ENVE404: Environmental Modeling

Homework 4

By Waqas Bin Hamed (2176923)

Submitted on: 11/11/2019

Instructors:

Sarp Çelebi

Bahar Evren

Question 1)

a)

Script

```
clc
clear all
close all

v = [0.715 0.14 0.372 0.368 0.824  0.309 0.41 0.301 1.01 0.608];
d = [0.335 0.512 0.268 0.130 0.264 0.338 0.268 0.105 0.227 0.123];
w = [13.6 2.23 10.8 1.2 11.1 8.83 10 5 6.8 21.5];
L_y = [605 3.83 379 15.9 575 201 382 191 322 817];
D_x = [29.5 0.0771 9.83 0.652 31.9 4.91 11 3.72 24.6 20.8];

Pe = ((L_y.*v)./D_x)';  %Peclet number calculation
wd = (w./d)';   %mean depth ratio calculation

T = table(wd,Pe);   %table of wd and Pe values
disp(T)
```

Results

```
      wd         Pe  
    ______     ______

    40.597     14.664
    4.3555     6.9546
    40.299     14.343
    9.2308     8.9742
    42.045     14.853
    26.124     12.649
    37.313     14.238
    47.619     15.455
    29.956      13.22
     174.8     23.882
```

b)

Script

```
clc
clear all
close all
format short g

v = [0.715 0.14 0.372 0.368 0.824  0.309 0.41 0.301 1.01 0.608];
d = [0.335 0.512 0.268 0.130 0.264 0.338 0.268 0.105 0.227 0.123];
w = [13.6 2.23 10.8 1.2 11.1 8.83 10 5 6.8 21.5];
L_y = [605 3.83 379 15.9 575 201 382 191 322 817];
D_x = [29.5 0.0771 9.83 0.652 31.9 4.91 11 3.72 24.6 20.8];

Pe = ((L_y.*v)./D_x)';  %Peclet number calculation
wd = (w./d)';   %mean depth ratio calculation

%Using the generalized least square method
Z = [ones(size(Pe)) log(wd)];
a = (Z'*Z)\(Z'*log(wd));

fprintf('Alpha:%8.4f\n', a(1))
fprintf('Beta:%8.4f\n', a(2))

PeE = a(1)+a(2)*log(wd);    %Pe estimates  calculation
dif = Pe - PeE; %difference between calculated and estimated Pe values
MQE = sqrt(sum(dif.*dif))/10;   %MQE calculation
fprintf('Mean Quadratic Error:%8.4f\n', MQE)
```

Results

```
Alpha:  0.0000
Beta:  1.0000
Mean Quadratic Error:  3.4852
```

c)

Script

```
clc
clear all
close all
format short g

v = [0.715 0.14 0.372 0.368 0.824  0.309 0.41 0.301 1.01 0.608];
d = [0.335 0.512 0.268 0.130 0.264 0.338 0.268 0.105 0.227 0.123];
w = [13.6 2.23 10.8 1.2 11.1 8.83 10 5 6.8 21.5];
L_y = [605 3.83 379 15.9 575 201 382 191 322 817];
D_x = [29.5 0.0771 9.83 0.652 31.9 4.91 11 3.72 24.6 20.8];

Pe = ((L_y.*v)./D_x)';  %Peclet number calculation
wd = (w./d)';   %mean depth ratio calculation

%Finding coefficients for 2nd order polynomial
pPoly=polyfit(wd,Pe,2);
%calculating MQE for 2nd order polynomial
PeE1 = pPoly(1).*(wd.^2)+pPoly(2).*wd+pPoly(3);
dif1 = Pe - (PeE1);
MQE_Poly = sqrt(sum(dif1.*dif1))/10;
fprintf('MQE for 2nd Order Polynomial:%8.4f\n', MQE_Poly)

%Finding coefficients for power function
pPow=polyfit(log(wd),log(Pe),1);
%calculating MQE for power function
PeE2 = exp(pPow(2)).*((wd).^pPow(1));
dif2 = Pe - (PeE2);
MQE_Pow = sqrt(sum(dif2.*dif2))/10;
fprintf('MQE for Power function:%8.4f\n', MQE_Pow)

%Finding coefficients for exponential function
pExp=polyfit(wd,log(Pe),1);
%calculating MQE for exponential function
PeE3 = exp(pExp(2)).*exp(pExp(1).*wd);
dif3 = Pe - (PeE3);
MQE_Exp = sqrt(sum(dif3.*dif3))/10;
fprintf('MQE for Exponential function:%8.4f\n', MQE_Exp)
```

Results

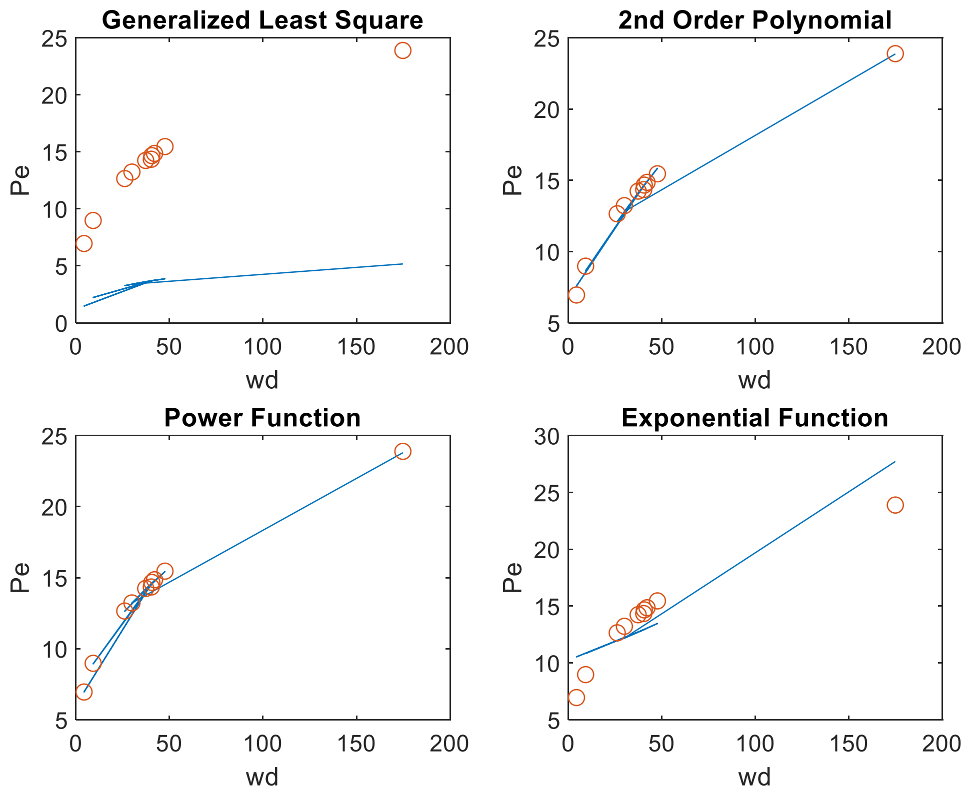
```
MQE for 2nd Order Polynomial:  0.1113
MQE for Power function:  0.0280
MQE for Exponential function:  0.6865
```

d)

Script

```
clc
clear all
close all
format short g

v = [0.715 0.14 0.372 0.368 0.824  0.309 0.41 0.301 1.01 0.608];
d = [0.335 0.512 0.268 0.130 0.264 0.338 0.268 0.105 0.227 0.123];
w = [13.6 2.23 10.8 1.2 11.1 8.83 10 5 6.8 21.5];
L_y = [605 3.83 379 15.9 575 201 382 191 322 817];
D_x = [29.5 0.0771 9.83 0.652 31.9 4.91 11 3.72 24.6 20.8];

Pe = ((L_y.*v)./D_x)';  %Peclet number calculation
wd = (w./d)';   %mean depth ratio calculation

%Using the generalized least square method for Pe estimates
Z = [ones(size(Pe)) log(wd)];
E = (Z'*Z)\(Z'*log(wd));
PeE = E(2)*log(wd) + E(1);

%Using the 2nd order polynomial for Pe estimates
pPoly=polyfit(wd,Pe,2);
PeE1 = pPoly(1).*(wd.^2)+pPoly(2).*wd+pPoly(3);

%Using the power function for Pe estimates
pPow=polyfit(log(wd),log(Pe),1);
PeE2 = exp(pPow(2)).*((wd).^pPow(1));

%Using the exponential for Pe estimates
pExp=polyfit(wd,log(Pe),1);
PeE3 = exp(pExp(2)).*exp(pExp(1).*wd);

subplot(2,2,1)
plot(wd,PeE,'-',wd,Pe,'o'); %plotting for generalized least square method
title('Generalized Least Square')
xlabel('wd')
ylabel('Pe')
subplot(2,2,2)
plot(wd,PeE1,'-',wd,Pe,'o');    %plotting for 2nd order polynomial
title('2nd Order Polynomial')
xlabel('wd')
ylabel('Pe')
subplot(2,2,3)
plot(wd,PeE2,'-',wd,Pe,'o');    %plotting for power function
title('Power Function')
xlabel('wd')
ylabel('Pe')
subplot(2,2,4)
plot(wd,PeE3,'-',wd,Pe,'o');    %plotting for exponential function
title('Exponential Function')
xlabel('wd')
ylabel('Pe')
```

Results



Generalized Least Square
2nd Order Polynomial
Power Function
Exponential Function
Pe
wd
0
50
100
150
200
0
5
10
15
20
25
30

e)

Script

```
clc
clear all
close all
format short g

v = [0.715 0.14 0.372 0.368 0.824  0.309 0.41 0.301 1.01 0.608];
d = [0.335 0.512 0.268 0.130 0.264 0.338 0.268 0.105 0.227 0.123];
w = [13.6 2.23 10.8 1.2 11.1 8.83 10 5 6.8 21.5];
L_y = [605 3.83 379 15.9 575 201 382 191 322 817];
D_x = [29.5 0.0771 9.83 0.652 31.9 4.91 11 3.72 24.6 20.8];

Pe = ((L_y.*v)./D_x)';  %Peclet number calculation
wd = (w./d)';   %mean depth ratio calculation

%Finding coefficients for Generalized Least Square method
Z = [ones(size(Pe)) log(wd)];
a = (Z'*Z)\(Z'*log(wd));
%calculating MQE for Generalized Least Square method
PeE = a(2)*log(wd);
dif = Pe - PeE;
MQE = sqrt(sum(dif.*dif))/10;

%Finding coefficients for 2nd order polynomial
pPoly=polyfit(wd,Pe,2);
%calculating MQE for 2nd order polynomial
PeE1 = pPoly(1).*(wd.^2)+pPoly(2).*wd+pPoly(3);
dif1 = Pe - (PeE1);
MQE_Poly = sqrt(sum(dif1.*dif1))/10;

%Finding coefficients for power function
pPow=polyfit(log(wd),log(Pe),1);
%calculating MQE for power function
PeE2 = exp(pPow(2)).*((wd).^pPow(1));
dif2 = Pe - (PeE2);
MQE_Pow = sqrt(sum(dif2.*dif2))/10;

%Finding coefficients for exponential function
pExp=polyfit(wd,log(Pe),1);
%calculating MQE for exponential function
PeE3 = exp(pExp(2)).*exp(pExp(1).*wd);
dif3 = Pe - (PeE3);
MQE_Exp = sqrt(sum(dif3.*dif3))/10;

%creating table for methods and their MQEs
MQEval = [MQE, MQE_Poly, MQE_Pow, MQE_Exp];
fprintf('Method                            MQE Calculated\n')
fprintf('Generalized Least Square  %8.4f\n',MQEval(1))
fprintf('2nd Order Polynomial      %8.4f\n',MQEval(2))
fprintf('Power function            %8.4f\n',MQEval(3))
fprintf('Exponential function      %8.4f\n',MQEval(4))

%finding out the best fit based on MQEs
if MQEval(1) == min(MQEval)
    fprintf('\nGeneralized Least Square Method is the best fit\n')
elseif MQEval(2) == min(MQEval)
        fprintf('\n2nd oder Polynomial function is the best fit\n')
elseif MQEval(3) == min(MQEval)
    fprintf('\nPower function is the best fit\n')
elseif MQEval(4) == min(MQEval)
        fprintf('\nExponential function is the best fit\n')
end
```

Results

```
Method                       MQE Calculated
Generalized Least Square       3.4852
2nd Order Polynomial           0.1113
Power function                 0.0280
Exponential function           0.6865

Power function is the best fit
```

Question 2)

a)

Function

```
%function for optimizing longitudinal dispersion coefficient
function f = fSSR(D,t,C_50)
v = 0.5;
C_0 = 80;
x = 50;

C = (C_0/2).*(erfc((x-v.*t)./(2*sqrt(D.*t)))+exp((v.*x)/D)*...
    erfc((x+v.*t)./(2*sqrt(D.*t))));
f = sum(sqrt(C_50-C));
end
```

Script

```
clear all
close all
clc

v = 0.5;
C_0 = 80;
x = 50;
t = 0:100:700;
C_50 = [0,40,59.6 69 73 76 77.5 78.5];

%using fuction fSSR.m to calculate D
D = fminsearch(@fSSR,1,[],t,C_50);
fprintf('The longitudinal dispersion coefficient(m^2/day): %8.4f\n',D)

%using D value to calculate C values at 50m
C = (C_0/2).*(erfc((x-v.*t)./(2*sqrt(D.*t)))+exp((v.*x)/D)*...
    (erfc((x+v.*t)./(2*sqrt(D.*t)))));
%calculating MQE using given and calculated C values at 50m
dif = C-C_50;
MQE = sqrt(sum(dif.*dif))/length(t);
fprintf('MQE: %8.4f\n',MQE)
```

Results

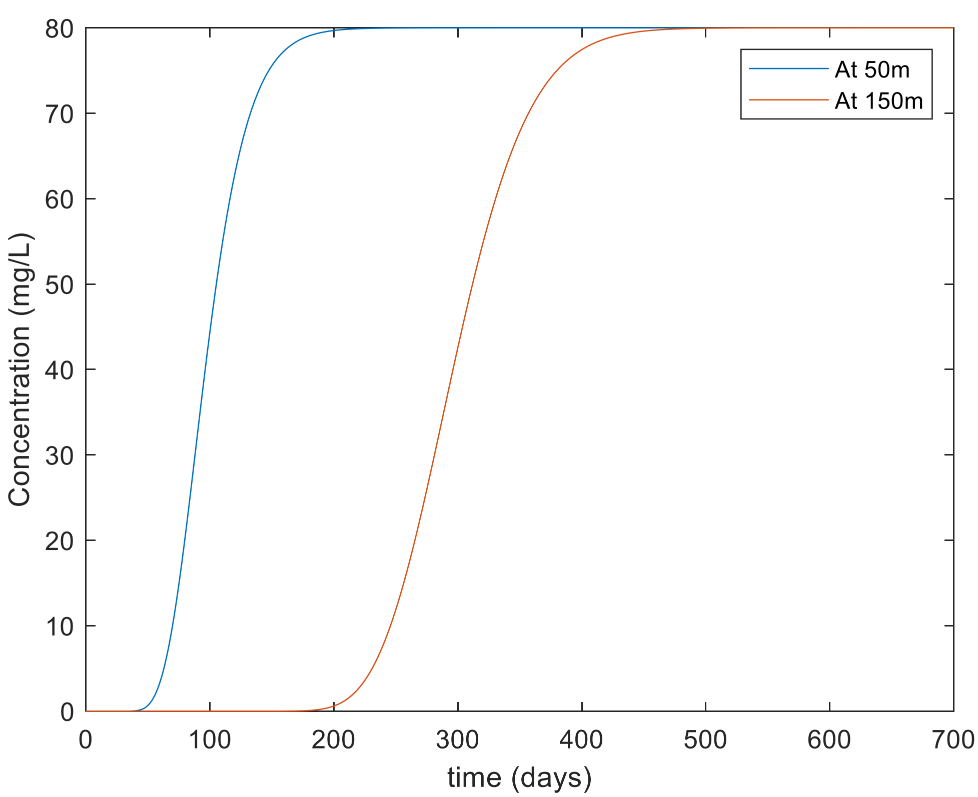
```
The longitudinal dispersion coefficient(m^2/day):   1.0000
MQE:   3.1064
```

b)

Script

```
clear all
close all
clc

v = 0.5;
C_0 = 80;
t = 0:700;
D = 1;  %calculated dispersion coefficient
x = [50 150];

%plotting C values at 50m and 150m
for i = 1:length(x)
    for j = 1:length(t)
    C(i,j) = (C_0/2).*(erfc((x(i)-v.*t(j))./(2*sqrt(D.*t(j))))+exp((v.*x(i)/D))*...
    erfc((x(i)+v.*t(j))./(2*sqrt(D.*t(j)))));
    end
    plot(t,C(i,:))
    xlabel('time (days)')
    ylabel('Concentration (mg/L)')
    hold on
end

legend('At 50m','At 150m')
```

Results



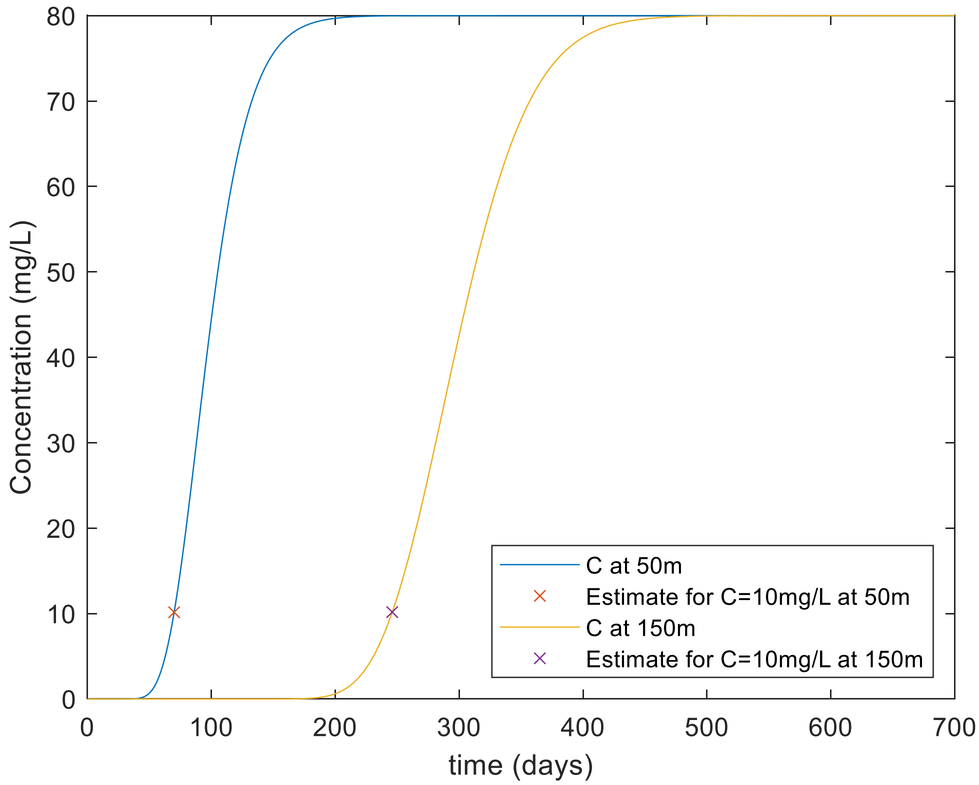
c)

Script

```
clear all
close all
clc

v = 0.5;
C_0 = 80;
t = 0:700;
D = 1;  %calculated dispersion coefficient
x = [50 150];


for i = 1:length(x)
    %plotting C values at 50m and 150m
    for j = 1:length(t)
    C(i,j) = (C_0/2).*(erfc((x(i)-v.*t(j))./(2*sqrt(D.*t(j))))+exp((v.*x(i)/D))*...
    erfc((x(i)+v.*t(j))./(2*sqrt(D.*t(j)))));
    end
    plot(t,C(i,:))
    xlabel('time (days)')
    ylabel('Concentration (mg/L)')
    hold on

    %estimating time for C to reach 10mg/L at 50m and 150m
    diff = abs(C-10);
    for h = 1:length(t)
        if diff(i,h) == min(diff(i,:))
            %plotting C values closest to 10mg/L
            plot(t(h),C(i,h),'x')
            fprintf('Time for concentration to reach 10mg/L at %dm: %d days\n'...
                ,x(i), t(h))
        end
    end
end

legend('C at 50m','Estimate for C=10mg/L at 50m','C at 150m',...
    'Estimate for C=10mg/L at 150m','Location','southeast')
```

Results

```
Time for concentration to reach 10mg/L at 50m: 70 days
Time for concentration to reach 10mg/L at 150m: 246 days
```



Concentration (mg/L)
time (days)
C at 50m
Estimate for C=10mg/L at 50m
C at 150m
Estimate for C=10mg/L at 150m

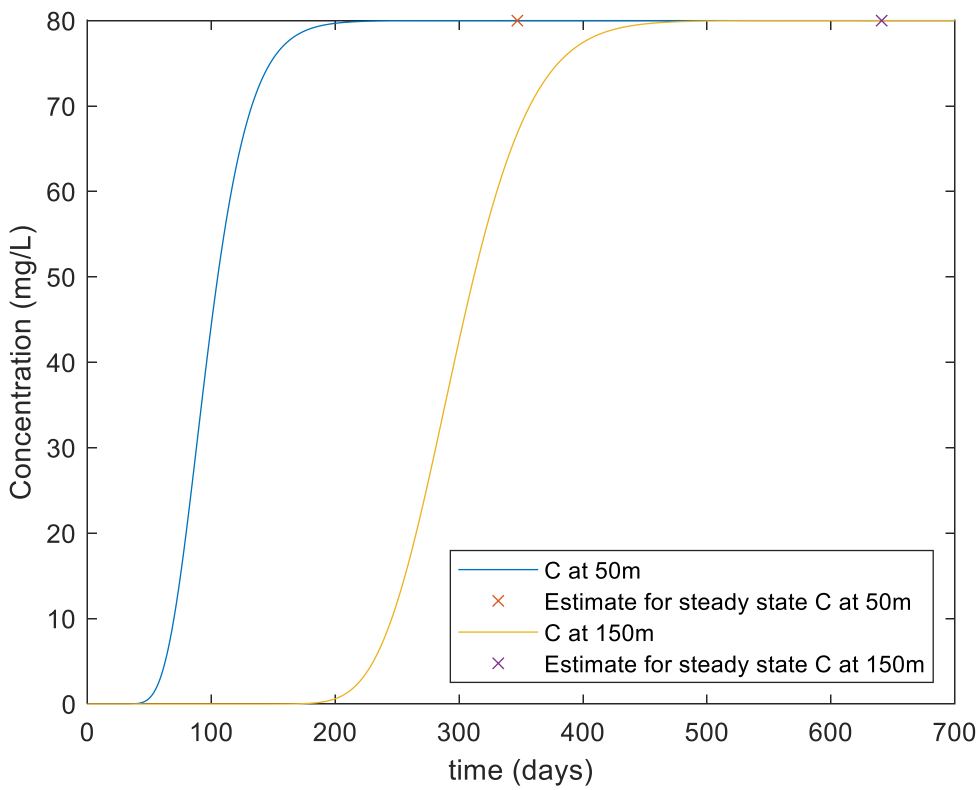
d)

Script

```
clear all
close all
clc

v = 0.5;
C_0 = 80;
t = 0:700;
D = 1;  %calculated dispersion coefficient
x = [50 150];


for i = 1:length(x)
    %plotting C values at 50m and 150m
    for j = 1:length(t)
    C(i,j) = (C_0/2).*(erfc((x(i)-v.*t(j))./(2*sqrt(D.*t(j))))+exp((v.*x(i)/D))*...
    erfc((x(i)+v.*t(j))./(2*sqrt(D.*t(j)))));
    end
    plot(t,C(i,:))
    xlabel('time (days)')
    ylabel('Concentration (mg/L)')
    hold on

    %estimating time for C to reach steadt state at 50m and 150m
    for h = 1:length(t)
        if round(C(i,h),4) == round(max(C(i,:)),4)
            %plotting steady state C values
            plot(t(h),C(i,h),'x')
            fprintf('Time for concentration to reach steady state at %dm: %d days\n'...
                ,x(i), t(h))
            break
        end
    end
end

legend('C at 50m','Estimate for steady state C at 50m','C at 150m',...
    'Estimate for steady state C at 150m','Location','southeast')
```

Results

```
Time for concentration to reach steady state at 50m: 347 days
Time for concentration to reach steady state at 150m: 641 days
```



Concentration (mg/L)
time (days)
0
10
20
30
40
50
60
70
80
0
100
200
300
400
500
600
700
C at 50m
Estimate for steady state C at 50m
C at 150m
Estimate for steady state C at 150m