MATHEMATICAL AND COMPUTER MODELING OF NONLINEAR BIOSYSTEMS I

COMPUTER LABORATORY II: Migrations in logistic equation, Alee effect, Holling response

Ph. D. Programme 2013/2014

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Project co-financed by European Union within the framework of European Social Fund

Logistic equation with migrations

Let us consider the following logistic model with emigration

$$\dot{N} = rN\left(1 - \frac{N}{K}\right) - m\,,$$

where all parameters are positive. From the lecture we know that if

$$rK/4 > m$$

and

$$N(0) < N_1^1 = \frac{rK - \sqrt{r^2K^2 - 4rmK}}{2r}$$

then the population become extinct in final time (all individuals emigrate in some time $\tilde{t} < \infty$).

Question: What is the time till extinction when we know parameters values and initial condition?

Excercise 1 - Event location

Consider the following logistic model with emigration

$$\dot{N} = rN\left(1 - \frac{N}{K}\right) - m\,.$$

Given the values of parameters r, K, m and initial condition estimate the time till extinction (check whether it is possible).

Hint: Use 'Event' property when setting integrator options.

Excercise 1 - Solution

```
function T = findZeroCross( init, r, K, m )

    T=0; %setting default output value

    delta=r^2*K^2-4*r*m*K;
    N1=(r*K-sqrt(delta))/(2*r); %positive steady state (first)

    if r*K/4>=m && init<N1 %checking if extinction is possible

        T=Inf;
        options=odeset('Events',@events);

        sol=ode45(@model,[0 T],init,options);

        T=sol.x(end); %returning estimated time
    else
        display('No extinction possible');
    end
```

Excercise 1 - Solution

```
    function y=model(~,x)
       y=r*x*(1-x/K)-m;
    end

    function [value,isterminal,direction] = events(~,y)
        value=y;
        isterminal=1;
        direction=0;
    end
end
```

Allee effect

The simplest form of the model reflecting the Allee effect reads

$$\dot{N} = rN(N - N_{cr})\left(1 - \frac{N}{K}\right),$$

compare Lecture 2.

It is easy to check that:

- if $N(0) < N_{Cr}$ then $N(t) \to 0$;
- if $N(0) > N_{Cr}$ then $N(t) \to K$.

Exercise 2 - time to rebuild population

Consider the following model

$$\dot{N} = rN(N - N_{cr})\left(1 - \frac{N}{K}\right).$$

Let us assume that at time t^* catastrophe occurred and $N(t^*) < K$ decreased by p percent.

Estimate time needed to rebuild the population to the level before catastrophe? (if it is possible)

Plot the solution.

Excercise 2 - Solution

```
function T = timeToRebuild( Ns, r, K, Ncr, p)

    T=0; %setting default output value

    if Ns*(1-p)>Ncr %checking if rebuilding is possible

        T=Inf;
        options=odeset('Events',@events);

        sol=ode45(@model,[0 T],Ns*(1-p),options);
        T=sol.x(end); %returning estimated time

        figure(1)
        clf
        hold on
        plot(sol.x,sol.y,'LineWidth',2);
        plot(sol.x([1 end]),[Ns Ns],'LineStyle','--');
        hold off
```

Excercise 2 - Solution

```
    else
        display('Population tends to zero');
    end

    function y=model(~,x)
       y=r*x*(x-Ncr)*(1-x/K);
    end

    function [value,isterminal,direction] = events(~,y)
        value=y-Ns;
        isterminal=1;
        direction=1;
    end
end
```

Holling response (type II)

When we consider the type II functional response (Holling disc equation) to predation in the logistic growth model, we consider the equation

$$\dot{N} = rN\left(1 - \frac{N}{K}\right) - m\frac{N}{1 + nN},$$

where all parameters are positive. It was show during the lecture that if $m < r$ there are two steady states: 0 and $\bar{N} < K$. Moreover, the latter is globally stable.

What is the dependence of $\bar{N}$ on m and n?

Exercise 3 - approximate the steady state

Consider the model

$$\dot{N} = rN\left(1 - \frac{N}{K}\right) - m\frac{N}{1 + nN} \,.$$

with $m < r$.

Given the values of r and K, plot the value of the positive steady state for $m \in [0\,,\ 0.9r]$ and $n \in [0, 2]$.

Excercise 3 - Solution

```
function plotSteadyState(r,K)

    mv=linspace(0*r,0.9*r,20);
    nv=linspace(0,2,20);
    out=zeros(length(mv),length(nv));

    for i=1:length(mv)
        for j=1:length(nv)
            m=mv(i); n=nv(j);
            %solving for the steady state
            out(i,j)=fzero(@(N)(r*N*(1-N/K)-m/(1+n*N)),K);
        end
    end

    figure(1) %plotting
    clf
    surface(nv,mv,out);

end
```