Modeling of Physical Systems

Application of box models to Upper Danube catchment simulation.

Dominik Katszer

28 April 2018

1 Aim of laboratory

The aim of the laboratory is to calculate a mean residence time of water in the Danuba river using modelled object called "black-box". This river was choosen because of accessibility to data accross many years. According to it, it is possible to connect water amount increasing with any occurrence.

2 Algorithm

In order to obtain information about river characteristics we are basing on tracer experiment. Basing on that experiment, it is possible to calculate any other response of the system. In this case , in order to calculate river catchments I used exponential model.

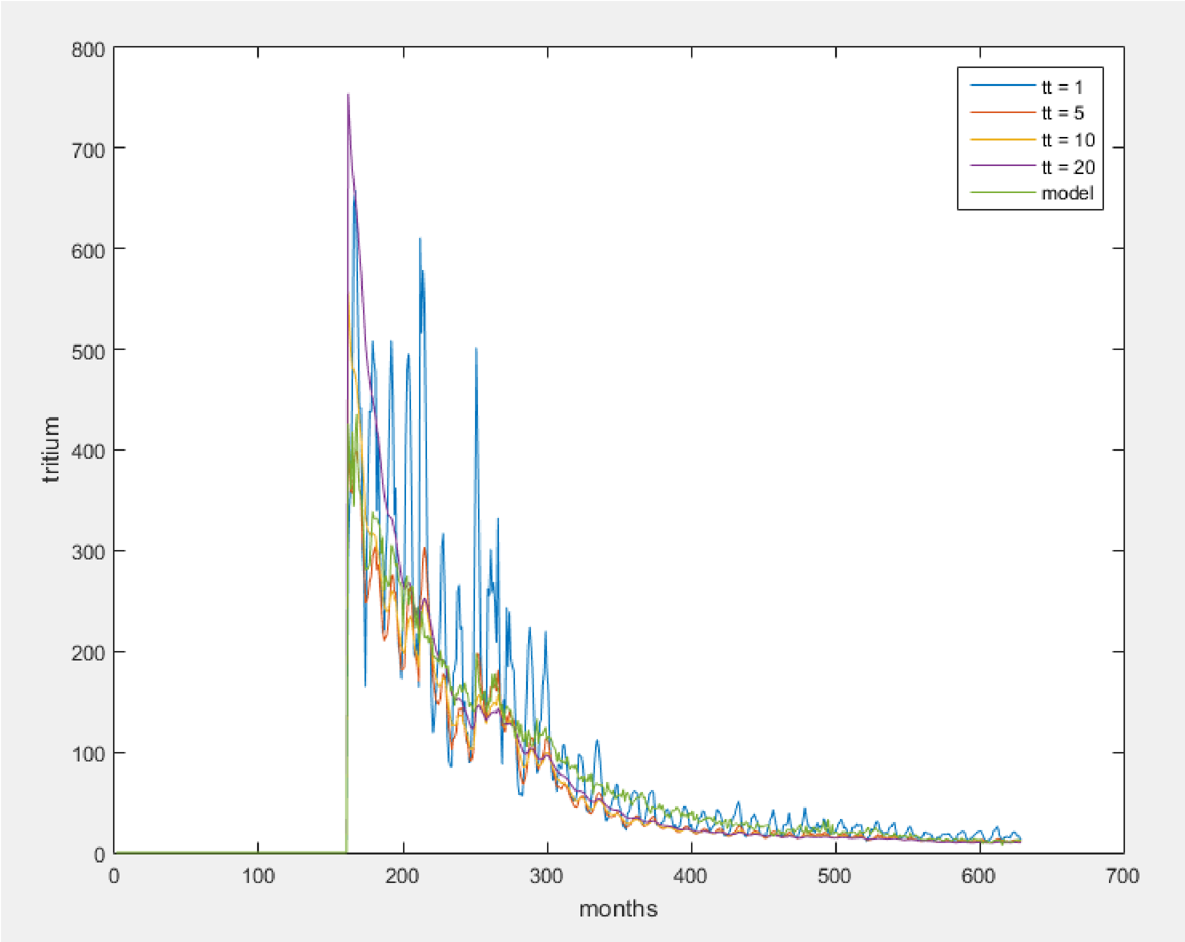
$$C(t) = \int_{-\infty}^{t} C_{in}(t') \cdot g(t - t') \cdot e^{-\lambda \cdot (t-t')} dt' \tag{1}$$

Where:

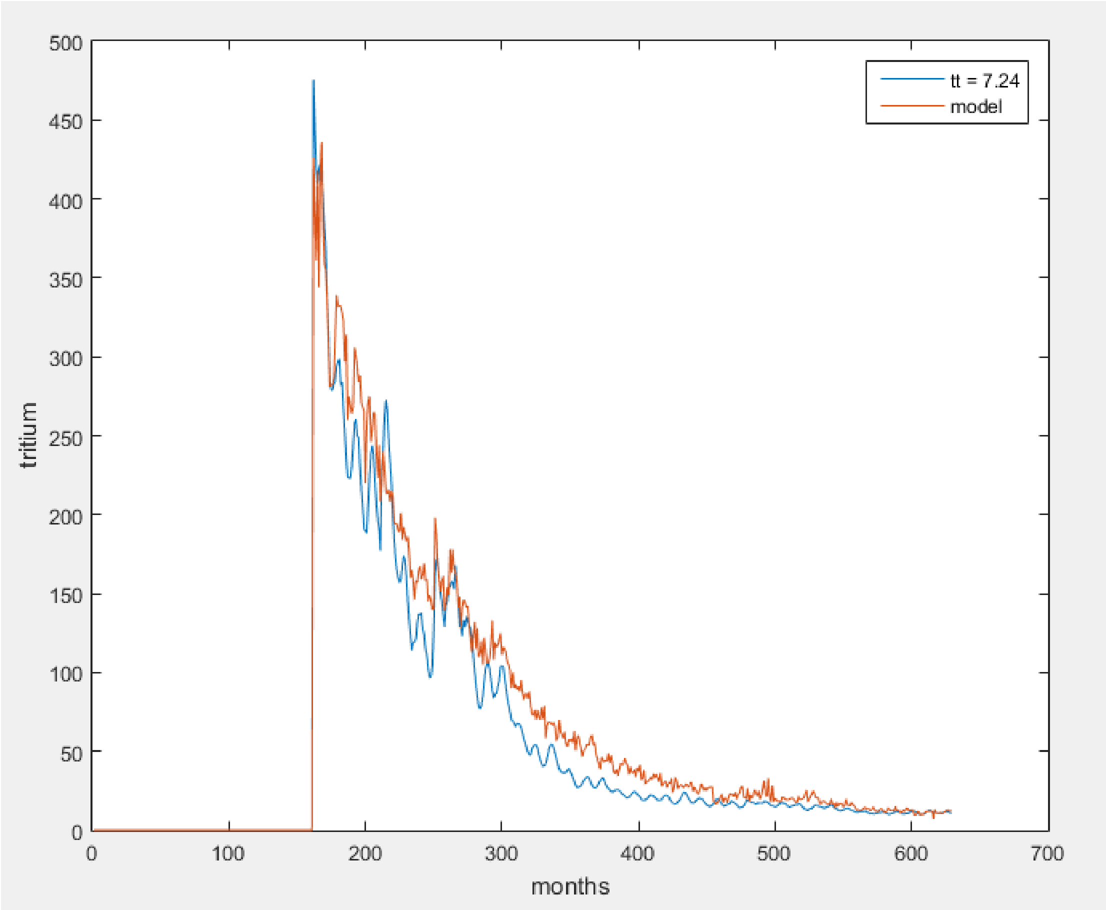
- t - time variable
- $C(t)$ - output function
- $C_{in}(t')$ - input function
- $g(t - t') = t_t^{-1} * e^{\frac{-(t-t')}{t_t}}$ - exponential model
- t_t - mean residence time
- $\lambda = 4.696 * 10^{-3}$ - radioactive decay constant

3 Results

By changing mean residence time t_t I tried to change the result of the simulation in order to be as similar to real data as possible. At the beggining manuall method was used and results are presented in the figure below.



After that automatic strategy was applying which calculates value of t_t basing on the difference between original data and calculated result.



4 Conclusions

This report covered simplified methods to calculate the mean residence time of water in the upper part of the Danube river using black box modeling with exponential transit function. This kind of exercise is difficult to make because it needs long term measurement to validate precision of calculation.

5 Source code

main.m

```
clear;
clc;

dunaj = importdata('dunaj.prn'); %Real data
opady = importdata('opady.prn');

tt_start=32;
[val diff] = fminunc(@forward, tt_start);
```

```
tt = val(1);
Cin = opady(:,2);
C(:,1) = blackBox(Cin, tt);
C(:,2) = dunaj(:,2);
plot(1:size(opady), C)
legend('tt = 7.24', 'model');
xlabel('months');
ylabel('tritium');
%%
dunaj = importdata('dunaj.prn'); %Real data
opady = importdata('opady.prn');

tt_ar = [1 5 10 20];
for ii=1:4
    Cin = opady(:,2);
    C(:,ii) = blackBox(Cin, tt_ar(ii));
end
C(:,5) = dunaj(:,2);
plot(1:size(opady), C)
legend('tt = 1', 'tt = 5', 'tt = 10', 'tt = 20', 'model');
xlabel('months');
ylabel('tritium');
```

blackBox.m

```
function Cout = blackBox( Cin, tt )
tmp = size(Cin);%opady and dunaj have the same size = 629
time = tmp(1);
C = zeros(time, 1);
T12 = 12.3; % T_1/2 half-life = 12.26 y
lambda = log(2)/(T12 * 12); %radioactive tritium decay constant
%lambda = 0.0565; %found in the internet

for t = 162:time % dunaj values are not 0 from 162 month
    suma = 0;
    for i = 1:t % i = t'
        f = Cin(i)*expoModel(tt,t,i)*exp(-lambda*(t-i));
        suma = suma + f;
    end
    C(t) = suma;
end

Cout = C;
end
```

expoModel.m

```
function result = expoModel( tt, t, i )
result = (1/tt)*exp(-(t-i)/tt);
end
```

forward.m

```
function diff = forward(tt)

dunaj = importdata('dunaj.prn'); %Real data
opady = importdata('opady.prn');
%The input function used in the modelling exercise is calculated as a
    ↪ mean value of tritium
%concentration in the precipitation (opady), weighted by the
    ↪ precipitation amount for each station

Cin = opady(:,2);
C = blackBox(Cin,tt);
diff = sqrt(sum((dunaj(:,2)-C).^2))

end
```