MATHEMATICAL AND COMPUTER MODELING OF NONLINEAR BIOSYSTEMS I

COMPUTER LABORATORY XIV: Model of self-repressing gene, Mackey-Glass model

Ph. D. Programme 2013/2014

PhD STUDIES

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

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

Mackey-Glass model

The Mackey-Glass equation is the nonlinear delayed differential equation

$$\frac{dx(t)}{dt} = \alpha \frac{x(t-\tau)}{1+x^n(t-\tau)} - gx(t) = F\left(x(t), x(t-\tau)\right)$$

where all parameters are positive.

We want to study behaviour of the solutions to the above equation.

We will perform the local stability analysis of the positive steady state and plot exemplary solutions.

Steady states of the model

In order to calculate steady states we search for the roots of the $F(\bar{x}, \bar{x})$ function, that is we solve the following equation

$$\alpha \frac{\bar{x}}{1 + \bar{x}^n} - g\bar{x} = 0\,.$$

The steady states are

$$\bar{x} = 0$$

and

$$\bar{x} = \sqrt[n]{\frac{\alpha}{g} - 1}\,,$$

for $\alpha > g$.

Linearisation around steady states

Next, we linearise the initial equation around the steady states. We calculate the derivatives of the $F(x(t), x(t-\tau))$ function with respect to both variables

$$\frac{\partial F(x(t), x(t-\tau))}{\partial x(t)} = -g$$

and

$$\frac{\partial F(x(t), x(t-\tau)}{\partial x(t-\tau)} = \alpha \frac{1-(n-1)x^n(t-\tau)}{(1+x^n(t-\tau))^2}\,.$$

We obtain the following equation around the trivial steady state

$$\frac{d\tilde{x}(t)}{dt} = -g\tilde{x}(t) + \alpha\tilde{x}(t-\tau)$$

and the following equation around the positive steady state

$$\frac{d\tilde{x}(t)}{dt} = -g\tilde{x}(t) + \alpha\frac{1-(n-1)\bar{x}}{(1+\bar{x}^n)^2}\tilde{x}(t-\tau) = -g\tilde{x}(t) + \beta\tilde{x}(t-\tau)\,.$$

Exercise 1

Positive steady state

$$\bar{x} = \sqrt[n]{\frac{\alpha}{g} - 1}\,,$$

Linearised equation

$$\frac{d\tilde{x}(t)}{dt} = -g\tilde{x}(t) + \alpha\frac{1-(n-1)\bar{x}}{\left(1+\bar{x}^n\right)^2}\tilde{x}(t-\tau) = -g\tilde{x}(t) + \beta\tilde{x}(t-\tau)\,.$$

Implement MATLAB function that for given parameters values returns the value of parameter β.

Input argument: structure with parameters values.

Exercise 1 - solution

```
function beta = calculateBeta( params )
    x = params.alpha/params.g;
    beta = params.alpha*(1-(params.n-1)*(x-1))/x^2;
end
```

Local stability of the positive steady state

We consider the linearised equation

$$\frac{d\tilde{x}(t)}{dt} = -g\tilde{x}(t) + \beta\tilde{x}(t-\tau)\,.$$

We calculate the characteristic pseudo-polynomial

$$W(\lambda) = \lambda + g - \beta \exp\left(-\lambda\tau\right)$$

and evaluate it at purely imaginary number

$$W(i\omega) = i\omega + g - \beta(\cos(\omega\tau) - i\sin(\omega\tau))\,,$$

where ω is positive real number.

Local stability of the positive steady state

We want to use Mikhailov criterion and hence we look at the total change in the argument of $W(i\omega)$ for $\omega \in [0, +\infty]$.

Hence, we look at the Mikhailov hodographs, that is the plots od $W(i\omega)$ in complex plane (with respect to changing ω). We plot

$$\Re(W(i\omega)) = g - \beta \cos(\omega\tau)$$

on the x-axis and

$$\Im(W(i\omega)) = \omega + \beta \sin(\omega\tau)$$

on the y-axis.

Exercise 2

Implement MATLAB function for plotting Mikhailov hodograph.

Input arguments:

- structure with parameters values
- maximal value of ω to draw
- number of plot mesh points

Exercise 2 - solution

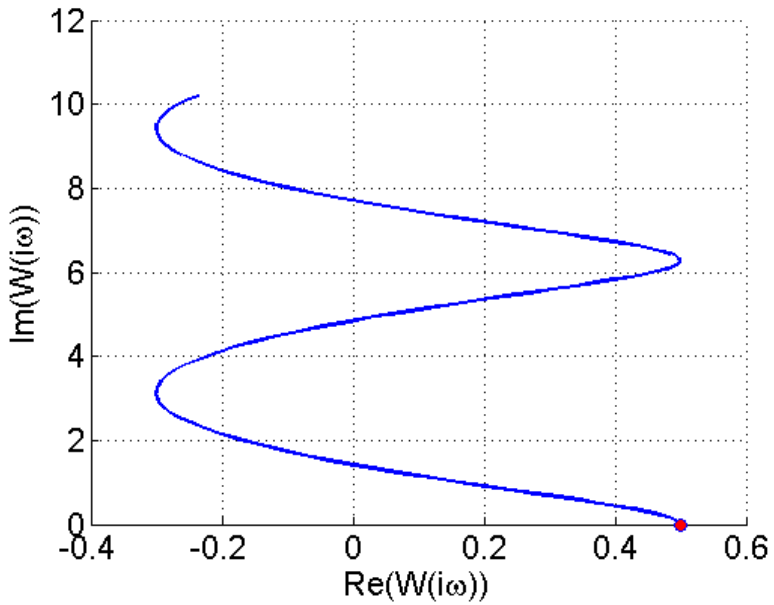
```
function plotHodograph( params, wmax, n)
    beta = calculateBeta(params);
    mesh = linspace(0,wmax,n);
    W = characteristic(mesh);

    set(0, 'DefaultAxesFontSize',16);
    figure(1)
    clf
    hold on
    plot(W(1,:),W(2,:),'LineWidth',2,'Color','b');
    plot(W(1,1),W(2,1),'Marker','o','MarkerFaceColor','r');
    hold off
    xlabel('Re(W(i\omega))') ylabel('Im(W(i\omega))')
    grid on;

    function W = characteristic(w)
        W=zeros(2,length(w));
        W(1,:) = params.g - beta*cos(w*params.tau);
        W(2,:) = w + beta*sin(w*params.tau);
    end
end
```

Exercise 2 - drawing exemplary hodograph

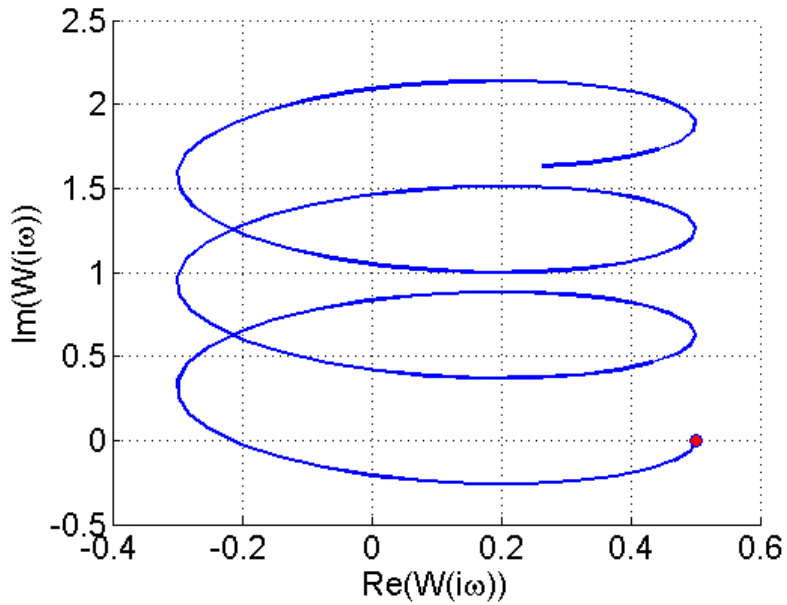
```
clear all;

params.alpha = 0.2;
params.n = 10;
params.g = 0.1;
params.tau = 1;

calculateBeta(params)

plotHodograph(params, 10, 100)
```

Exemplary hodographs

$\tau = 1$



$\tau = 10$



The steady state is stable if the curve crosses imaginary axis above zero (at all crossing points).

Exercise 3

Implement MATLAB function that determines the stability of the positive steady state.

Input arguments:

- structure with parameters values

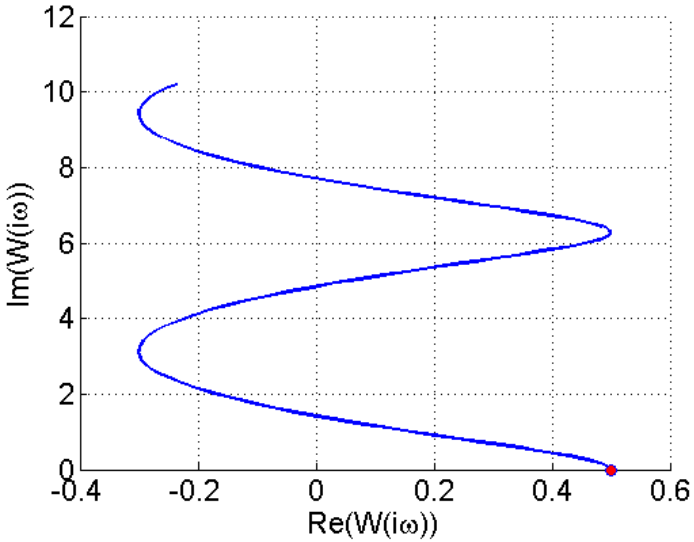
Hint: find the first intersection point with the imaginary axis and check sign of its imaginary part.

Exercise 3 - solution

```
function stab = determineStability( params  )

    beta = calculateBeta(params);

    wR = [0 pi/params.tau];

    intersec = fzero(@F,wR);

    stab = (intersec + beta*sin(intersec*params.tau))>0;

    function z = F(w)
        z = params.g - beta*cos(w*params.tau);
    end

end
```

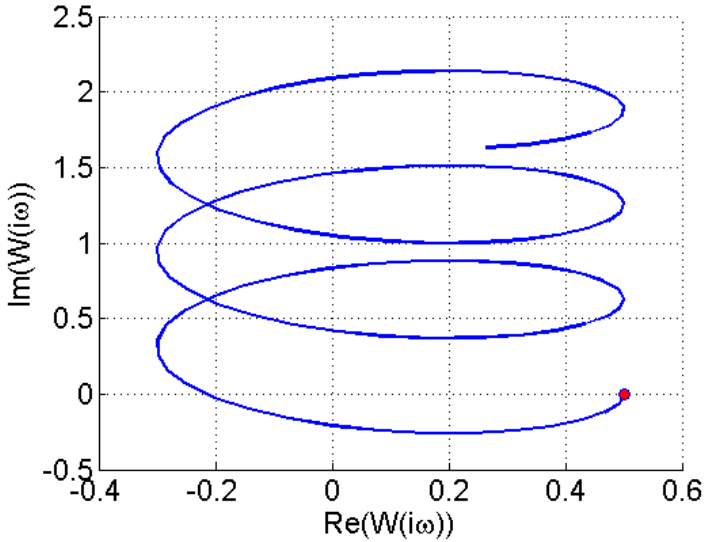
Determining stability on the exemplary hodographs

$\tau = 1$



Stable

$\tau = 10$



Unstable

Exercise 4

Implement MATLAB function that finds the maximal value of τ for which positive steady state remains stable.

Input arguments:

- structure with parameters values

Exercise 4 - solution

```
function tauCr = calculateCritical( params )

    tauCr = 0;
    step = 1;
    direction = 1;

    while step > 1e-5
        params.tau = tauCr + direction*step;
        if ~determineStability(params)
            step = step/2;
        else
           tauCr = tauCr + step;
        end
    end
end
```

It occurs that the value of critical τ is ≈ 4.7082.

Exercise 5

Implement MATLAB function for calculating the solution to the Mackey-Glass equation and draw solutions for τ below and above the critical value found in previous exercise.

Input arguments

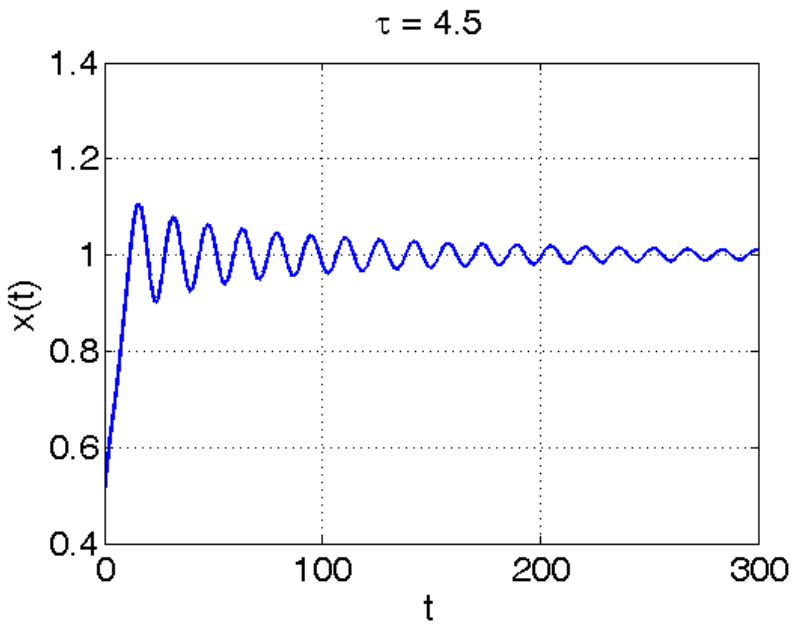
- time horizon for the solution
- structure with parameters values
- initial condition

Remark: In order to solve the equation we need to impose an initial function on interval $[-\tau, 0]$ (in case of ODEs it was sufficient to impose only values at $t = 0$).

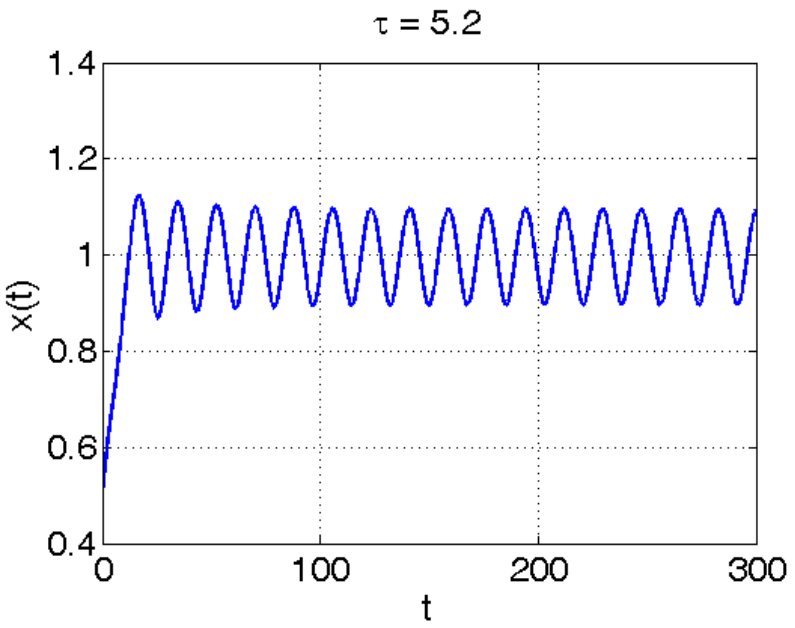
Exercise 5 - solution

```
function sol = solutionMackeyGlass( T, params, init )

    opt = ddeset('RelTol',1e-8,'AbsTol',1e-8);
    sol = dde23(@model, params.tau, init, [0 T],opt);

    function y = model(~,x,xlag)
       y = params.alpha*xlag/(1+xlag^params.n)-params.g*x;
    end

end
```

Exemplary solutions

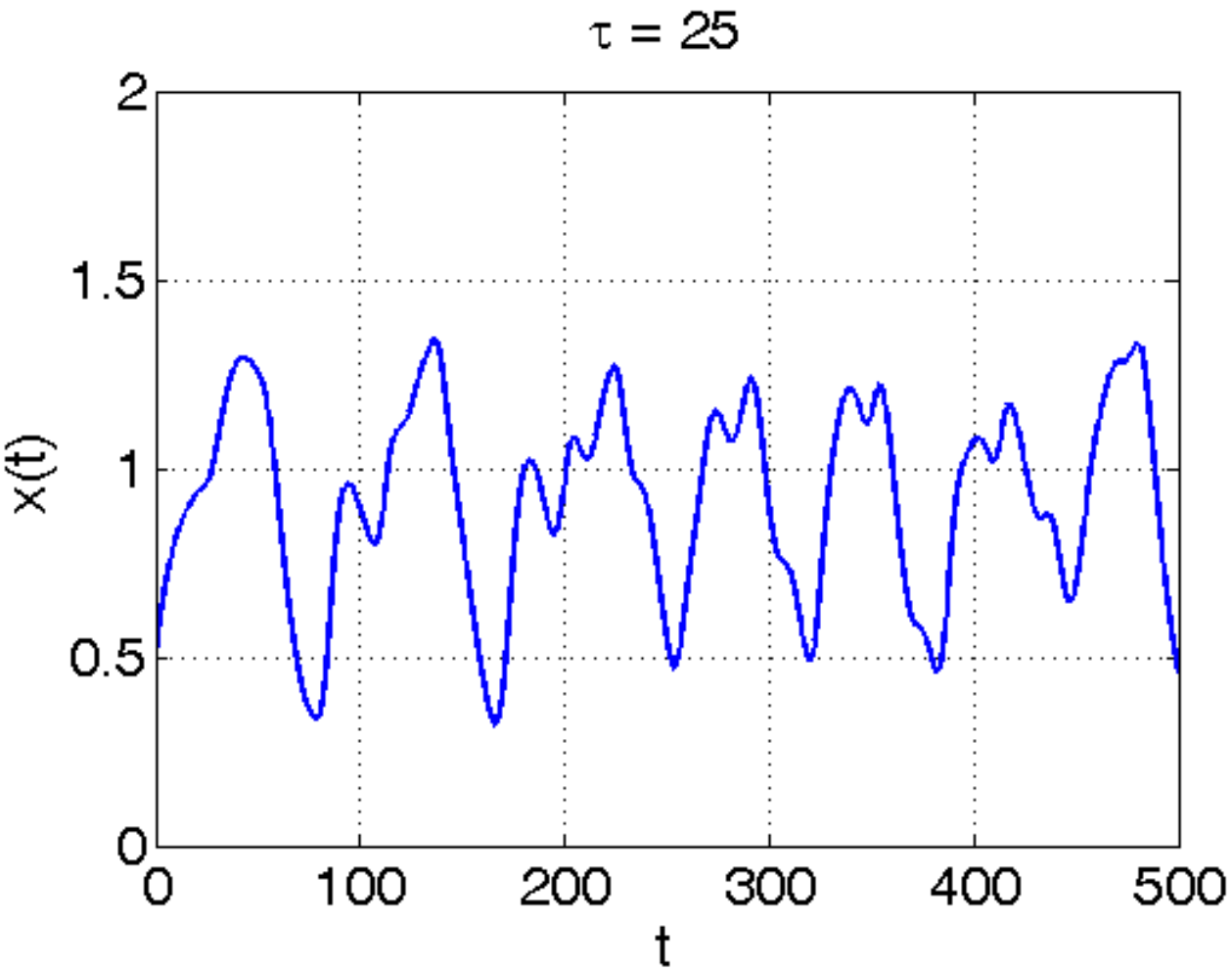


Stable



Unstable

Large delay



For larger τ we see the onset of chaos.