323;
K=5532*rain*0.3;
h=0:0.02:0.2;
N0=250000;
runlen=99;
Populations=zeros(length(h),runlen+1);
% rand('seed',0)
for i=1:length(h)
    n=[N0];
    for t=1:runlen
        k=K(floor(rand*34)+1);
        a=n(t)*((R*k)/(k+(R-1.0)*n(t)))*(1-h(i));
        if a<100000
            a=100000;
        end
        n=[n a];
    end
    Populations(i,:)=n;
end

figure(1);
plot(1:runlen+1,Populations);
legend(num2str(h'));
xlabel('Time'); ylabel('Population Size');
```