MEC 104

MATLAB Assignment

Student ID: 1931254
Full name: Zhongnan Li
Email address: Zhongnan.li19@student.xjtlu.edu.cn

Matlab Assignment

Problem 1

P1-1

1. Equation derivations

1.1 The equation used in the coding

There is no equations used in the coding, since all the questions in P1-1 is the simple matrix representation.

1.2 Coefficients and source terms

The condition of this question is the matrix A, which equals to [3 7 -4 12;-5 9 10 2;6 13 8 11;15 5 4 1]

2. Computational conditions

There is no computational conditions here.

3. Main programme

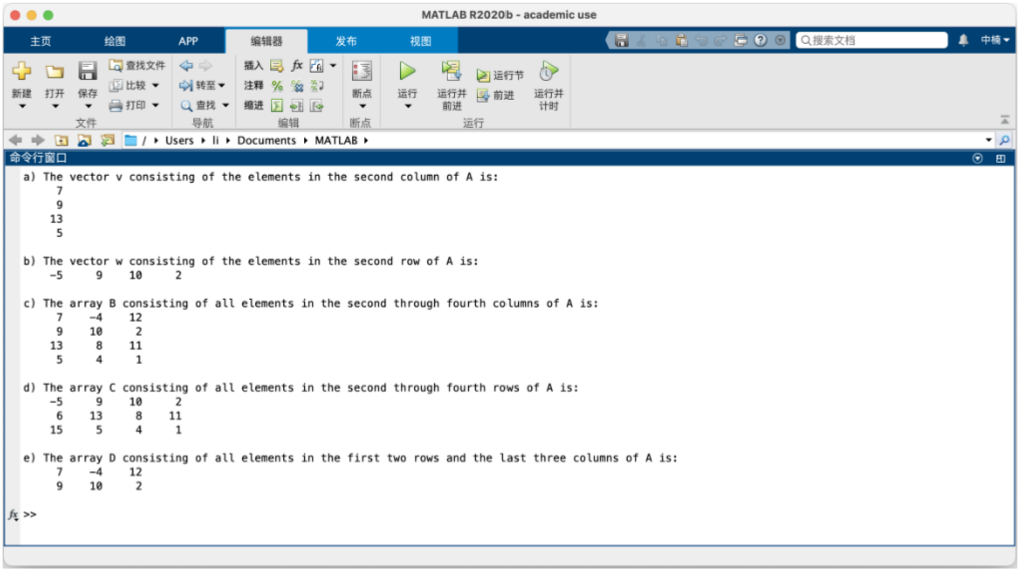
```
%Problem 1 Matrix Operation
%P1-1
clear all
close all
clc

A=[3 7 -4 12;-5 9 10 2;6 13 8 11;15 5 4 1];
%(a)
disp('a) The vector v consisting of the elements in the second column of A is:');
v=A(:,2);
disp(v);
%(b)
disp('b) The vector w consisting of the elements in the second row of A is:');
w=A(2,:);
disp(w);
%(c)
disp('c) The array B consisting of all elements in the second through fourth columns of A is:');
B=A(:,2:4);
disp(B);
%(d)
disp('d) The array C consisting of all elements in the second through fourth rows of A is:');
C=A(2:4,:);
disp(C);
%(e)
disp('e) The array D consisting of all elements in the first two rows and the last three columns of A is:');
```

```
D=A(1:2,2:4);
disp(D);
```

4. Functions

There is no need for functions in this question.

5. Results



6. Flow charts

There is no need for flow charts in this question.

P1-2

1. Equation derivations

1.1 The equation used in the coding

There is no equations used in the coding, since all the questions in P1-2 is the simple matrix representation.

1.2 Coefficients and source terms

The source terms of this question are the matrix A and B.

A=[1 4 2;2 4 100;7 9 7;3 pi 42] and B=log(A)

2. Computational conditions

There is no computational conditions here.

3. Main programme

```
%Problem 1 Matrix Operation
```

```
%P1-2
clear all
close all
clc

A=[1 4 2;2 4 100;7 9 7;3 pi 42];
B=log(A);
%(a)
disp('a) The second row of B is: ')
a=B(2,:);
disp(a);
%(b)
disp('b) The sum of the second row of B is: ');
disp(sum(a));
%(c)
disp('c) The product of the second column of B multiply the first column of A element by element is: ');
disp(B(:,2).*A(:,1));
%(d)
disp('d) The maximum value in the vector resulting from question c) is: ');
disp(max(B(:,2).*A(:,1)));
%(e)
disp('e) the sum of the elements of the resulting vector is:');
disp(sum(sum(A(1,:)./B(1:3,3))));
```

4. Functions

4.1 Explanations

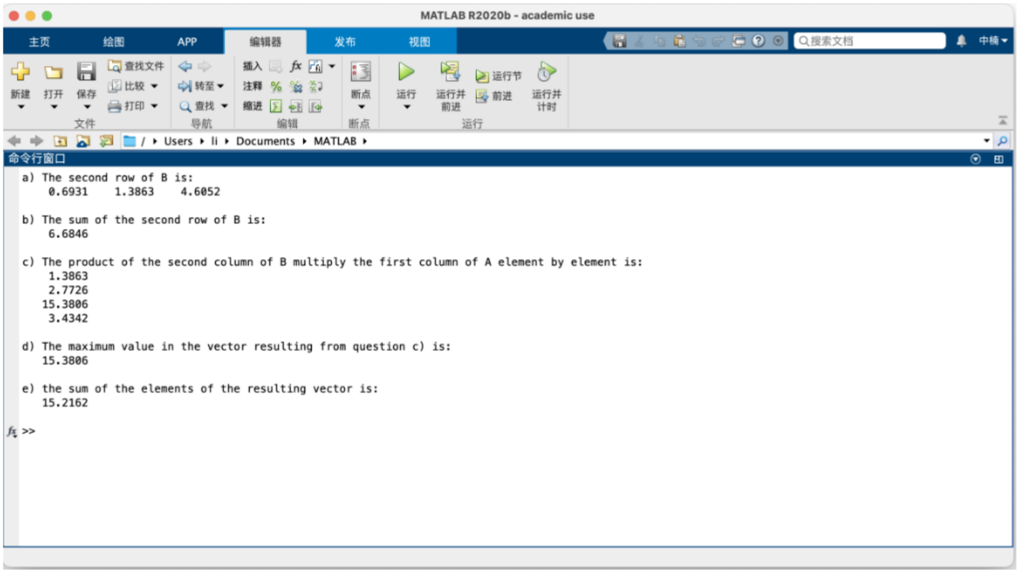
The function sum is used to find the sums of number in brackets, and the function max is to find the maximum value in the numbers in brackets.

4.2 Codes

```
disp(sum(a));
disp(max(B(:,2).*A(:,1)));
disp(sum(sum(A(1,:)./B(1:3,3))));
```

5. Results

5.1 Results figure



5.2 Comments and Analysis

For the question (e), the requirement is evaluate the sum of the elements of the resulting vector. Therefore, we need to use "sum" twice to get a sum number. If only one "sum" used, the result will be a vector.

6. Flow charts

There is no need for flow charts in this question.

P1-3

1. Equation derivations

1.1 The equation used in the coding

$$\frac{1}{\sqrt{34}}T_1 - \frac{3}{\sqrt{34}}T_2 + \frac{1}{\sqrt{42}}T_3 = 0$$

$$\frac{3}{\sqrt{35}}T_1 + 0T_2 + \frac{4}{\sqrt{42}}T_3 = 0$$

$$\frac{5}{\sqrt{35}}T_1 + \frac{5}{\sqrt{34}}T_2 + \frac{5}{\sqrt{42}}T_3 = mg = 1$$

1.2 Coefficients and source terms

Here we first assume mg as 1 and when we get the values of T1, T2, T3, we multiply mg to the values.

2. Computational conditions

There is no computational conditions here.

3. Main programme

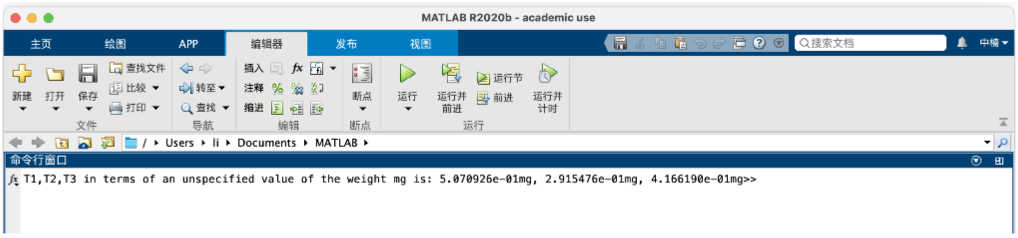
```
%Problem 1 Matrix Operation
%P1-3
clear all
close all
clc

A=[1/sqrt(35) -3/sqrt(34) 1/sqrt(42);3/sqrt(35) 0 -4/sqrt(42);5/sqrt(35) 5/sqrt(34) 5/sqrt(42)];
B=[0;0;1];
T=A\B;
T1=T(1);
T2=T(2);
T3=T(3);
fprintf("T1,T2,T3 in terms of an unspecified value of the weight mg is: %dmg, %dmg, %dmg",T1,T2,T3);
```

4. Functions

There is no need for functions in this question.

5. Results

5.1 Results figure



5.2 Comments and Analysis

The left division method is used here since the matrix A is a square matrix and |A| does not equals to 0.

6. Flow charts

There is no need for flow charts in this question.

P1-4

1. Equation derivations

1.1 The equation used in the coding

$$(R_1)q + (1)T_1 + (0)T_2 + (0)T_3 = T_i$$
$$(R_2)q + (-1)T_1 + (1)T_2 + (0)T_3 = 0$$
$$(R_3)q + (0)T_1 + (-1)T_2 + (1)T_3 = 0$$
$$(R_4)q + (0)T_1 + (0)T_2 + (-1)T_3 = -T_0$$

1.2 Coefficients and source terms

$$R_1 = 0.036\ K/W, R_2 = 4.01\ K/W, R_3 = 0.408\ K/W, R_4 = 0.038\ K/W,$$

$$T_i = 20℃, T_0 =-10℃, A = 10m^2$$

2. Computational conditions

There is no computational conditions here.

3. Main programme

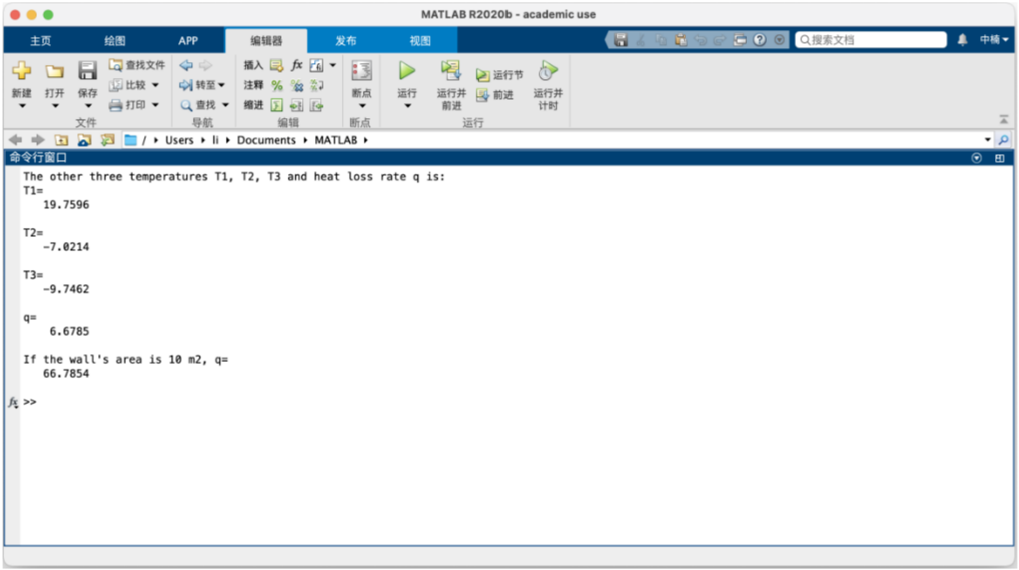
```
%Problem 1 Matrix Operation
%P1-4
clear all
close all
clc

R1=0.036;
R2=4.01;
R3=0.408;
R4=0.038;
Ti=20;
T0=-10;

A=[R1 1 0 0;R2 -1 1 0;R3 0 -1 1;R4 0 0 -1];
B=[Ti;0;0;-T0];
X=A\B;
q1=X(1);
q2=X(1)*10;
T1=X(2);
T2=X(3);
T3=X(4);
disp("The other three temperatures T1, T2, T3 and heat loss rate q is:");
disp("T1=");disp(T1);
disp("T2=");disp(T2);
disp("T3=");disp(T3);
disp("q=");disp(q1);
disp("If the wall's area is 10 m2, q=");disp(q2);
```

4. Functions

There is no need for functions in this question.

5. Results

5.1 Results figure



5.2 Comments and Analysis

In this question, we should first abstract out the mathematical relationships among five different temperature T and heat flow rate q, and then we can get four Quaternary system of first order equations. Therefore, we can built the matrix A which contain the coefficients of q, T1, T2, T3, and matrix B which contain the values in the right side of four equations.

Moreover, when the wall's area is 10 m2, the calculated heat flow rate is 10 times of the previous q.

6. Flow charts

There is no need for flow charts in this question.

Problem 2

P5-1

1. Equation derivations

1.1 The equation used in the coding

$$x_1 = \theta,$$

$$\dot{x}_1 = x_2 \qquad (1),$$

$$\dot{x}_2 = -\frac{g \sin x_1}{L} + \frac{a(t)\cos x_1}{L} \qquad (2)$$

1.2 Coefficients and source terms

$$g = 9.81\ \mathrm{m/s^2},\ L = 1\ \mathrm{m},\ \theta'(0) = 0,$$

$$(a)\ a = 5\ m/s^2, \theta(0) = 0.5\ rad$$
$$(b)\ a = 5\ m/s^2, \theta(0) = 3\ rad$$
$$(c)\ a = 0.5t\ m/s^2, \theta(0) = 3\ rad$$

2. Computational conditions

The time t ranges from 0 to 10s for three cases.

3. Main programme

```
%Problem 2
%P5-1
clear all
close all
clc

%(a)
[t,x]=ode45(@Motion1,[0 10],[0.5 0]);
subplot(3,1,1);
plot(t,x(:,1),'r-');
xlabel('time(t)')
ylabel('angle')
title('Plot of Problem 2 P5-1')

%(b)
[t,x]=ode45(@Motion2,[0 10],[3 0]);
subplot(3,1,2);
plot(t,x(:,1),'b-');
xlabel('time(t)')
ylabel('angle')

%(c)
[t,x]=ode45(@Motion3,[0 10],[3 0]);
subplot(3,1,3);
plot(t,x(:,1),'k-');
xlabel('time(t)')
ylabel('angle')

function [xdot]= Motion1(t,x)
g=9.81;
L=1;
a=5;
xdot=zeros(2,1);
xdot(1)=x(2);
xdot(2)=a*cos(x(1))/L-g*sin(x(1))/L;
```

```
end

function [xdot]= Motion2(t,x)
g=9.81;
L=1;
a=5;
xdot=zeros(2,1);
xdot(1)=x(2);
xdot(2)=a*cos(x(1))/L-g*sin(x(1))/L;
end

function [xdot]= Motion3(t,x)
g=9.81;
L=1;
a=0.5*t;
xdot=zeros(2,1);
xdot(1)=x(2);
xdot(2)=a*cos(x(1))/L-g*sin(x(1))/L;
end
```

4. Functions

4.1 Explanations

There are three functions which are Motion1, Motion2, Motion3 in this code for three different cases. Moreover, there is also an ode45 function used in the code.
These three functions are built to define the ODEs to be solved in MATLAB environment and ode45 function is used to solve the ODEs.
The ODEs defined by the following functions are:

$$\dot{x}_1 = x_2 \qquad (1),$$

$$\dot{x}_2 = -\frac{g\sin x_1}{L} + \frac{a(t)\cos x_1}{L} \qquad (2)$$

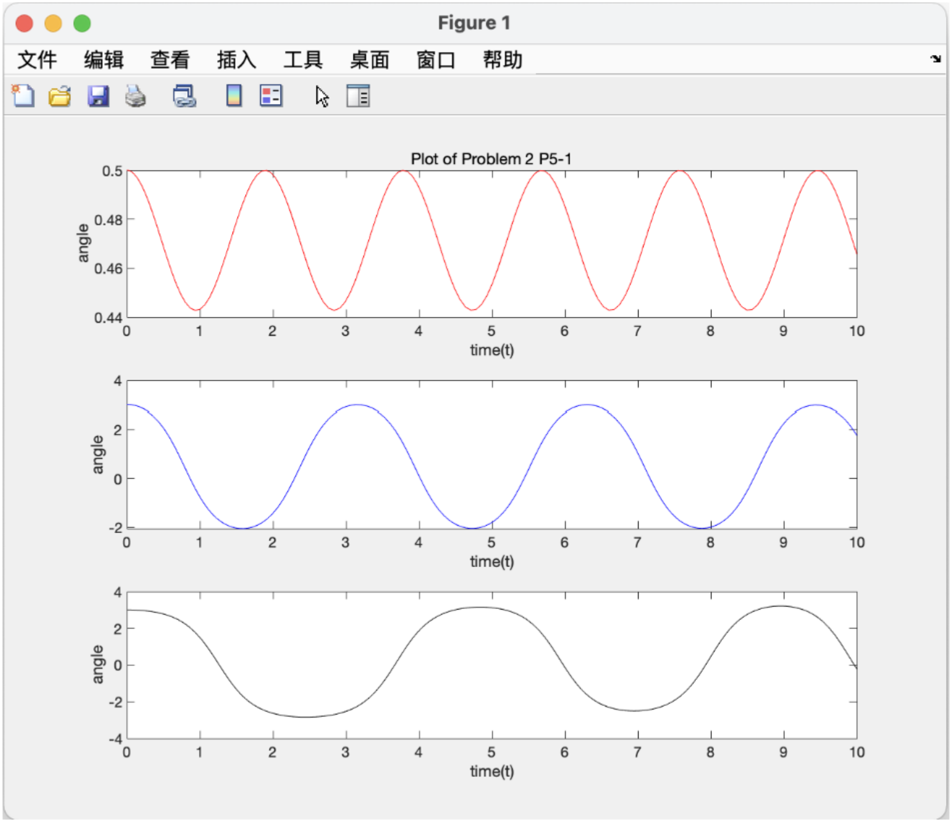
4.2 Codes

```
% Function for the case 1
function [xdot]= Motion1(t,x)
g=9.81;
L=1;
a=5;
xdot=zeros(2,1);
xdot(1)=x(2);
xdot(2)=a*cos(x(1))/L-g*sin(x(1))/L;
end
```

```matlab
% Function for the case 2
function [xdot]= Motion2(t,x)
g=9.81;
L=1;
a=5;
xdot=zeros(2,1);
xdot(1)=x(2);
xdot(2)=a*cos(x(1))/L-g*sin(x(1))/L;
end

% Function for the case 3
function [xdot]= Motion3(t,x)
g=9.81;
L=1;
a=0.5*t;
xdot=zeros(2,1);
xdot(1)=x(2);
xdot(2)=a*cos(x(1))/L-g*sin(x(1))/L;
end

% ode45 function for all three cases
[t,x]=ode45(@Motion1,[0 10],[0.5 0]);
```

5. Results

5.1 Results figure



5.2 Comments and Analysis

For this kind of ODEs Solving question, the first step is rewriting the 2nd order ODE in the question into two 1st order ODEs by letting $x1=\theta$, $x2=\dot{x}_1$, therefore we have another 1st order ODE: $xdot_2 = -\frac{g\sin x_1}{L} + \frac{a(t)\cos x_1}{L}$.

The second step is defining the ODEs to be solved in MATLAB environment, which can be done by creating function files (Motion1, Motion2, Motion3) as shown in 4.2.

The third step is using ode45 function to solve the equation.

In the end, the figures showing the angle of the motion for a pendulum as a function of time.

6. Flow charts

There is no need for flow charts in this question.

P5-2

1. Equation derivations

1.1 The equation used in the coding

$$x_1 = y,$$

$$u = f(t) = 10\sin\omega t$$

$$\dot{x}_1 = x_2 \qquad (1),$$

$$\dot{x}_2 = -\frac{75}{3}x_1 + \frac{0}{3}x_2 + \frac{1}{3}u \qquad (2)$$

1.2 Coefficients and source terms

$$y(0) = y'(0) = 0$$

(a) $\omega = 1$ rad/s

(b) $\omega = 5$ rad/s

(c) $\omega = 10$ rad/s

2. Computational conditions

The time t ranges from 0 to 20s for three cases.

3. Main programme

```
%Problem 2
%P5-2
clear all
close all
clc

%(a)
[ta,xa]= ode45(@Spring1,[0 20],[0,0]);
subplot(3,1,1)
plot(ta, xa(:,1),'r-')
ylabel('Position (m)')
title('Plot of Problem 2 P5-2')
text(0.3,-0.15,'\omega = 1.0 rad/s')

%(b)
[tb,xb]= ode45(@Spring2,[0 20],[0,0]);
subplot(3,1,2)
plot(tb,xb(:,1),'b-')
ylabel('Position (m)')
text(0.3,-5,'\omega = 5.0 rad/s')

%(c)
[tc,xc]= ode45(@Spring3,[0 20],[0,0]);
subplot(3,1,3)
plot(tc,xc(:,1),'k-')
xlabel('Time t(sec)'),ylabel('Position (m)')
text(0.3,-0.1,'\omega = 10.0 rad/s')
```

```
function [xdot]=Spring1(t,x)
omega=1;
u=10*sin(omega*t);
xdot(1)=x(2);
xdot(2)=-75*x(1)/3-0*x(2)/3+1*u/3;
A=[0 1;-75/3 -0/3];
B=[0;1/3];
xdot=A*x+B*u;
end

% ode45 function for all three cases
function [xdot]=Spring2(t,x)
omega=5;
u=10*sin(omega*t);
xdot(1)=x(2);
xdot(2)=-75*x(1)/3-0*x(2)/3+1*u/3;
A=[0 1;-75/3 -0/3];
B=[0;1/3];
xdot=A*x+B*u;
end

function [xdot]=Spring3(t,x)
omega=10;
u=10*sin(omega*t);
xdot(1)=x(2);
xdot(2)=-75*x(1)/3-0*x(2)/3+1*u/3;
A=[0 1;-75/3 -0/3];
B=[0;1/3];
xdot=A*x+B*u;
end
```

4. Functions

4.1 Explanations

There are three functions which are Spring1, Spring2, Spring3 in this code for three different cases. Moreover, there is also an ode45 function used in the code.

These three functions are built to define the ODEs to be solved in MATLAB environment and ode45 function is used to solve the ODEs.

The ODEs defined by the following functions are:

$$\dot{x}_1 = x_2 \qquad (1),$$

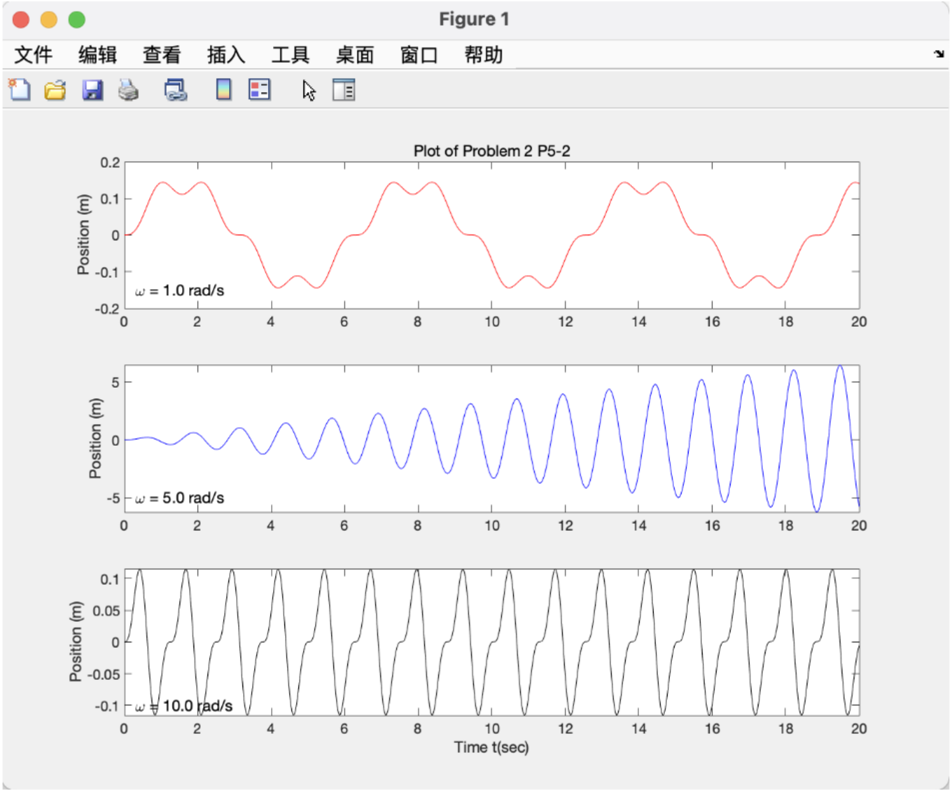
$$\dot{x}_2 = -\frac{75}{3}x_1 + \frac{0}{3}x_2 + \frac{1}{3}u \qquad (2)$$

4.2 Codes

```
% Function for the case 1
function [xdot]=Spring1(t,x)
omega=1;
u=10*sin(omega*t);
xdot(1)=x(2);
xdot(2)=-75*x(1)/3-0*x(2)/3+1*u/3;
A=[0 1;-75/3 -0/3];
B=[0;1/3];
xdot=A*x+B*u;
end

 % Function for the case 2
function [xdot]=Spring2(t,x)
omega=5;
u=10*sin(omega*t);
xdot(1)=x(2);
xdot(2)=-75*x(1)/3-0*x(2)/3+1*u/3;
A=[0 1;-75/3 -0/3];
B=[0;1/3];
xdot=A*x+B*u;
end

% Function for the case 3
function [xdot]=Spring3(t,x)
omega=10;
u=10*sin(omega*t);
xdot(1)=x(2);
xdot(2)=-75*x(1)/3-0*x(2)/3+1*u/3;
A=[0 1;-75/3 -0/3];
B=[0;1/3];
xdot=A*x+B*u;
end

[ta,xa]= ode45(@Spring1,[0 20],[0,0]);
```

5. Results

5.1 Results figure



5.2 Comments and Analysis

Since the ODE is linear, the matrix method is used in this question.

First step is still rewriting the 2nd order ODE in the question into two 1st order ODEs by letting x1=y, x2=$\dot{x}_1$ (1), $u = f(t) = 10\sin\omega t$, therefore we have another 1st order ODE:

$\dot{x}_2 = -\frac{75}{3}x_1 - \frac{0}{3}x_2 + \frac{1}{3}u$ (2).

Then the equations (1) (2) can be written in the matrix form.

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\frac{75}{3} & -\frac{0}{3}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{3}\end{bmatrix}u$$

Then $xdot = Ax + Bu$.

The next step is also defining the ODEs to be solved in MATLAB environment, which can be done by creating function files (Spring1, Spring2, Spring3) as shown in 4.2.

The third step is using ode45 function to solve the equation.

In the end, the figures showing the position of a certain mass connected to a spring as a function of time.

6. Flow charts

There is no need for flow charts in this question.

Problem 3

1. Equation derivations

1.1 The equation used in the coding

$$L\dot{i} = -Ri - K_e\omega + v(t) \quad (1),$$

$$J\dot{\omega} = K_T i - c\omega \quad (2)$$

$$v(t) = \begin{cases} 400t, & 0 \le t < 0.05 \\ 20, & 0 \le t \le 0.05 \\ -400(t-0.2)+20, & 0.2 \le t \le 0.25 \\ 0, & t > 0.25 \end{cases}$$

1.2 Coefficients and source terms

$$R = 0.8\,\Omega,$$

$$L = 0.003\,\mathrm{H},$$

$$K_T = 0.05\,\mathrm{N\cdot m/A},$$

$$K_e = 0.05\,\mathrm{V\cdot s/rad},$$

$$c = 0,$$

$$J = 8\times10^{-5}\,\mathrm{kg\cdot m^2}$$

2. Computational conditions

For both two questions, the initial conditions are $i(0) = 0$ and $\omega(0) = 0$.

3. Main programme

```
%Problem 3
clear all
close all
clc

%(a) Motor current
[ta,xa]=ode45(Ccurrent1,[0 0.16],[0,0]);
subplot(4,1,1)
plot(ta,xa(:,1))
xlabel('Time (sec)')
ylabel('Motor current(A)')
title('Plots of Problem 3')

%(a) Rotational velocity
[ta,xa]=ode45(@Current1,[0 0.16],[0,0]);
subplot(4,1,2)
plot(ta,xa(:,2))
xlabel('Time (sec)')
```

```
ylabel('Rotational velocity (rad/sec)')

%(b) Rotational velocity
[tb,xb]=ode45(@Current2,[0 0.3],[0,0]);
subplot(4,1,3)
plot(tb,xb(:,2))
xlabel('Time (sec)')
ylabel('Rotational velocity (rad/sec)')

%(b) Applied Motor voltage
t=linspace(0,0.3,1000);
for k= 1:1000
if t(k)<0.05
v(k)=400*t(k);
elseif t(k)>0.05 && t(k)<=0.2
v(k)=20;
elseif t(k)>0.2 && t(k)<=0.25
v(k)=-400*(t(k)-0.2)+ 20;
else
v(k)= 0;
end
end

subplot(4,1,4)
plot(t,v)
xlabel('Time (sec)')
ylabel('Applied Motor voltage(v)')
axis([0 0.3 0 25])

function xdot = Current1(t,x)
R=0.8;
L=0.003;
Kt=0.05;
Ke=0.05;
c=0;
J=8*10^-5;
v=20;
A=[-R/L -Ke/L;Kt/J -c/J];
B=[1/L;0];
xdot=A*x+B*v;
end


function xdot = Current2(t,x)
```

```
R=0.8;
L=0.003;
Kt=0.05;
Ke=0.05;
c=0;
J=8*10^-5;
A = [-R/L -Ke/L;Kt/J -c/J];
for k= 1:1000
if t<0.05
v=400*t;
elseif t>0.05 && t<=0.2
v=20;
elseif t>0.2 && t<=0.25
v=-400*(t-0.2)+20;
else
v=0;
end
B=[1/L; 0];
xdot=A*x+B*v;
end
end
```

4. Functions

4.1 Explanations

There are two functions which are Current1, Current2 in this code for two different cases. Moreover, there is also an ode45 function used in the code.

These two functions are built to define the ODEs to be solved in MATLAB environment and ode45 function is used to solve the ODEs.

The ODEs defined by the following functions are:

$$L\dot{i} = -Ri - K_e\omega + v(t) \quad (1),$$

$$J\dot{\omega} = K_T i - c\omega \quad (2)$$

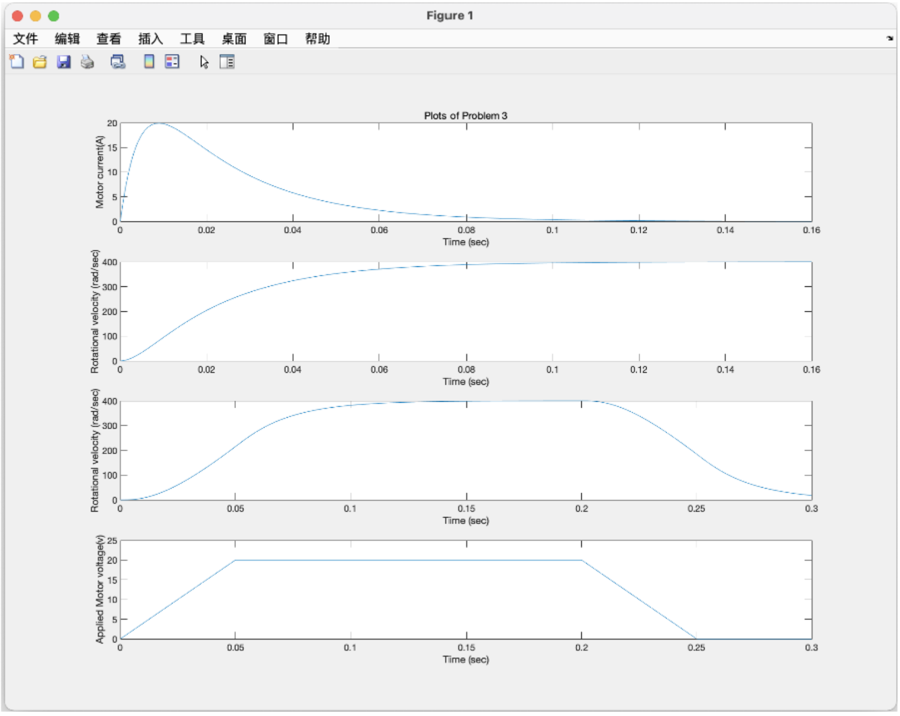
4.2 Codes

```
function xdot = Current1(t,x)
R=0.8;
L=0.003;
Kt=0.05;
Ke=0.05;
c=0;
J=8*10^-5;
v=20;
A=[-R/L -Ke/L;Kt/J -c/J];
B=[1/L;0];
xdot=A*x+B*v;
```

```
end


function xdot = Current2(t,x)
R=0.8;
L=0.003;
Kt=0.05;
Ke=0.05;
c=0;
J=8*10^-5;
A = [-R/L -Ke/L;Kt/J -c/J];
for k= 1:1000
if t<0.05
v=400*t;
elseif t>0.05 && t<=0.2
v=20;
elseif t>0.2 && t<=0.25
v=-400*(t-0.2)+20;
else
v=0;
end
B=[1/L; 0];
xdot=A*x+B*v;
end
end
```

5. Results

5.1 Results figure



5.2 Comments and Analysis

Since the ODE is linear, the matrix method is used in this question.

First step is rewriting two 1nd order ODEs:

$$\dot{i} = -\frac{R}{L}i - \frac{K_e}{L}\omega + \frac{1}{L}v(t) \quad (1),$$

$$\dot{\omega} = \frac{K_T}{J}i - \frac{c}{J}\omega \quad (2)$$

Then the equations (1) (2) can be written in the matrix form.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{K_e}{L} \\ \frac{K_T}{J} & -\frac{c}{J} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} v$$

Then $xdot = Ax + Bv$.

The next step is also defining the ODEs to be solved in MATLAB environment, which can be done by creating function files (Current1, Current2) as shown in 4.2.

The third step is using ode45 function to solve the equation.

In the end, the figures showing the motor's speed and current as a function of time.

6. Flow charts

There is no need for flow charts in this question.

Problem 4

1. Equation derivations

1.1 The equation used in the coding

$$\rho A \frac{dh}{dt} = q + \frac{1}{R_l} SSR(p_l - p) - \frac{1}{R_r} SSR(p - p_r)$$

$$h01\, SSR(\Delta p) = \{ \begin{array}{ll} \sqrt{\Delta p} & if\ \Delta p > 0 \\ -\sqrt{\Delta p} & if\ \Delta p < 0 \end{array}$$

1.2 Coefficients and source terms

$\rho = 1.94\ \mathrm{slug/ft^3}$: the fluid mass density

$A_1 = 3\ \mathrm{ft}^2$: the bottom area of the tank1

$A_2 = 5\ \mathrm{ft}^2$: the bottom area of the tank2

$qin1$: the mass flow rate of the flow source

p_L: supply pressure from the left-hand side tube

p_R: supply pressure from the right-hand side tube

R_L: flow resistance from the left-hand side tube

$R_{L1} = 0.000001\ \mathrm{ft}^{-1} * \mathrm{sec}^{-1},\ R_{L2} = 30\ \mathrm{ft}^{-1} * \mathrm{sec}^{-1}$

R_R: flow resistance from the right-hand side tube

$R_{R1} = 30\ \mathrm{ft}^{-1} * \mathrm{sec}^{-1},\ R_{R2} = 40\ \mathrm{ft}^{-1} * \mathrm{sec}^{-1}$

Δp: the pressure difference across the flow resistances.

h01 = 2 ft, h02 = 5 ft: Initial height in Tank 1

$g = 32.2\ \mathrm{ft} * \mathrm{sec}^{-2}$

2. Computational conditions

Time step from 0s, and the simulation is 20s.

3. Main programme

```
%Problem 4
clear all;
close all;
clc;

A1 = 3;%A2 Bottom Area of Tank 1
A2 = 5;%A2 Bottom Area ot Tank 1
R_L1= 1E6;%No flow through the left-hand inlet
```

```
R_R1 = 30;% ftA(-1)*sA(-1)
R_L2=30;%One resistance between tanks: R_R1= RL2=R1
R_R2 = 40;% ftA(-1)*sA(-1)
rho = 1.94;% kg/m3
g= 32.2;% m/s2
h01 = 2;%Initial height in Tank 1
h02 = 5;%Initial height in Tank 1
qmi1 = 0.5;%Input flow into Tank 1
timestep1 = 0;%Time when Input flow into Tank 1 is turned on
```

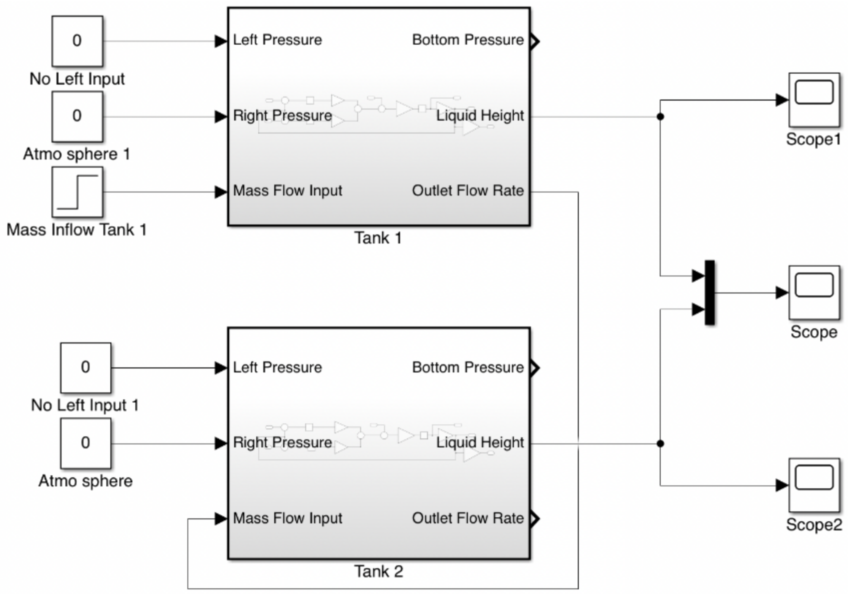
(The main code part is written for defining the initial values for parameters.)

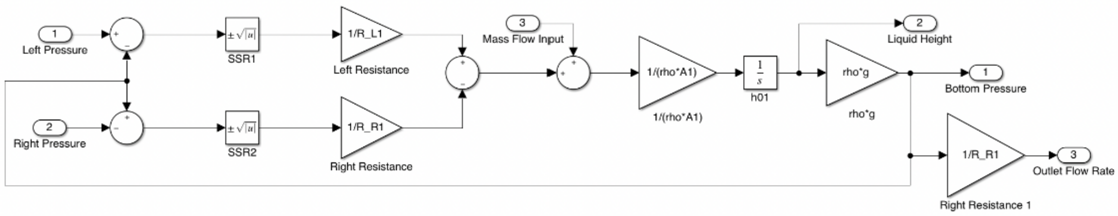
4. Functions

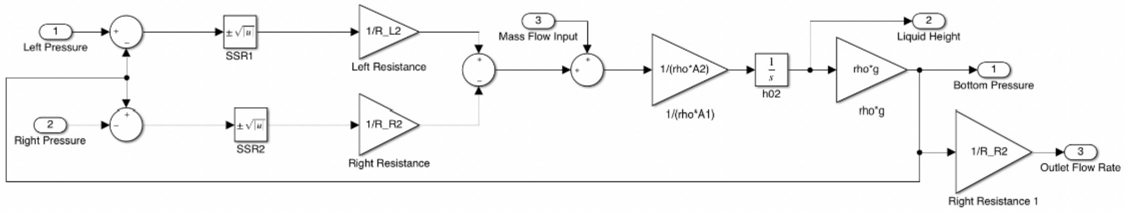
4.1 Explanations

The first picture is the model for hydraulic system, and the following two figures which are Tank 1 and Tank 2.

4.2 Codes

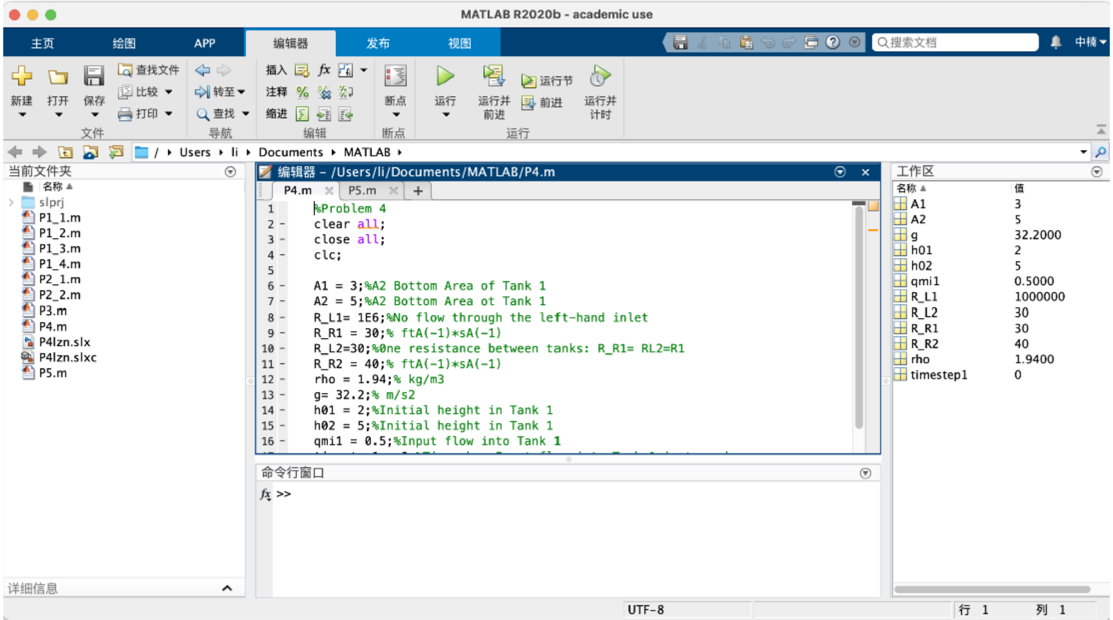


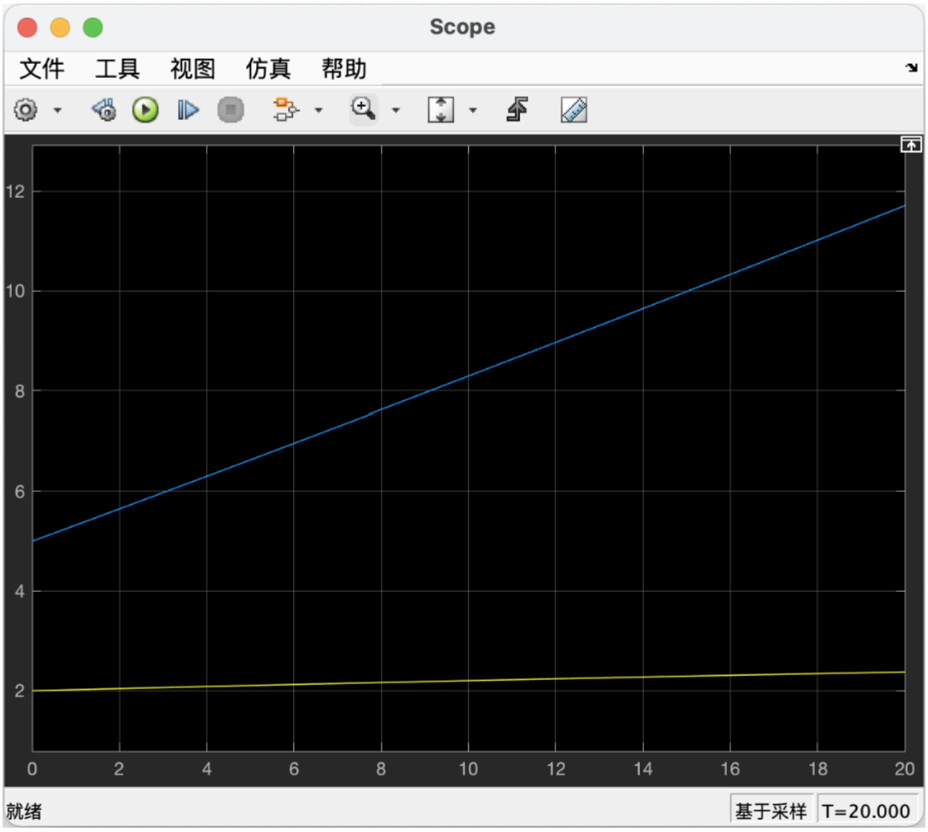




5. Results

5.1 Results figure





5.2 Comments and Analysis

First, run the codes and store the parameters in the work area. And then run the simulation. The figure is the scope from the simulation.

6. Flow charts

Since the simulation represents the process of the question.

Problem 5

1. Equation derivations

1.1 The equation used in the coding

The distribution of electrical potential governed by the Laplace equation in the uniform medium inside the area can be expressed as:

$$\frac{\partial^2 \varphi}{x^2} + \frac{\partial^2 \varphi}{y^2} = 0$$

Apply the finite different method (FDM) to the Partial Differential Equation (PDE) above by using iterative method, which can deduced to the following equation:

$$\varphi_{i,j}^{n+1} = \varphi_{i,j}^{n+1} + \frac{\omega}{4}\left(\varphi_{i,j+1}^{n} + \varphi_{i,j-1}^{n+1} + \varphi_{i-1,j}^{n+1} + \varphi_{i+1,j}^{n} - 4\varphi_{i,j}^{n}\right)$$

The numerical iteration needs to define a relaxation factor which can be estimated by the following equation:

$$\omega = \frac{2}{1 + \sqrt{1 - \left[\dfrac{\cos\left(\dfrac{\pi}{m1}\right) + \cos\left(\dfrac{\pi}{m2}\right)}{2}\right]^2}}$$

1.2 Coefficients and source terms

mesh size h,

m1=a/h, m2=b/h,

hx = m1+1 (with i = 1, 2, 3 … hx),

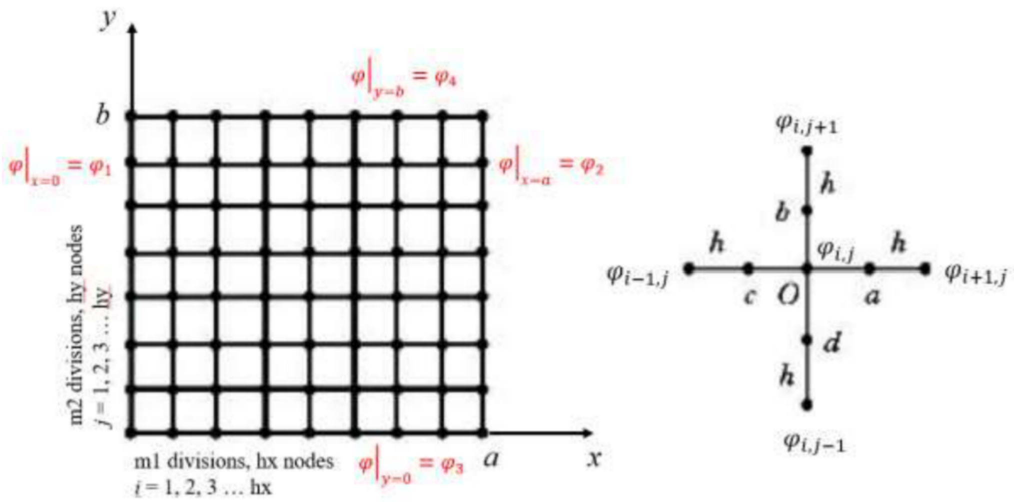
hy = m2+1 (with j = 1, 2, 3 … hy)

2. Computational conditions

Boundary:

Case1: φ|x=0 = φ1, j =0, φ|x=a = φ1, j =0, φ|y=0 = φ1, j =0, φ|y=b = φ1, j =100

Case2: φ|x=0 = φ1, j =0, φ|x=a = φ1, j =100, φ|y=0 = φ1, j =50, φ|y=b = φ1, j =100



(a) The computational domain (b) Node (i, j) and its surrounding nodes

3. Main programme

```
%Problem 5
clear all;
close all;
clc;

%Define parameter lamda
lamda=1;
%Define h for mesh size
h=input('Please specify the mesh size: ');
%Define parameters for computational domain
a=input('Please specify the boundary for x– dimension: ');
b=input('Please specify the boundary for y– dimension: ');
%Define parameter omega
omega=2/(1+sqrt(1–((cos(pi/(a/h))+cos(pi/(b/h)))/2)^2));
%Define numbers of divided length in x & y directions
m1=a/h;
m2=b/h;
%Nodes in x & y directions
hx=m1+1;
hy=m2+1;
%Define and initialize a matrix for voltage u
u=ones(hy,hx);
%Define boundary conditions
u(1,:)=input('Please specify boundary condition for y= 0: ');
u(hy,:)=input('Please specify boundary condition for y = b: ');
u(:,1)=input('Please specify boundary condition for x= 0: ');
u(:,hx)=input('Please specify boundary condition for x = a: ');
%Define domain of the figure
x=0:h:a;
y=0:h:b;
%Call Voltage function to calculate and return the result matrix
u2 = Voltage (hx,hy, omega, u);
%Draw the coordinates and the graph
[X1,Y1]=meshgrid(x,y);
subplot(1,2,1)
mesh(X1,Y1,u2)
axis([0 a 0 b])
%Define show the coordinates labels
xlabel('Dimension x(cm)')
ylabel('Dimension y(cm)')
zlabel('Voltage Phi(V)')
```

```
%Draw the colored Top view figure
subplot(1,2,2)
contour(u2,32)
[C,h]=contour(X1,Y1,u2 );
clabel(C, h)
%Define show the coordinates labels
xlabel('Dimension x(cm)')
ylabel('Dimension y(cm)')

%Finite Difference Method
function u2 = Voltage(hx,hy,omega,u)
m=1;%error value
k=0;%count for the times in loops
u2=u;%copy the matrix to the result matrix
while(m>10e-5)
 k=k+1;
 m=0;
 for j=2:hx-1
  for i=2:hy-1
   %apply iterative equation
   u2(i,j)=u(i,j)+omega*0.25*(u2(i-1,j)+u(i+1,j)+u2(i,j-1)+u(i,j+1)-4*u(i,j));
   t=(abs(u2(i,j)-u(i,j)));%calculate error
   if(t>m)%compare to last error
     m=t;%pass error value
   end
  end
 end
u=u2;%pass the calculated values to the last matrix
end
display(k)%show loop times
end
```

4. Functions

4.1 Explanations

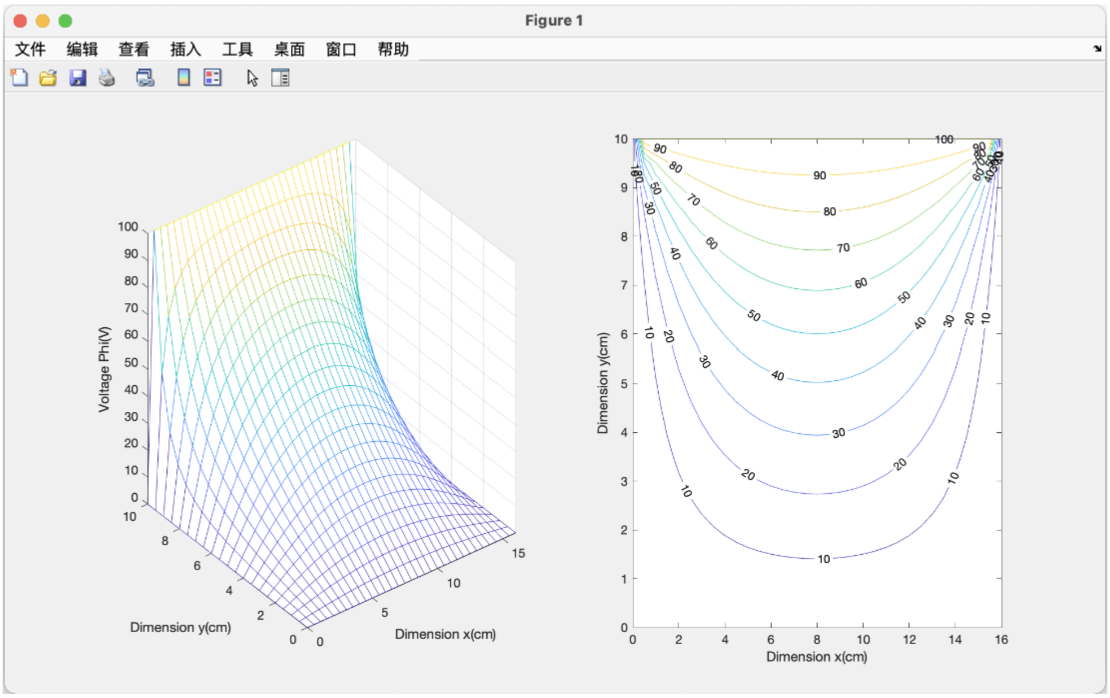
4.2 Codes

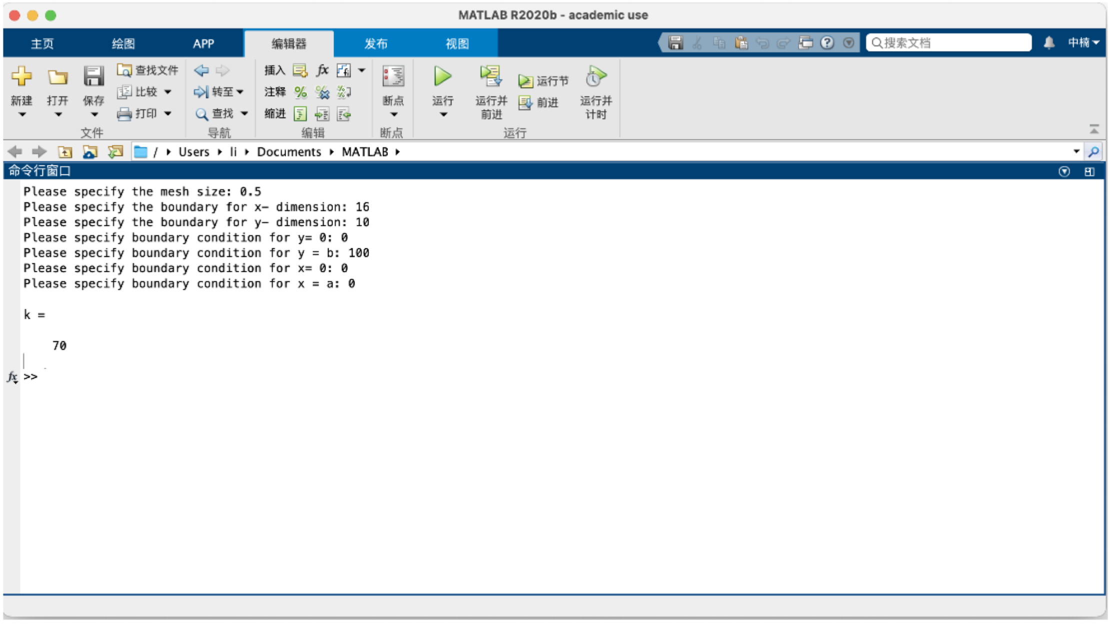
```
%Finite Difference Method
function u2 = Voltage(hx,hy,omega,u)
m=1;%error value
k=0;%count for the times in loops
u2=u;%copy the matrix to the result matrix
while(m>10e-5)
 k=k+1;
```

```
 m=0;
 for j=2:hx-1
  for i=2:hy-1
   %apply iterative equation
   u2(i,j)=u(i,j)+omega*0.25*(u2(i-1,j)+u(i+1,j)+u2(i,j-1)+u(i,j+1)-4*u(i,j));
   t=(abs(u2(i,j)-u(i,j)));%calculate error
   if(t>m)%compare to last error
     m=t;%pass error value
   end
  end
 end
u=u2;%pass the calculated values to the last matrix
end
display(k)%show loop times
end
```

5. Results

5.1 Results figure




MATLAB R2020b - academic use
命令行窗口
Please specify the mesh size: 0.5
Please specify the boundary for x- dimension: 16
Please specify the boundary for y- dimension: 10
Please specify boundary condition for y= 0: 0
Please specify boundary condition for y = b: 100
Please specify boundary condition for x= 0: 0
Please specify boundary condition for x = a: 0
k =
70
>>

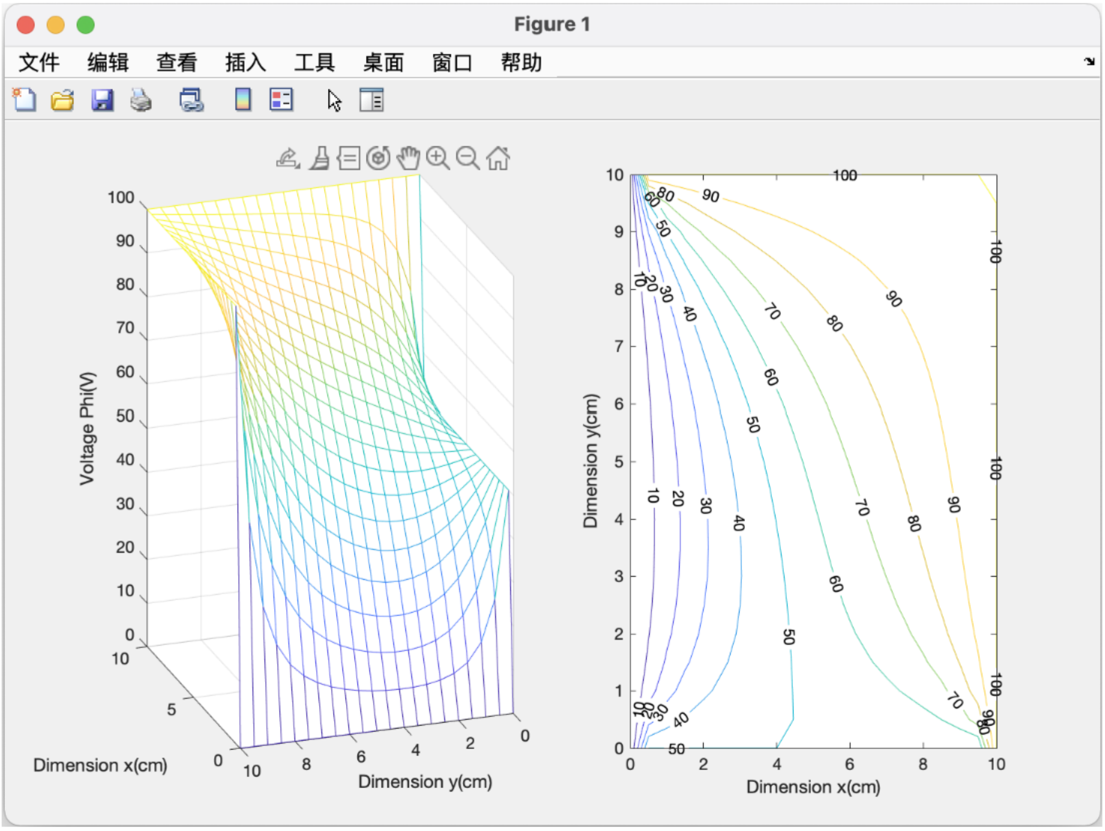
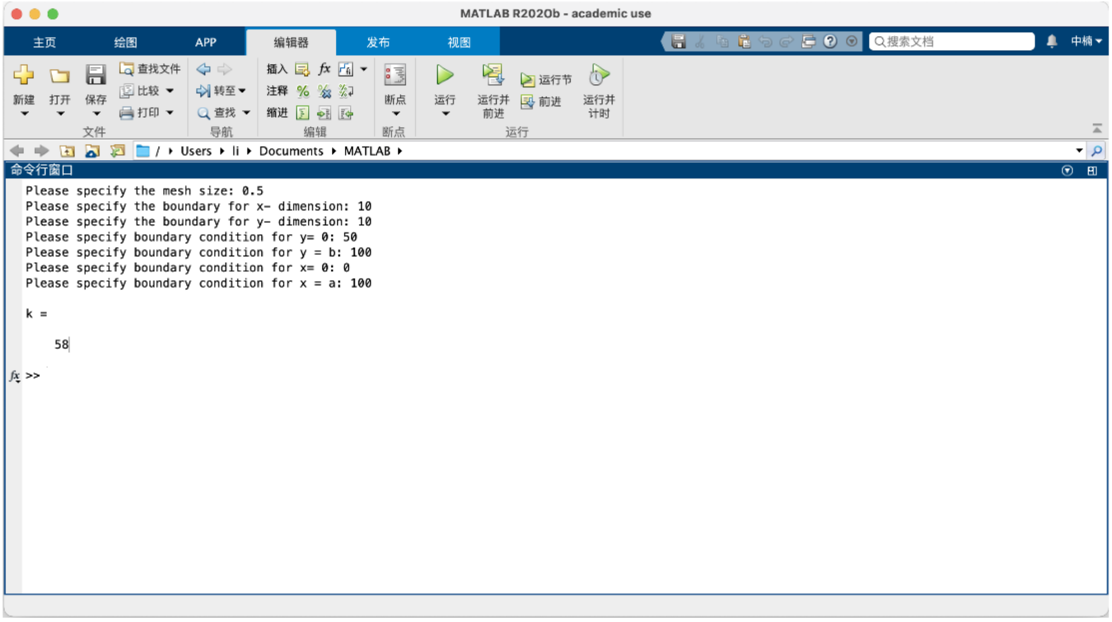


Figure 1
文件 编辑 查看 插入 工具 桌面 窗口 帮助
Voltage Phi(V)
Dimension x(cm)
Dimension y(cm)
Dimension y(cm)
Dimension x(cm)



5.2 Comments and Analysis

This function is used to calculate the result of each point on the surface using an iterative method for the equation below.

6. Flow charts