LAB REPORT: LAB 6
Taylor model explicit method
MODELING OF PHYSICAL SYSTEMS

Patryk Gaczyski

Tuesday 24th April, 2018

1 Aim

The aim of this laboratory was to simulate, visualize and analyze simplified Taylor model of one dimensional river pollutants transportation process using "QUICKEST explicit method".

2 Simulation description

Pollutants transportation process can be modeled by following Taylor equation:

$$\frac{\partial c}{\partial t} + U\frac{\partial c}{\partial x} - D\frac{\partial^2 c}{\partial x^2} = 0 \tag{1}$$

Such model requires to define boundary conditions as well as initial conditions:

1. boundary condition - left side - $c(0,t) = 0$
2. boundary condition - right side - $\frac{\partial c}{\partial x}(L,t) = 0$
3. initial condition - $c(x,0) = \begin{cases} 0 & \text{for} \quad x \neq x_i \\ \frac{m}{w \cdot d \cdot dx} & \text{for} \quad x = x_i \end{cases}$

where:

- c - tracers concentration function
- t - time variable
- U = 0.1$[\frac{m}{s}]$- advection coefficient
- x - displacement
- D = 0.01$[\frac{m^2}{s}]$ - dispersion coefficient

- L = 100$[m]$ - river length
- m = 1$[kg]$ - mass of injected tracer
- w = 5$[m]$ - river width
- d = 1$[m]$ - river depth

Given differential equation would be hard to model within matlab script, which is why numerical approach was chosen over the analytical one. In this case, explicit quickest method provides fairly easy formula to implement, and most importantly it provides iterative interface (next function value is being calculated based on previously calculated one). Following this approach, one can propose following formula:

$$c_j^{n+1} = c_j^n + [C_d(1-C_a) - \frac{C_a}{6}(C_a^2 - 3C_a + 2)]c_{j+1}^n - [C_d(2-3C_a) - \frac{C_a}{2}(C_a^2 - 2C_a - 1)]c_j^n + [C_d(1-3C_a) - \frac{C_a}{2}(C_a^2 - C_a - 2)]c_{j-1}^n + [C_d C_a - \frac{C_a}{6}(C_a^2 - 1)]c_{j-2}^n \quad (2)$$

where:

- C_a - advective Courant number
- C_d - diffusive Courant number

2.1 Quickest method implementation

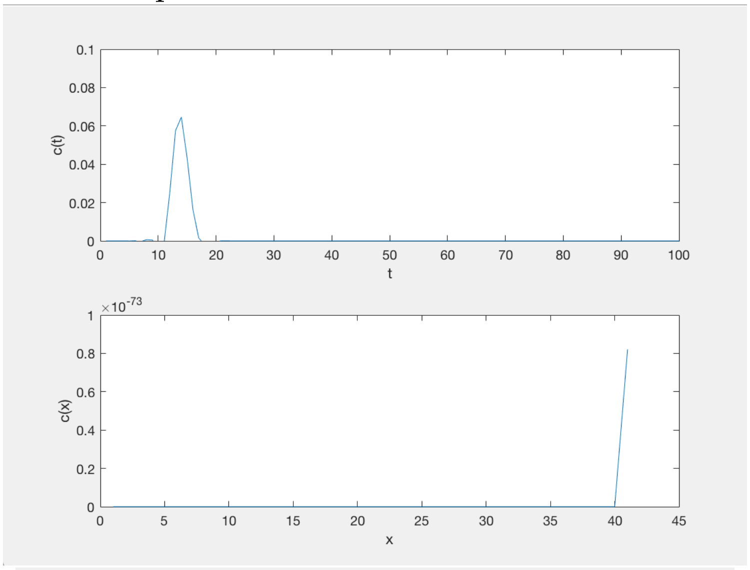
To realize provided algorithm, simple matlab script were developed:

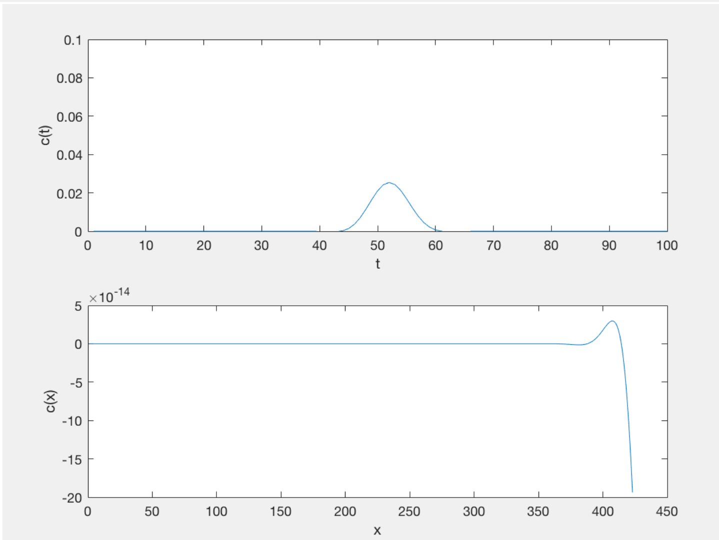
Listing 1: Matlab script for explicit taylor method

```
% define parameters
len     = 100;     % m
wid     = 5;       % m
dep     = 1;       % m

t       = 1000;    % s
dt      = 1;   % s
dx      = 1;   % m
nt      = t/dt;    % num
U       = 0.1;     % m/s  [advection coefficient ]
D       = 0.01;    % m2/s [dispersion coefficient]
inject  = 10;      % m
measure = 90;      % m
tracer  = 1;       % kg

```

```
Ca = U * dt / dx;
Cd = D * dt / dx^2;
c = [];

% initial condition
for j = 1:(len/dx)
    if j*dx == inject
        c(1,j) = tracer/(dx*dep*wid);
    else
        c(1, j) = 0;
    end
end

% calculate in real time
for i = 1:(t/dt)
    for j = 3:(len/dx)-1
        c(i+1, j) = c(i, j) + ...
            (Cd*(1-Ca) - (Ca/6)*(Ca^2 - 3*Ca + 2))*c(i,j
               +1) - ...
            (Cd*(2 - 3*Ca) - (Ca/2)*(Ca^2 - 2*Ca - 1))*c(i
               ,j) + ...
            (Cd*(1 - 3*Ca) - (Ca/2)*(Ca^2 - Ca - 2))*c(i,j
               -1) + ...
            (Cd*Ca + (Ca/6)*(Ca^2 - 1))*c(i,j-2);
    end

    % plot results
    subplot(2,1,1);
    plot(1:(len/dx), c(i,:));
    xlabel('t');
    ylabel('c(t)');
    xlim([0 100]);
    ylim([0 0.1]);
    subplot(2,1,2);
    plot(c(:,measure/dx));
    xlabel('x');
    ylabel('c(x)');
    drawnow;
    pause(0.1);
end
```

This script simulates, how 1kg of tracer is being moved by the river. At runtime, it produces constant animation, showing tracer spatial distribution over time and change of tracer at given point (measure point) over time. Below, two pictures from different timestamps were shown:



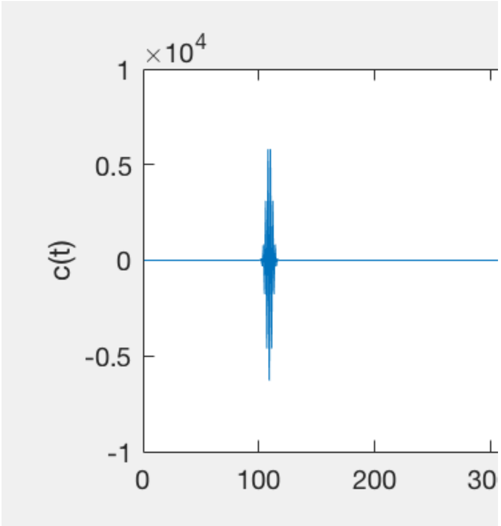


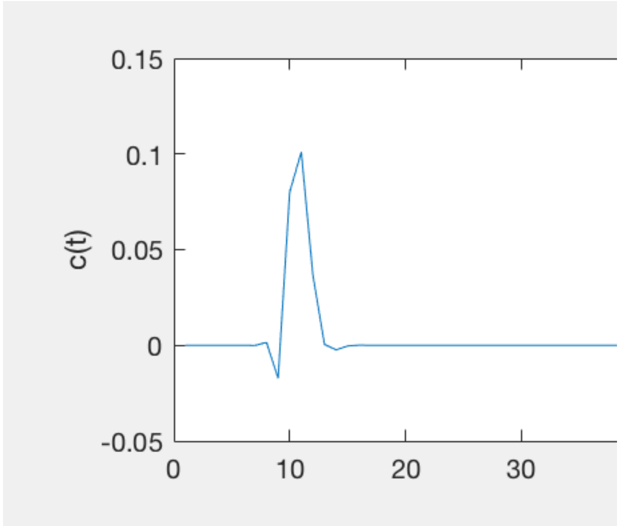
It is clearly visible that the "plateau" moves towards right over time, it makes sense as tracer flows within river. Also, as river and time flows, it is observable that amount of tracer at measure point variates in time (I guess it oscillates, sometime it increases, sometime it decreases).

2.2 Numerical stability

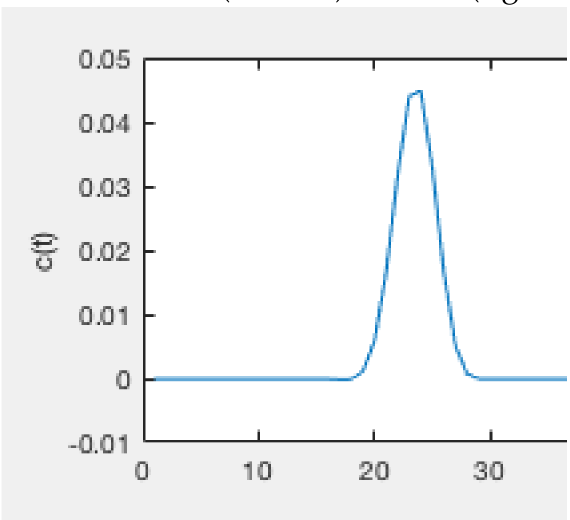
Due to numerical algorithm properties, there is a change that for given parameters, this numerical model will remain unstable. To check such behavior, it should be observable that for small change, one would see huge change in obtained results. For this purpose, various parameters were checked for such behavior.

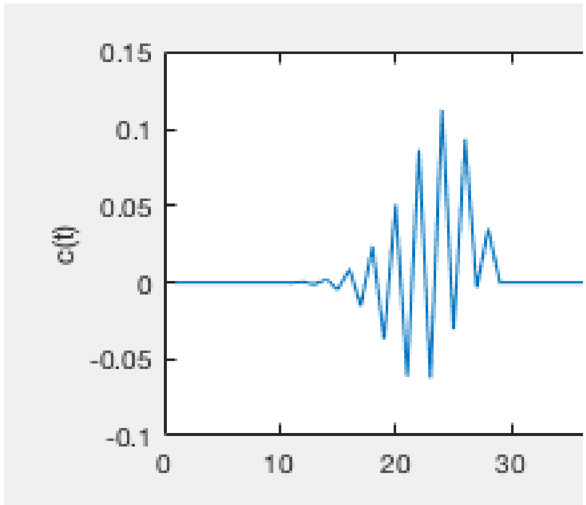
1. dx between 0.1m (left one) and 1m (right one)



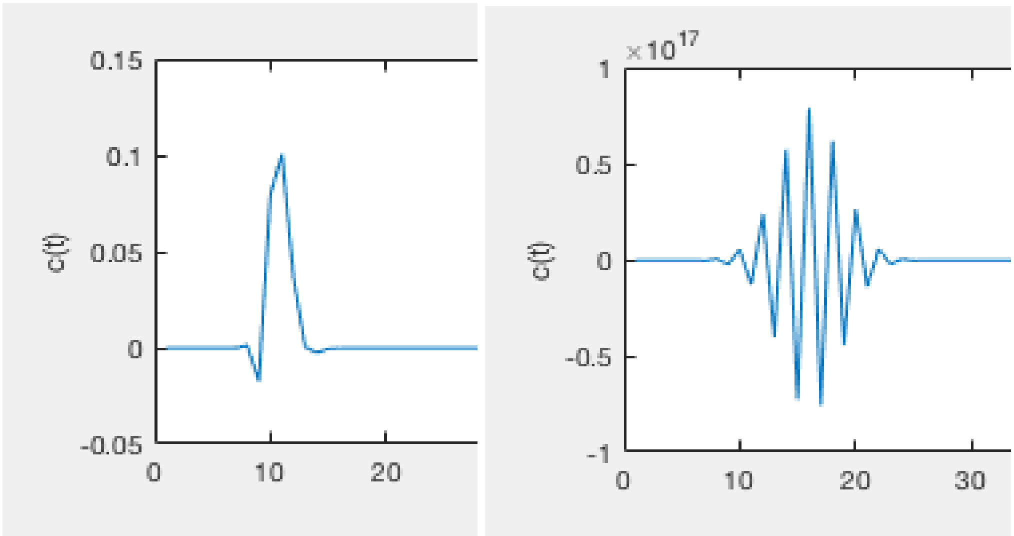


2. dt between 15s (left one) and 16s (right one)

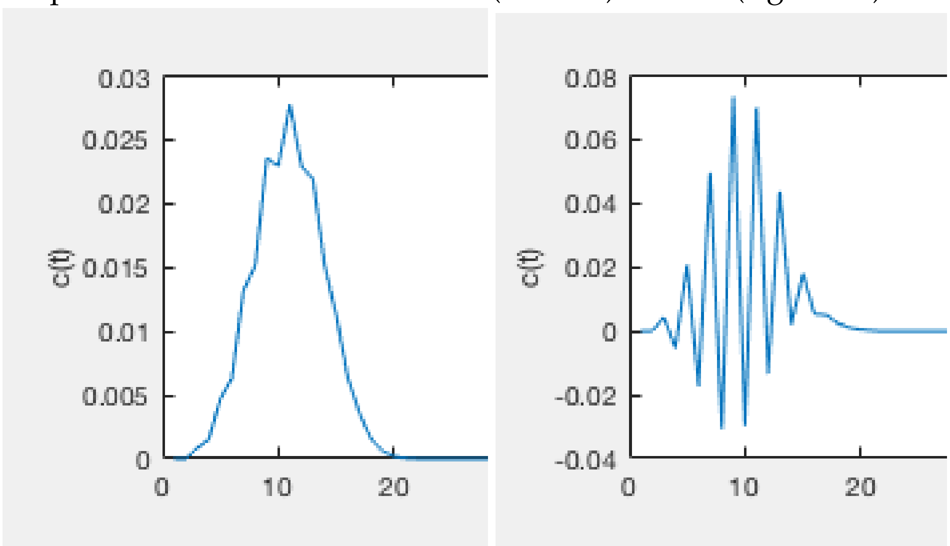




3. Advection coefficient between 0.1 (left one) and 5 (right one)



4. Dispersion coefficient between 0.5 (left one) and 0.6 (right one)



It is clearly visible that this model has tendencies for being numerically unstable and careful parameters choice is very important.

2.3 Mass conservation law

Mass conservation law states that, for closed system to all transfers of matter and energy, the mass of the system must remain constant over time. To assert, that this holds, matlab script can be expanded with vector of integrals of amount of tracer in river over time.

Listing 2: Matlab script for explicit taylor method

```
...
sums = [];
% calculate in real time
for i = 1:nt
    for j = 3:(len/dx)-1
        c(i+1, j) = c(i, j) + ...
            (Cd*(1-Ca) - (Ca/6)*(Ca^2 - 3*Ca + 2))*c(i,j+1) - ...
            (Cd*(2 - 3*Ca) - (Ca/2)*(Ca^2 - 2*Ca - 1))*c(i,j) + ...
            (Cd*(1 - 3*Ca) - (Ca/2)*(Ca^2 - Ca - 2))*c(i,j-1) + ...
            (Cd*Ca + (Ca/6)*(Ca^2 - 1))*c(i,j-2);
    end
        sums(i) = sum(c(i,:));
    ...
end

K>> length(unique(sums))

ans =

     1
```

Which means, that all integrals stored in "sums" vector has same value (or roughly the same).

3 Conclusion

This report covered simplified simulation of river pollution using quickest Taylor model implementation. It proved its easiness to implement, and good accuracy in numerical approach to this problem, which was confirmed by mass conservation law. However, it is highly vulnerable to osculations and instability with unwise parameters choosing. All in all, author of this report is satisfied with obtained results.