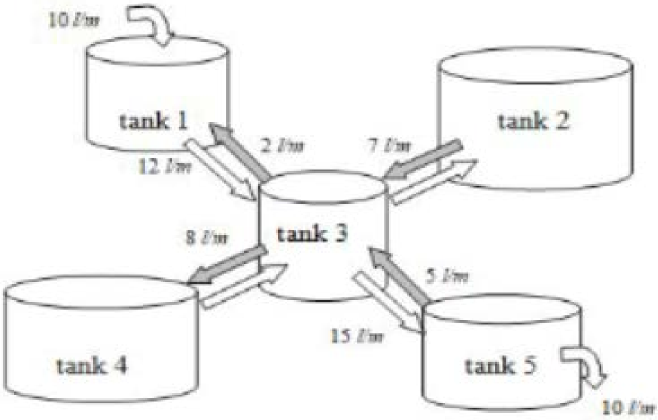
Mathematical modeling Project #1-b

2016039034 박준형
2018021794 염태은

1. Goal

1) Set up linear systems of differential equations x'=Ax+f
2) Find the concentration of salt in each tank at time = 5,10,15,20 (min)
3) After several minutes have elapsed, which tank has the higher concentraion of salt
4) Find its limiting concentration

2. Problem and setting



$V_1 = 500, V_2 = 1000, V_3 = 400, V_4 = 700, V_5 = 500$

The liquid inside each tank is kept well stirred. A brine solution that has a concentration of 5kg/l of salt flows into tank 1 at a rate of 10l/min. The flow rate of liquid from one tank to another tank and the concentration of salt flows in/out are given in figure.

$$\frac{dx_1}{dt} = \frac{x_3}{400}*2 - \frac{x_1}{500}*12 + 10*5$$

$$\frac{dx_2}{dt} = \frac{x_3}{400}*7 - \frac{x_2}{1000}*7$$

$$\frac{dx_3}{dt} = \frac{x_1}{500}*12 + \frac{x_2}{1000}*7 + \frac{x_4}{700}*8 + \frac{x_5}{500}*5 - \frac{x_3}{400}*32$$

$$\frac{dx_4}{dt} = \frac{x_3}{400}*8 - \frac{x_4}{700}*8$$

$$\frac{dx_5}{dt} = \frac{x_3}{400}*15 - \frac{x_5}{500}*15$$

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ x'_4 \\ x'_5 \end{bmatrix} = \begin{bmatrix} \frac{-12}{500} & 0 & \frac{1}{200} & 0 & 0 \\ 0 & \frac{-7}{1000} & \frac{7}{400} & 0 & 0 \\ \frac{12}{500} & \frac{7}{1000} & \frac{-32}{400} & \frac{8}{700} & \frac{5}{500} \\ 0 & 0 & \frac{8}{400} & \frac{-8}{700} & 0 \\ 0 & 0 & \frac{15}{400} & 0 & \frac{-15}{500} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} + \begin{bmatrix} 50 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

$$A = \begin{bmatrix} \frac{-12}{500} & 0 & \frac{1}{200} & 0 & 0 \\ 0 & \frac{-7}{1000} & \frac{7}{400} & 0 & 0 \\ \frac{12}{500} & \frac{7}{1000} & \frac{-32}{400} & \frac{8}{700} & \frac{5}{500} \\ 0 & 0 & \frac{8}{400} & \frac{-8}{700} & 0 \\ 0 & 0 & \frac{15}{400} & 0 & \frac{-15}{500} \end{bmatrix}. \qquad x(0) = \begin{bmatrix} 10 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \qquad f = \begin{bmatrix} 50 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Now, let's see the solution of Linear system of differential equations form

$$X' = Ax + f$$

Suppose that $f = Ay$, then

$$\begin{aligned} X' &= Ax + f \\ &= Ax + Ay \\ &= A(x + y) \end{aligned}$$

Let eigen-values be λ_i and eigenvector be v_i where $i = (1,2,\cdots,n)$.

Then, we obtain the the solution of Linear system of differential equations form of $X' = Ax + f$ as

$$\begin{aligned} x + y &= c_1 V_1 e^{\lambda_1 t} + c_2 V_2 e^{\lambda_2 t} + \cdots + c_n V_n e^{\lambda_n t} \\ x &= c_1 V_1 e^{\lambda_1 t} + c_2 V_2 e^{\lambda_2 t} + \cdots + c_n V_n e^{\lambda_n t} - y \end{aligned}$$

Then, by using initial value, $x(0)$, we can find c_i where $i = (1,2,\cdots,n)$,

since $f = Ay$ and $y = A\backslash bf$.

We will complete the differential equation by finding the eigen values and eigen vectors.

Use the following methods and find the values of c_i by using the initial value $x(0)$.

3. Methods to find eigenvalue and eigenvector

1) Power Method

Power Method is an iterative technique used to determine the dominant eigenvalue of matrix.

$$
\begin{aligned}
y_{i+1} &= Ax_i \\
x_{i+1} &= y_{i+1}/||y_{i+1}||_2 && \text{(approximate eigenvector)} \\
\mu_{i+1} &= x_{i+1}^H A x_{i+1} && \text{(approximate eigenvalue)} \\
i &= i+1
\end{aligned}
$$

2) Inverse Power Method

Inverse Power Method is an iterative technique used to determine the smallest absolute eigenvalue of a matrix.

$$
\begin{aligned}
y_{i+1} &= A^{-1}x_i \\
x_{i+1} &= y_{i+1}/\| y_{i+1} \|_2 && \text{(approximate eigenvector)} \\
u_{i+1} &= x_{i+1}^T A x_{i+1} && \text{(approximate eigenvalue)} \\
i &= i+1
\end{aligned}
$$

3) Shifted Inverse Power Method

Shifted Inverse power method is a modification of the Power method that gives a faster convergence. it is used to determine eigenvalue of A that is closest to a specified number s(shifting number). To find s, we used Gershgorin circle theorem that all eigen values in Gershgorin's circle.

This circle has center λ_i and radius $\sum_{j=1}^{n} |a_{ij}| - |\lambda_i|$

$$
\begin{aligned}
y_{i+1} &= (A - \sigma I)^{-1} x_i \\
x_{i+1} &= y_{i+1}/||y_{i+1}||_2 && \text{(approximate eigenvector)} \\
\mu_{i+1} &= x_{i+1}^H A x_{i+1} && \text{(approximate eigenvalue)} \\
i &= i+1
\end{aligned}
$$

4) QR Iteration

For orthogonal matrix Q,

$$A_{k+1} = R_k Q_k = Q_k^{-1} Q_k R_k Q_k = Q_k^{-1} A_k Q_k = Q_k^{\mathsf{T}} A_k Q_k.$$

Then A and A(k+1) is similar and have same eigenvalue.

$$\begin{aligned} &A_i = Q_i R_i \qquad \text{(QR decomposition)} \\ &A_{i+1} = R_i Q_i \\ &i = i+1 \end{aligned}$$

5) Shifted QR Iteration

Choose a shift s, it is faster than QR iteration.

$$\begin{aligned} &\text{Choose a shift } \sigma_i \\ &A_i - \sigma_i I = Q_i R_i \qquad \text{(QR decomposition)} \\ &A_{i+1} = R_i Q_i + \sigma_i \\ &i = i+1 \end{aligned}$$

If σ_i is an exact eigenvalue of A, then it can be shown that

$$A_{i+1} = R_i Q_i + \sigma_i = \begin{bmatrix} A' & a \\ 0 & \sigma_i \end{bmatrix}.$$

This means that the algorithm converges in one iteration. If more eigenvalues are wanted, we can apply the algorithm again for the $n-1$ by $n-1$ matrix A'.

6) Hessenberg Shifted QR Iteration

Algorithm:
Compute Q such that $Q^T A Q = H$ (Hessenberg reduction)
Let $H_0 = H$
$i = 0$
repeat
  Choose a shift σ_i
  $H_i - \sigma_i I = Q_i R_i$ (QR decomposition)
  $H_{i+1} = R_i Q_i + \sigma_i$
  $i = i+1$
until convergence

4. eigenvalues and eigenvectors

1) Power method

step	1	2	...	7	8
eigen value λ_1	-0.0665	-0.0956	...	-0.0921	-0.0921
eigen vector v_1	-0.7071	-0.1063	...	-0.0607	0.0606
	0	-0.1699	...	-0.1695	0.1694
	0.7071	0.8520	...	0.8239	-0.8237
	0	-0.2019	...	-0.2043	0.2043
	0	-0.4395	...	-0.497	0.4974

2) Inverse Power Method

step	1	2	...	12	13
eigen value λ_5	-0.0023	-0.0027	...	-0.0030	-0.0030
eigen vector v_5	-0.341	0.0831	...	-0.0454	-0.0454
	-0.682	0.794	...	-0.8292	-0.8292
	-0.2728	0.2094	...	-0.1908	-0.1908
	-0.4774	0.4825	...	-0.4514	-0.4514
	-0.341	0.2934	...	-0.2648	-0.2648

3) Shifted Inverse Power Method

$\boldsymbol{\sigma = -0.03}$

step	1	2	...	8	9
eigen value λ_2	-0.0291	-0.0274	...	-0.0268	-0.0268
eigen vector v_2	0.3846	0.2219	...	0.1491	0.149
	0	0.0491	...	0.0736	0.0737
	0	-0.0646	...	-0.0833	-0.0834
	0	0.0695	...	0.1084	0.1084
	-0.9231	-0.9692	...	-0.9766	-0.9766

$\boldsymbol{\sigma = -0.024}$

step	1	2	...	99	100
eigen value λ_3	-0.0239	-0.0223	...	-0.0214	-0.0214
eigen vector v_3	0.1183	0.8223	...	0.3566	0.357
	-0.1547	-0.2702	...	-0.2244	-0.2245
	0.1503	0.191	...	0.1848	0.1848
	-0.2391	-0.4575	...	-0.3702	-0.3704
	0.9393	-0.0703	...	0.807	0.8067

$\boldsymbol{\sigma = -0.007}$

step	1	2	...	13	14
eigen value λ_4	-0.0083	-0.0080	...	-0.0092	-0.0092
eigen vector v_4	-0.28	0.1163	...	-0.0277	0.0277
	0.96	-0.2566	...	0.6638	-0.6632
	0	0.1956	...	-0.0821	0.0823
	0	0.8835	...	-0.728	0.7285
	0	0.319	...	-0.1479	0.1482

4) QR & QR with Shift

Result of eigen values are same with above.

- Number of iteration of QR : 14
- Number of iteration of Shifted QR : 18
- Number of iteration of Hessenberg QR : 4

5. Solution of LDE (Result)

eigen value	λ_1	λ_2	λ_3	λ_4	λ_5
	-0.0921	-0.0268	-0.0214	-0.003	-0.0092
eigen vector	v_1	v_2	v_3	v_4	v_5
	0.0606	0.149	0.357	-0.0454	0.0277
	0.1694	0.0737	-0.2245	-0.8292	-0.6632
	-0.8237	-0.0834	0.1848	-0.1908	0.0823
	0.2043	0.1084	-0.3704	-0.4514	0.7285
	0.4974	-0.9766	0.8067	-0.2648	0.1482

[figure 1]

Let $x_i\,(t)$ denote the amount of salt in tank i at time t. Since we already find the eigen values $(\lambda_1, \cdots, \lambda_5)$ and eigen vectors $(\mathbb{v}_1, \cdots, \mathbb{v}_5)$ by above methods, we can use the general solution,

$$\mathrm{X}(t) = c_1\mathbb{v}_1 e^{\lambda_1 t} + c_2\mathbb{v}_2 e^{\lambda_2 t} + c_3\mathbb{v}_3 e^{\lambda_3 t} + c_4\mathbb{v}_4 e^{\lambda_4 t} + c_5\mathbb{v}_5 e^{\lambda_5 t}$$

Then, $$\mathrm{X}(t) = c_1\begin{bmatrix} 0.0606 \\ 0.1694 \\ -0.8237 \\ 0.2043 \\ 0.4974 \end{bmatrix} e^{-0.0921t} + c_2\begin{bmatrix} 0.149 \\ 0.0737 \\ -0.0834 \\ 0.1084 \\ -0.9766 \end{bmatrix} e^{-0.0268t} + c_3\begin{bmatrix} 0.357 \\ -0.2245 \\ 0.1848 \\ -0.3704 \\ 0.8067 \end{bmatrix} e^{-0.0214t} + c_4\begin{bmatrix} -0.0454 \\ -0.8292 \\ -0.1908 \\ -0.4514 \\ -0.2648 \end{bmatrix} e^{-0.003t} + c_5\begin{bmatrix} 0.0277 \\ -0.6632 \\ 0.0823 \\ 0.7285 \\ 0.1482 \end{bmatrix} e^{-0.0092t} - \begin{bmatrix} -2500 \\ -5000 \\ -2000 \\ -3500 \\ -2500 \end{bmatrix}$$

And $$\mathrm{X}(0) = \begin{bmatrix} 0.0606 & 0.149 & 0.357 & -0.0454 & 0.0277 \\ 0.1694 & 0.0737 & -0.2245 & -0.8292 & -0.6632 \\ -0.8237 & -0.0834 & 0.1848 & -0.1908 & 0.0823 \\ 0.2043 & 0.1084 & -0.3704 & -0.4514 & 0.7285 \\ 0.4974 & -0.9766 & 0.8067 & -0.2648 & 0.1482 \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} - \begin{bmatrix} -2500 \\ -5000 \\ -2000 \\ -3500 \\ -2500 \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Therefore, we can find $c_1, \cdots, c_5$ by solving

$$\begin{bmatrix} 0.0606 & 0.149 & 0.357 & -0.0454 & 0.0277 \\ 0.1694 & 0.0737 & -0.2245 & -0.8292 & -0.6632 \\ -0.8237 & -0.0834 & 0.1848 & -0.1908 & 0.0823 \\ 0.2043 & 0.1084 & -0.3704 & -0.4514 & 0.7285 \\ 0.4974 & -0.9766 & 0.8067 & -0.2648 & 0.1482 \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} -2490 \\ -5000 \\ -2000 \\ -3500 \\ -2500 \end{bmatrix}$$

Then, $\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} -193.3 \\ -3519.9 \\ -4339.3 \\ 8021.8 \\ -1462.1 \end{bmatrix}$.

So, we can construct formulas of amount as following,

$\mathrm{x}_1(t) = (-193.3)(0.0606)e^{-0.0921t} + (-3519.9)(0.149)e^{-0.0268t} + (-4339.3)(0.357)e^{-0.0214t} + (8021.8)(-0.0454)e^{-0.003t} + (-1462.1)(0.0277)e^{-0.0092t} + 2500$

$\mathrm{x}_2(t) = (-193.3)(0.1694)e^{-0.0921t} + (-3519.9)(0.0737)e^{-0.0268t} + (-4339.3)(-0.2245)e^{-0.0214t} + (8021.8)(-0.8292)e^{-0.003t} + (-1462.1)(-0.6632)e^{-0.0092t} + 5000$

$\mathrm{x}_3(t) = (-193.3)(-0.8237)e^{-0.0921t} + (-3519.9)(-0.0834)e^{-0.0268t} + (-4339.3)(0.1848)e^{-0.0214t} + (8021.8)(-0.1908)e^{-0.003t} + (-1462.1)(0.0823)e^{-0.0092t} + 2000$

$\mathrm{x}_4(t) = (-193.3)(0.2043)e^{-0.0921t} + (-3519.9)(0.1084)e^{-0.0268t} + (-4339.3)(-0.3704)e^{-0.0214t} + (8021.8)(-0.4514)e^{-0.003t} + (-1462.1)(0.7285)e^{-0.0092t} + 3500$

$\mathrm{x}_5(t) = (-193.3)(0.4974)e^{-0.0921t} + (-3519.9)(-0.9766)e^{-0.0268t} + (-4339.3)(0.8067)e^{-0.0214t} + (8021.8)(-0.2648)e^{-0.003t} + (-1462.1)(0.1482)e^{-0.0092t} - 2500$

6. Concentration of salt in each tank

We have to find the concentration of salt in each tank at time $t = 5, 10, 15, 20\ (min)$ for question (1).

By substituting numbers for each t, we can find the amount of salt in each tank at time $t = 5, 10, 15, 20\ (min)$.

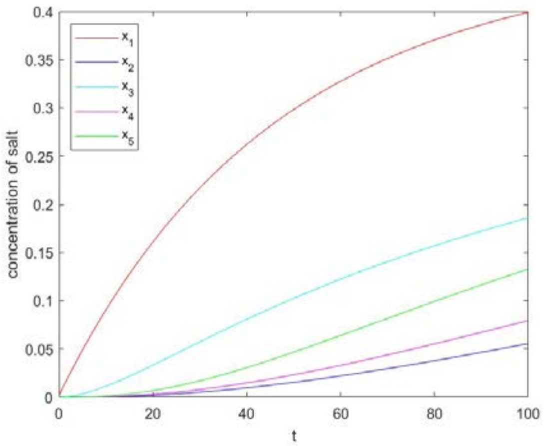
t	5	10	15	20	25
x_1	244.5361	453.1141	638.9808	804.8716	953.1234
x_2	1.2009	4.3785	10.6064	20.3933	33.905
x_3	13.9896	45.4425	86.6308	132.7669	180.9129
x_4	1.1276	4.4701	11.1942	21.8059	36.4151
x_5	0.9631	5.9416	16.6721	33.5862	56.3993

[figure 2]

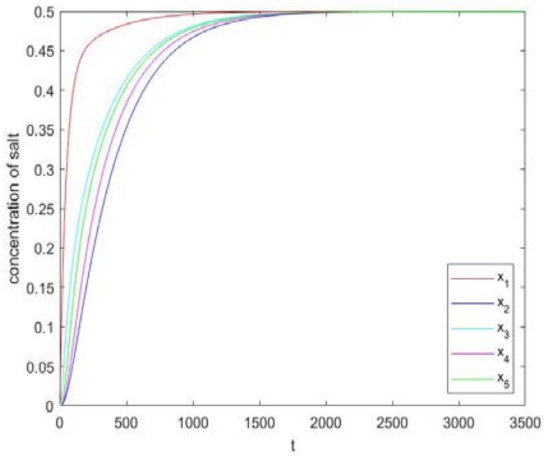
Then the concentration of salt in each tank at time $t = 5, 10, 15, 20\ (min)$ is

t	5	10	15	20	25
x_1	0.048907	0.090623	0.127796	0.160974	0.190625
x_2	0.00012	0.000438	0.001061	0.002039	0.003391
x_3	0.003497	0.011361	0.021658	0.033192	0.045228
x_4	0.000161	0.000639	0.001599	0.003115	0.005202
x_5	0.000193	0.001188	0.003334	0.006717	0.01128

[figure 3]



[figure 4]



[figure 5]

[figure 4] is the concentration graph for t=0:100. We can know that [Tank 1] has the highest concentration of salt after several minutes.

[figure 5] is the concentration graph for t=0:3500. As we can see, the concentration of every tanks are converge to 0.5. For detail, refer to following table.

t	1500	2000	2500	3000	3500
x_1	0.4992	0.4998	0.5	0.5	0.5
x_2	0.4926	0.4984	0.4996	0.4999	0.5
x_3	0.4957	0.4991	0.4998	0.5	0.5
x_4	0.4943	0.4987	0.4997	0.4999	0.5
x_5	0.4953	0.4989	0.4998	0.4999	0.5

[figure 6]

Hence, limiting concentration of salt is $0.5\ \mathrm{g}/\ell$
We can conclude that as time goes by, the concentrations of the brine in the tanks are same with the brine's concentration that was flowing in. It means all tanks are full of brine that flows into tank1

7. Matlab Code

1) Power Method

```
A=[-12/500 0 1/200 0 0 ; 0 -0.007 7/400 0 0 ; 12/500 0.007 -32/400 8/700 0.01 ; 0 0 0.02 -8/700 0 ; 0 0 15/400 0 -0.03]
u=[10 0 0 0 0 ]'
epsilon=0.0001
m1=1;
v=A*u;
m2=norm(v);
err=abs(m1-m2);
 while err>epsilon
   v=A*u;
   m2=norm(v);
   u=v/m2
   lambda = u'*A*u
   err=abs(m1-m2);
   m1=m2;

 end
fprintf('\n The greatest eigenvalue is %2.4f \n',lambda);
disp(' The corresponding eigenvector is:');
disp(u);
```

2) Inverse Power Method

```
A=[-12/500 0 1/200 0 0 ; 0 -0.007 7/400 0 0 ; 12/500 0.007 -32/400 8/700 0.01 ; 0 0 0.02 -8/700 0 ; 0 0 15/400 0 -0.03];
u=[10 0 0 0 0 ]';
epsilon=0.001;
m1=1;
v=A\u;
m2=norm(v);
err=abs(m1-m2);
 while err>epsilon
   v=A\u;
   m2=norm(v);
   u=v/m2
   lambda = u'*A*u
   err=abs(m1-m2);
   m1=m2;

 end
fprintf('\n The greatest eigenvalue is %2.4f \n',lambda);
disp(' The corresponding eigenvector is:');
disp(u);
```

3) Shifted Inverse Power Method

```
A = [-12/500 0 1/200 0 0 ; 0 -0.007 7/400 0 0 ; 12/500 0.007 -32/400 8/700 0.01 ; 0 0 0.02 -8/700 0 ; 0 0 15/400 0 -0.03];
s = -0.007;
q = [ 10 0 0 0 0 ]';
tol = 0.0001;

relerr = 0;
ll = s;

I = eye(size(A));

for i=1:100
   z = (A-s*I)\q;
   q = z/norm(z,2)
   lambda = q'*A*q
   if (abs( lambda - ll )/abs(ll))<tol
       break
   end
   ll = lambda;

end
fprintf('\n The greatest eigenvalue is %2.4f \n',lambda);
disp(' The corresponding eigenvector is:');
disp(q)
```

4) QR iteration

```
function A = qrstep0(A)
epsilon = 0.0001;
[Q,R] = qr(A);
[m,n] = size(A);
i=0;
for j=1:n-1
while max(abs(diag(A,-n+j)))>epsilon
[m,n]=size(A);
rho = 0;
[Q,R]=qr(A-rho*eye(n,n));
A = R*Q + rho*eye(n,n);
i=i+1
end
end
```

5) Shifted QR iteration

```
function A = qrstep(A)
epsilon = 0.0001;
[Q,R] = qr(A);
[m,n] = size(A);
i=0;
for j=1:n-1
while max(abs(diag(A,-n+j)))>epsilon
[m,n]=size(A);
rho = A(n,n);
[Q,R]=qr(A-rho*eye(n,n));
A = R*Q + rho*eye(n,n);
i=i+1
end
end
```

6) Hessenberg Shifted QR iteration

```
function H = myhessen2(A)

[Q,R] = qr(A);
[m,n] = size(A);
H = transpose(Q)*A*Q;
epsilon = 0.0001;
i=0
for j=1:n-2
 while max(abs(diag(H,-n+j)))>epsilon
[m,n]=size(H);
rho = H(n,n);
[Q,R]=qr(H-rho*eye(n,n));
H = R*Q + rho*eye(n,n);
i=i+1
 end
end
```

7) Find C ($c_1, \cdots, c_5$)

```
A = [-12/500 0 1/200 0 0 ; 0 -0.007 7/400 0 0 ; 12/500 0.007 -32/400 8/700 0.01 ; 0 0 0.02 -8/700 0 ; 0 0 15/400 0 -0.03];

%eigen vectors
f_c = [0.0606   0.149   0.357   -0.0454 0.0277
;0.1694 0.0737  -0.2245 -0.8292 -0.6632
;-0.8237    -0.0834 0.1848  -0.1908 0.0823
;0.2043 0.1084  -0.3704 -0.4514 0.7285
;0.4974 -0.9766 0.8067  -0.2648 0.1482
];
b = [50 0 0 0 0]';
k = [-2490 -5000 -2000 -3500 -2500]';
y = A\b;
C = f_c\k;
e_v = [-0.0921  -0.0268 -0.0214 -0.003  -0.0092];
```

8) Drawing graph [figure 4&5]

```
tt=0:3000;
i=1;
x1_t = @(t) ((f_c(i,1)*C(1)*exp(e_v(1)*t))+(f_c(i,2)*C(2)*exp(e_v(2)*t))+(f_c(i,3)*C(3)*exp(e_v(3)*t))+(f_c(i,4)*C(4)*exp(e_v(4)*t))+(f_c(i,5)*C(5)*exp(e_v(5)*t))-y(i));
plot(tt,x1_t(tt),'r');
hold on;
i=2;
x2_t = @(t) ((f_c(i,1)*C(1)*exp(e_v(1)*t))+(f_c(i,2)*C(2)*exp(e_v(2)*t))+(f_c(i,3)*C(3)*exp(e_v(3)*t))+(f_c(i,4)*C(4)*exp(e_v(4)*t))+(f_c(i,5)*C(5)*exp(e_v(5)*t))-y(i));
plot(tt,x2_t(tt),'b');
hold on;
i=3;
x3_t = @(t) ((f_c(i,1)*C(1)*exp(e_v(1)*t))+(f_c(i,2)*C(2)*exp(e_v(2)*t))+(f_c(i,3)*C(3)*exp(e_v(3)*t))+(f_c(i,4)*C(4)*exp(e_v(4)*t))+(f_c(i,5)*C(5)*exp(e_v(5)*t))-y(i));
plot(tt,x3_t(tt),'c');
hold on;
i=4;
x4_t = @(t) ((f_c(i,1)*C(1)*exp(e_v(1)*t))+(f_c(i,2)*C(2)*exp(e_v(2)*t))+(f_c(i,3)*C(3)*exp(e_v(3)*t))+(f_c(i,4)*C(4)*exp(e_v(4)*t))+(f_c(i,5)*C(5)*exp(e_v(5)*t))-y(i));
plot(tt,x4_t(tt),'m');
hold on;
i=5;
x5_t = @(t) ((f_c(i,1)*C(1)*exp(e_v(1)*t))+(f_c(i,2)*C(2)*exp(e_v(2)*t))+(f_c(i,3)*C(3)*exp(e_v(3)*t))+(f_c(i,4)*C(4)*exp(e_v(4)*t))+(f_c(i,5)*C(5)*exp(e_v(5)*t))-y(i));
plot(tt,x5_t(tt),'g');
legend({'x_1','x_2','x_3','x_4','x_5'},'Location','southeast')
```