FLOW ROUTING CODES

```
% Solution of 1-D Flow Routing Equation by Bright Takyi (10878402)
% Flow Routing Equation; Q_t = -c*Q_x + u*Q_xx
% Initial and Boundary Conditions;
% Q(x,0)=Q0, Q(0,t)=Q1, dQ/dx=0
%**************************************************************************
% Initializing Parameters
clear
close all
clc

L = 10000; % lenght of river [m]
%T = 21*24*3600; % simulation time [sec]
T = 7200; % simulation time [sec]
dt = 1;
dx = 20;
n = T/dt;
m = L/dx;
%oneday = 24*3600; % converting one-day into seconds
%m = 100; % number of subintervals of length of river
%n = 15; % number of subintervals of simulation time
%n = 720; % number of subintervals of simulation time
%dx = L/m; % space step-size
%dt = T/n; % time step-size
x = linspace(0,L,m+1); % space discretization
%dx = x(2)-x(1);
t = linspace(0,T,n+1); % time discretization
rx = dt/(2.*dx);
rxx = dt/(dx^2);
c = 3.*ones(m+1,1); % wave celerity [m/s]
u = 0.003.*ones(m+1,1); % diffusivity coefficient [m^3/s]
%u1 = xlsread('C:\Users\G\Desktop\Diffusivity.xls','A4:B23');
%u=repmat(u1',m+1,1);% diffusivity coefficient [m^3/s]

% Setting up Q matrix, initial and boundary conditions
Q = zeros(m+1,n+1);
Q(:,1) = 50; % initial condition
Q(1,:) = 70; % boundary condition
Qa = Q(1);

% For loop
for k = 2:n+1   % temporal loop
    aaa = zeros(m+1,1);
    bbb = zeros(m+1,1);
    ddd = zeros(m+1,1);
    B = zeros(m,m);
    F = zeros(m+1,1);
    for i = 2:m+1   % spatial loop
          aaa(i) = -c(i)*rx-0.5*(u(i-1)*rxx+u(i)*rxx);
          if i==m+1
              bbb(i) = 1+0.5*(u(i-1)*rxx+u(i)*rxx+u(i-1)*rxx+u(i)*rxx);
              ddd(i) = -0.5*u(i-1)*rxx-0.5*u(i)*rxx+c(i)*rx;
          else   % for i < m+1
              bbb(i) = 1+0.5*(u(i+1)*rxx+u(i)*rxx+u(i-1)*rxx+u(i)*rxx);
              ddd(i) = -0.5*u(i+1)*rxx-0.5*u(i)*rxx+c(i)*rx;
```

```
          end

        if i==2
            F(i) = Q(i,k-1)-aaa(i)*Qa;
            B(i-1,i-1) = bbb(i);
            B(i-1,i) = ddd(i);
        elseif i==m+1
            B(i-1,i-1) = bbb(i);
            B(i-1,i-2) = aaa(i)+ddd(i);
            F(i) = Q(i,k-1);
        else
            B(i-1,i-2) = aaa(i);
            B(i-1,i-1) = bbb(i);
            B(i-1,i) = ddd(i);
            F(i) = Q(i,k-1);

        end
    end



    Q(2:end,k) = (B\F(2:end,1));
end
figure
plot(t(1:1001),Q(1,1:1001),'LineWidth',3);
hold on
plot(t(1:1001),Q(26,1:1001),'LineWidth',3);
plot(t(1:1001),Q(51,1:1001),'LineWidth',3);
plot(t(1:1001),Q(151,1:1001),'LineWidth',3);
%plot(t(1:1001),Q(1201,1:1001),'LineWidth',3);
xlabel('Time [sec]')
ylabel('River Discharge (Q)[m^3/s]')
legend('x = 0 m','x = 500 m', 'x = 1000 m', 'x = 3000
m','location','NorthEast')
%title('Graph of River Discharge for various distance')

%figure
%plot(t,Q,'LineWidth',3)
%xlabel('Time [sec]')
%ylabel('River Discharge (Q)[m^3/s]')
%legend('x = 0km','x = 1km','x = 2km','x = 4km','x = 6km','x = 8km','x =
10km','location','NorthWest')
%title('Graph of River Discharge for various distance')



figure
%plot(x,Q, 'LineWidth',3)
plot(x,Q(:,101),'LineWidth',3);
hold on
plot(x,Q(:,201),'LineWidth',3);
plot(x,Q(:,501),'LineWidth',3);
plot(x,Q(:,1001),'LineWidth',3);
%plot(x,Q(:,2501),'LineWidth',3);
xlabel('distance [m]')
```

```
ylabel('River Discharge (Q)[m^3/s]')
%title('Graph of River Discharge for various times')
legend('After 100 sec','After 200 sec', 'After 500 sec','After 1000 sec')
```