ENVE404: Environmental Modeling
Homework 6

By Waqas Bin Hamed (2176923)

Submitted on: 6/12/2019

Instructors:

Sarp Çelebi
Bahar Evren

Question 1)

Governing Equation: $D\frac{d^2C}{dx^2} - v\frac{dC}{dx} - kC = 0$

Boundary condition 1: $vC_{in} = vC_{x=0} - D\frac{dC}{dx}_{x=0}$

Boundary condition 2: $\left(\frac{dC}{dx}\right)_{x=L} = 0$

For finite difference method:

$$\frac{dC_i}{dx} = \frac{C_i - C_{i-1}}{\Delta x}$$

$$\frac{d^2C}{dx^2} = \frac{C_{i-1} - 2C_i + C_{i+1}}{\Delta x^2}$$

Using this, the governing equation can be written as

$$D\left(\frac{C_{i-1} - 2C_i + C_{i+1}}{\Delta x^2}\right) - v\left(\frac{C_i - C_{i-1}}{\Delta x}\right) - kC_i = 0$$

Rearranging the equation

$$\left(\frac{D}{\Delta x^2} + \frac{v}{\Delta x}\right)C_{i-1} + \left(-\frac{2D}{\Delta x^2} - \frac{v}{\Delta x} - k\right)C_i + \left(\frac{D}{\Delta x^2}\right)C_{i+1} = 0$$

Simplifying the equation

$$a_1C_{i-1} + a_2C_i + a_3C_{i+1} = 0$$

Where $a_1 = \frac{D}{\Delta x^2} + \frac{v}{\Delta x}$, $a_2 = -\frac{2D}{\Delta x^2} - \frac{v}{\Delta x} - k$, $a_3 = \frac{D}{\Delta x^2}$

Nodes 1 to 7 respond to distance values from 0 to 120m with an increment of 20m.

Boundary condition 1 can be written as

$$vC_{i=1} - D\left(\frac{C_{i=1} - C_{i=2}}{\Delta x}\right) = vC_{in}$$

$$\left(v + \frac{D}{\Delta x}\right)C_{i=1} + \left(-\frac{D}{\Delta x}\right)C_{i=2} = vC_{in}$$

Simplifying the condition

$$b_1C_{i=1} + b_2C_{i=2} = vC_{in}$$

Where $b_1 = v + \frac{D}{\Delta x}$, $b_2 = -\frac{D}{\Delta x}$

Boundary condition 2 can be written as

$$\frac{C_7 - C_6}{\Delta x} = 0$$

Simplifying the condition

$$C_{i=6} - C_{i=7} = 0$$

The matrix required to solve the system:

$$A = \begin{bmatrix} b_1 & b_2 & 0 & 0 & 0 & 0 & 0 \\ a_1 & a_2 & a_3 & 0 & 0 & 0 & 0 \\ 0 & a_1 & a_2 & a_3 & 0 & 0 & 0 \\ 0 & 0 & a_1 & a_2 & a_3 & 0 & 0 \\ 0 & 0 & 0 & a_1 & a_2 & a_3 & 0 \\ 0 & 0 & 0 & 0 & a_1 & a_2 & a_3 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \mathrm{b} = \begin{bmatrix} vC_{in} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \mathrm{C} = \begin{bmatrix} C_{i=1} \\ C_{i=2} \\ C_{i=3} \\ C_{i=4} \\ C_{i=5} \\ C_{i=6} \\ C_{i=7} \end{bmatrix}$$

Solving $A * C = b$ will give us the concentration values.

a)

Function for Boundary Value Problem (same for part b)

```
function dCdx = bvp(x,C)

D = 4500;    %dispersion coefficient in m^2/hr
v = 75; %velocity in m/hr
k = 1.5;    %reaction rate in hr^-1

%C(1) = C
%C(2) = C'

dCdx(1) = C(2);
dCdx(2) = (v/D)*C(2)+(k/D)*C(1);

dCdx = dCdx(:);
```

Function for Boundary Conditions (same for part b)

```
function res = bc(C0,CL)

D = 4500;    %dispersion coefficient in m^2/hr
v = 75; %velocity in m/hr
Cin = 85;   %inflow concentration in mol/L

res(1) = v*Cin-v*C0(1)+D*C0(2);
res(2) = CL(2);

res = res(:);
```

Script

```
clc
clear all
close all

D = 4500;    %dispersion coefficient in m^2/hr
v = 75; %velocity in m/hr
Cin = 85;   %inflow concentration in mol/L
k = 1.5;    %reaction rate in hr^-1
L = 120;    %length of reactor in m

%Solving the BVP using bvp4c-----------------------------------------------

solinit=bvpinit(0:120,[0 0]);
sol=bvp4c(@bvp,@bc,solinit);

%Solving the BVP using Shooting Method-------------------------------------

err = [100 100]; %intial value for error
iv1(1) = 0;    %intial value for C at x=0
g = 2;

while err(g-1) > 5  %tolerance is 5%

    if g < 4
        iv1(g) = input('Guess C at x=0:');
    else
        iv1(g) = iv1(g-1)+(-1*err(g-1))*((iv1(g-1)-iv1(g-2))/(err(g-1)...
            -err(g-2)));    %next value is predicted using linear extrapolation
    end

    iv2 = (v/D)*(iv1(g)-Cin); %calculating dC/dt using boundary condition at x=0
    [x,shC] = ode45(@bvp,[0:120],[iv1(g) iv2]);
    err(g) = abs((shC(120,2)-0)*100);   %calculates error

    if err(g) > 5
        if g < 4
            fprintf('The error for your guess is %8.4f%%\n', err(g))
        else
            fprintf('Guess using linear extrapolation: %8.4f\n', iv1(g))
            fprintf('Error for this guess: %8.4f%%\n', err(g))
        end
    end

    g = g+1;
end

%Solving the BVP using Finite Difference Method---------------------------

delx = 20; %distance between nodes in m

a1 = D/(delx^2)+v/delx;
a2 = -2*D/(delx^2)-v/delx-k;
a3 = D/(delx^2);

b1 = v+D/delx;
b2 = -D/delx;
```

```
b = zeros(7,1);
b(1) = v*Cin;
A = diag(ones(length(b),1)*a2)+diag(ones(length(b)-1,1)*a3,1)+diag...
    (ones(length(b)-1,1)*a1,-1);
A(1,:)=[b1; b2; 0; 0; 0; 0; 0];
A(length(b),:)=[0; 0; 0; 0; 0; -1; 1];
finC = A\b;


%Creating table-----------------------------------------------------------
xT = 0:20:120;

VarNames = {'x','C (mol/L) using bvp4c','C (mol/L) using Shooting Method',...
    'C (mol/L) using Finite Difference'};

fprintf('\n')
T = table(xT',sol.y(1,xT+1)',shC(xT+1,1),finC,'VariableNames',VarNames);
disp(T)
```

Results

The first guess was 30, second guess was 10.

```
Guess C at x=0:30
The error for your guess is 1186.0909%
Guess C at x=0:10
The error for your guess is 2364.8903%
```

x	C (mol/L) using bvp4c	C (mol/L) using Shooting Method	C (mol/L) using Finite Difference
0	50.114	50.124	53.788
20	39.792	39.807	43.384
40	31.755	31.778	35.297
60	25.618	25.657	29.22
80	21.153	21.221	25.013
100	18.306	18.424	22.739
120	17.262	17.47	22.739

b)

Script

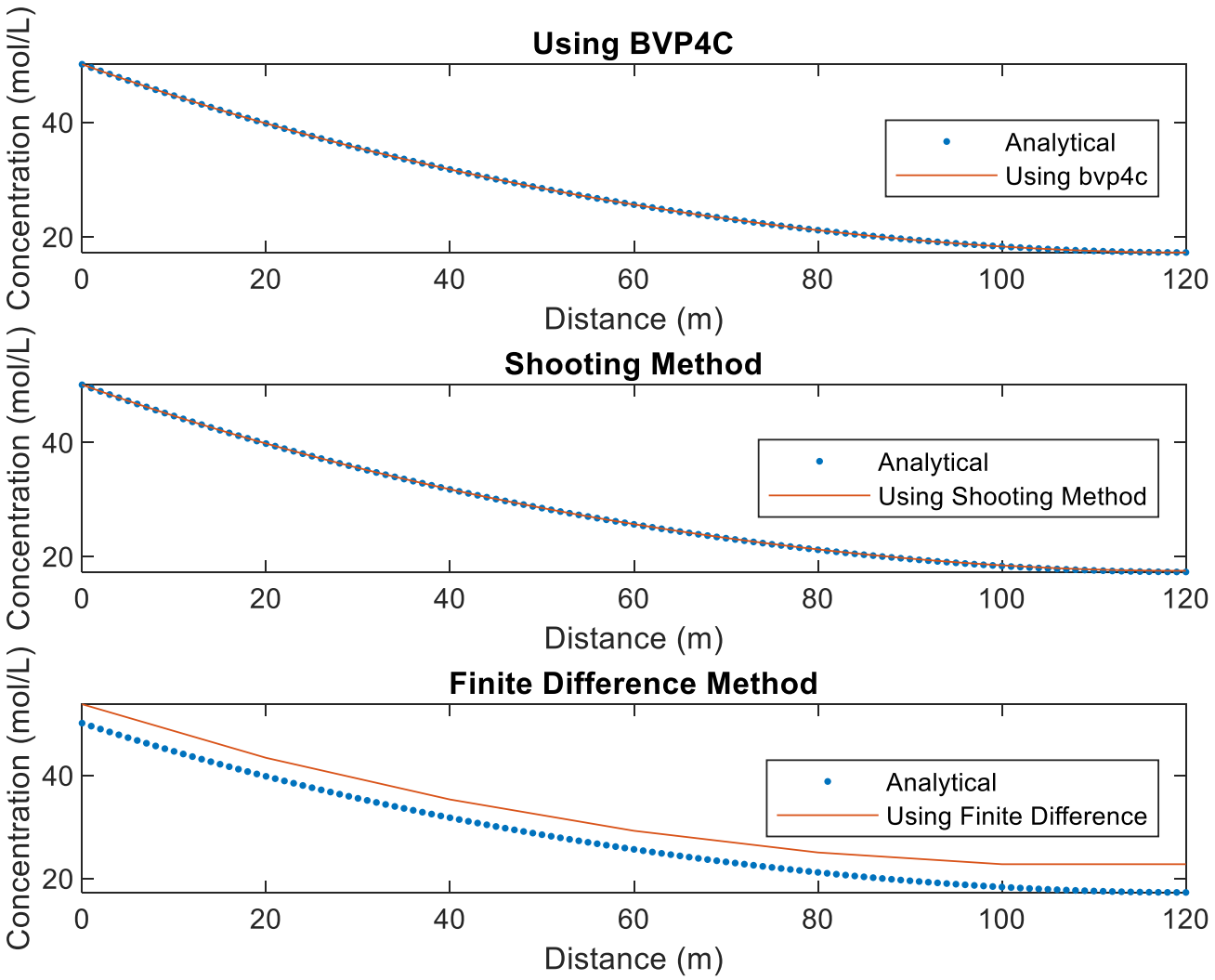
```
clc
clear all
close all

D = 4500;    %dispersion coefficient in m^2/hr
v = 75; %velocity in m/hr
Cin = 85;   %inflow concentration in mol/L
k = 1.5;    %reaction rate in hr^-1
L = 120;    %length of reactor in m

%Solving the BVP using bvp4c-----------------------------------------------

solinit=bvpinit(0:120,[0 0]);
sol=bvp4c(@bvp,@bc,solinit);

%Solving the BVP using Shooting Method-------------------------------------

err = [100 100]; %intial value for error
iv1(1) = 0;    %intial value for C at x=0
g = 2;

while err(g-1) > 5  %tolerance is 5%

    if g < 4
        iv1(g) = input('Guess C at x=0:');
    else
        iv1(g) = iv1(g-1)+(-1*err(g-1))*((iv1(g-1)-iv1(g-2))/(err(g-1)...
            -err(g-2)));    %next value is predicted using linear extrapolation
    end

    iv2 = (v/D)*(iv1(g)-Cin); %calculating dC/dt using boundary condition at x=0
    [x,shC] = ode45(@bvp,[0:120],[iv1(g) iv2]);
    err(g) = abs((shC(120,2)-0)*100);   %calculates error

    if err(g) > 5
        if g < 4
            fprintf('The error for your guess is %8.4f%%\n', err(g))
        else
            fprintf('Guess using linear extrapolation: %8.4f\n', iv1(g))
            fprintf('Error for this guess: %8.4f%%\n', err(g))
        end
    end

    g = g+1;
end

%Solving the BVP using Finite Difference Method--------------------------

delx = 20; %distance between nodes in m

a1 = D/(delx^2)+v/delx;
a2 = -2*D/(delx^2)-v/delx-k;
a3 = D/(delx^2);
```

```
b1 = v+D/delx;
b2 = -D/delx;

b = zeros(7,1);
b(1) = v*Cin;
A = diag(ones(length(b),1)*a2)+diag(ones(length(b)-1,1)*a3,1)+diag...
    (ones(length(b)-1,1)*a1,-1);
A(1,:)=[b1; b2; 0; 0; 0; 0; 0];
A(length(b),:)=[0; 0; 0; 0; 0; -1; 1];
finC = A\b;


%Analytical Solution-------------------------------------------------------

l1 = (v/(2*D))*(1-sqrt(1+(4*k*D)/(v^2)));
l2 = (v/(2*D))*(1+sqrt(1+(4*k*D)/(v^2)));

C = ((v*Cin)/((v-D*l1)*l2*exp(l2*L)-(v-D*l2)*l1*exp(l1*L)))*(l2*exp...
    (l2*L)*exp(l1*x)-l1*exp(l1*L)*exp(l2*x));

%Plotting the graphs-------------------------------------------------------

subplot(3,1,1)
plot(x,C,'.')
hold on
plot(x,sol.y(1,:)')
title(sprintf('Using BVP4C'))
xlabel('Distance (m)')
ylabel('Concentration (mol/L)')
legend('Analytical','Using bvp4c','location','east')

subplot(3,1,2)
plot(x,C,'.')
hold on
plot(x,shC(:,1))
title(sprintf('Shooting Method'))
xlabel('Distance (m)')
ylabel('Concentration (mol/L)')
legend('Analytical','Using Shooting Method','location','east')

subplot(3,1,3)
plot(x,C,'.')
hold on
xT = 0:20:120;
plot(xT,finC)
title(sprintf('Finite Difference Method'))
xlabel('Distance (m)')
ylabel('Concentration (mol/L)')
legend('Analytical','Using Finite Difference','location','east')
```

Results

The first guess was 30, second guess was 10.

```
Guess C at x=0:30
The error for your guess is 1186.0909%
Guess C at x=0:10
The error for your guess is 2364.8903%
```



Bonus)

Script

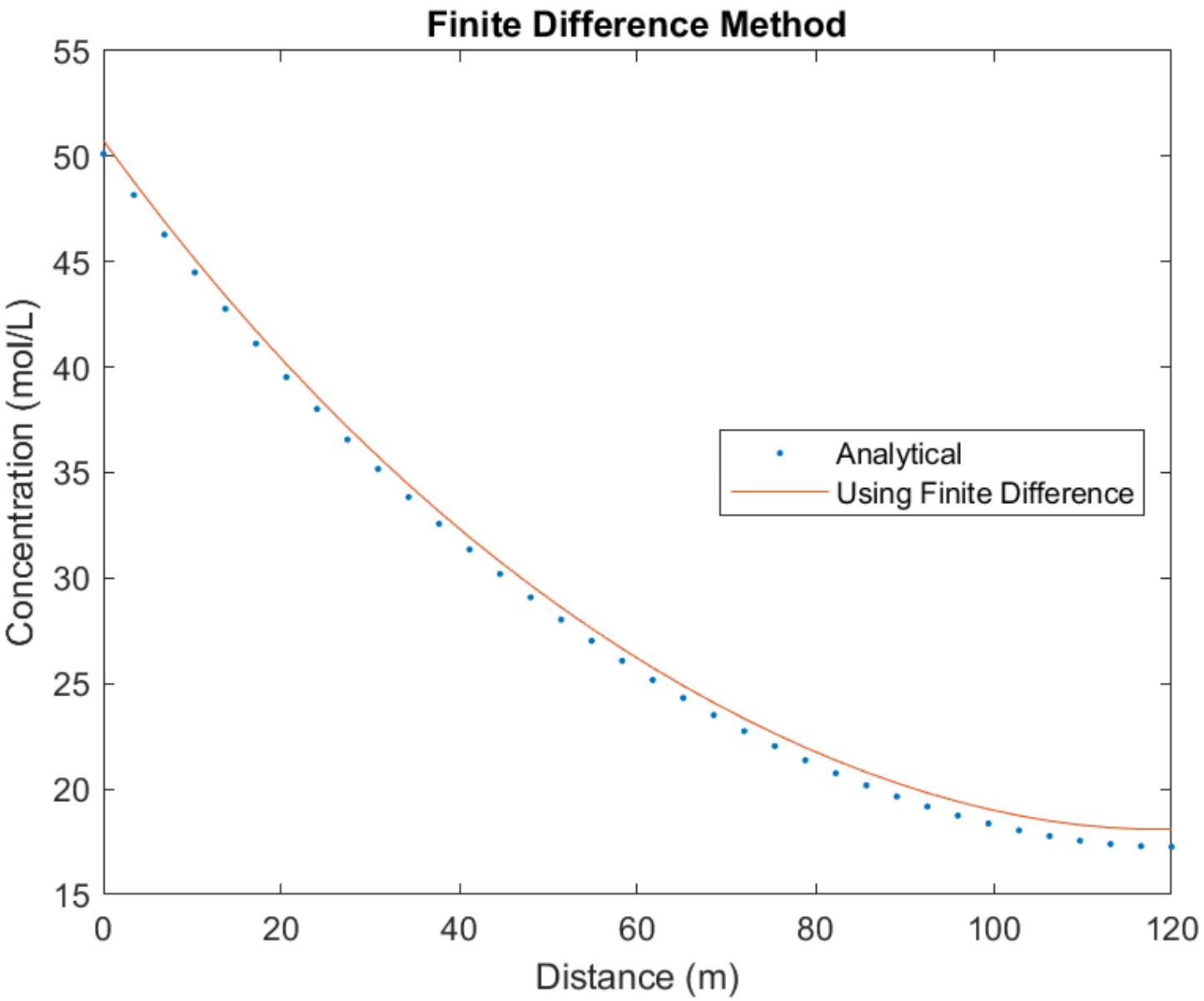
```
clc
clear all
close all

k = 1.5;    %reaction rate in hr^-1
L = 120;    %length of reactor in m
D = 4500;    %dispersion coefficient in m^2/hr
v = 75; %velocity in m/hr
Cin = 85;   %inflow concentration in mol/L

n = 7;   %intial number of nodes
delx = 20;  %intial distance between nodes in m

a1 = D/(delx^2)+v/delx;
a2 = -2*D/(delx^2)-v/delx-k;
a3 = D/(delx^2);

b1 = v+D/delx;
b2 = -D/delx;

err = 100;

while err > 5   %tolerance is 5% for maximum error

    delx = L/(n-1); % m

    a1 = D/(delx^2)+v/delx;
    a2 = -2*D/(delx^2)-v/delx-k;
    a3 = D/(delx^2);

    b1 = v+D/delx;
    b2 = -D/delx;

    b = zeros(n,1);
    b(1) = v*Cin;
    A = diag(ones(length(b),1)*a2)+diag(ones(length(b)-1,1)*a3,1)+diag...
        (ones(length(b)-1,1)*a1,-1);
    A1b = [b1 b2];
    A(1,:) = [A1b zeros(1,n-2)];
    A11 = [-1 1];
    A(length(b),:)=[zeros(1,n-2) A11];
    finC = A\b;


    l1 = (v/(2*D))*(1-sqrt(1+(4*k*D)/(v^2)));
    l2 = (v/(2*D))*(1+sqrt(1+(4*k*D)/(v^2)));

    x = linspace(0,120,n);
    C = ((v*Cin)/((v-D*l1)*l2*exp(l2*L)-(v-D*l2)*l1*exp(l1*L)))*(l2*exp...
    (l2*L)*exp(l1.*x)-l1*exp(l1*L)*exp(l2.*x));
    C = C';


    ercol= abs((C-finC)./C)*100;    %error at each node
    err = max(abs(ercol));  %maximum error
    n = n + 1;

end
```

```
fprintf('For a tolerance of 5%%, %d nodes are required.\n',n-1);

VarNames = {'x','C (mol/L) using Finite Difference','C (mol.L)','Error'};
T = table(x',finC,C,ercol,'VariableNames',VarNames);
disp(T)

plot(x,C,'.')
hold on
plot(x,finC)
title(sprintf('Finite Difference Method'))
xlabel('Distance (m)')
ylabel('Concentration (mol/L)')
legend('Analytical','Using Finite Difference','location','east')
```

Results

```
For a tolerance of 5%, 36 nodes are required.
```

x	C (mol/L) using Finite Difference	C (mol.L)	Error
0	50.746	50.114	1.2614
3.4286	48.789	48.161	1.303
6.8571	46.911	46.288	1.3455
10.286	45.109	44.491	1.3892
13.714	43.381	42.768	1.434
17.143	41.725	41.116	1.4802
20.571	40.137	39.533	1.5279
24	38.616	38.017	1.5773
27.429	37.16	36.564	1.6285
30.857	35.765	35.174	1.6819
34.286	34.431	33.843	1.7376
37.714	33.156	32.571	1.7959
41.143	31.938	31.356	1.857
44.571	30.775	30.195	1.9214
48	29.667	29.088	1.9894
51.429	28.612	28.034	2.0612
54.857	27.608	27.03	2.1373
58.286	26.655	26.077	2.2181
61.714	25.752	25.172	2.3041
65.143	24.898	24.316	2.3957
68.571	24.094	23.507	2.4933
72	23.337	22.746	2.5975
75.429	22.629	22.032	2.7087
78.857	21.969	21.365	2.8273
82.286	21.357	20.745	2.9538
85.714	20.795	20.172	3.0884
89.143	20.281	19.646	3.2315
92.571	19.818	19.169	3.3832
96	19.405	18.741	3.5434
99.429	19.046	18.364	3.7121
102.86	18.74	18.038	3.8887
106.29	18.49	17.767	4.0727
109.71	18.299	17.55	4.2632
113.14	18.168	17.392	4.4589
116.57	18.101	17.295	4.6583
120	18.101	17.262	4.8595


Finite Difference Method
Concentration (mol/L)
Distance (m)
Analytical
Using Finite Difference
55
50
45
40
35
30
25
20
15
0
20
40
60
80
100
120