MATHEMATICAL AND COMPUTER MODELING OF NONLINEAR BIOSYSTEMS I

COMPUTER LABORATORY VI: Lorenz attractor (solutions, behavior)

Ph. D. Programme 2013/2014

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

Sensititity analysis

Sensitivity analysis helps to understand how uncertainty due to input parameters propagates through the modelled system.

A basic approach to measuring of the sensitivity at some fixed time point is to calculate the partial derivative of systems solution with respect to the parameters

$$S_{ij}(t^*) = \frac{\partial y_i(t^*)}{\partial \alpha_j},$$

when α_j represent the investigated parameter and y_i is the i*th* solution component.

Positive (negative) value of $S_{ij}(t^*)$ indicates that a small increase in the α_j value will cause in increase (decrease) in $y_i(t^*)$. The larger is the absolute value of $S_{ij}(t^*)$ the larger is the change induced by the parameter perturbation.

Excercise 1 - estimating sensitivity indices

Using the fact that the derivative may be numerically approximated using finite differences

$$F'(y) \approx \frac{F(y+h) - F(y)}{h},$$

for small h, implement MATLAB procedure for estimating the sensitivity indices $S_{i,j}$ at given t^* for the first solution coordinate and selected parameters/initial conditions.

Input arguments

- model handle (or name in a string);
- final time for the model solution;
- nominal parameters and initial conditions (single structure);
- cell with parameters names for which we want to estimate sensitivity;
- vector of steps h for estimation of respective sensitivities.

Output arguments

- vector containing the sensitivity indices.

Excercise 1 - solution

```
function sens = estimateSensitivity(model, T, nominalParams,...
 sensParams, diffStep )

    sens = zeros(1,length(sensParams));

    nominalSol = feval(model, T, nominalParams);

    for i = 1:length(sensParams)
       pertParams=nominalParams;
       pertParams.(sensParams{i})=pertParams.(sensParams{i})+...
       diffStep(i);
       pertSol = feval(model, T, pertParams);
       sens(i) = (F(pertSol) - F(nominalSol))/diffStep(i);
    end

    function y = F(sol)
       y = sol.y(1,end);
    end
end
```

Excercise 2 - sensitivities in logistic equation

Calculate sensitivity indices for the logistic equation

$$\dot{N} = rN(1 - \frac{N}{K})$$

with respect to parameters r, K and initial condition $N(0)$.

Nominal values: $r = 0.5,\ K = 100,\ N(0) = 1$

Final time for the solution: $T_f = 15$

Plot the sensitivity indices for $h = 0.1/2^i,\ \ i = 1, ..., 10$.
Use the logarithmic scale for x-axis.

Excercise 2 - solution (part 1)

Model definition

```
function sol = logistic( T, params )

    opt=odeset('RelTol',1e-8,'AbsTol',1e-8);
    sol = ode45(@model,[0 T],params.x0,opt);

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

Excercise 2 - solution (part 2)

Script

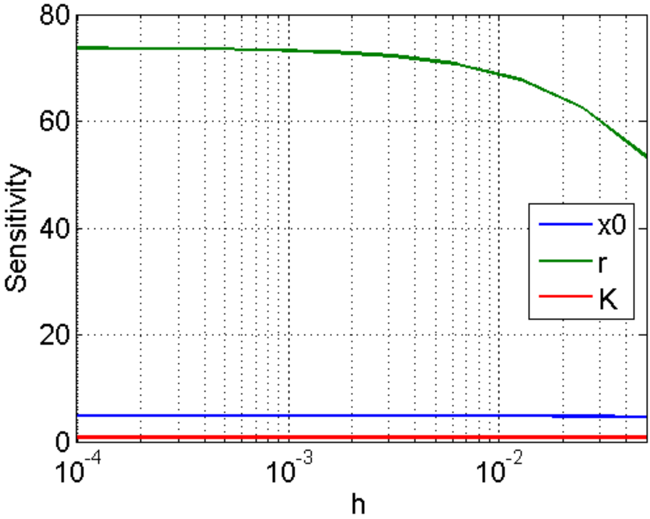
```
clear all

params.r = 0.5;
params.K = 100;
params.x0 = 1;

Tend = 15;

hmesh = 0.1./2.^(1:10);
sens = zeros(length(hmesh),3);

for i = 1:length(hmesh)
    sens(i,:) = estimateSensitivity('logistic', Tend, params,...
    {'x0','r','K'}, hmesh(i)*ones(1,3) );
end

plotSensitivity(sens, hmesh, {'x0','r','K'}, 1);
```

Excercise 2 - solution (part 3)

Plotting function

```
function plotSensitivity( sens, hmesh, params, f )

figure(f)
clf
plot(hmesh, sens', 'LineWidth',2);

legend(params);
xlabel('h - difference step')
ylabel('Sensitivity')

xlim([hmesh(end) hmesh(1)])

set(gca,'XScale','log')

grid on;
box on;
end
```

Excercise 2 - result



We observe divergence of the derivative approximation.

Lorenz model

The famous Lorenz model reads

$$\begin{aligned} \dot{x} &= -ax + ay, \\ \dot{y} &= rx - y - xz, \\ \dot{z} &= -bz + xy, \end{aligned}$$

Let us take the following nominal parameters values

$$r = 28, \; a = 10, \; b = 8/3$$

and initial conditions

$$x(0) = y(0) = z(0) = 1 \, .$$

We are interested in the first solution component at $T_f = 500$.

Excercise 3 - sensitivities in the Lorenz model

Calculate the sensitivity of the solution to the Lorenz model

$$\begin{aligned}\dot{x} &= -ax + ay,\\ \dot{y} &= rx - y - xz,\\ \dot{z} &= -bz + xy,\end{aligned}$$

with respect to initial condition $x(0)$ (nominal settings on the previous slide).

Plot the sensitivity indices for $h = 0.1/2^i, \; i = 1, ..., 10$.
Use the logarithmic scale for x-axis.

Excercise 3 - solution (part 1)

Model definition

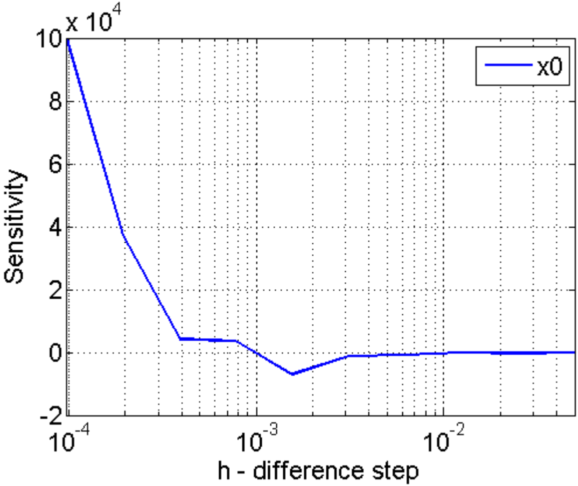
```
function sol = lorenz( T, params )

    opt=odeset('RelTol',1e-8,'AbsTol',1e-8);
    sol = ode45(@model,[0 T],...
                [params.x0 params.y0 params.z0],opt);

    function y = model(~,x)
       y = zeros(3,1);
       y(1) = -params.a*x(1) + params.a*x(2);
       y(2) = params.r*x(1)-x(2)-x(1)*x(3);
       y(3) = -params.b*x(3)+x(1)*x(2);
    end
end
```

Excercise 3 - solution (part 2)

Script

```
clear all
params.r = 28;
params.a = 10;
params.b = 8/3;
params.x0 = 1;
params.y0 = 1;
params.z0 = 1;
Tend = 500;
hmesh = 0.1./2.^(1:10);
sens = zeros(size(hmesh));
for i = 1:length(hmesh)
    display(i);
    sens(i) = estimateSensitivity('lorenz', Tend, params,...
    {'x0'}, hmesh(i) );
end
plotSensitivity(sens, hmesh, {'x0'}, 2);
```

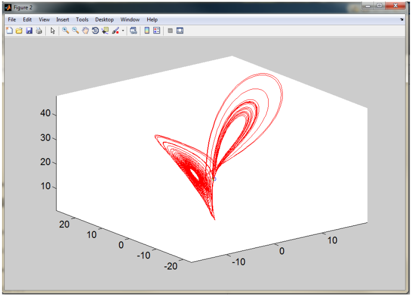
Excercise 3 - result



We <u>don't</u> observe divergence of the derivative approximation.

Excercise 4 - dynamics of trajectory in 3D

Solve the Lorenz model for the nominal parameters values and plot the solution in 3D using comet3 function.



Is the solution behaviour predictable?

Excercise 4 - solution

```
clear all

params.r = 28;
params.a = 10;
params.b = 8/3;

params.x0 = 1;
params.y0 = 1;
params.z0 = 1;

Tend = 500;

sol = lorenz(Tend,params);
figure(3)
comet3(sol.y(1,:),sol.y(2,:),sol.y(3,:))
```

Excercise 5 - nominal vs. perturbed trajectory

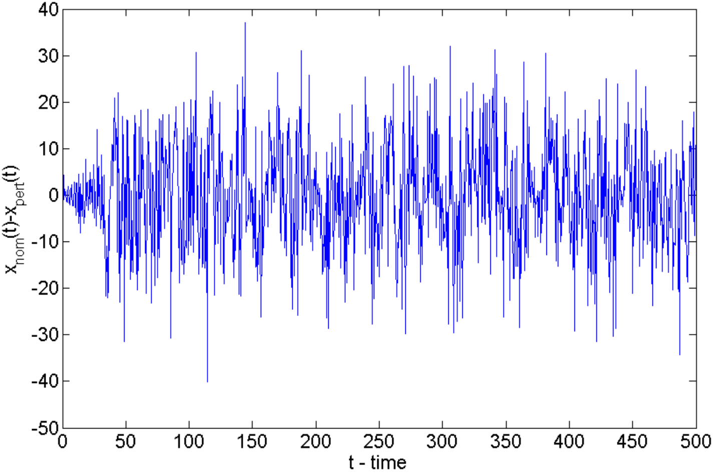
Plot the difference between the nominal solution to the Lorenz model (solution for nominal parameters values) and the perturbed solution obtained for $x(0) = 1 + 10^{-5}$.

Is the difference between solutions small?

Excercise 5 - solution

```
clear all

params.r = 28; params.a = 10; params.b = 8/3;
params.x0 = 1; params.y0 = 1; params.z0 = 1;
Tend = 500;

sol = lorenz(Tend,params);
params.x0 = 1+1e-5;
solPert = lorenz(Tend,params);

tmesh=linspace(0,Tend,1000);
s1 = deval(sol,tmesh);
s2 = deval(solPert, tmesh);

plot(tmesh, s1(1,:)-s2(2,:))
xlabel('t - time')
ylabel('x_{nom}(t)-x_{pert}(t)')
```

Excercise 5 - result



Summary

In a Lorenz model we observe phenomenon called the butterfly effect, that is small perturbation in the initial conditions causes in tremendous changes in the solution.