MATLAB-ASSIGNMENT-1

Student ID Number: 2251625

Student Full Name: Yukun.Zheng22

Student Email Address: Yukun.Zheng22@student.xjtlu.edu.cn

Problem 1. Plotting (10 Marks)

P 1-1 (5 Marks)

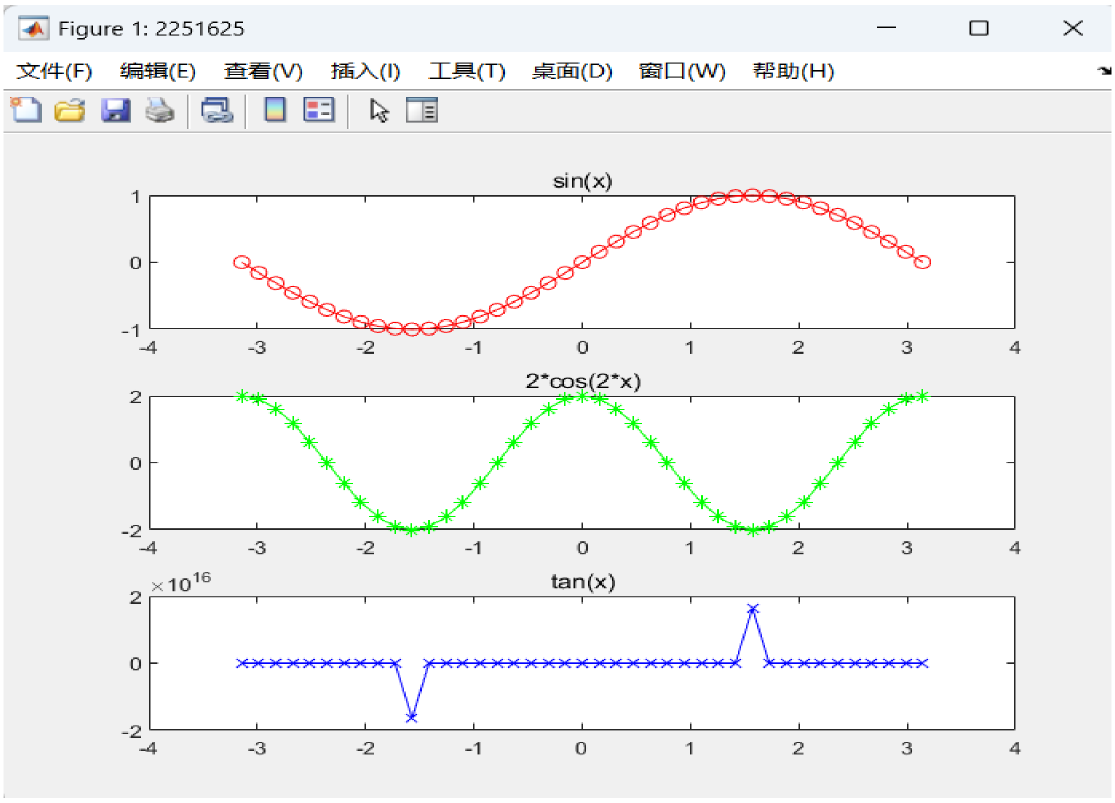
Write a script that can generate the figure shown below using the conditions and provide a screenshot of the figure.

Answer:

Part1: Coding

```
x = -pi:pi/20:pi;
figure('Name','2251625');
y1 = sin(x);
y2 = 2*cos(2*x);
y3 = tan(x);
h1 = subplot(3,1,1);
plot(x, y1, 'r-o');
title('sin(x)');
h2 = subplot(3,1,2);
plot(x, y2, 'g-*');
title('2*cos(2*x)');
h3 = subplot(3,1,3);
plot(x, y3, 'b-x');
title('tan(x)');
axis([-pi, pi, -1, 1]);
ylim(h3, [-2e16, 2e16]);
xlim(h3, [-4, 4]);
```

Part 2: Results Display



Part 3: Comments and analysis of the results

The code first defines the range of x values, then calculates the y values for the sine, 2*cosine(2x), and tangent functions. It then creates three subplots, one for each function, using the subplot function and handle. Each subplot is customized with a plot of the corresponding function and a title. Finally, the axis limits are set for all subplots, with special attention given to the third subplot to account for the wide range of the tangent function.

P 1-2 (5 Marks)

The Arrhenius equation is widely used to describe plenty of physical phenomena. Based on the equation and conditions shown below, write a script that can generate the figure of k as a function of T.

Answer:

Part 1: Coding

```
Q=900;
R=8.314;
k0=15;
x=300:25:1000;
y=k0*exp(-Q./(R*x));
plot(x,y)
title('Reaction-rate constants for chemical reactions');
xlabel('Temperature(K)');
ylabel('Reaction-rate constants(/s)');
```

Part 2: Results Display

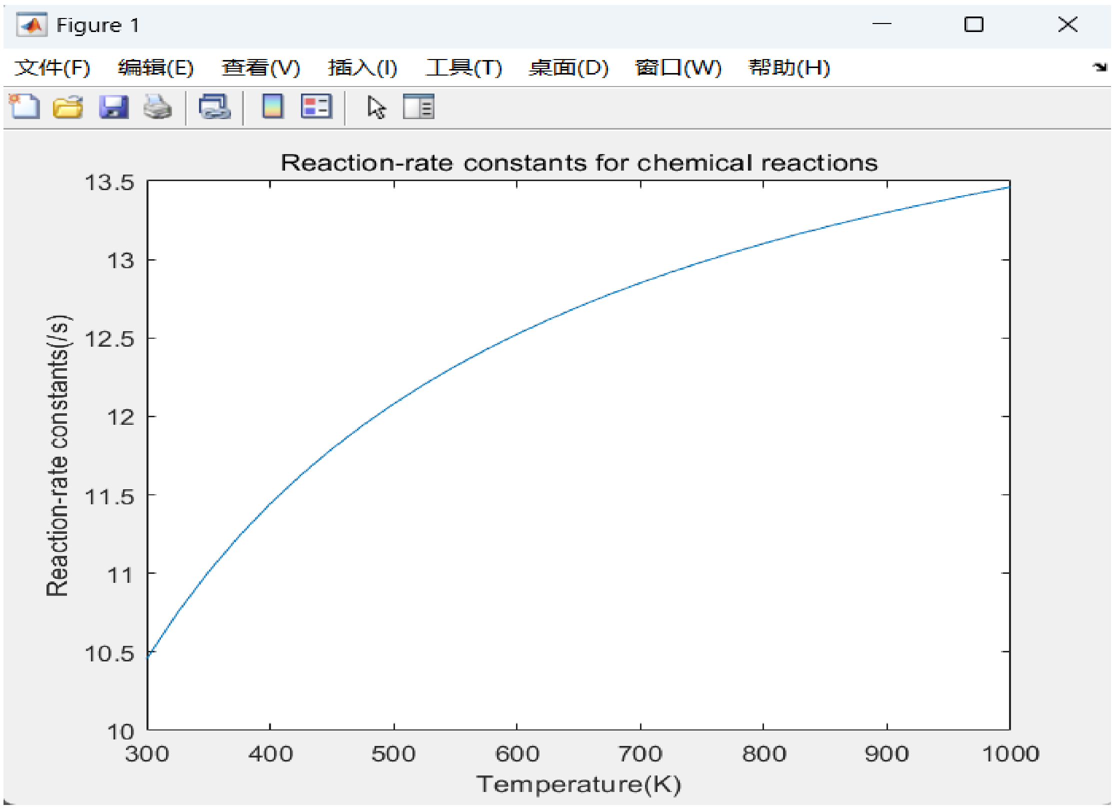


Figure 1
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
Reaction-rate constants for chemical reactions
Reaction-rate constants(/s)
Temperature(K)
13.5
13
12.5
12
11.5
11
10.5
10
300
400
500
600
700
800
900
1000

Part 3: Comments and analysis of the results

I initializes the values for Q, R, and k0, which represent the activation energy, gas constant, and the pre-exponential factor for a chemical reaction rate constant, respectively. I then define a range of x values, representing the temperatures in Kelvin. Next, it calculates the y values using the Arrhenius equation, which models the reaction rate constant as a function of temperature. The plot function is then used to visualize the relationship between temperature and reaction rate constants. Finally, the plot is labeled with a title, x-axis label, and y-axis label for clarity.

P 2-1 (5 Marks, 1 Mark for each small question)

Type this matrix in MATLAB and use MATLAB to carry out the following instructions.

$$\boldsymbol{AA} = \begin{bmatrix} 1 & 7 & -3 & 8 \\ -3 & -7 & 10 & 8 \\ 6 & 4 & 91 & 14 \\ 17 & 10 & 3 & -1 \end{bmatrix}$$

a) Create a vector v consisting of the elements in the third column of AA.

b) Create a vector w consisting of the elements in the second row of AA.

c) Create a 4 x 3 array BB consisting of all elements in the second through fourth columns of AA.

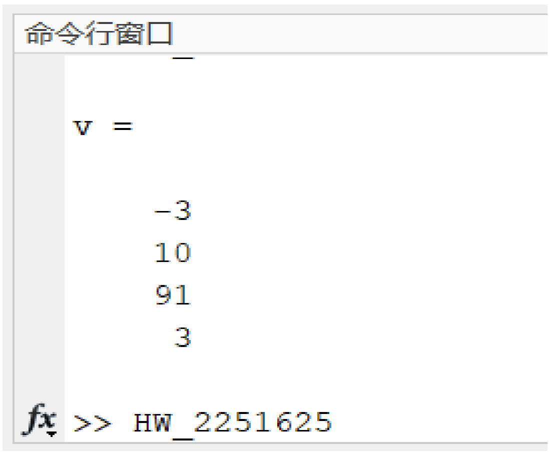
d) Create a 3 x 4 array CC consisting of all elements in the second through fourth rows of AA.

e) Create a 2 x 3 array DD consisting of all elements in the first two rows and the last three columns of AA.

Answers:

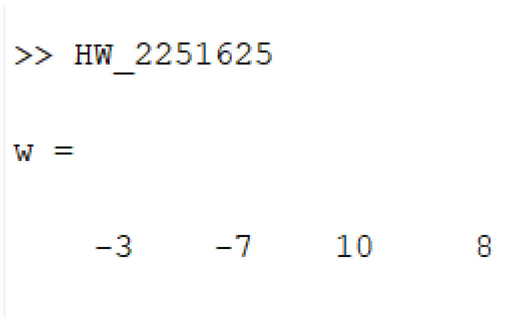
Part1: Coding and Results Display

```
AA=[1 7 -3 8;-3 -7 10 8;6 4 91 14;17 10 3 -1];
%(a)
v=AA(:,3)
```



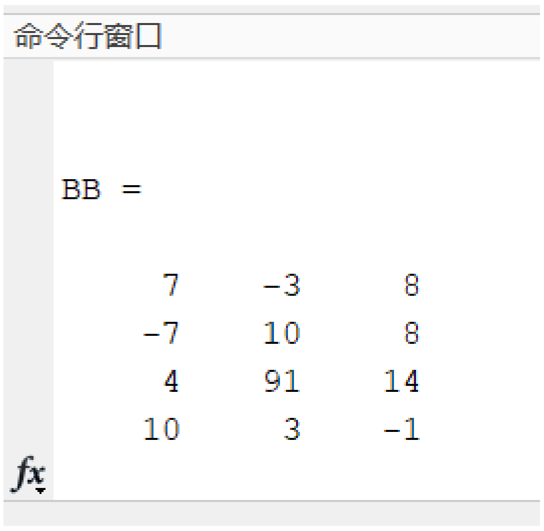
```
v =

    -3
    10
    91
     3

>> HW_2251625
```

```
%(b)AA
w=AA(2,:)
```



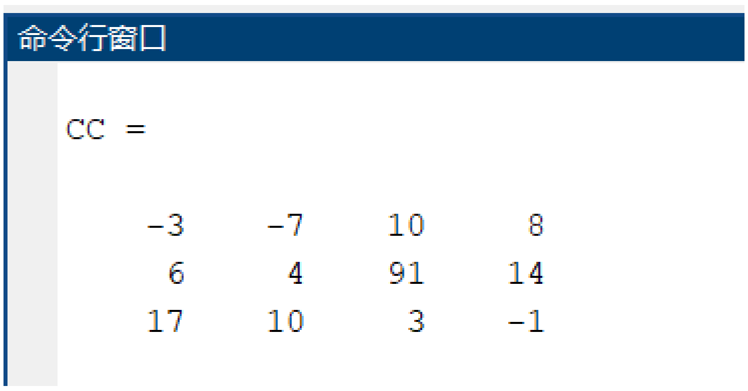
```
>> HW_2251625

w =

    -3    -7    10     8
```

```
%(c)4*3 BB
BB=zeros(4,3);
for ii=1:1:4
    for jj=1:1:3
        BB(ii,jj)=AA(ii,jj+1)
    end
end
disp(BB)
```



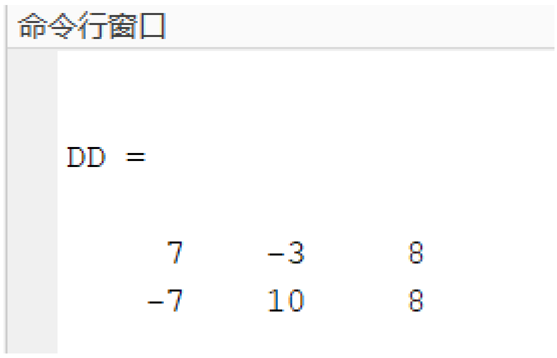
```
BB =

     7    -3     8
    -7    10     8
     4    91    14
    10     3    -1
```

```
%(d) 3*4
CC=zeros(3,4);
for ii=1:1:3
    for jj=1:1:4
        CC(ii,jj)=AA(ii+1,jj)
    end
end
disp(CC)
```



```
CC =

    -3    -7    10     8
     6     4    91    14
    17    10     3    -1
```

```
%(e)2*3 array
DD=zeros(2,3);
for ii=1:1:2
    for jj=1:1:3
        DD(ii,jj)=AA(ii,jj+1)
    end
end
disp(DD)
```



```
DD =

     7    -3     8
    -7    10     8
```

Part2: Comments and analysis of the results

I begin with initializing a matrix AA with four rows and four columns. Subsequently, I perform a series of operations on this matrix to extract specific elements, create new matrices, and compute their values.

(a) If “:” on the left side of the index, it usually means "all rows"; else ”:” on the right side of the index, it usually means "all columns", and the comma is used to separate rows and columns. `(:,3)` the elements in the third column

(b) Similar to A, `(2,:)` means the elements in the second row

(c) Two for loops to get the elements needed like C-Language learnt in last semester.

(d) Same method as (c).

(e) My code performs a combination of all elements in the first two rows and the last three columns via two for loops. *Notice that I set the upper limit “2” to avoid outstripping the limit of the array.*

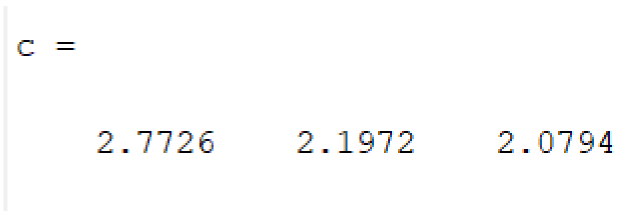
P 2-2 (5 Marks, 1 Mark for each small question)
Consider the following arrays:

$$\boldsymbol{A} = \begin{bmatrix} 2 & 5 & 6 \\ 2 & 3 & 25 \\ 16 & 9 & 8 \\ 12 & 2\pi & 42 \end{bmatrix}, \qquad \boldsymbol{B} = \ln \boldsymbol{A}$$

Write MATLAB expressions to do the following.
a) Select just the third row of B.
b) Evaluate the sum of the second row of B.
c) Multiply the second column of B and the first column of A element by element.
d) Evaluate the maximum value in the vector resulting from element-by-element multiplication of the second column of B with the first column of A.
e) Use element-by-element division to divide the first row of A by the first three elements of the third column of B. Evaluate the sum of the elements of the resulting vector.

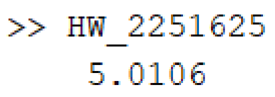
Answers:
Part 1: Coding and results display

```
A=[2 5 6;2 3 25;16 9 8;12 2*pi 42];
B=log(A);
%(a)
c=B(3,:)
```



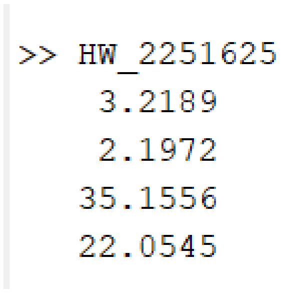
```
c =

    2.7726    2.1972    2.0794
```

```
%(b)
sum=0;
for ii=1:1:3
    sum=sum+B(2,ii);
end
disp(sum)
```



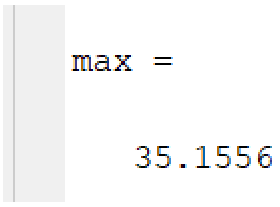
```
>> HW_2251625
    5.0106
```

```
%(c)
D=B(:,2);
E=A(:,1);
MULI=D.*E;
disp(MULI)
```



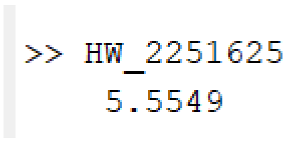
```
>> HW_2251625
    3.2189
    2.1972
   35.1556
   22.0545
```

```
%(d)
max=0;
for ii=1:1:3
    max=MULI(ii)
    if MULI(ii+1)>max
        max=MULI(ii+1)
    end
end
disp(max)
```



```
max =

   35.1556
```

```
%(e)
F=A(1,:);
H=B(:,3);
G=[0;0;0];
for ii=1:1:3
    G(ii)=H(ii);
end
sum=0;
for jj=1:1:3
    sum=sum+F(jj)./G(jj);
end
disp(sum)
```



```
>> HW_2251625
    5.5549
```

Part2: Comments and analysis of the results

I first initialize the matrix A with a set of numerical values. Subsequently, it applies the natural logarithm function to each element of A, resulting in the matrix B. Then, the code performs several operations on these matrices:

(a) If “:” on the left side of the index, it usually means "all rows"; else ”:” on the right side of the index, it usually means "all columns", and the comma is used to separate rows and columns.

(b) The sum of the elements in the second row of B is computed by a for loop and iteratively adding each element. The result is then displayed.

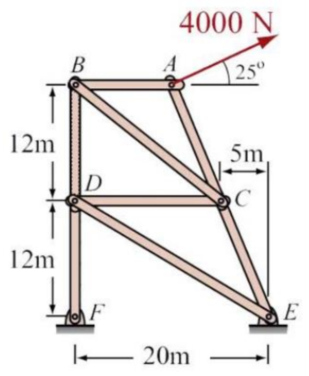
(c) My code calculates the element-wise product of the second column of B and the first column of A. The resulting vector MULI is then displayed.

(d) The maximum element of MULI is determined by iterating through the vector and comparing each element with the current maximum. *Notice that I set the upper limit “3” to avoid outstripping the limit of the array.*

(e) My code performs an element-wise division of the first row of A by the first three elements of the third column of B via two for loops.

P 2-3 (5 Marks)

The figure shows the forces in the eight-member truss. The forces can be determined by solving a system of linear equations using the sets of equations shown below. Using the equations shown below, set the system of linear equations and get the solutions of the forces.



1) $0.9231F_{AC} = 1690;$
2) $-F_{AB} - 0.3846F_{AC} = 3625;$
3) $F_{AB} - 0.7809F_{BC} = 0;$
4) $0.6247F_{BC} - F_{BD} = 0;$
5) $F_{CD} - 0.8575F_{DE} = 0;$
6) $F_{BD} - 0.5145F_{DE} - F_{DF} = 0;$
7) $0.3846F_{CE} - 0.3856F_{AC} - 0.7809F_{BC} - F_{CD} = 0;$
8) $0.9231F_{AC} + 0.6247F_{BC} - 9231F_{CE} = 0.$

Figure 3. Forces in the eight-member truss for P 2-3.

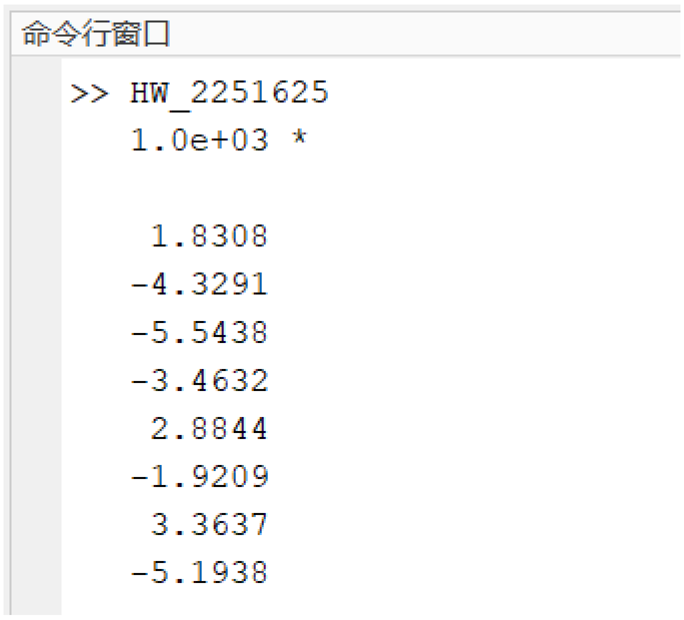
Answers:

Part1: Coding

```
A1=[0.9231,0,0,0,0,0,0,0];
A2=[-0.3846,-1,0,0,0,0,0,0];
A3=[0,1,-0.7809,0,0,0,0,0];
A4=[0,0,0.6247,-1,0,0,0,0];
A5=[0,0,0,0,1,0,-0.8575,0];
A6=[0,0,0,1,0,0,-0.5145,-1];
A7=[-0.3856,0,-0.7809,0,-1,0.3846,0,0];
A8=[0.9231,0,0.6247,0,0,-9231,0,0];
A=[A1;A2;A3;A4;A5;A6;A7;A8];
```

```
b=[1690;3625;0;0;0;0;0;0];
x=A\b;
disp(x)
```

Part2: Results Display

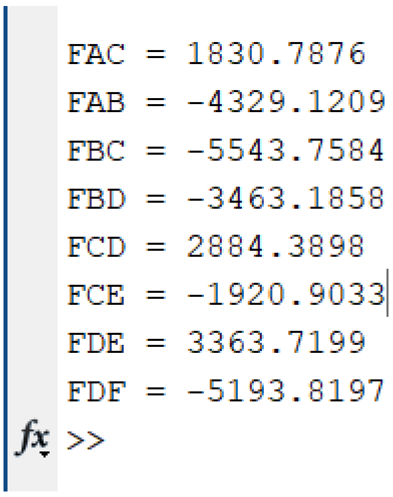


```
>> HW_2251625
   1.0e+03 *

    1.8308
   -4.3291
   -5.5438
   -3.4632
    2.8844
   -1.9209
    3.3637
   -5.1938
```

Part3: Comments and analysis of the results

In this question, we are required to finish achieving the answer of a matrix by MATLAB instead of utilizing linear algebra to get the solution of it. Moreover, there is something we can do to make the results clearer. I explored the MATLAB help section and find a way to better perform the answer logically. Let me present my code.

```
%create a structure like C-Language
solution = struct();
% assign name for each solution
solution.FAC = x(1);
solution.FAB = x(2);
solution.FBC = x(3);
solution.FBD = x(4);
solution.FCD = x(5);
solution.FCE = x(6);
solution.FDE = x(7);
solution.FDF = x(8);
% display the results
disp(['FAC = ' num2str(solution.FAC)]);
disp(['FAB = ' num2str(solution.FAB)]);
disp(['FBC = ' num2str(solution.FBC)]);
disp(['FBD = ' num2str(solution.FBD)]);
disp(['FCD = ' num2str(solution.FCD)]);
disp(['FCE = ' num2str(solution.FCE)]);
disp(['FDE = ' num2str(solution.FDE)]);
disp(['FDF = ' num2str(solution.FDF)]);
```



```
FAC = 1830.7876
FAB = -4329.1209
FBC = -5543.7584
FBD = -3463.1858
FCD = 2884.3898
FCE = -1920.9033
FDE = 3363.7199
FDF = -5193.8197
>>
```

Making use of the structure provides a clearer way to present the answers to the users.

Problem 3 (30 Marks)

P 3-1 (15 Marks, 10 Marks for programming and 5 Marks for displaying results)

The equation of motion for a pendulum whose base is accelerating horizontally with an acceleration a(t) is

$$L\ddot{\theta} + g\sin\theta = a(t)\cos\theta$$

Suppose that g =9.81 m/s2, L=1.5m, and θ (0) =0. Plot 0(t) for 0 ≤ t ≤ 10s for the following three cases:

a) The acceleration is constant: a = 5 m/s2, and θ (0) = 0.5 rad.

b) The acceleration is constant: a = 6m/s2, and θ (0) = 3 rad.

c) The acceleration is linear with time: a =0.5t m/s2, and θ (0) = 3 rad.

Answers:

(a)

Part 1: Coding

Self-defined function

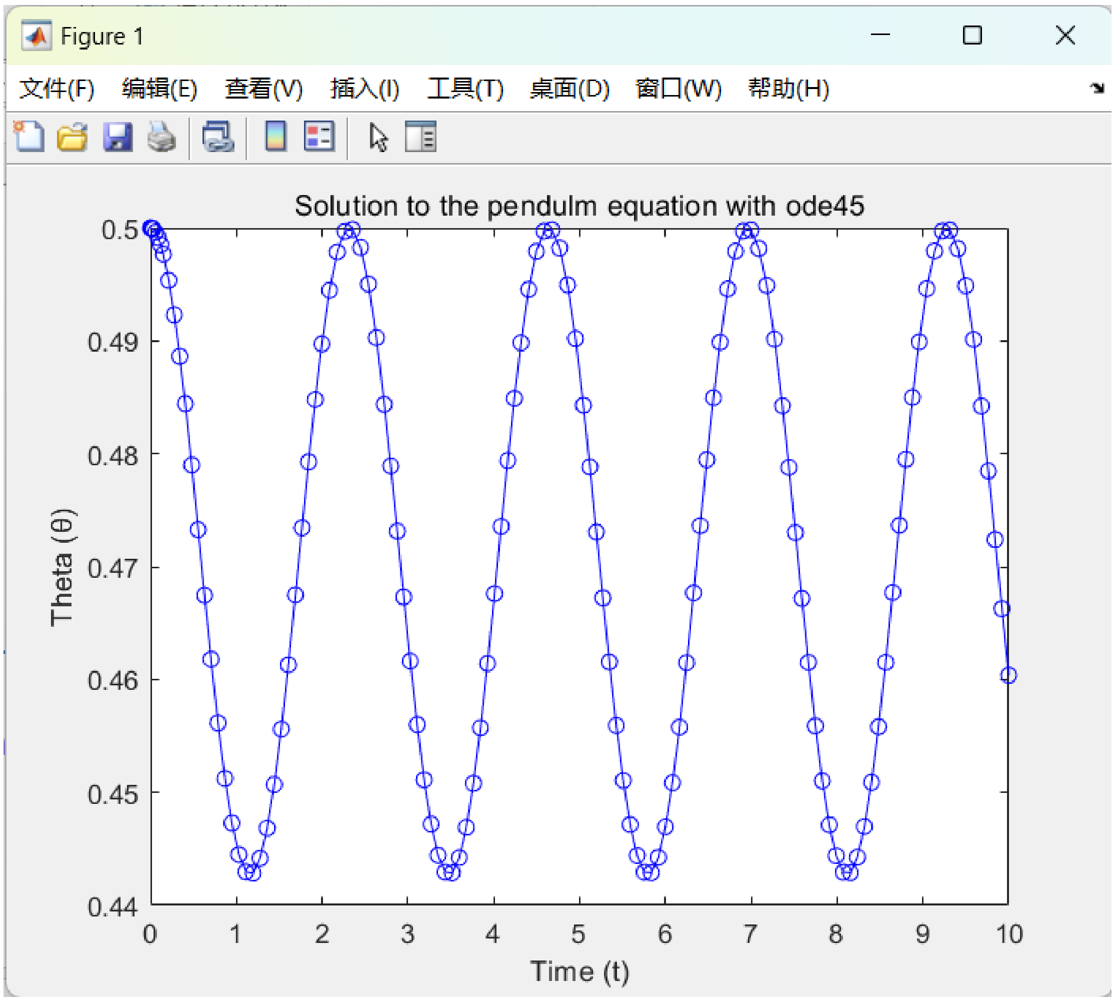
```
function dthetadt = pendulum(t, theta)
%pendulum
%A function that defines motion function
    g = 9.81;
    L = 1.5;
    a = 5;
    dthetadt = [theta(2); -g*sin(theta(1))/L + a*cos(theta(1))/L];
```

```
end
```

Script

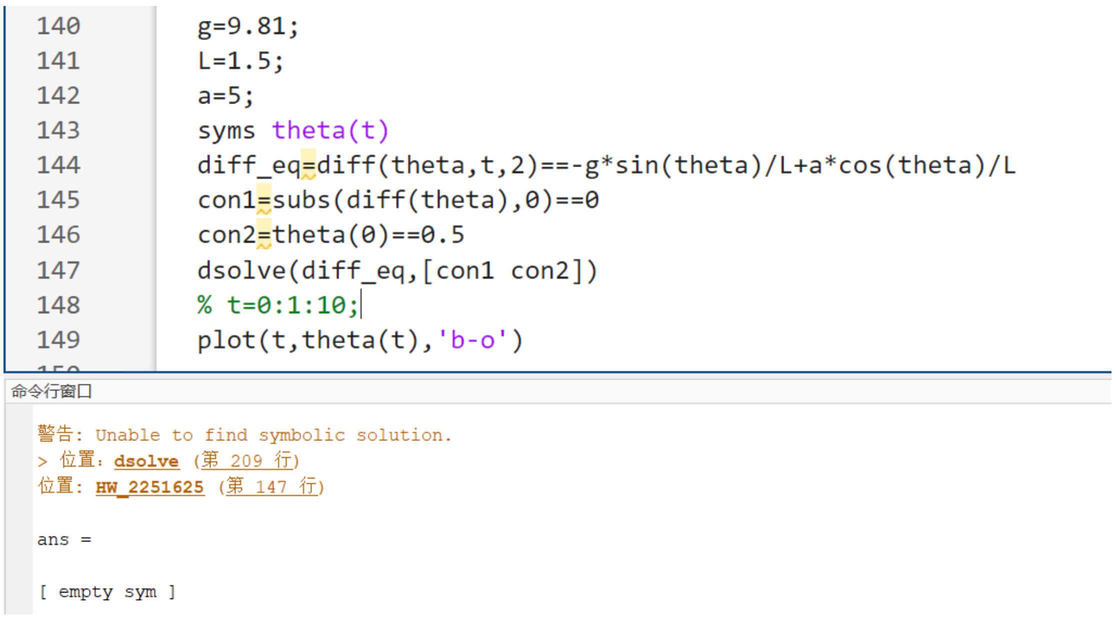
```
theta0 = [0.5; 0];
[t, theta] = ode45(@pendulum, [0 10], theta0);
plot(t, theta(:,1), 'b-o')
title('Solution to the pendulm equation with ode45')
xlabel('Time (t)')
ylabel('Theta (θ)')
```

Part 2: Results Display



Part3: Comments and analysis of the results

First, I try to use the method of “dsolve” to achieve the results of the solution like below:


```
g=9.81;
L=1.5;
a=5;
syms theta(t)
diff_eq=diff(theta,t,2)==-g*sin(theta)/L+a*cos(theta)/L
con1=subs(diff(theta),0)==0
con2=theta(0)==0.5
dsolve(diff_eq,[con1 con2])
% t=0:1:10;
plot(t,theta(t),'b-o')
```

```
命令行窗口
警告: Unable to find symbolic solution.
> 位置: dsolve (第 209 行)
位置: HW_2251625 (第 147 行)

ans =

[ empty sym ]
```

Only to be told that the result is empty, I guess that it may be attributed to the lack of symbol solution of this ode, which means that we have to find the **Numerical** solution of this ode. So I tried another method—"ode45", and get the graph as I mentioned in the preceding part. I first split the second order ode into two equations with each equation containing the first derivative in my draft paper. Then I define a function pendulum to write this equation. Finally, I called ode 45 and plot the graph in my favorite color—blue in the script.

One more thing I would like to report is that I attempted to use the symbol θ in the MATLAB, but it fails the recognize. Thus, I had no choice but to spell the English of it.

(b)

Answers:

Part 1: Coding

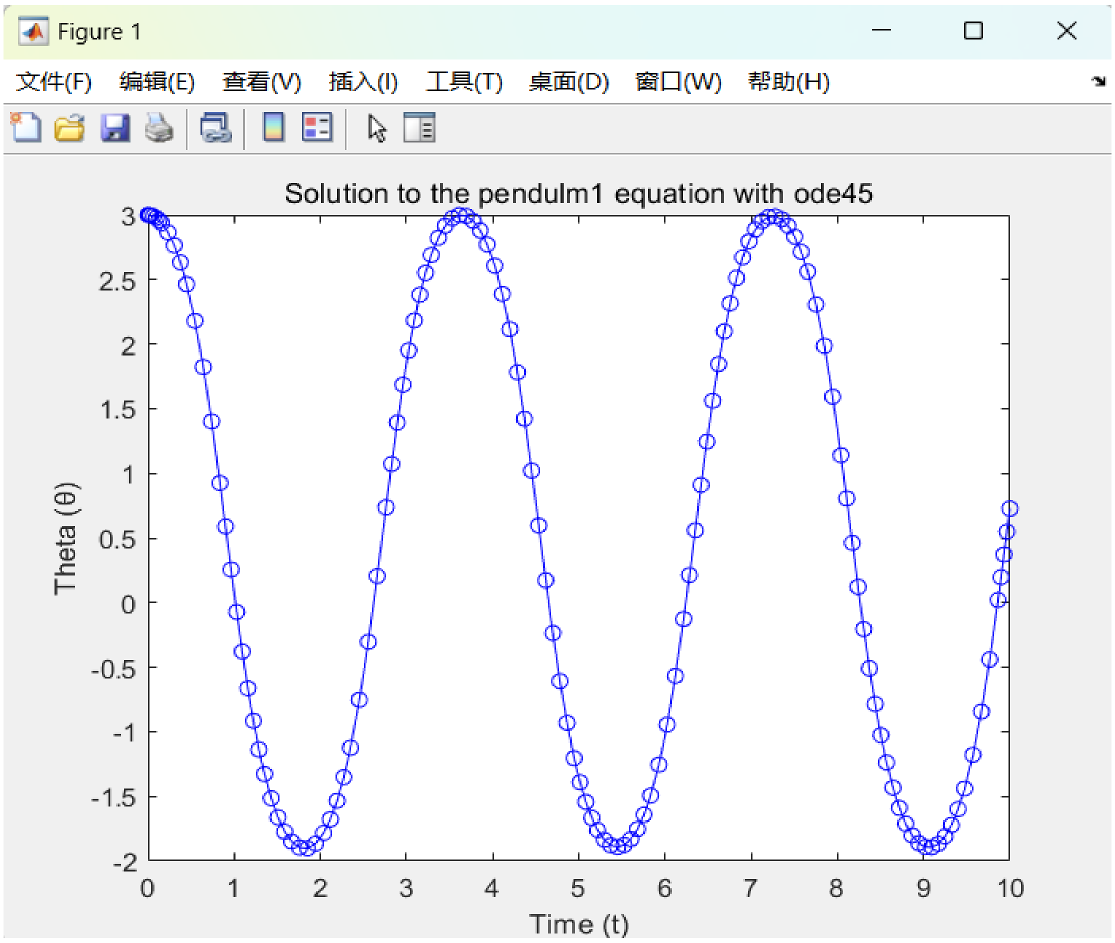
Self-defined function

```
function dthetadt = pendulum1(t, theta)
%pendulum
%A function that defines motion function
    g = 9.81;
    L = 1.5;
    a = 6;
    dthetadt = [theta(2); -g*sin(theta(1))/L + a*cos(theta(1))/L];
end
```

Script

```
theta0 = [3; 0];
[t, theta] = ode45(@pendulum1, [0 10], theta0);
plot(t, theta(:,1), 'b-o')
title('Solution to the pendulm1 equation with ode45')
xlabel('Time (t)')
ylabel('Theta (θ)')
```

Part 2: Results and Display



Part3: Comments and analysis of the results

After finishing the preceding example, it is easy for me to get the graph of it. Similarly, I first split the second order ode into two equations with each equation containing the first derivative in my draft paper. Then I define a function pendulm1(add 1 just for avoiding replication) to write this equation. Finally, I called ode 45 and plot the graph in my favorite color—blue in the script.

In our tutorial, we are told to use "tspan" to replace the time interval. Here, for the consideration of the simplicity and clearness, I adopt to make the time interval explicit instead of "tspan".

(c)

Answers:

Part1: Coding

Self-defined function 1

```
function dthetadt = pendulum2(t, theta, a_func)
    g = 9.81;
    L = 1.5;
    a = a_func(t);
    dthetadt = [theta(2); -g*sin(theta(1))/L + a*cos(theta(1))/L];
end
```

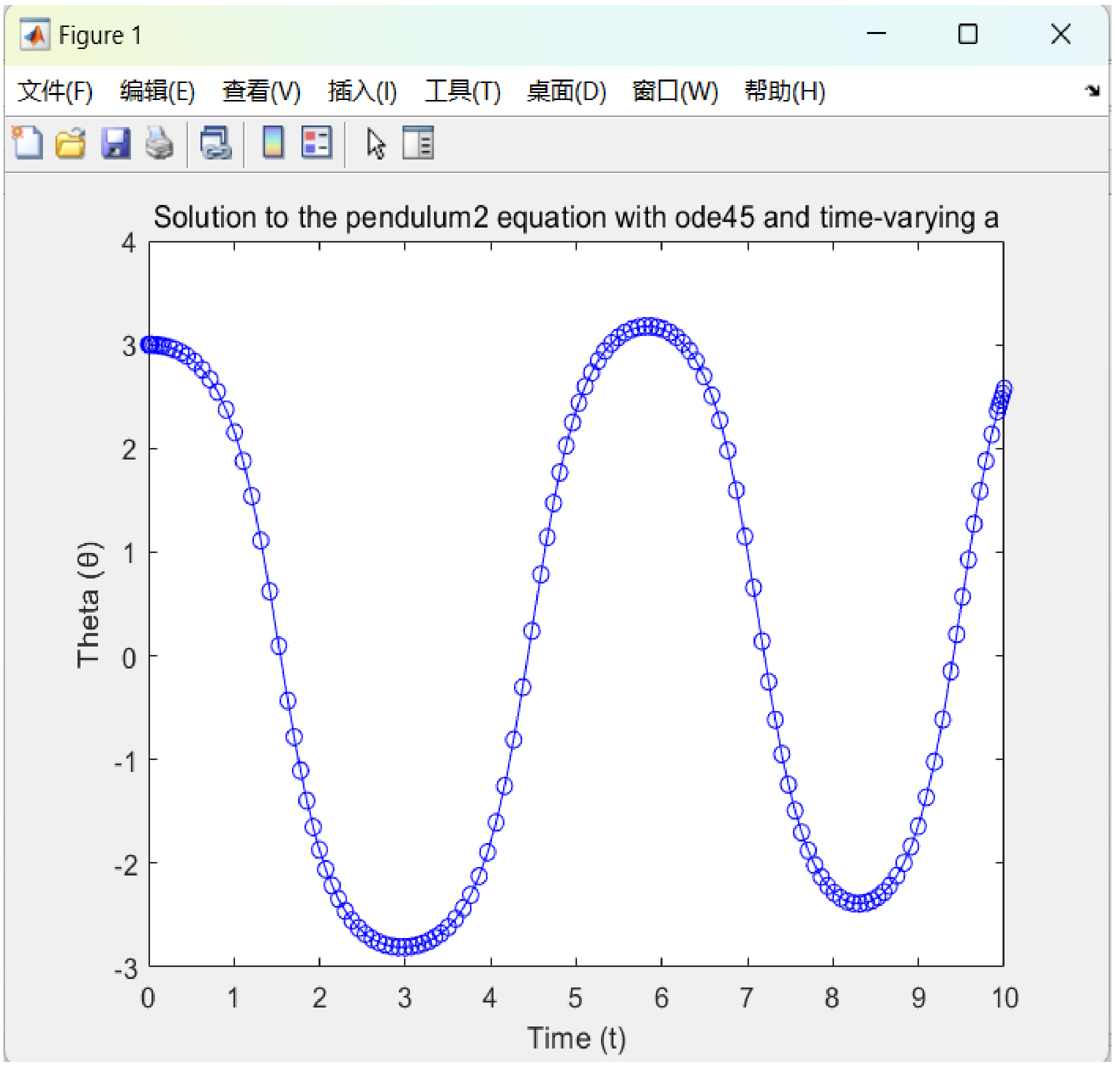
Self-defined function 2

```
function a = a_func(t)
    a = 0.5 * t;
end
```

Script

```
theta0 = [3; 0];
[t, theta] = ode45(@(t, theta) pendulum2(t, theta, @a_func), [0 10],
theta0);
plot(t, theta(:,1), 'b-o')
title('Solution to the pendulum2 equation with ode45 and time-varying a')
xlabel('Time (t)')
ylabel('Theta (θ)')
```

Part 2: Results and Display



Part3: Comments and analysis of the results

The difference between this question and the previous two lies in the acceleration. In this sample, “a” is not a constant, but time-varying. Hence, I treat it as a variable regarding time “t”, and let it be a function like θ. Fortunately, it works within my anticipation.

P 3-2 (15 Marks, 10 Marks for programming and 5 Marks for displaying results)
The following equation describes the motion of a certain mass connected to a spring, with no friction

$$3\ddot{y} + 60y = f(t)$$

where f(t) is an applied force. Suppose the applied force is sinusoidal with a frequency of w rad/s and an amplitude of 10 N: f(t) = 10 sin(wt).

Suppose that the initial conditions are y(0) = y(0) =0. Plot y(t) for $0 \leq t \leq 20$ s. Do this for the following three cases. Compare the results of each case:

a) w=2 rad/s

b) w=5rad/s

c) w= 10 rad/s

Answers:

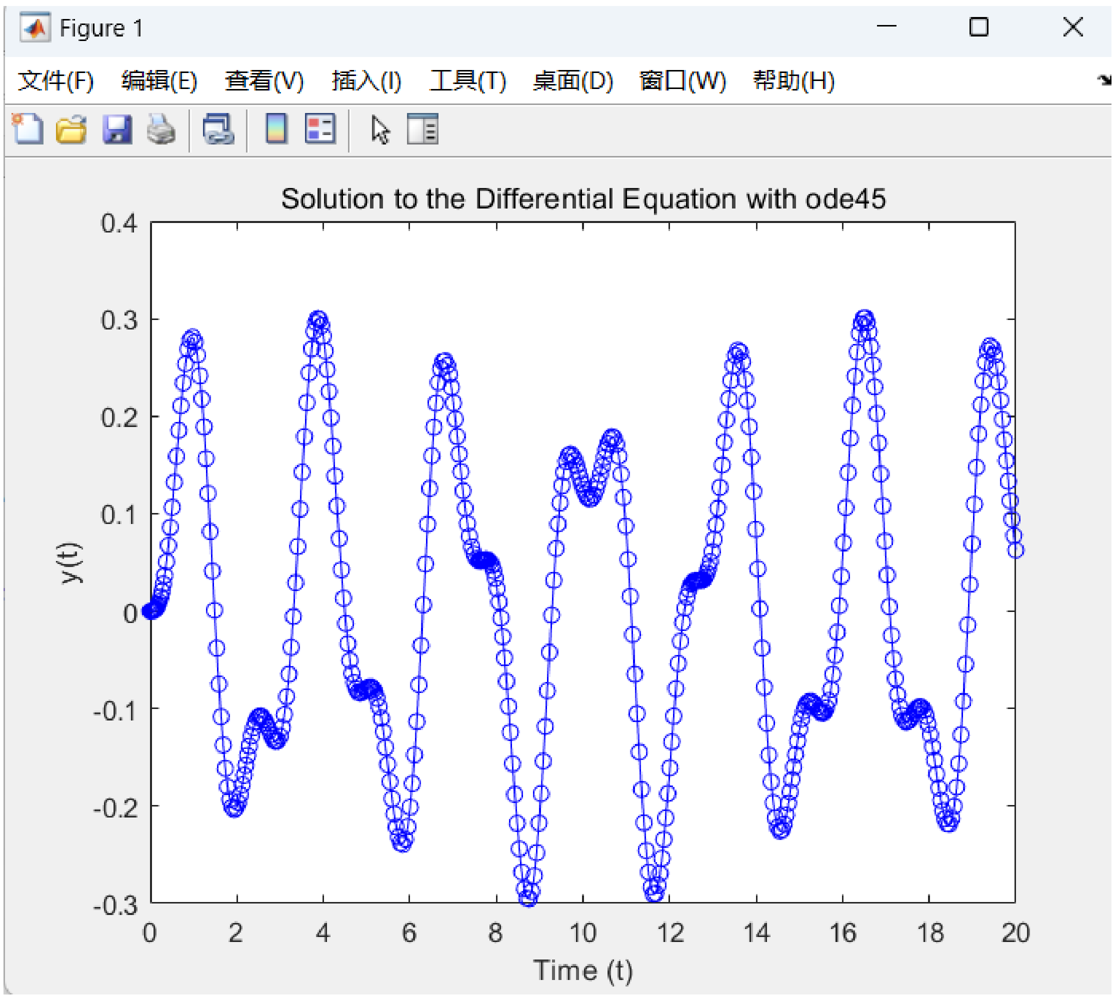
Part 1: Coding and Results Display

Self-defined functions:

```
function dydt = spring(t, y,omega)
%spring
%A function that the certain mass of spring
    dydt = [y(2);-60*y(1)/3 + 10*sin(omega*t)/3];
end
```

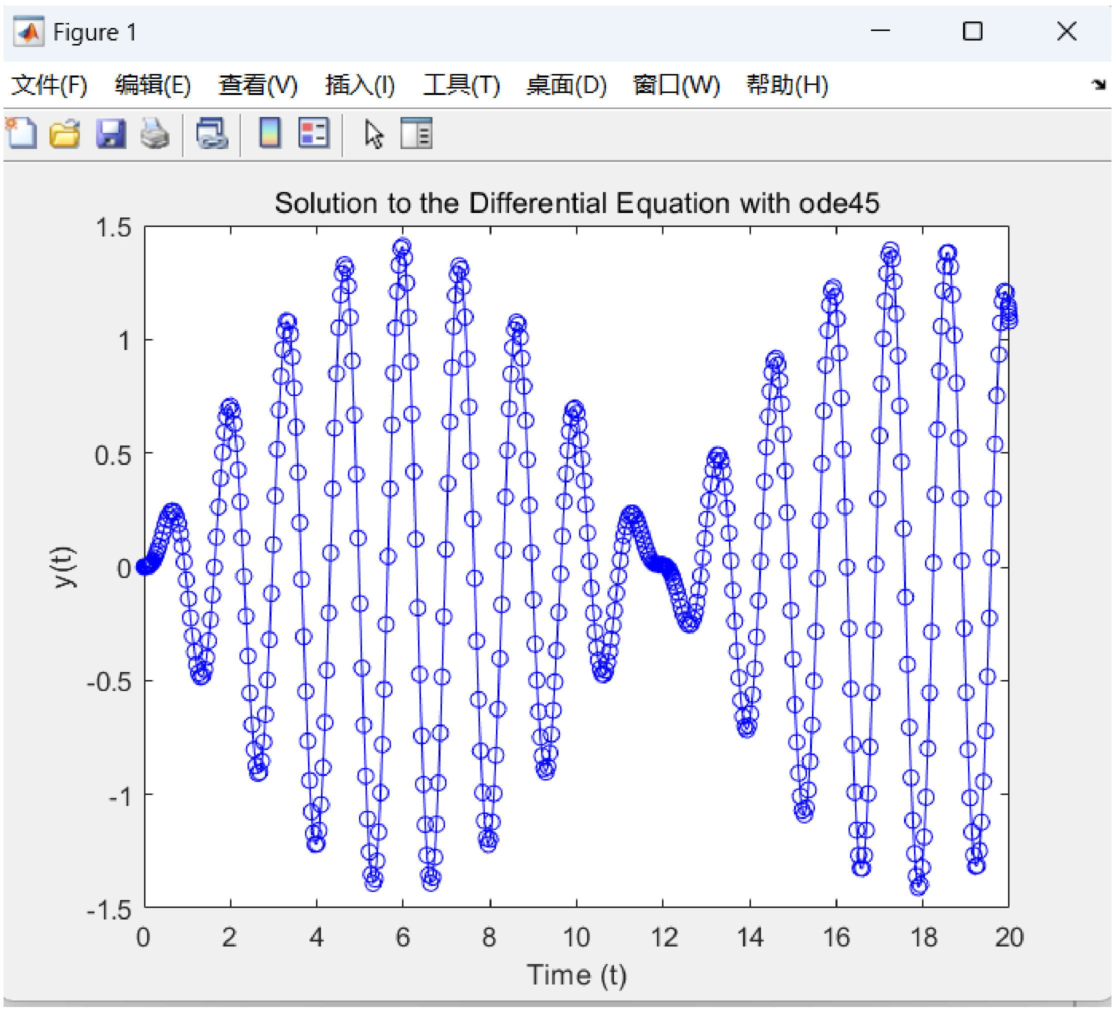
Script for (a)

```
omega = 2;
y0 = [0; 0];
[t, y] = ode45(@(t, y) spring(t, y, omega), [0 20], y0);
plot(t, y(:,1), 'b-o')
title('Solution to the Differential Equation with ode45')
xlabel('Time (t)')
ylabel('y(t)')
```



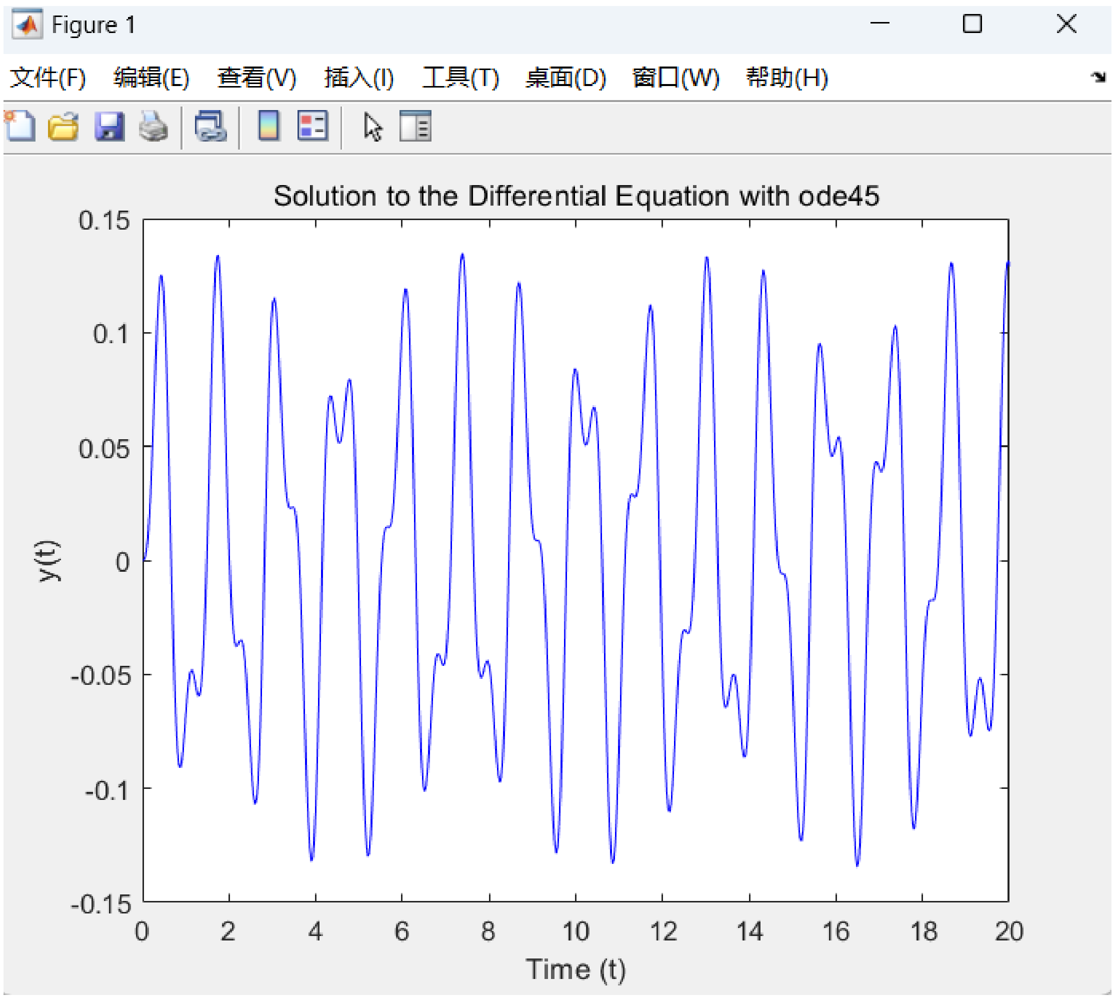
Script for (b)

```
omega = 5;
y0 = [0; 0];
[t, y] = ode45(@(t, y) spring(t, y, omega), [0 20], y0);
plot(t, y(:,1), 'b-o')
title('Solution to the Differential Equation with ode45')
xlabel('Time (t)')
ylabel('y(t)')
```



Script for (c)

```
omega = 10;
y0 = [0; 0];
[t, y] = ode45(@(t, y) spring(t, y, omega), [0 20], y0);
plot(t, y(:,1), 'b-')
title('Solution to the Differential Equation with ode45')
xlabel('Time (t)')
ylabel('y(t)')
```



Part2: Comments and analysis of the results

Inspired by the previous question, I spent less time for coding in this question. Also, on account of the fact that ω is not a function of time, I don't have to define another function like the question before. The function "spring" here is to define the formula for calculation like before.

Notice that the type of my line of graph for (c) is '-' instead of 'o'. The reason is that the graph may be hard to distinguish because of the large density of "x".

Problem 4 (25 Marks, with 1) 15 Marks, 2) 10 Marks)

Consider a circuit system as shown in Figure 4.

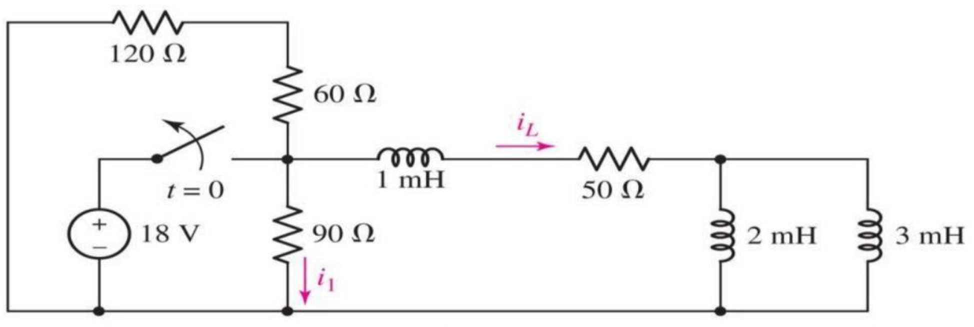


Figure 4. A circuit with multiple resistors and inductors for Problem 4.

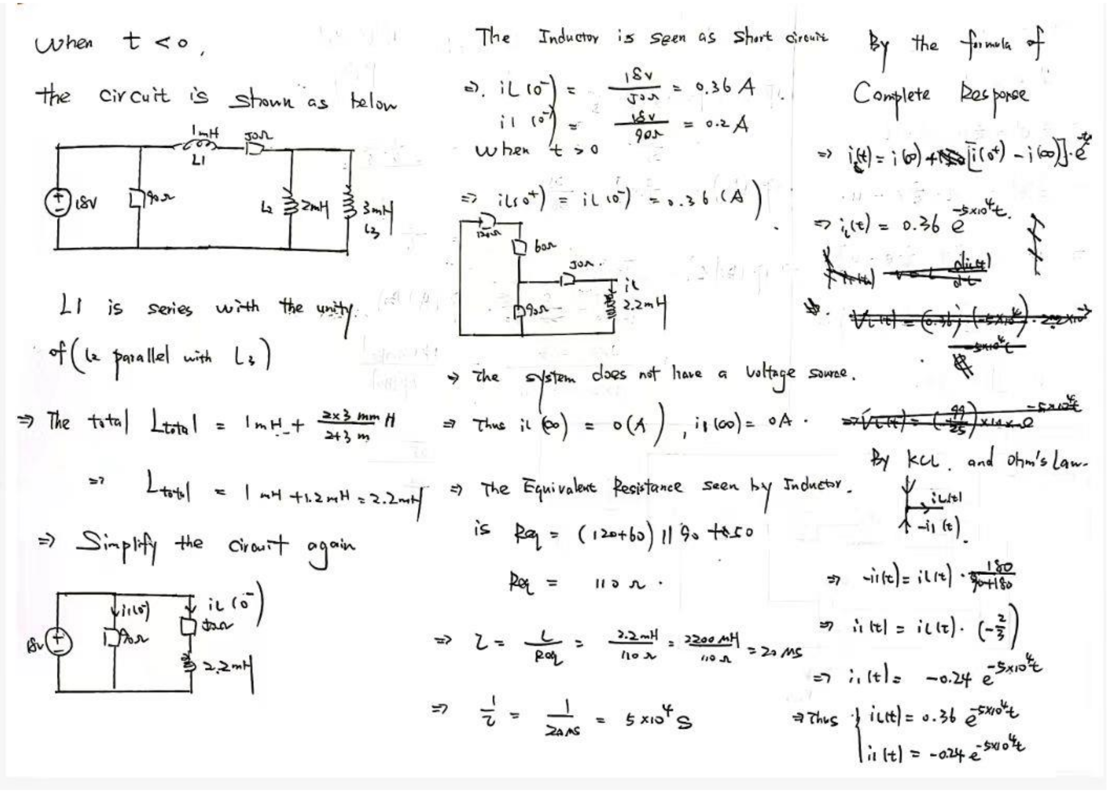
Question:

1) Derive the equations for $i_1(t)$ and $i_L(t)$ for all t.

2) Obtain the plots of $i_1(t)$ and $i_L(t)$ for all t.

Answer:

(1)



When $t < 0$,

the circuit is shown as below

1mH
50Ω
L1
18V
90Ω
L2
2mH
3mH
L3

L1 is series with the unity of (L2 parallel with L3)

$\Rightarrow$ The total $L_{total} = 1\,mH + \frac{2\times 3}{2+3}\,mH$

$\Rightarrow$ $L_{total} = 1\,mH + 1.2\,mH = 2.2\,mH$

$\Rightarrow$ Simplify the circuit again

18V
$i_1(0^-)$
90Ω
$i_L(0^-)$
50Ω
2.2mH

The Inductor is seen as short circuit

$\Rightarrow$ $i_L(0^-) = \frac{18V}{50\Omega} = 0.36\,A$

$i_1(0^-) = \frac{18V}{90\Omega} = 0.2\,A$

when $t > 0$

$\Rightarrow$ $i_L(0^+) = i_L(0^-) = 0.36\,(A)$

120Ω
60Ω
50Ω
90Ω
i_L
2.2mH

$\Rightarrow$ The system does not have a voltage source.

$\Rightarrow$ Thus $i_L(\infty) = 0\,(A)$, $i_1(\infty) = 0\,A$.

$\Rightarrow$ The Equivalent Resistance seen by Inductor.

is $R_{eq} = (120+60)\,||\,90 + 50$

$R_{eq} = 110\,\Omega$.

$\Rightarrow$ $\tau = \frac{L}{R_{eq}} = \frac{2.2\,mH}{110\,\Omega} = \frac{2200\,\mu H}{110\,\Omega} = 20\,\mu s$

$\Rightarrow$ $\frac{1}{\tau} = \frac{1}{20\,\mu s} = 5\times 10^4\,S$

By the formula of Complete Response

$\Rightarrow$ $i_L(t) = i(\infty) + [i(0^+) - i(\infty)]\cdot e^{-\frac{t}{\tau}}$

$\Rightarrow$ $i_L(t) = 0.36\,e^{-5\times10^4 t}$.

~~$v_L(t) = L\frac{di_L(t)}{dt}$~~

~~$V_L(t) = (0.36)\cdot(-5\times10^4)\cdot 2.2\times10^{-3}\,e^{-5\times10^4 t}$~~

~~$\Rightarrow V_L(t) = (-\frac{99}{25})\times 1\times e^{-5\times10^4 t}$~~

By KCL and Ohm's Law.

$\Rightarrow$ $-i_1(t) = i_L(t)\cdot\frac{180}{90+180}$

$\Rightarrow$ $i_1(t) = i_L(t)\cdot(-\frac{2}{3})$

$\Rightarrow$ $i_1(t) = -0.24\,e^{-5\times10^4 t}$

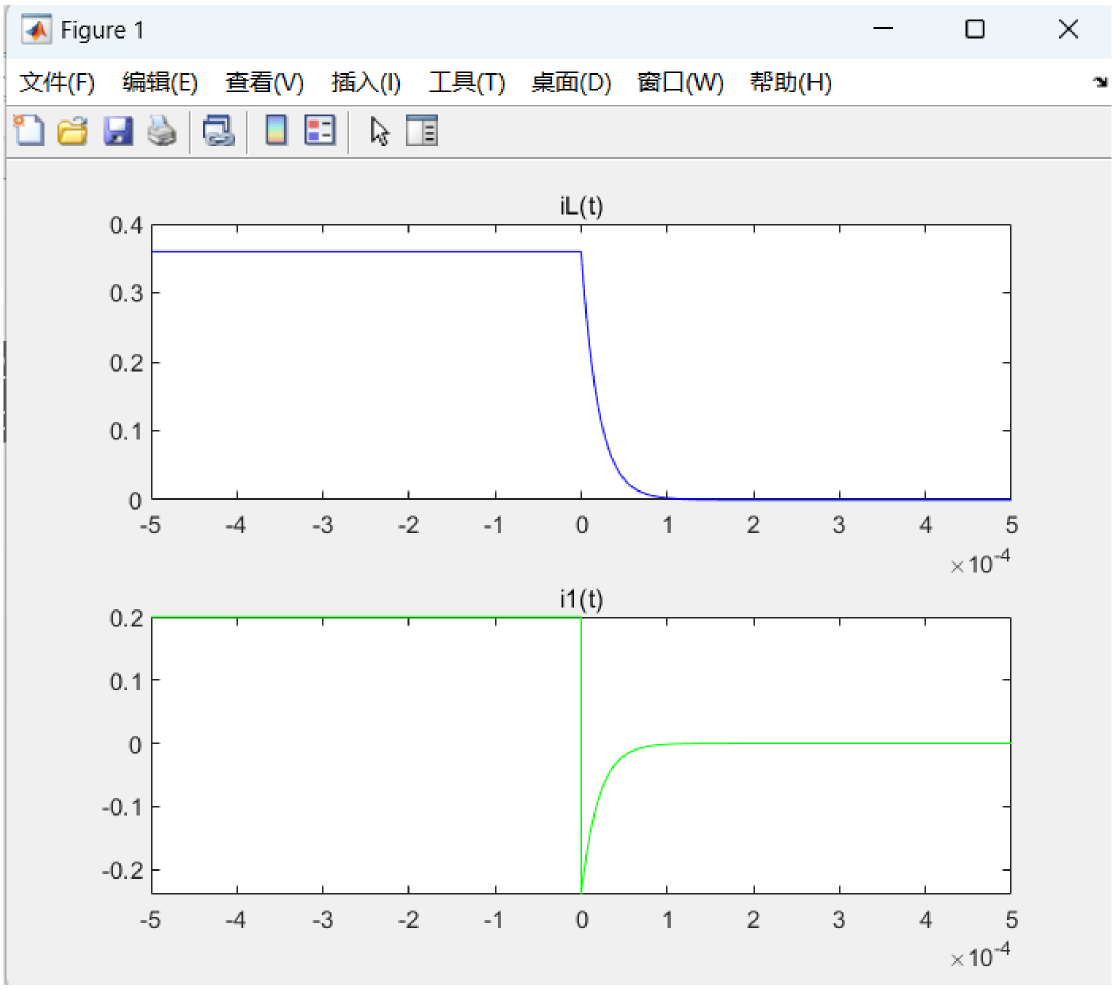
$\Rightarrow$ Thus $\begin{cases} i_L(t) = 0.36\,e^{-5\times10^4 t} \\ i_1(t) = -0.24\,e^{-5\times10^4 t} \end{cases}$

(2)

Part 1: Coding

```
t1 = -5e-4:1e-9:0;
iL1 = 0.36 + 0*t1;
i11 = 0.2 + 0*t1;
t2 = 0:1e-9:5e-4;
iL2 = 0.36 * exp(-5e4 * t2);
i12 = -0.24 * exp(-5e4 * t2);
t = [t1, t2];
iL = [iL1, iL2];
i1 = [i11, i12];
h4 = subplot(2,1,1);
plot(t, iL, 'b-');
title('iL(t)');
h5 = subplot(2,1,2);
plot(t, i1, 'g-');
title('i1(t)');
```

Part 2: Results and Display



Part3: Comments and analysis of the results

By analyzing the two cases which are "t<0" and "t>0", we are able to find the complete response of $i_L(t)$. We know that $VL(t) = L * di/dt$, and the $V_L(t)$ is also the voltage across the 90Ω. Thus, we can calculate the corresponding $i_1(t)$. After that, we move to the MATLAB, and attempt to put them into 1 figure by using the "subplot". To personalize each graph, I also use the handle to modify each graph.

Notice that we have to plot the graph both for "t<0" and "t>0" to meet the demand of all time in the question.

Problem 5 (20 Marks, with 1) 5 Mark, 2) 5 Marks, 3) 10 Marks)

1) In Simulink, generate a model that simulates 5sin(2pi * 60t) with time between 0 and 0.2 seconds and show the result using the scope of Simulink.

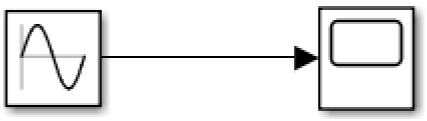
2) In Simulink, generate a model that simulates 2sin(50t) with time between 0 and 5 seconds and show the result using the scope of Simulink.

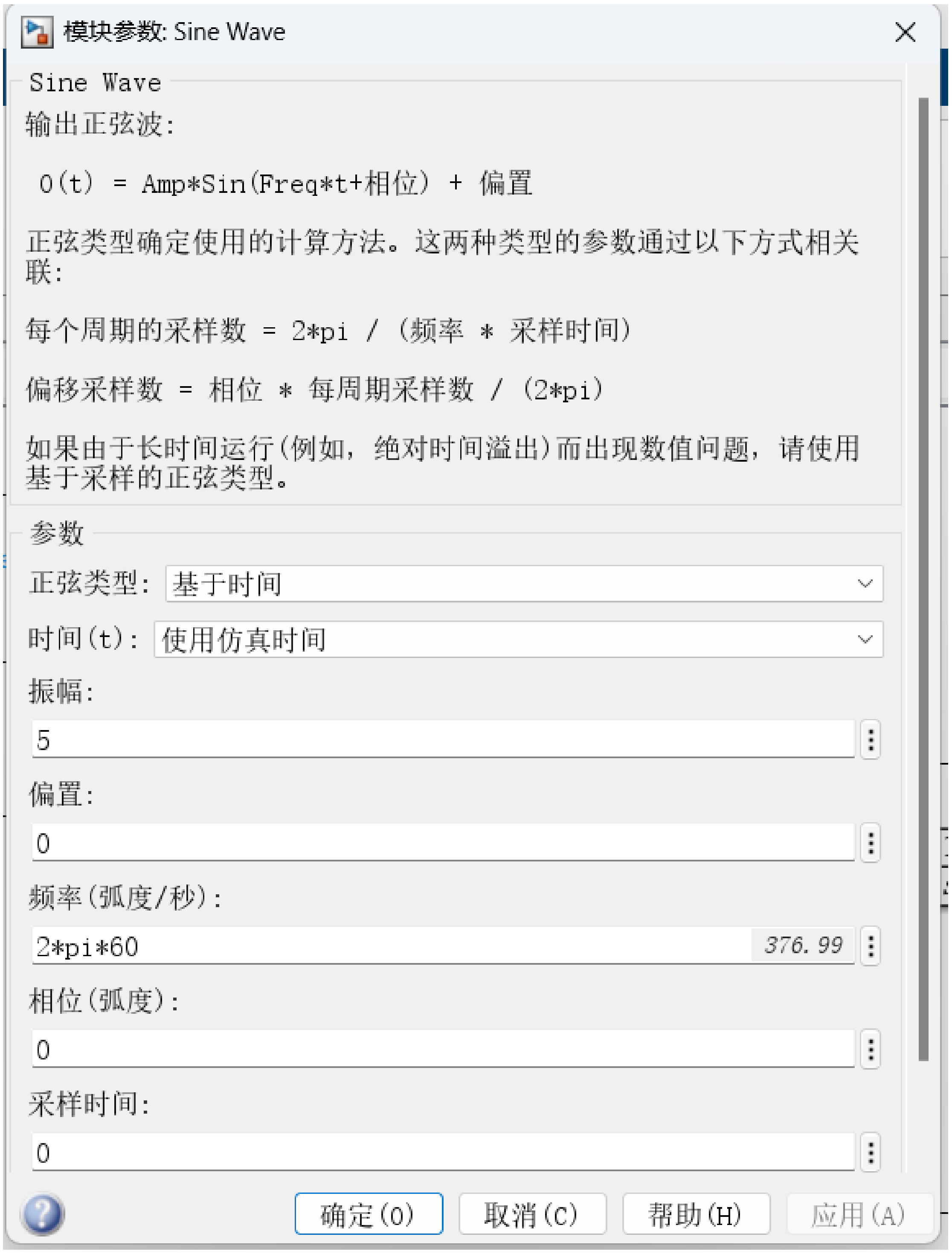
3) Using Simulink, generate a model that can solve $1/2u''+2/5u'+u=t$ with the initial conditions of u (0) =1 u' (0) = 2. Set the simulation time between 0 to 10 seconds and show the result using the scope of Simulink.

Answers:

(a)

Part 1: Parameters Setting and Results Display





模块参数: Sine Wave

Sine Wave

输出正弦波:

O(t) = Amp*Sin(Freq*t+相位) + 偏置

正弦类型确定使用的计算方法。这两种类型的参数通过以下方式相关联:

每个周期的采样数 = 2*pi / (频率 * 采样时间)

偏移采样数 = 相位 * 每周期采样数 / (2*pi)

如果由于长时间运行(例如, 绝对时间溢出)而出现数值问题, 请使用基于采样的正弦类型。

参数

正弦类型: 基于时间

时间(t): 使用仿真时间

振幅:

5

偏置:

0

频率(弧度/秒):

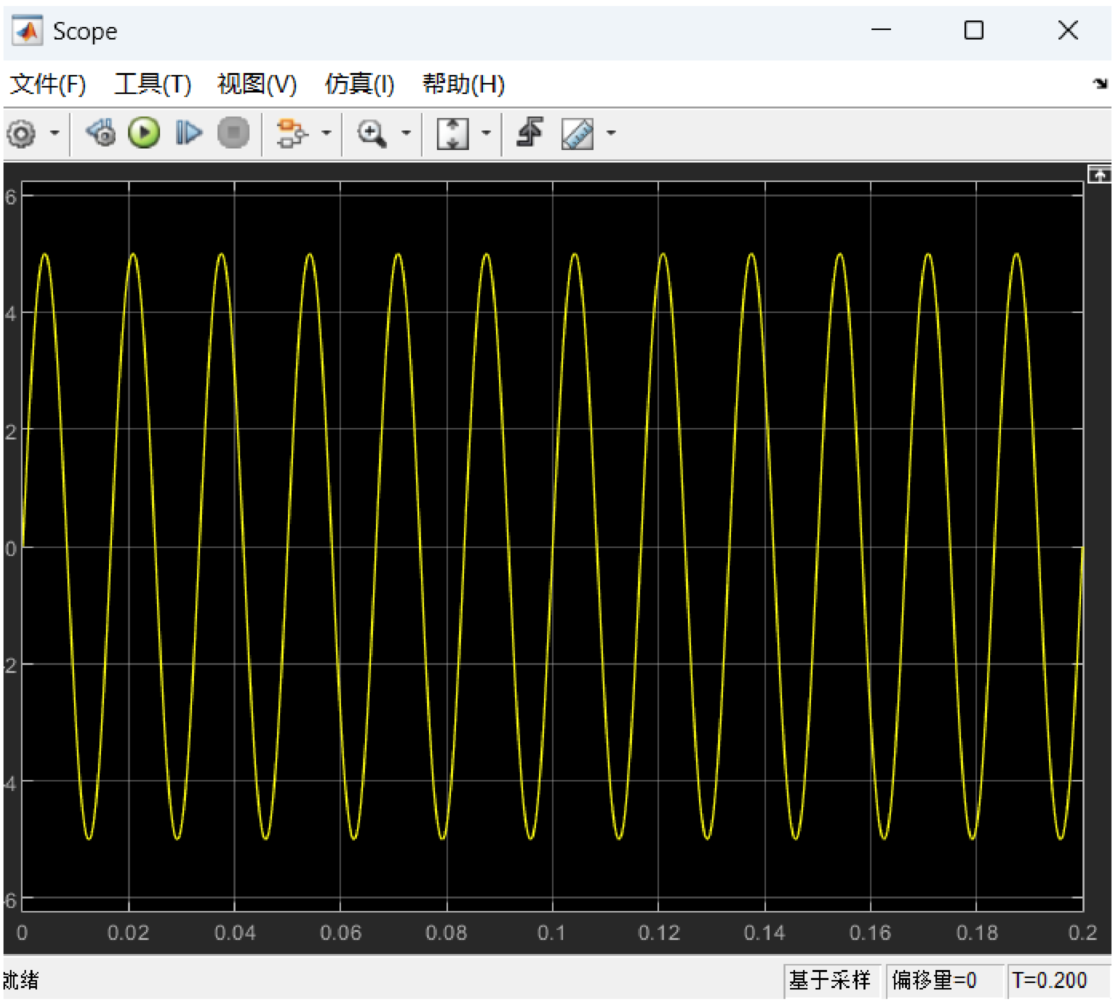
2*pi*60 376.99

相位(弧度):

0

采样时间:

0

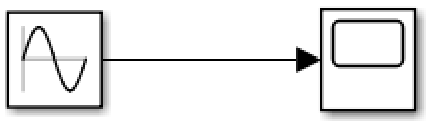


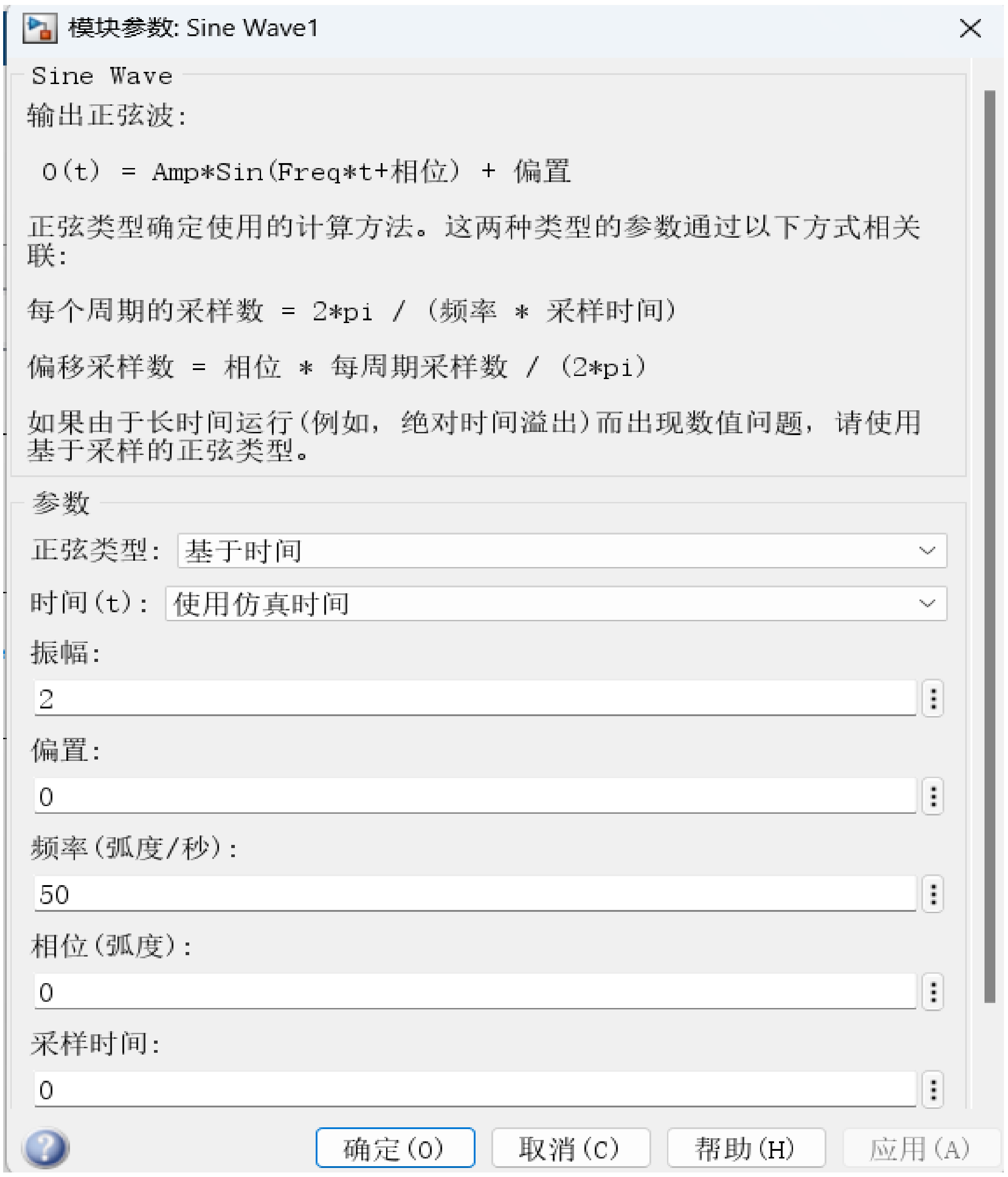
Part2: Comments and analysis of the results

I search the library for sine wave and scope first. Then, I set the parameters including the Amplitude, the phase, and the time duration. Finally, I click the "RUN" and scope sequentially, and caught the graph as above.

(b)

Part 1: Parameters Setting and Results Display




模块参数: Sine Wave1

Sine Wave

输出正弦波:

O(t) = Amp*Sin(Freq*t+相位) + 偏置

正弦类型确定使用的计算方法。这两种类型的参数通过以下方式相关联:

每个周期的采样数 = 2*pi / (频率 * 采样时间)

偏移采样数 = 相位 * 每周期采样数 / (2*pi)

如果由于长时间运行(例如, 绝对时间溢出)而出现数值问题, 请使用基于采样的正弦类型。

参数

正弦类型: 基于时间

时间(t): 使用仿真时间

振幅:

2

偏置:

0

频率(弧度/秒):

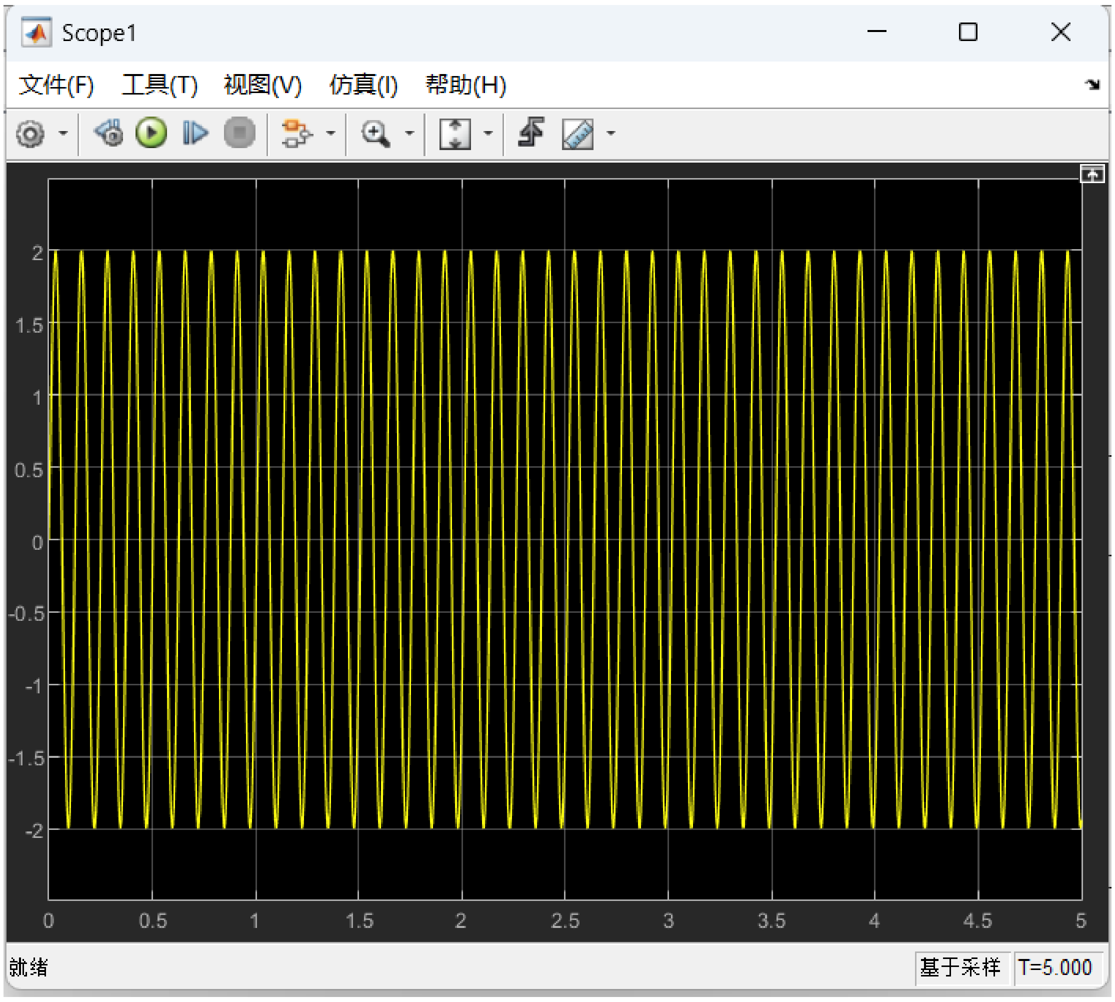
50

相位(弧度):

0

采样时间:

0

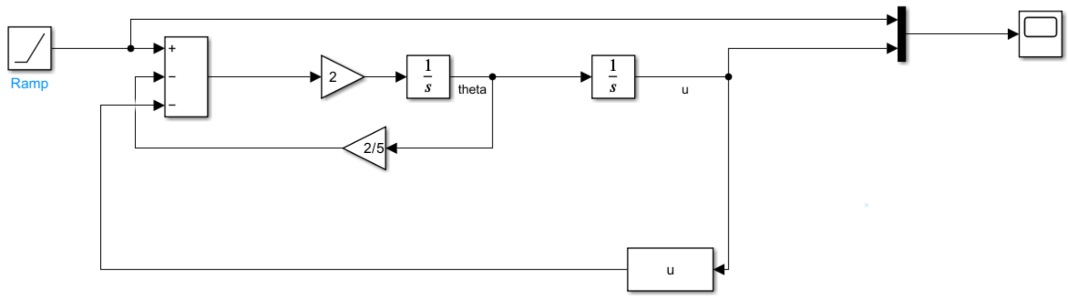


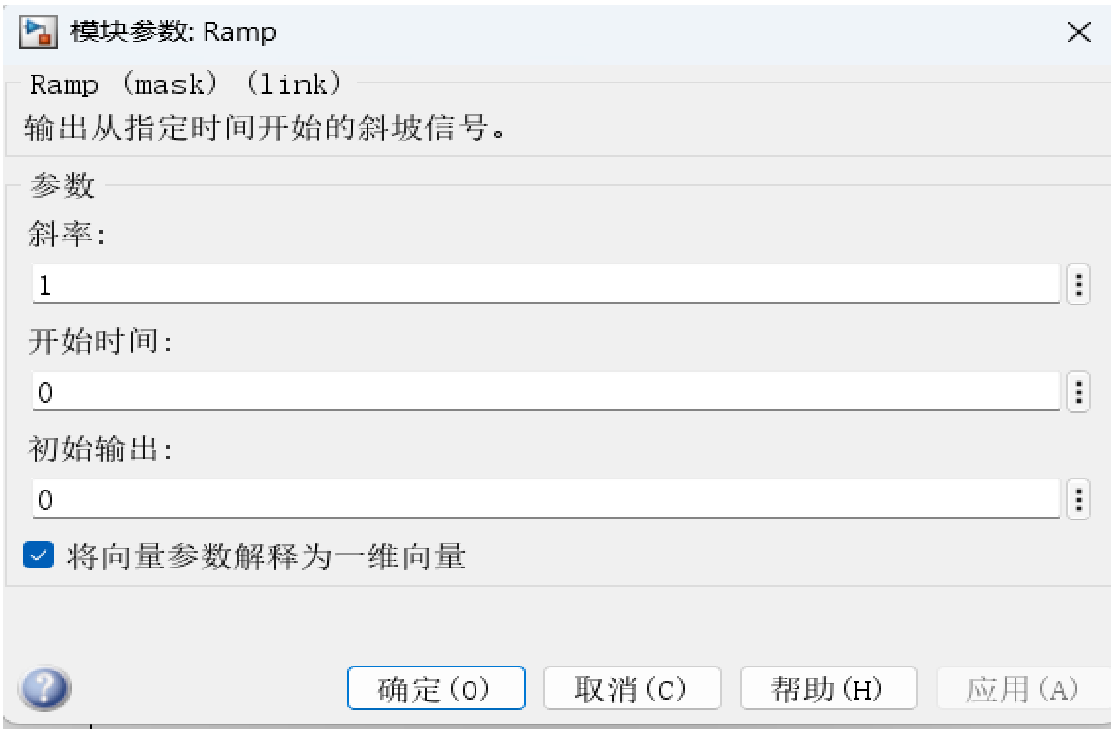
Part2: Comments and analysis of the results

The discrepancy between this question and the (a) only lies in the parameters-setting. So, I will not reclaim the same comments again.

(c)

Part 1: Parameters Setting and Results Display




模块参数: Ramp

Ramp (mask) (link)

输出从指定时间开始的斜坡信号。

参数

斜率:

1

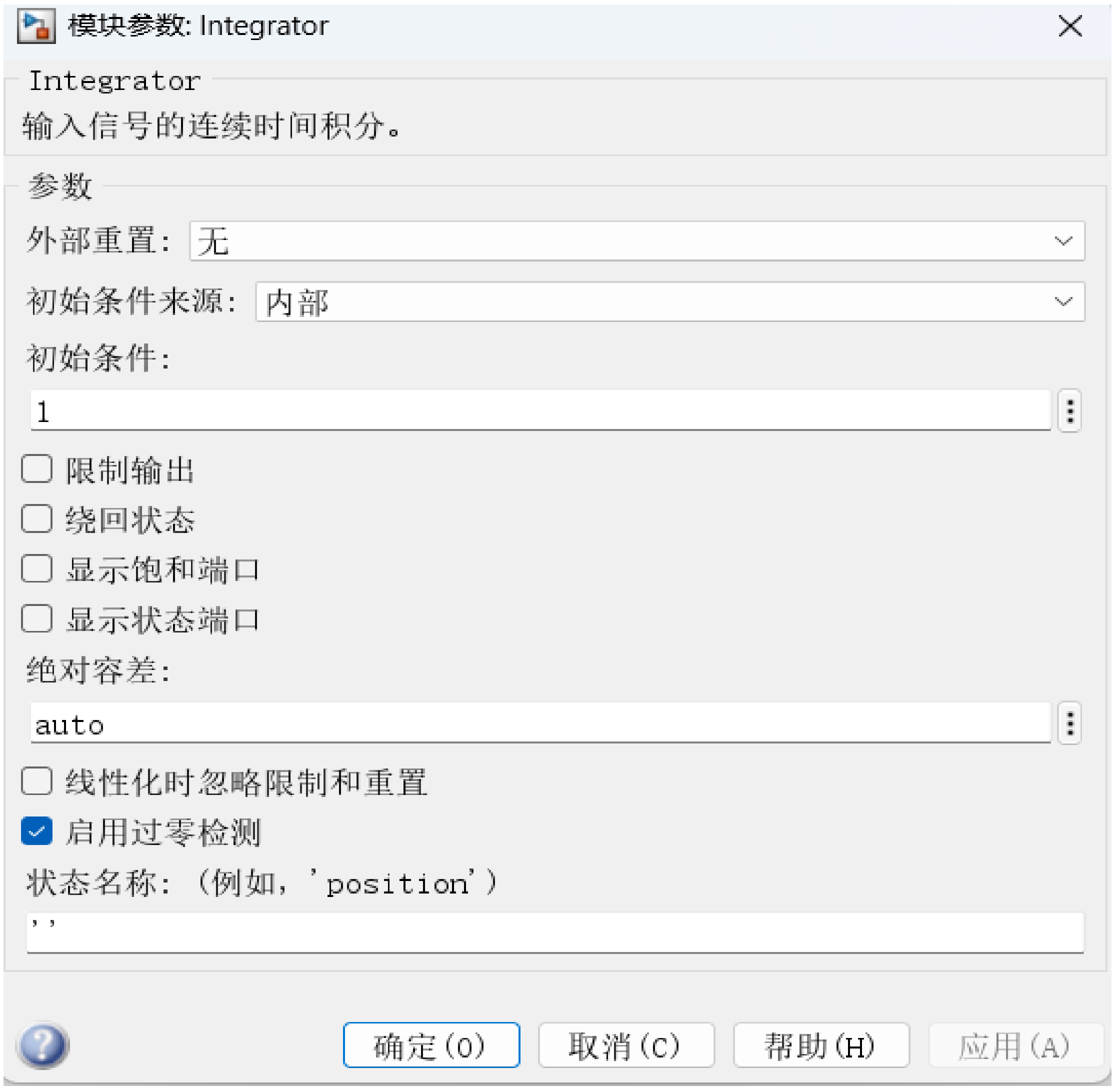
开始时间:

0

初始输出:

0

☑ 将向量参数解释为一维向量


模块参数: Integrator

Integrator

输入信号的连续时间积分。

参数

外部重置: 无

初始条件来源: 内部

初始条件:

1

☐ 限制输出

☐ 绕回状态

☐ 显示饱和端口

☐ 显示状态端口

绝对容差:

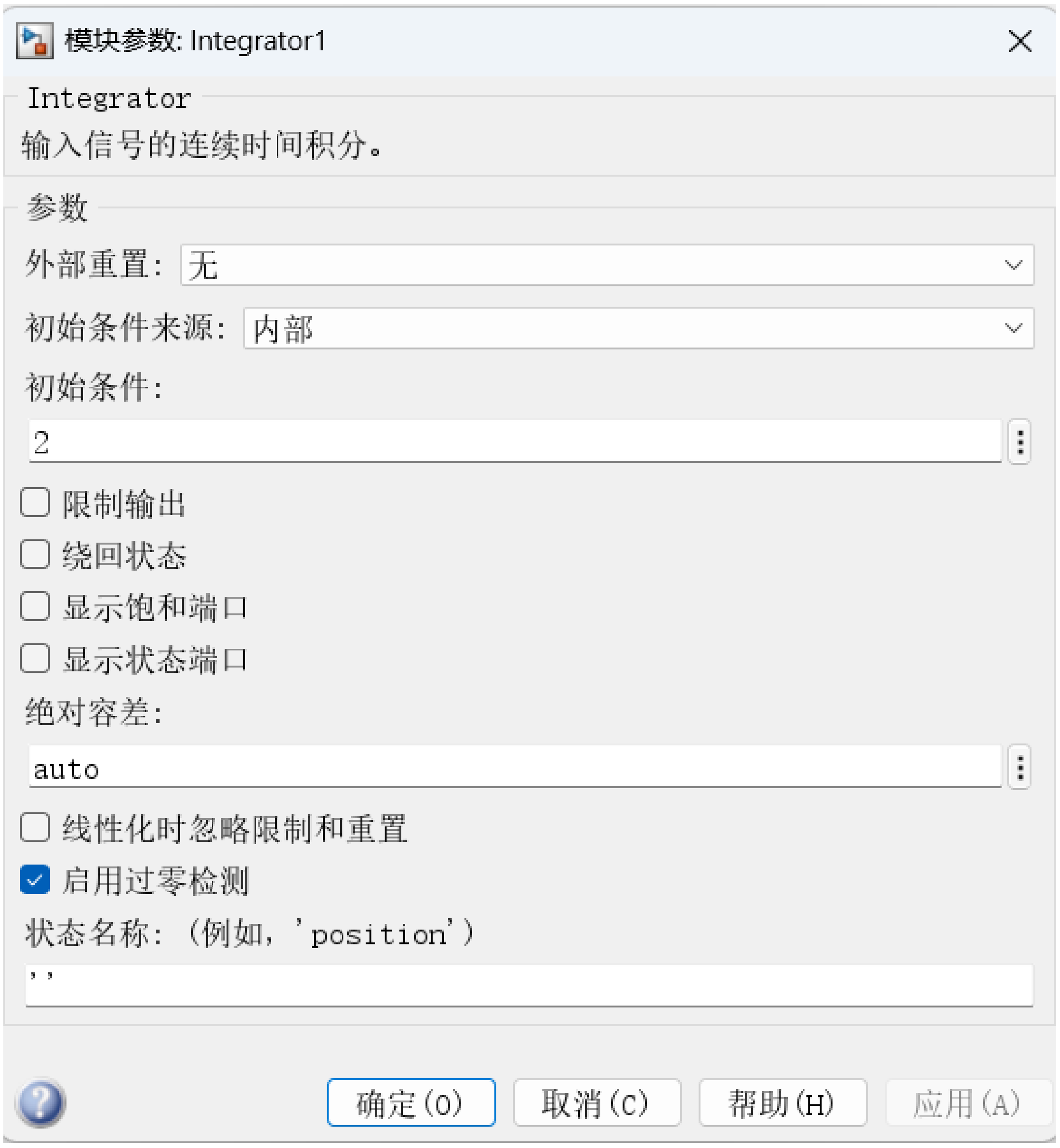
auto

☐ 线性化时忽略限制和重置

☑ 启用过零检测

状态名称: (例如, 'position')

''



Integrator

输入信号的连续时间积分。

参数

外部重置: 无

初始条件来源: 内部

初始条件:

2

- [ ] 限制输出
- [ ] 绕回状态
- [ ] 显示饱和端口
- [ ] 显示状态端口

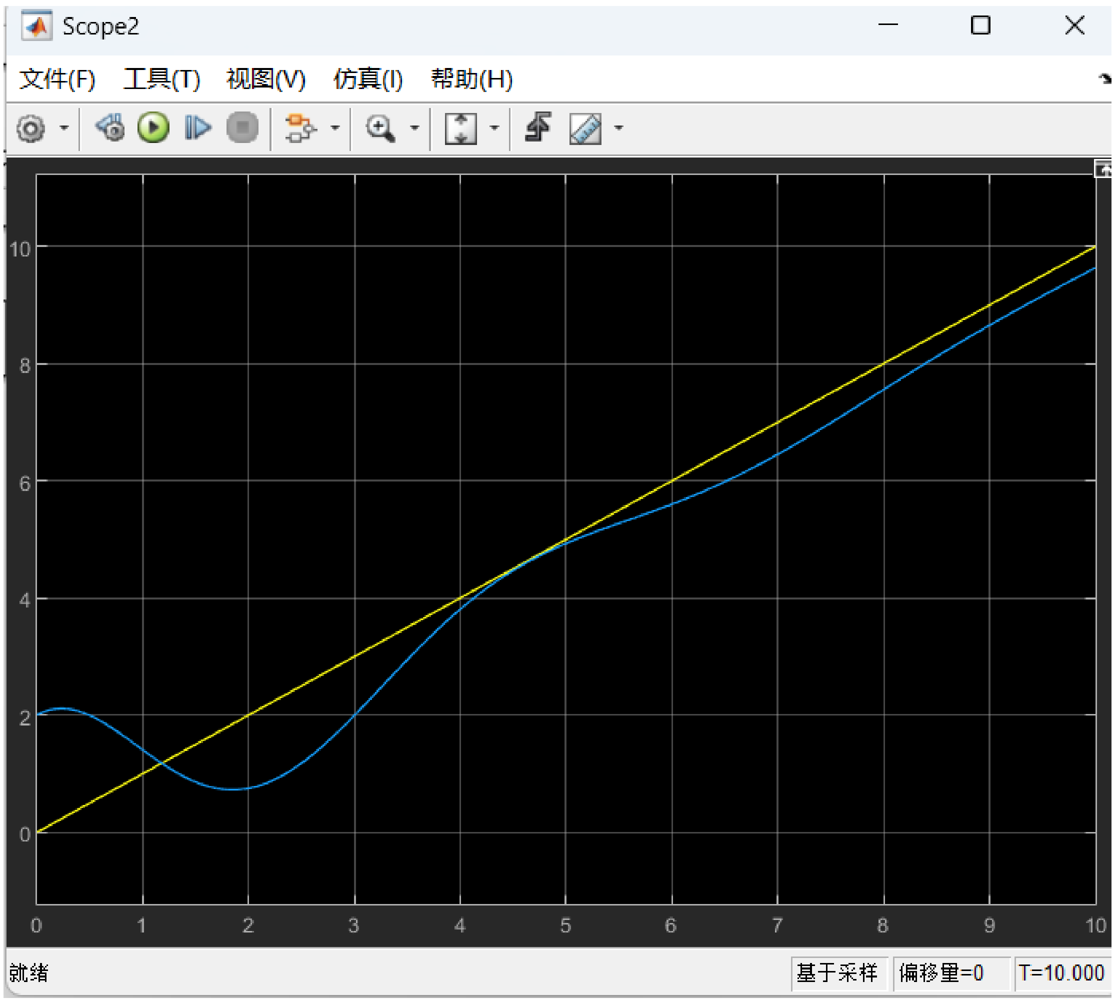
绝对容差:

auto

- [ ] 线性化时忽略限制和重置
- [x] 启用过零检测

状态名称: (例如, 'position')

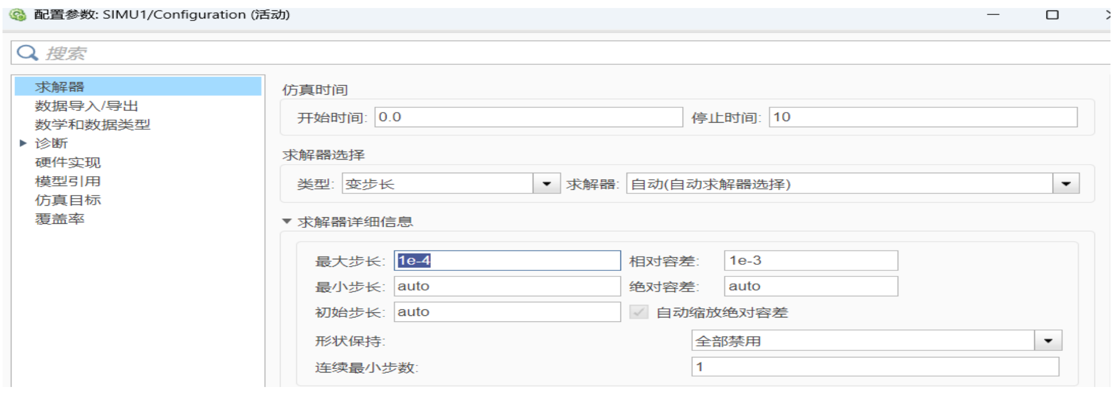
''



Part2: Comments and analysis of the results

Inspired by the PowerPoint of MATLAB-Tutorial 4 p59, I was attempting to solve this question by the emulation based on my understanding. Firstly, I split the second-order ODE into 2 parts which guaranteed that both of the two parts only contained the first derivatives and set $\theta = u'$. Next, by observing the ode [1/2u''+2/5u'+u=t], I simplify it into $\theta = \int \left(t - \frac{2}{5} * \theta - u\right) dt$ with $u = \int \theta\, dt$. After that, the "Add" module can be designed as one "+" linking "Signal Generator" and two "-" linking theta with gain of 2; linking u with Fcn "u", and u is the integral of θ. Ultimately, I finish the construction of the SIMULINK and achieve the results as I show above.

Notice that we have to modify the maximum step to 1e-4 to catch the correct graph as below.


配置参数: SIMU1/Configuration (活动)

搜索

求解器
数据导入/导出
数学和数据类型
诊断
硬件实现
模型引用
仿真目标
覆盖率

仿真时间

开始时间: 0.0　停止时间: 10

求解器选择

类型: 变步长　求解器: 自动(自动求解器选择)

求解器详细信息

最大步长: 1e-4　相对容差: 1e-3
最小步长: auto　绝对容差: auto
初始步长: auto　[✓] 自动缩放绝对容差
形状保持: 全部禁用
连续最小步数: 1