# ENVE404: Environmental Modeling

# Homework 3

By Waqas Bin Hamed (2176923)

**Submitted on:** 4/11/2019

**Instructors:**

Sarp Çelebi

Bahar Evren

**Question 1)**

**a)**

**Steady-state Mass Balances**

**Lake A:** $Q_{w1}C_{w1} - Q_{12}C_a - Q_{i1}C_a - V_a r_a = 0$

**Lake B:** $Q_{w2}C_{w2} + Q_{12}C_a - Q_{23}C_b - Q_{i2}C_b - V_b r_b = 0$

**Lake C:** $Q_{w3}C_{w3} + Q_{23}C_b - Q_{34}C_c - Q_{i3}C_c - V_c r_c = 0$

**Lake D:** $Q_{w4}C_{w4} + Q_{34}C_c - Q_{i4}C_d - Q_{4r}C_d - V_d r_d = 0$

where $r = kC^2$

# Code

```
clear all
close all
clc

Q_w = [60 30 52.08 24]; %incoming flowrates in m^3/day
Q_i = [50 35 53 20];     %outgoing flowrates in m^3/day

Q_12=Q_w(1)-Q_i(1); %other flowrates in m^3/day
Q_23=Q_w(2)+Q_12-Q_i(2);
Q_34=Q_w(3)+Q_23-Q_i(3);
Q_4r=Q_w(4)+Q_34-Q_i(4);

C_w = [15 8 14 12]; %incoming concentrations in mg/L
C_w = C_w.*(10^3);  %in mg/m^3

V = [5.2 4.1 7.3 2.7];
V = V.*(10^6);  %lake volumes in 10^6m^3

k = [0.535 0.467 0.952 0.368];  %decay rate in m^3y^-1kg^-1
k = k.*((10^-6)*(1/365));   %in m^3d^-1mg^-1

syms C_a C_b C_c C_d;

r = [k(1)*(C_a)^2 k(2)*(C_b)^2 k(3)*(C_c)^2 k(4)*(C_d)^2]; %decay rates

%steady-state mass balances for each lake
MB_a= (Q_w(1)*C_w(1))-(Q_i(1)*C_a)-(Q_12*C_a)-V(1)*r(1);
MB_b=(Q_12*C_a)+(Q_w(2)*C_w(2))-(Q_i(2)*C_b)-(Q_23*C_b)-V(2)*r(2);
MB_c=(Q_23*C_b)+(Q_w(3)*C_w(3))-(Q_i(3)*C_c)-(Q_34*C_c)-V(3)*r(3);
MB_d=(Q_34*C_c)+(Q_w(4)*C_w(4))-(Q_i(4)*C_d)-(Q_4r*C_d)-V(4)*r(4);

[solC_a,solC_b,solC_c,solC_d] = solve([MB_a == 0, MB_b == 0, MB_c == 0, MB_d == 0], [C_a, C_b,...
    C_c, C_d]);

solC_a = double(solC_a);
solC_b = double(solC_b);
solC_c = double(solC_c);
solC_d = double(solC_d);

% choosing only the positive real roots
for i = 1:length(solC_a)
    if  ((solC_a(i) >= 0) && (real(solC_a(i)) == solC_a(i))&&...
            (solC_b(i) >= 0) && (real(solC_b(i)) == solC_b(i))&&...
            (solC_c(i) >= 0) && (real(solC_c(i)) == solC_c(i))&&...
            (solC_d(i) >= 0) && (real(solC_d(i)) == solC_d(i)))
        C_a = solC_a(i);
        C_b = solC_b(i);
        C_c = solC_c(i);
        C_d = solC_d(i);
    end
end
```

```
disp(['Lake A concentration (mg/m^3) = ' num2str(C_a)])
disp(['Lake B concentration (mg/m^3) = ' num2str(C_b)])
disp(['Lake C concentration (mg/m^3) = ' num2str(C_c)])
disp(['Lake D concentration (mg/m^3) = ' num2str(C_d)])
```

# Results

```
Lake A concentration (mg/m^3) = 7621.3505
Lake B concentration (mg/m^3) = 4837.0096
Lake C concentration (mg/m^3) = 4967.2077
Lake D concentration (mg/m^3) = 6667.9078
```

# b)

# Code

```
clear all
close all
clc

Q_w = [60 30 52.08 24]; %incoming flowrates in m^3/day
Q_i = [50 35 53 20];    %outgoing flowrates in m^3/day

syms Q_23 Q_34 Q_4r;

Q_12=Q_w(1)-Q_i(1);

CEq_2=Q_23-0.4*Q_4r==Q_w(2)+Q_12-Q_i(2);    %continuity eqs. for flowrates
CEq_3=-1*Q_23+Q_34==Q_w(3)-Q_i(3);
CEq_4=-1*Q_34+Q_4r==Q_w(4)-Q_i(4);

[A,B] = equationsToMatrix([CEq_2,CEq_3,CEq_4]);  %calculation of pther flowrates
x = A\B;
Q_23 = double(x(1,1));
Q_34= double(x(2,1));
Q_4r= double(x(3,1));

C_w = [15 8 14 12]; %incoming concentrations in mg/L
C_w = C_w.*(10^3);  %in mg/m^3

V = [5.2 4.1 7.3 2.7];
V = V.*(10^6);  %lake volumes in 10^6m^3

k = [0.535 0.467 0.952 0.368];  %decay rate in m^3y^-1kg^-1
k = k.*((10^-6)*(1/365));   %in m^3d^-1mg^-1

syms C_a C_b C_c C_d;


r = [k(1)*(C_a)^2 k(2)*(C_b)^2 k(3)*(C_c)^2 k(4)*(C_d)^2]; %decay rates

%steady-state mass balances for each lake
MB_a= (Q_w(1)*C_w(1))-(Q_i(1)*C_a)-(Q_12*C_a)-V(1)*r(1);
MB_b=(Q_12*C_a)+(Q_w(2)*C_w(2))+(0.4*Q_4r*C_d)-(Q_i(2)*C_b)-(Q_23*C_b)...
    -V(2)*r(2); %modified equation
MB_c=(Q_23*C_b)+(Q_w(3)*C_w(3))-(Q_i(3)*C_c)-(Q_34*C_c)-V(3)*r(3);
MB_d=(Q_34*C_c)+(Q_w(4)*C_w(4))-(Q_i(4)*C_d)-(Q_4r*C_d)-V(4)*r(4);

[solC_a,solC_b,solC_c,solC_d] = solve([MB_a == 0, MB_b == 0, MB_c == 0, MB_d == 0], [C_a, C_b,...
    C_c, C_d]);

solC_a = double(solC_a);
solC_b = double(solC_b);
solC_c = double(solC_c);
solC_d = double(solC_d);

% choosing only the positive real roots
for i = 1:length(solC_a)
```

```
    if  ((solC_a(i) >= 0) && (real(solC_a(i)) == solC_a(i))&&...
            (solC_b(i) >= 0) && (real(solC_b(i)) == solC_b(i))&&...
            (solC_c(i) >= 0) && (real(solC_c(i)) == solC_c(i))&&...
            (solC_d(i) >= 0) && (real(solC_d(i)) == solC_d(i)))
        C_a = solC_a(i);
        C_b = solC_b(i);
        C_c = solC_c(i);
        C_d = solC_d(i);
    end
end

disp(['Lake A concentration (mg/m^3) = ' num2str(C_a)])
disp(['Lake B concentration (mg/m^3) = ' num2str(C_b)])
disp(['Lake C concentration (mg/m^3) = ' num2str(C_c)])
disp(['Lake D concentration (mg/m^3) = ' num2str(C_d)])
```

## Results

```
Lake A concentration (mg/m^3) = 7621.3505
Lake B concentration (mg/m^3) = 4931.7236
Lake C concentration (mg/m^3) = 4968.3301
Lake D concentration (mg/m^3) = 6536.0764
```

The riverine concentration before pumping = 6667.91mg/ $m^3$

The riverine concentration after pumping = 6536.08mg/$m^3$

The pumping of water back into Lake B doesn't have a significant difference on the riverine concentration.

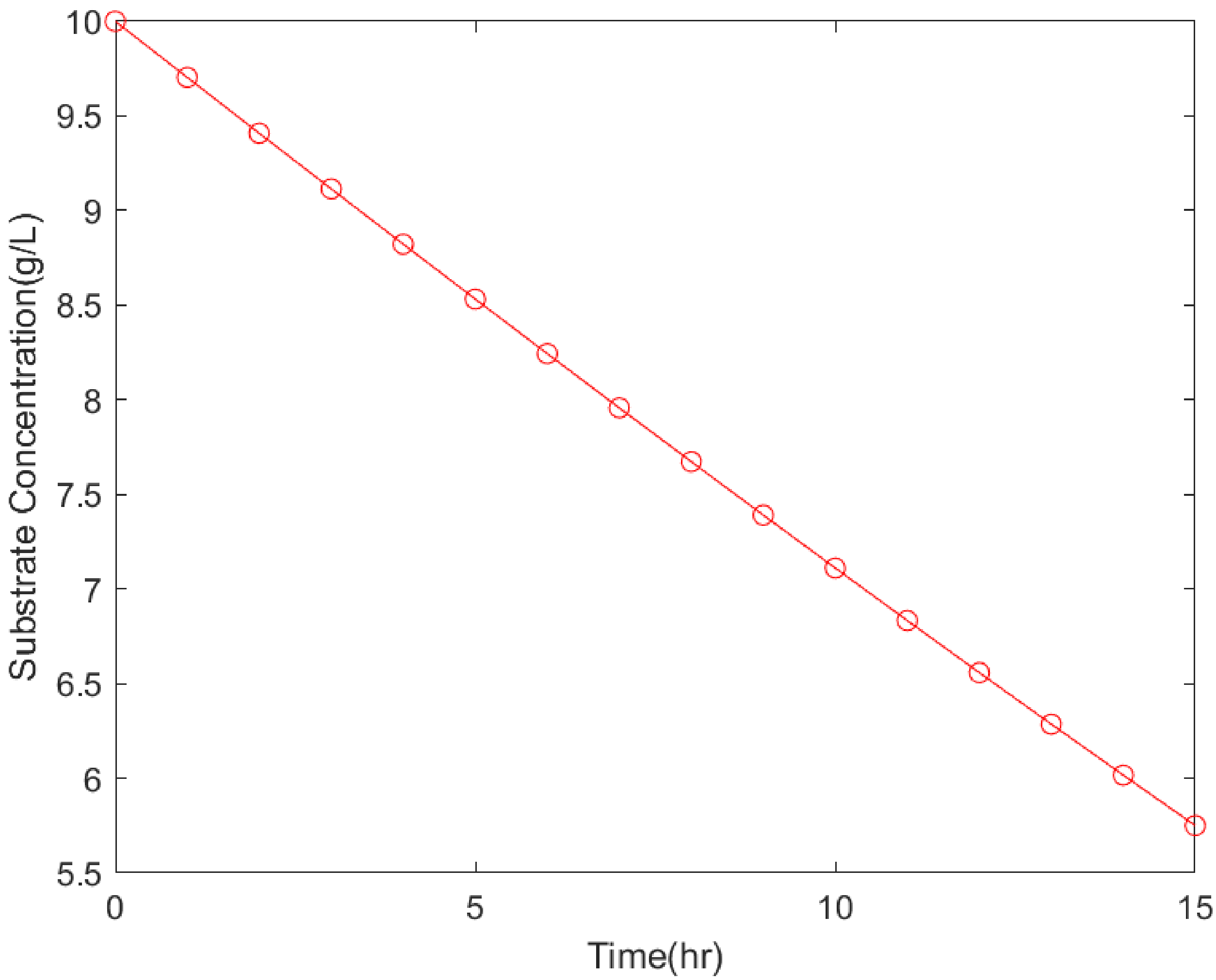
## Question 2)

## Code

```
close all
clear all
clc

subC = zeros(1,15);
X_0=0.5;    %g/L
S_0=10; %g/L
Y=0.35;
K_s=2;  %mg/L
K_0=0.25;   %1/hr

for t = 0:15

    f = @(S)K_s*log(S/S_0)+S-S_0+(X_0*K_0*t/Y);
        S = fsolve(f,1);
        subC(t+1) = S;

end

plot(0:15, subC,'-r o')
xlabel('Time(hr)');
ylabel('Substrate Concentration(g/L)');
subC = [0:15; subC];
fprintf(1, ' Time(hr)\t\t Substrate Concentration(g/L)\n') % Column Titles
fprintf(1, '%2d\t\t         %2d\n', subC) % Write Rows
```

## Results

| Time(hr) | Substrate Concentration(g/L) |
|---|---|
| 0 | 1.000000e+01 |
| 1 | 9.703130e+00 |
| 2 | 9.407805e+00 |
| 3 | 9.114097e+00 |
| 4 | 8.822083e+00 |
| 5 | 8.531845e+00 |
| 6 | 8.243470e+00 |
| 7 | 7.957053e+00 |
| 8 | 7.672692e+00 |
| 9 | 7.390495e+00 |
| 10 | 7.110575e+00 |
| 11 | 6.833055e+00 |
| 12 | 6.558065e+00 |
| 13 | 6.285744e+00 |
| 14 | 6.016244e+00 |
| 15 | 5.749724e+00 |




**Question 3)**

**a)**

**Mass Balances**

**Tank 2:** $\frac{dC_1}{dt} = W - C_2Q_2 - D(C_2 - C_1) - kC_2V_2$

**Tank 1:** $\frac{dC_1}{dt} = C_2Q_2 + D(C_1 - C_2) - C_1Q_{out} - kC_1V_1$

**Code**

```
clear all
close all
clc

syms Q_2     %Flowrates
syms Q_out

syms V_1     %Volumes
syms V_2

syms C_1     %Concentrations
syms C_2

syms W      %Mass

syms k      %decay constant
syms D      %diffusive flow rate (L^3/T)


%Mass Balances
MB_2=W-Q_2*C_2-D*(C_2-C_1)-k*C_2*V_2;
fprintf('Tank 2 Mass Balance: \ndC_2/dt= %s\n',char(MB_2));
MB_1=Q_2*C_2+D*(C_2-C_1)-k*C_1*V_1-Q_out*C_1;
fprintf('\nTank 1 Mass Balance: \ndC_1/dt= %s\n',char(MB_1));
```

**Results**

```
Tank 2 Mass Balance:
dC_2/dt= W + D*(C_1 - C_2) - C_2*Q_2 - C_2*V_2*k

Tank 1 Mass Balance:
dC_1/dt= C_2*Q_2 - D*(C_1 - C_2) - C_1*Q_out - C_1*V_1*k
```

**b)**

**Code**

```
clear all
close all
clc


W=1825*10^9;     %Mass

Q_2=7*10^9;      %Flowrates
Q_out=16*10^9;

V_1=3507*10^9;  %Volumes
V_2=8*10^9;

A_c=1.7*10^5;    %cross-sectional area between tanks
k_L=1.48*10^5;  %diffusion mass-transfer coefficient
D=k_L*A_c;  %diffusive flowrate

k=(0.693/43.8)*365; %decay constant

%concentrations
syms C_1
syms C_2

%mass balances
MB_1=Q_2*C_2+D*(C_2-C_1)-k*C_1*V_1-Q_out*C_1;
MB_2=W-Q_2*C_2-D*(C_2-C_1)-k*C_2*V_2;

[A, b] = equationsToMatrix ([MB_1, MB_2],[C_1, C_2]);
x = A\b;
C_1 = double(x(1,1));
C_2 = double(x(2,1));

fprintf(['Tank 1 concentration = ' num2str(C_1)]);
fprintf(['\nTank 2 concentration = ' num2str(C_2)]);
fprintf(['\n']);
```

## Results

```
Tank 1 concentration = 0.036926
Tank 2 concentration = 23.3018
```