MATLAB 第 8 次作业参考答案

几点反馈：

- 作业的参考答案来自同学们的作业，只是稍作调整，为保护隐私，隐去姓名。
- 这一次作业存在三个普遍的问题。①没有意识到 BVP 可能存在不止一个解；②Richardson 外推又错了；③有限元法的推导。

第一题

示例 1：

homework8_1_1.m

(a) 有多个解，好多人都忽略了

```
clear
clc
u=linspace(-20,20);
for k=1:length(u)
    v(k)=Fun(u(k));
end
plot(u,v)
hold on
plot(u,zeros(length(u)),'--','Color','r')
hold off
xlabel('初始一阶导  s')
ylabel('F(s)')
title('选取初值的依据')

a1=-5;
b1=-4;
TOL=0.5e-6;
s1=Bisection(@Fun,a1,b1,TOL);
a2=-1;
b2=1;
s2=Bisection(@Fun,a2,b2,TOL);

f=@(t,y) [y(2);2*exp(-2*y(1)).*(1-t.^2)];
y1_0=[0;s1];
h=0.01;
t=0:h:1;
y1(:,1)=y1_0;
for i=1:length(t)-1
    y1(:,i+1)=y1(:,i)+h*f(t(i),y1(:,i));
end

y2_0=[0;s2];
y2(:,1)=y2_0;
for i=1:length(t)-1
```

```
    y2(:,i+1)=y2(:,i)+h*f(t(i),y2(:,i));
end

figure
hold on
plot(t,y1(1,:))
plot(t,y2(1,:))
xlabel('t')
ylabel('y')
title('solution')
Legend1=['s1 = ',num2str(s1)];
Legend2=['s2 = ',num2str(s2)];
legend(Legend1,Legend2)

function r=Fun(s)
    f=@(t,y) [y(2);2*exp(-2*y(1)).*(1-t.^2)];
    y0=[0;s];
    h=0.01;
    t=0:h:1;
    y(:,1)=y0;
    for i=1:length(t)-1
        y(:,i+1)=y(:,i)+h*f(t(i),y(:,i));
    end
    r=y(1,end)-log(2);
end

function x=Bisection(f,a,b,TOL)
    % f 是函数句柄，最好是匿名函数
    % a 是区间左端，b 是区间右端
    % TOL=0.5e-p  精确到小数点后 p 位
    %  检验 a 和 b 是否满足条件
    if(sign(f(a))==sign(f(b)))
        error('wrong input');
    end

    while (b-a)/2 > TOL
        c=(a+b)/2;
        fc=f(c);
        if(fc==0)
            break;
        end
        if sign(fc)==sign(f(b))
            b=c;
        else
```

```
            a=c;
        end
    end
    x=(a+b)/2;
end
```

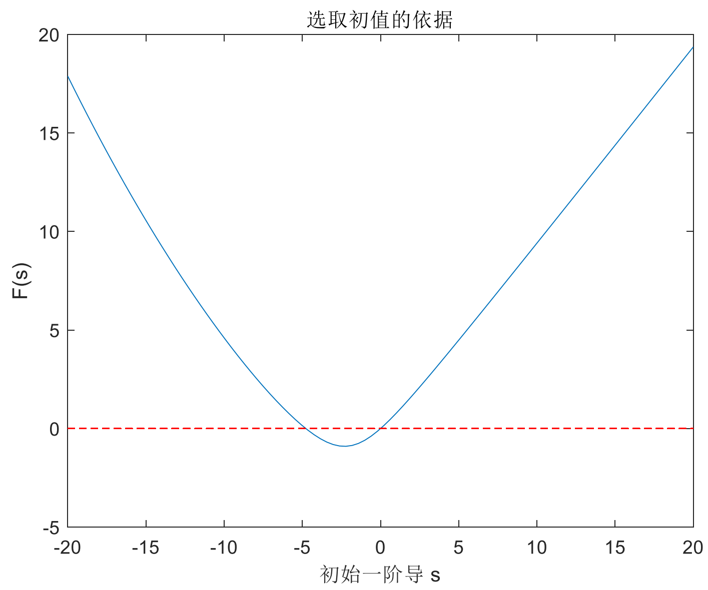
homework8_1_2.m

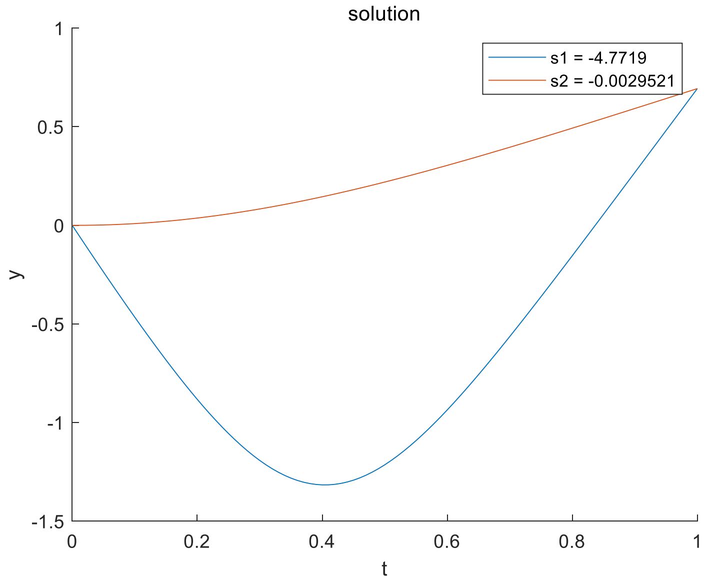
```
clear
clc

u=linspace(-20,20);
for k=1:length(u)
    v(k)=Fun(u(k));
end
plot(u,v)
hold on
plot(u,zeros(length(u)),'--','Color','r')
hold off
xlabel('初始一阶导  s')
ylabel('F(s)')
title('选取初值的依据')

s=fzero(@Fun,1);

f=@(t,y) [y(2);sin(y(2))];
y0=[1;s];
h=0.01;
t=0:h:1;
y(:,1)=y0;
for i=1:length(t)-1
    y(:,i+1)=y(:,i)+h*f(t(i),y(:,i));
end
figure
plot(t,y(1,:))
xlabel('t')
ylabel('y')
title('solution fzero')

figure
plot(t,y(1,:))

bcfcn=@(ya,yb) [ya(1)-1;yb(1)+1];
guess=@(x) [sin(x);sin(x)];
tmesh = linspace(0,1,1/h+1);
```

```
solinit = bvpinit(tmesh, guess);
sol = bvp4c(f, bcfcn, solinit);

hold on
Y=sol.y(1,:);
plot(sol.x,Y,'*','MarkerSize',2)
hold off
xlabel('t')
ylabel('y')
title('solution')
legend('fzero','bvp4c')

function r=Fun(s)
    f=@(t,y) [y(2);sin(y(2))];
    y0=[1;s];
    h=0.01;
    t=0:h:1;
    y(:,1)=y0;
    for i=1:length(t)-1
        y(:,i+1)=y(:,i)+h*f(t(i),y(:,i));
    end
    r=y(1,end)+1;
end
```

对于问题 a，首先画出 F(s)随 s 的变化关系图，发现有两个 s 值满足条件，因此，可以得到两组解。

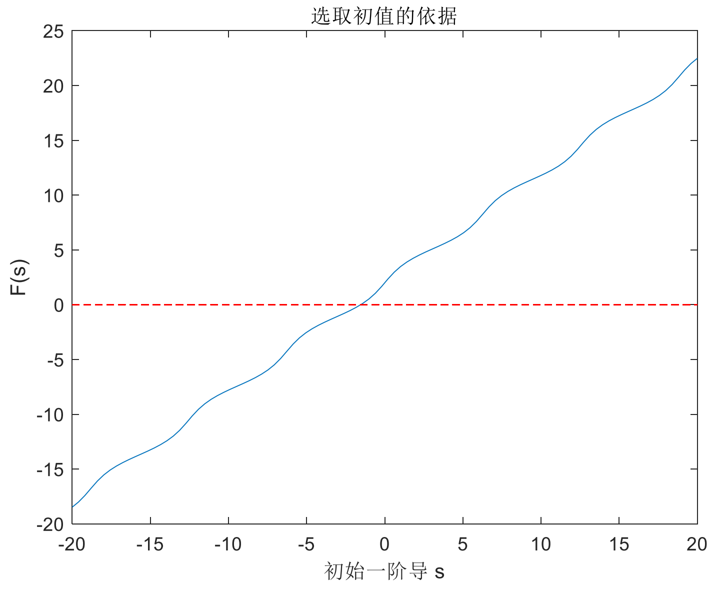
执行 **homework8_1_1.m**，得到如下结果：

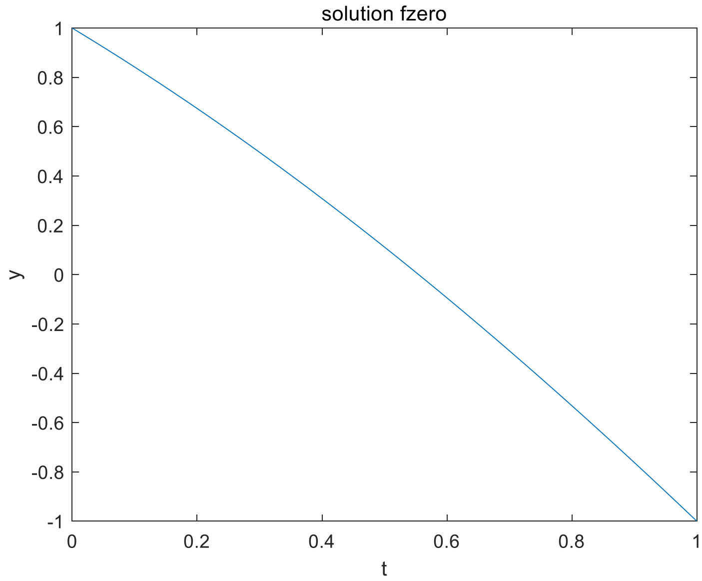


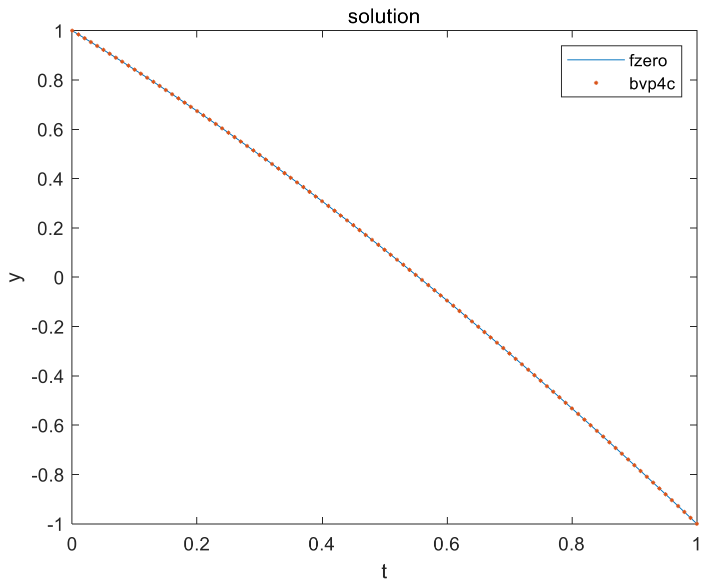


上图展示了得到的两组解，分别对应两个初始一阶导的值。

对于问题 b，同样首先画出 F(s)随 s 的变化关系图，作为选取初值的依据。执行 **homework8_1_2.m**，得到如下结果：







可见，使用 fzero 和 bvp4c 得到的结果非常接近，从图像上看不出差别。

示例 2：

（1）（a）令

$$y' = v$$

则问题转化为寻找 s，使得

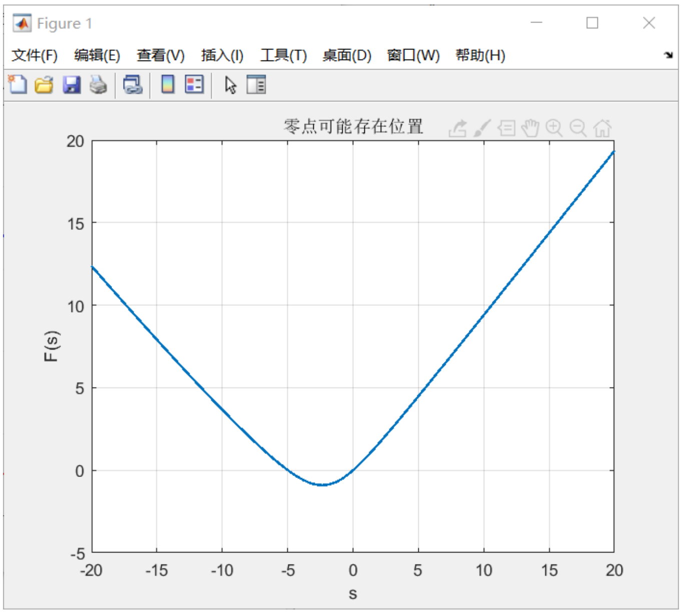
$$F(s) = \begin{cases} y' = v \\ v' = 2e^{-2y}(1-t^2) \\ y(0) = 0 \\ v(0) = s \end{cases}$$

$$F(s) = 0$$

此处采用四阶龙格库塔方法求解 IVP

先判断零点的位置：

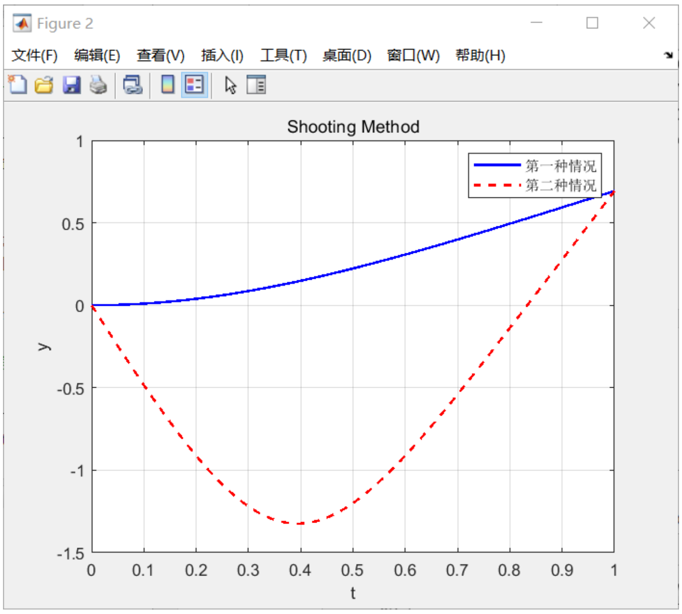
图像：



$$F(-1) = -0.5890$$
$$F(1) = 0.7729$$
$$F(-4) = -0.5056$$
$$F(-6) = 0.6407$$

二分法开始区间为$[-1\ 1]$和$[-6-4]$

图像：



代码：

函数 1（向量值四阶龙格库塔法）：[hw8_1\rungexl.m]

```
%向量值龙格库塔法
function [t,y]=rungexl(f,inter,t0,y0,h)
n=(inter(2)-inter(1))/h;
t(1)=t0;
y(1,:)=y0;
t=zeros(n+1,1);
```

```
for i=1:n
        t(i+1)=t(i)+h;
        k1(i,:)=f(t(i),y(i,:));
        k2(i,:)=f(t(i)+h/2,y(i,:)+h/2*k1(i,:));
        k3(i,:)=f(t(i)+h/2,y(i,:)+h/2*k2(i,:));
        k4(i,:)=f(t(i)+h,y(i,:)+h*k3(i,:));
        y(i+1,:)=y(i,:)+h/6*(k1(i,:)+2*k2(i,:)+2*k3(i,:)+k4(i,:));
end
end
```

函数 2（二分法函数）：hw8_1\BisectionMethod.m

```
%二分法实现
function xc=BisectionMethod(f,a,b,TOL)
while(b-a)>TOL
        c=(a+b)/2;
        fc=f(c);
        if(fc==0)
                break;
        end
        if sign(fc)==sign(f(b))
                b=c;
        else
                a=c;
        end
end
xc=(a+b)/2;
end
```

函数 3（F（s））：hw8_1\F.m

```
function z=F(s)
inter=[0 1];
t0=0;
yb=log(2);
y0=[0 s];
f=@(t,y)[y(2);2*exp(-2*y(1)).*(1-t^2)];
h=0.01;
[t,y]=rungexl(f,inter,t0,y0,h);
z=y(end,1)-yb;
end
```

主程序：hw8_1\Mhw8_1_1.m

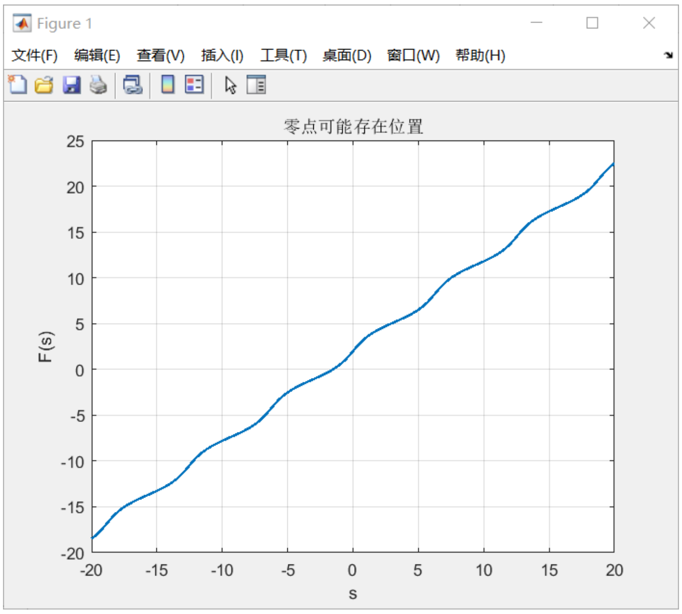
```
clear all;
clc;
addpath(genpath('.'));

s=-20:0.1:20;%先看零点的情况
for i=1:length(s)
```

```
    m(i)=F(s(i));
end
plot(s,m,'linewidth',1.5);
xlabel('s');
ylabel('F(s)');
title('零点可能存在位置');
grid on

a=-1;%第一种情况
b=1;
TOL=0.5e-6;%小数点后六位
s=BisectionMethod(@F,a,b,TOL);
inter=[0 1];
t0=0;
y0=[0 s];
f=@(t,y)[y(2);2*exp(-2*y(1)).*(1-t^2)];
h=0.01;
[t,y]=rungexl(f,inter,t0,y0,h);

a1=-6;%第二种情况
b1=-4;
TOL=0.5e-6;%小数点后六位
s1=BisectionMethod(@F,a1,b1,TOL);
inter=[0 1];
t01=0;
y01=[0 s1];
f=@(t,y)[y(2);2*exp(-2*y(1)).*(1-t^2)];
h=0.01;
[t1,y1]=rungexl(f,inter,t01,y01,h);

figure
plot(t,y(:,1),'b-','linewidth',1.5);
hold on
plot(t1,y1(:,1),'r--','linewidth',1.5);
grid on
xlabel('t');
ylabel('y');
title('Shooting Method');
legend('第一种情况','第二种情况')
axis([0 1 -1.5 1]);
rmpath(genpath('.'))
```

（b）同上

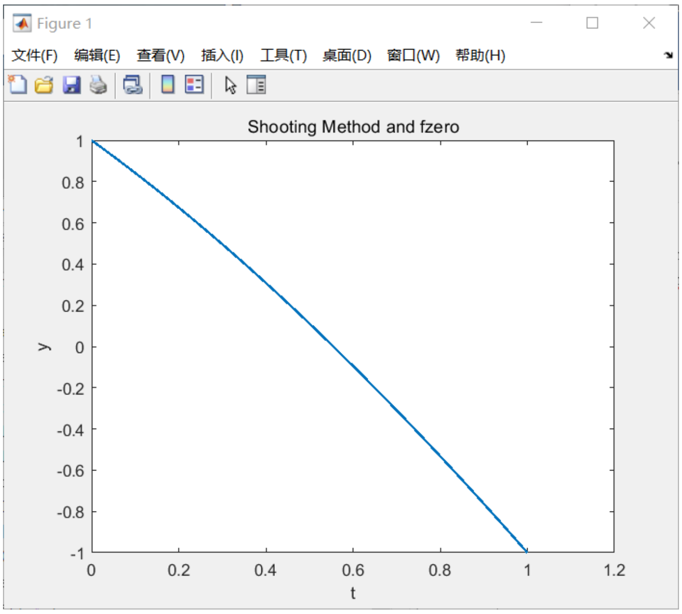
$$F_1(s)=\begin{cases} y'=v \\ v'=\sin v \\ y(0)=0 \\ v(0)=s \end{cases}$$

先判断零点存在位置：



$$F_1(-2)=-0.3785$$
$$F_1(-1)=0.5295$$

图像：



代码：

函数 1：（F1(s)）[hw8_1\F1.m](hw8_1\F1.m)

```
function z=F1(s)
inter=[0 1];
t0=0;
yb=-1;
```

```
y0=[1 s];
f=@(t,y)[y(2);sin(y(2))];
h=0.01;
[t,y]=rungexl(f,inter,t0,y0,h);
z=y(end,1)-yb;
end
```

函数 2：（龙格库塔向量）hw8_1\rungexl.m

```
function [t,y]=rungexl(f,inter,t0,y0,h)
n=(inter(2)-inter(1))/h;
t(1)=t0;
y(1,:)=y0;
t=zeros(n+1,1);
for i=1:n
    t(i+1)=t(i)+h;
    k1(i,:)=f(t(i),y(i,:));
    k2(i,:)=f(t(i)+h/2,y(i,:)+h/2*k1(i,:));
    k3(i,:)=f(t(i)+h/2,y(i,:)+h/2*k2(i,:));
    k4(i,:)=f(t(i)+h,y(i,:)+h*k3(i,:));
    y(i+1,:)=y(i,:)+h/6*(k1(i,:)+2*k2(i,:)+2*k3(i,:)+k4(i,:));
end
end
```

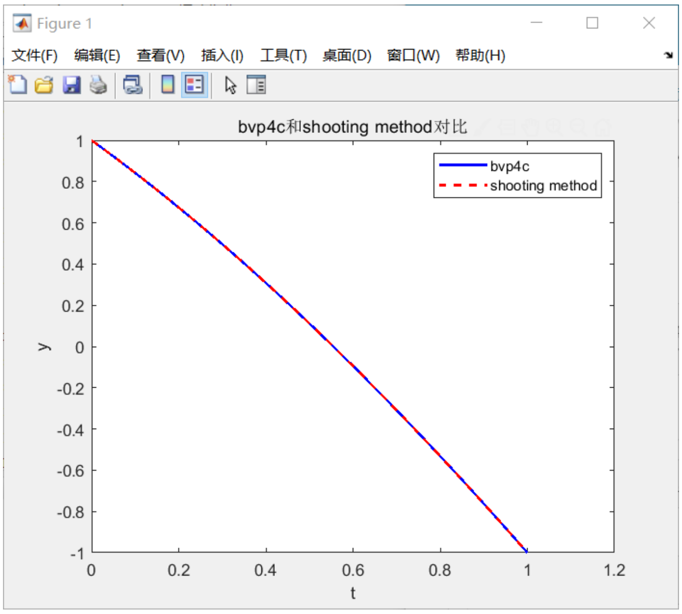
主程序：hw8_1\Mhw8_1_2.m

```
clear all;
clc;
addpath(genpath('.'));
s=fzero(@F1,[-2 -1]);
inter=[0 1];
t0=0;
yb=-1;
y0=[1 s];
f=@(t,y)[y(2);sin(y(2))];
h=0.01;
[t,y]=rungexl(f,inter,t0,y0,h);
plot(t,y(:,1),'linewidth',1.5);
xlabel('t');
ylabel('y');
title('Shooting Method and fzero');
rmpath(genpath('.'))
```

（2）使用 bvp4c 时需要将微分方程重写为包含两个一阶 ODE 的方程组，即

$$y' = v$$
$$v' = \sin v$$
$$y(0) = 0$$
$$v(0) = s$$

图像：



可以比较看到，bvp4c 和 shooting method 得到的图像几乎是重叠的，查询 bvp4c 的帮助文档后，看到其算法为：bvp4c 是一个有限差分代码，此代码实现 3 阶段 Lobatto IIIa 公式。这是配置公式，并且配置多项式会提供在整个 [a,b] 中具有一致四阶精度的C^1连续解。网格选择和误差控制均基于连续解的残差。

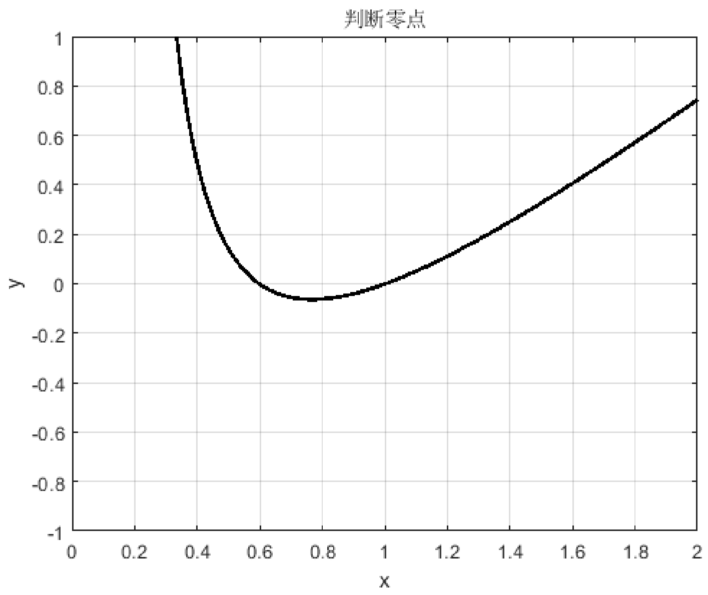
代码如下：

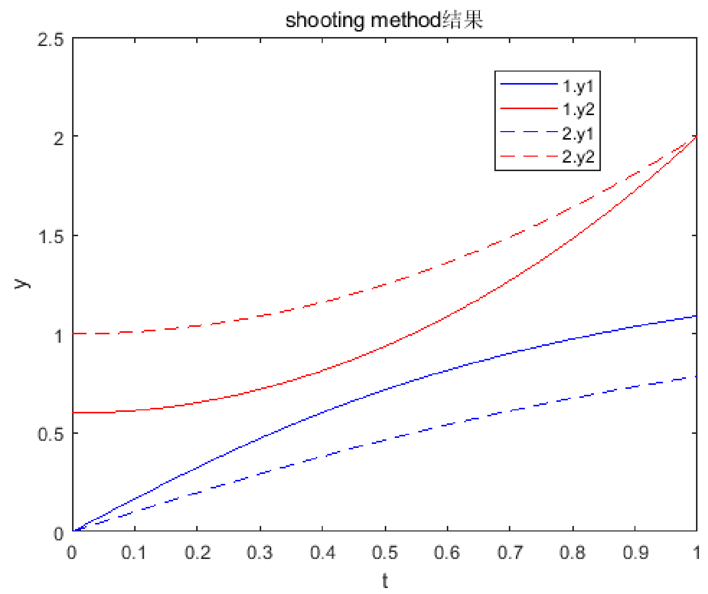
主程序：hw8_1\Mhw8_2_1.m

```
clear all,clc;
addpath(genpath('.'));
s=fzero(@F1,[-2 -1]);
inter=[0 1];
t0=0;yb=-1;y0=[1 s];
f=@(t,y)[y(2);sin(y(2))];
res=@(ya,yb)[ya(1)-1;yb(1)+1];
solinit=bvpinit(linspace(0,1,5),[1 0]);
sol=bvp4c(f,res,solinit);
x1=linspace(0,1,1000);
y1=deval(sol,x1);
plot(x1,y1(1,:),'b-','linewidth',1.5);
hold on
h=0.01;
[t,y]=rungexl(f,inter,t0,y0,h);
plot(t,y(:,1),'r--','linewidth',1.5);
xlabel('t');
ylabel('y');
title('bvp4c 和 shooting method 对比');
legend('bvp4c','shooting method')
rmpath(genpath('.'))
```

第二题

示例 1：





此处也有两个解

work2.m

```
function work2
h = 0.01;
tspan = linspace(0,1,1/0.01+1);
w = zeros(1/0.01+1,2);%y1,y2
%判断 0 点
x0 = 0:0.01:2;
y0 = zeros(1,length(x0));
figure(1)
for i = 1:1:length(x0)
    w = pro(tspan,w,x0(i));
    y0(i) = w(end,2)-2;
```

```
end
plot(x0,y0,'-k','LineWidth',2)%零点在-6 -4  以及  -1 1 之间
grid on
title('判断零点');
xlabel('x')
ylabel('y')
axis([0 2 -1 1])
figure(2)
a = 0;b = 0.8;%零点 0.6 1
tol = 1e-6;
w = pro(tspan,w,a);
af = w(end,2)-2;
w = pro(tspan,w,b);
bf = w(end,2)-2;
while (b-a)/2 > tol
        w = pro(tspan,w,a);
        af = w(end,2)-2;
        w = pro(tspan,w,b);
        bf = w(end,1)-log(2);
        c = a/2 + b/2;
        w = pro(tspan,w,c);
        cf = w(end,2)-2;
        if cf == 0
            break
        elseif cf * af < 0
            b = c;
        else
            a = c;
        end
end
plot(tspan,w(:,1),'b-');hold on
plot(tspan,w(:,2),'r-');hold on
a = 0.8;b = 1.2;%零点 0.6 1
tol = 1e-6;
w = pro(tspan,w,a);
af = w(end,2)-2;
w = pro(tspan,w,b);
bf = w(end,2)-2;
while (b-a)/2 > tol
        w = pro(tspan,w,a);
        af = w(end,2)-2;
        w = pro(tspan,w,b);
        bf = w(end,1)-log(2);
        c = a/2 + b/2;
```

```
        w = pro(tspan,w,c);
        cf = w(end,2)-2;
        if cf == 0
            break
        elseif cf * af < 0
            b = c;
        else
            a = c;
        end
end
plot(tspan,w(:,1),'b--');hold on
plot(tspan,w(:,2),'r--');
legend('1.y1','1.y2','2.y1','2.y2')
title('shooting method 结果');
xlabel('t')
ylabel('y')

function y = pro(tspan,w,x)
w(1,:) = [0 x];
for i = 1:1:length(w)-1
    w(i+1,:) = rk(tspan(i),w(i,:));
end
y = w;
function z = rk(t,w)
z = zeros(1,2);
h = 0.01;
k1 = f(t,w);
k2 = f(t+h/2,w+h*k1/2);
k3 = f(t+h/2,w+h*k2/2);
k4 = f(t+h,w+h*k3);
z = w + h*(k1+2*k2+2*k3+k4)/6;
function k = f(t,w)
h = 0.01;
k = zeros(1,2);
k(1) = 1/w(2);
k(2) = t+tan(w(1));
```

示例 2：

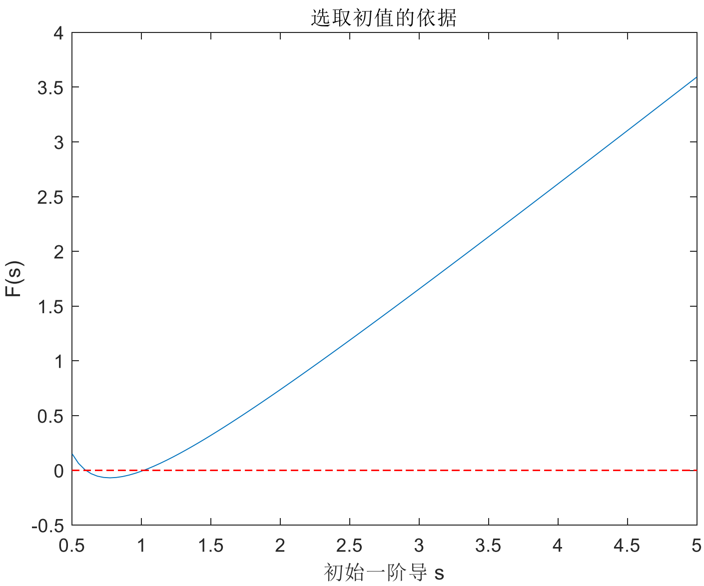
homework8_2.m

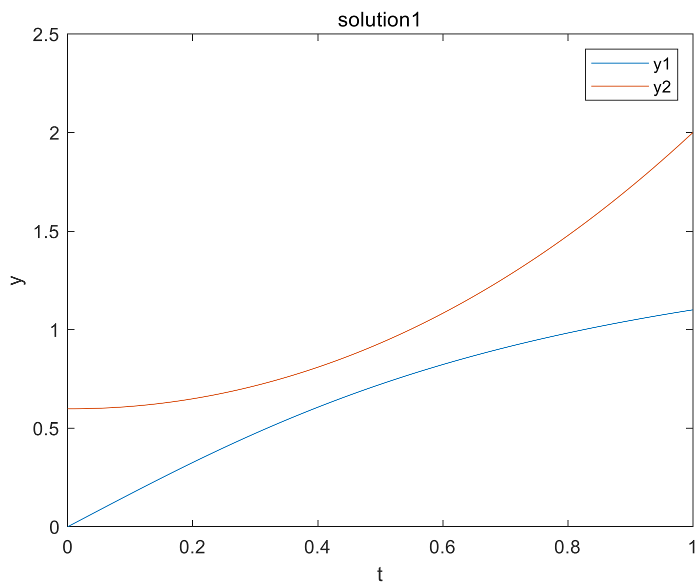
```
clear
clc

u=linspace(0.5,5);
for k=1:length(u)
```

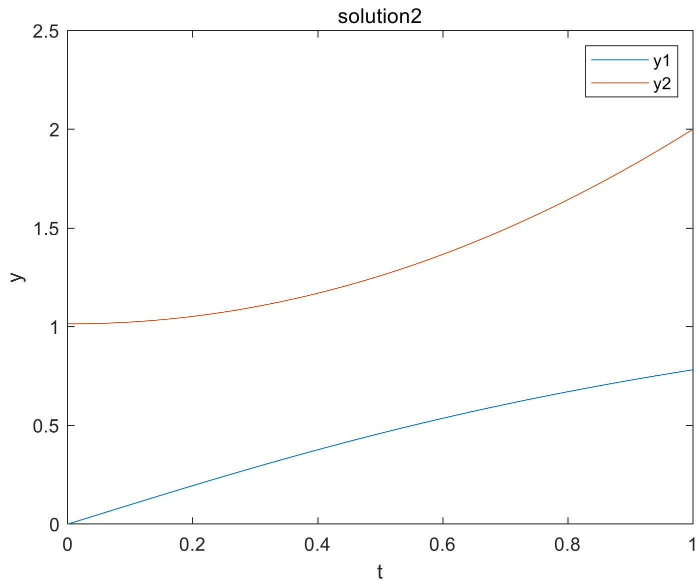
```
    v(k)=Fun(u(k));
end
plot(u,v)
hold on
plot(u,zeros(length(u)),'--','Color','r')
hold off
xlabel('初始一阶导  s')
ylabel('F(s)')
title('选取初值的依据')

figure
opt=optimset('Display','off');

s1=fsolve(@Fun,0.6,opt);
s2=fsolve(@Fun,1,opt);

f=@(t,y) [1./y(2);t+tan(y(1))];
h=0.01;
t=0:h:1;

y1_0=[0;s1];
y1(:,1)=y1_0;
for i=1:length(t)-1
    y1(:,i+1)=y1(:,i)+h*f(t(i),y1(:,i));
end

y2_0=[0;s2];
y2(:,1)=y2_0;
for i=1:length(t)-1
    y2(:,i+1)=y2(:,i)+h*f(t(i),y2(:,i));
end

plot(t,y1)
legend('y1','y2')
xlabel('t')
ylabel('y')
title('solution1')

figure
plot(t,y2)
legend('y1','y2')
xlabel('t')
ylabel('y')
title('solution2')
```

```
function r=Fun(s)
    f=@(t,y) [1./y(2);t+tan(y(1))];
    y0=[0;s]; y(:,1)=y0;
    h=0.01; t=0:h:1;
    for i=1:length(t)-1
        y(:,i+1)=y(:,i)+h*f(t(i),y(:,i));
    end
    r=y(2,end)-2;
end
```

首先画出 F(s)和 s 的关系图，作为选取初值的依据，发现有两个 s 值满足条件。
执行 homework8_2.m，得到如下结果：






solution2
y1
y2
y
t
2.5
2
1.5
1
0.5
0
0
0.2
0.4
0.6
0.8
1

第三题

示例 1：

(1)

$$y'' = 4y$$

其特征根为：$r^2 - 4 = 0, r = 2, r = -2;$

故其通解为

$$y_2 = c_1 e^{2t} + c_2 e^{-2t}$$

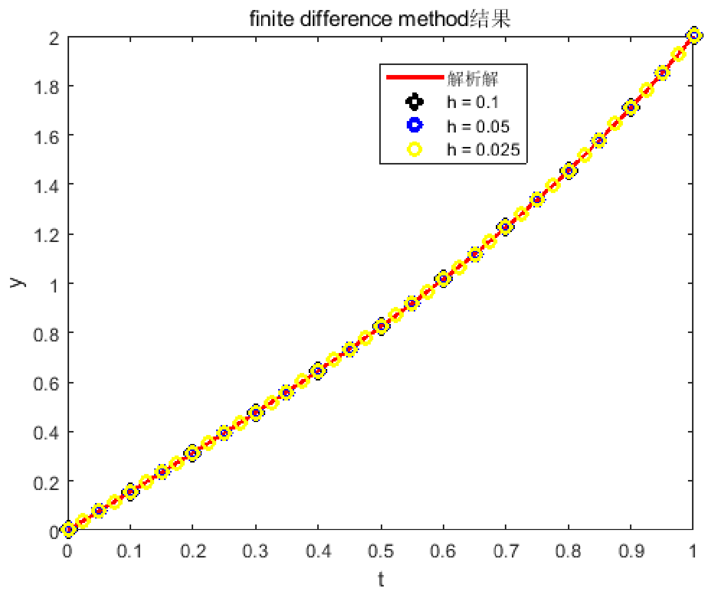
故原非齐次线性微分方程的解为：

$$y = y_1 + y_2 = c_1 e^{2t} + c_2 e^{-2t} + t$$

代入边界条件得

$$y = y_1 + y_2 = \frac{1}{e^2 - e^{-2}} e^{2t} - \frac{1}{e^2 - e^{-2}} e^{-2t} + t$$

(2)



work3.m

```
function work3
%解析解
xspan = linspace(0,1,1/0.025+1);
y = exp(2*xspan)/(exp(2)-exp(-2)) + exp(-2*xspan)/(exp(-2)-exp(2)) + xspan;
plot(xspan,y,'r-','LineWidth',2);hold on
fprintf('解析解,y(0.5) = %12.10f\n',y(0.5/0.025+1));
%finite difference
%y'' = 4y-4t
hselect = [0.1 0.05 0.025];
for k = 1:1:length(hselect)
    h = hselect(k);
    t = linspace(0,1,1/h+1);
    w = zeros(size(t));
    w(1) = 0;w(end) = 2;
    n = length(t)-2;
```

```
        q = zeros(n,n);
        p = zeros(n,1);
        q(1,1) = -2-4*h^2;q(1,2) = 1;p(1) = -4*h^2*t(2);
        q(n,n) = -2-4*h^2;q(n,n-1) = 1;p(n) = -2-4*h^2*t(n+1);
        for i = 2:1:n-1
            q(i,i-1) = 1;
            q(i,i) = -2-4*h^2;
            q(i,i+1) = 1;
            p(i) = -4*h^2*t(i+1);
        end
        w(2:end-1) = q\p;
        switch k
            case 1
                plot(t,w,'ok','LineWidth',4);hold on
                fprintf('h = 0.1 时,y(0.5) = %12.10f\n',w(0.5/h+1));
            case 2
                plot(t,w,'ob','LineWidth',3);hold on
                fprintf('h = 0.05 时,y(0.5) = %12.10f\n',w(0.5/h+1));
            case 3
                plot(t,w,'oy','LineWidth',2);
                fprintf('h = 0.025 时,y(0.5) = %12.10f\n',w(0.5/h+1));
        end
end
legend('解析解','h = 0.1','h = 0.05','h = 0.025')
title('finite difference method 结果');
xlabel('t')
ylabel('y')
```

(3)

```
解析解,y(0.5) = 0.8240271368
h = 0.1时,y(0.5) = 0.8244366633
h = 0.05时,y(0.5) = 0.8241298496
h = 0.025时,y(0.5) = 0.8240528358
```

这里的外推直接外推和代回二阶导再外推结果是一样的，所以都算是对的。
但是大家对外推的理解可能还不是太透彻，第二步的外推也是错了很多，希望大家重新回顾一下。

根据 richardson extrapolation 有

$F_2(h) = 0.8244366633$		
$F_2\left(\frac{h}{2}\right) = 0.8241298496$	$F_4(h) = 0.8240275784$	
$F_2\left(\frac{h}{4}\right) = 0.8240528358$	$F_4(h/2) = 0.8240271645$	$F_6(h) = 0.8240271369$

计算

$$F_4(h) = \frac{4F_2\left(\frac{h}{2}\right) - F_2(h)}{3} = 0.8240275784$$

$$F_4(h/2) = \frac{4F_2\left(\frac{h}{4}\right) - F_2\left(\frac{h}{2}\right)}{3} = 0.8240271645$$

$$F_6(h) = \frac{2^4 F_4\left(\frac{h}{2}\right) - F_4(h)}{2^4 - 1} = 0.8240271369$$

示例 2：

homework8_3.m

```
clear all
clc

syms t y(t)
y=dsolve(diff(y,2)==4*(y-t),y(0)==0,y(1)==2);
pretty(y)
Y=matlabFunction(y);

u=linspace(0,1,10000);
v=Y(u);
plot(u,v)
hold on

h1=0.1;
t1=0:h1:1;
n1=length(t1)-2;
A1=zeros(n1);
A1(1,1)=-(4*h1^2+2);
A1(1,2)=1;
A1(n1,n1-1)=1;
A1(n1,n1)=-(4*h1^2+2);
b1=zeros(n1,1);
b1(1)=-4*h1^2*t1(2);
b1(end)=-4*h1^2*t1(end-1)-2;
for k=2:n1-1
    A1(k,k-1)=1;
    A1(k,k)=-(4*h1^2+2);
    A1(k,k+1)=1;
    b1(k)=-4*h1^2*t1(k+1);
end
y1=transpose(A1\b1);
y1=[0,y1,2];
plot(t1,y1,'*','MarkerSize',8)

h2=0.05;
```

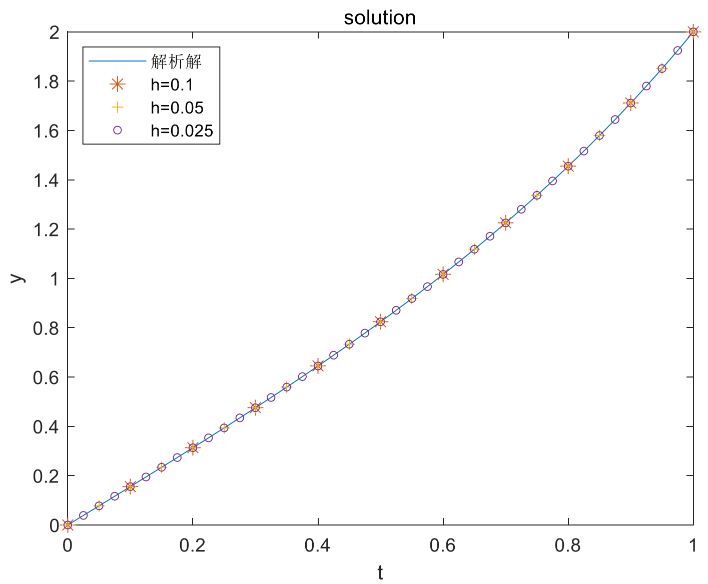
```
t2=0:h2:1;
n2=length(t2)-2;
A2=zeros(n2);
A2(1,1)=-(4*h2^2+2);
A2(1,2)=1;
A2(n2,n2-1)=1;
A2(n2,n2)=-(4*h2^2+2);
b2=zeros(n2,1);
b2(1)=-4*h2^2*t2(2);
b2(end)=-4*h2^2*t2(end-1)-2;
for k=2:n2-1
    A2(k,k-1)=1;
    A2(k,k)=-(4*h2^2+2);
    A2(k,k+1)=1;
    b2(k)=-4*h2^2*t2(k+1);
end
y2=transpose(A2\b2);
y2=[0,y2,2];
plot(t2,y2,'+','MarkerSize',6)

h3=0.025;
t3=0:h3:1;
n3=length(t3)-2;
A3=zeros(n3);
A3(1,1)=-(4*h3^2+2);
A3(1,2)=1;
A3(n3,n3-1)=1;
A3(n3,n3)=-(4*h3^2+2);
b3=zeros(n3,1);
b3(1)=-4*h3^2*t3(2);
b3(end)=-4*h3^2*t3(end-1)-2;
for k=2:n3-1
    A3(k,k-1)=1;
    A3(k,k)=-(4*h3^2+2);
    A3(k,k+1)=1;
    b3(k)=-4*h3^2*t3(k+1);
end
y3=transpose(A3\b3);
y3=[0,y3,2];
plot(t3,y3,'o','MarkerSize',4)
hold off
legend('解析解','h=0.1','h=0.05','h=0.025','Location','northwest')
xlabel('t')
ylabel('y')
```

```
title('solution')

t=0.5;
a=y1(t1==t);
fa=4*(a-t);
b=y2(t2==t);
fb=4*(b-t);
c=y3(t3==t);
fc=4*(c-t);
f=(16*(4*fc-fb)/(2^2-1)-(4*fb-fa)/(2^2-1))/(2^4-1);
result=f/4+t;
fprintf('Richardson extrapolation Result\n')
format long
disp(result)
format
```

执行 homework8_3.m，得到如下结果：

命令行窗口

```
        exp(-2 t) exp(2)    exp(2 t) exp(2)
t - ---------------- + ---------------
          exp(4) - 1          exp(4) - 1


Richardson extrapolation Result
   0.824027136988094
```



利用符号运算求得解析解，在命令行输出。即

$$y = t - \frac{e^{-2t+2}}{e^4 - 1} + \frac{e^{2t+2}}{e^4 - 1}$$

Richardson Extrapolation 计算公式推导如下：

$$y'' = \frac{y(t-h) - 2y(t) + y(t+h)}{h^2} + c_2 h^2 + o(h^4) = F_2(h)$$

$$y'' = \frac{4F_2(\frac{h}{2}) - F_2(h)}{2^2 - 1} + c_4 h^4 + o(h^6) = F_4(h)$$

$$y'' = \frac{16F_4(\frac{h}{2}) - F_4(h)}{2^4 - 1} + o(h^6)$$

$$F(h) = 4(y(t,h) - t)$$

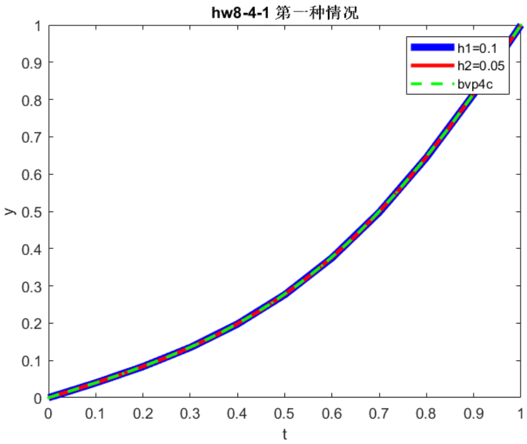
$$y'' = \frac{16\frac{4(4(y(t,\frac{h}{4}) - t)) - (4(y(t,\frac{h}{2}) - t))}{2^2 - 1} - \frac{4(4(y(t,\frac{h}{2}) - t)) - (4(y(t,h) - t))}{2^2 - 1}}{2^4 - 1} + o(h^6)$$

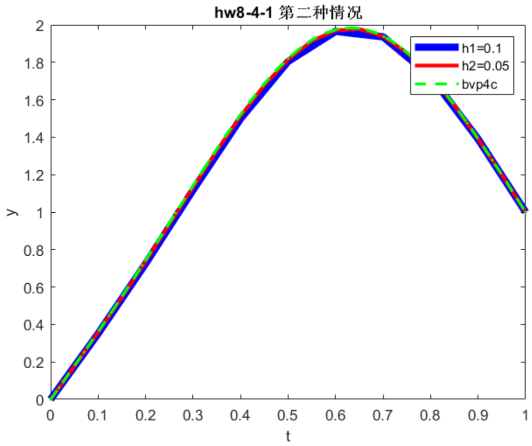
计算得到 y'' 之后，利用 $y'' = 4(y - t)$，得到 y 的值。

```
Richardson extrapolation Result
    0.824027136988094
```

第四题

（1）





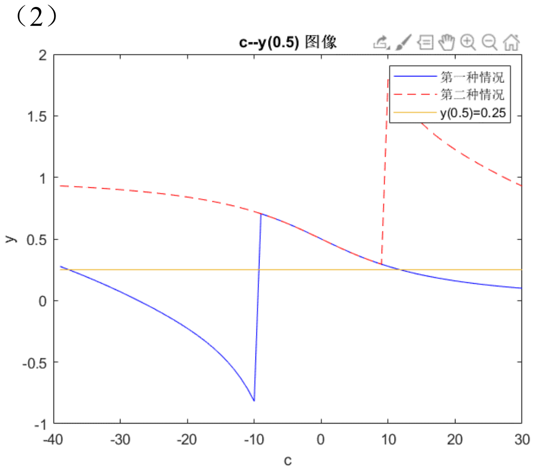
Hw8_4_1.m

```
%%
clc;clear;
%-(1)
h1 = 0.1;
h2 = 0.05;
tspan = [0 1];
bv = [0 1];
TOL = 1e-5;
n1 = (tspan(2)-tspan(1))/h1-1;
n2 = (tspan(2)-tspan(1))/h2-1;
% 第一种 initial guess
w11 = nlbvpfdm4T(tspan,h1,bv,TOL,zeros(n1,1));
w12 = nlbvpfdm4T(tspan,h2,bv,TOL,zeros(n2,1));
% 第二种 initial guess
w21 = nlbvpfdm4T(tspan,h1,bv,TOL,ones(n1,1));
w22 = nlbvpfdm4T(tspan,h2,bv,TOL,ones(n2,1));
f = @(t,y)[y(2);10*y(1)*(1-y(1))];
bcfun = @(ya,yb)[ya(1);yb(1)-1];
solinit1 = bvpinit(linspace(0,1,20),[0;0]);
w3s1 = bvp4c(f,bcfun,solinit1);
solinit2 = bvpinit(linspace(0,1,20),[1;1]);
w3s2 = bvp4c(f,bcfun,solinit2);
figure(1)
plot(tspan(1):h1:tspan(2),w11,'b-','LineWidth',5)
hold on
plot(tspan(1):h2:tspan(2),w12,'r-','LineWidth',2.5)
plot(w3s1.x,w3s1.y(1,:),'g.--','LineWidth',2)
legend({'h1=0.1' 'h2=0.05' 'bvp4c'})
title('hw8-4-1  第一种情况')
xlabel('t'),ylabel('y')
figure(2)
plot(tspan(1):h1:tspan(2),w21,'b-','LineWidth',5)
hold on
plot(tspan(1):h2:tspan(2),w22,'r-','LineWidth',2.5)
plot(w3s2.x,w3s2.y(1,:),'g.--','LineWidth',2)
legend({'h1=0.1' 'h2=0.05' 'bvp4c'})
title('hw8-4-1  第二种情况')
xlabel('t'),ylabel('y')
```

两个小题都有两个解，第二小题没有人做出第二种情况，这里需要注意一下，确实比较有迷惑性。

```
nlbvpfdm4T.m
function w = nlbvpfdm4T(tspan,h,bv,TOL,w0)
n = (tspan(2)-tspan(1))/h-1; % matrix and vector size
w1 = w0-DF(w0,n,h)\F(w0,n,h,bv);
while norm(w0-w1)>TOL
    w0 = w1;
    w1 = w0-DF(w0,n,h)\F(w0,n,h,bv);
end
w = w1;
w = [bv(1);w;bv(2)]; % column vector
end

F.m
% define vector-valued function F(w)
function Fv = F(w,n,h,bv)
Fv = zeros(n,1);
for i = 1:n
    if i == 1
        Fv(i) = bv(1)-(2+10*h^2)*w(i)+10*h^2*w(i)^2+w(i+1);
    elseif i == n
        Fv(i) = w(i-1)-(2+10*h^2)*w(i)+10*h^2*w(i)^2+bv(2);
    else
        Fv(i) = w(i-1)-(2+10*h^2)*w(i)+10*h^2*w(i)^2+w(i+1);
    end
end
end

DF.m
% define jacobian matrix DF(w)
function DFv = DF(w,n,h)
d0 = zeros(1,n);
for i = 1:n
    d0(i) = 20*h^2*w(i)-(2+10*h^2);
end
DFv = diag(d0)+diag(ones(1,n-1),1)+diag(ones(1,n-1),-1);
End
```

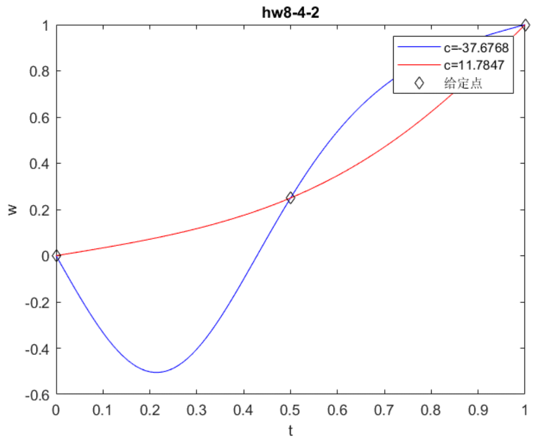
（2）



比较有迷惑性的一道题

绘制 c—y(0.5)图像，发现 c 可取的范围为第一种情况下的 (-38,-37) (-10,-9) (11,12)，其中(-10,-9)区间的这个点在当前步长下无法求取，故只求第一、三个

第一个c值为：c=-37.6767578125
第二个c值为：c=11.7846679688



Hw8_4_2.m

```
clc;clear;close all
h = 0.01;
tspan = [0 1];
bv = [0 1];
TOLnt = 1e-5;
TOLbs = 0.5e-6;
c = -39:30;
n = (tspan(2)-tspan(1))/h-1;
% 第一种情况
w0_1 = zeros(n,1); % 第一种 initial guess
w1 = zeros(n+2,length(c));
% 第二种情况
w0_2 = ones(n,1); % 第二种 initial guess
w2 = zeros(n+2,length(c));
for i = 1:length(c)
    w1(:,i) = nlbvpfdm4T_hyper(tspan,h,bv,TOLnt,w0_1,c(i));
    w2(:,i) = nlbvpfdm4T_hyper(tspan,h,bv,TOLnt,w0_2,c(i));
end
% 取 t=0.5
idx = 0.5/h+1;
w1sel = w1(idx,:);
w2sel = w2(idx,:);
figure(1)
plot(c,w1sel,'b-',c,w2sel,'r--')
hold on
fplot(@(x)0.25,[c(1),c(end)])
title('c--y(0.5) 图像')
legend({'第一种情况' '第二种情况' 'y(0.5)=0.25'})
xlabel('c'),ylabel('y')
cinit1 = [-38,-37];
cinit3 = [11,12];
c1 = mybisect(@Fc,cinit1(1),cinit1(2),TOLbs);
c3 = mybisect(@Fc,cinit3(1),cinit3(2),TOLbs);
fprintf('第一个 c 值为: c=%.10f\n 第二个 c 值为: c=%.10f\n',c1,c3)
w_c1 = nlbvpfdm4T_hyper(tspan,h,bv,TOLnt,w0_1,c1);
w_c3 = nlbvpfdm4T_hyper(tspan,h,bv,TOLnt,w0_1,c3);
figure(2)
plot(tspan(1):h:tspan(2),w_c1,'b-')
hold on
plot(tspan(1):h:tspan(2),w_c3,'r-')
plot([0 0.5 1],[0 0.25 1],'kd')
```

```
legend({['c=' num2str(c1)] ['c=' num2str(c3)] '给定点'})
title('hw8-4-2')
xlabel('t'),ylabel('w')
```

Fc.m

```
function wsel = Fc(c)
h = 0.01;
tspan = [0 1];
bv = [0 1];
TOLnt = 1e-5;
n = (tspan(2)-tspan(1))/h-1;
w0_1 = zeros(n,1); %  第一种 initial guess
idx = 0.5/h+1; % t=0.5
w = nlbvpfdm4T_hyper(tspan,h,bv,TOLnt,w0_1,c)-0.25;
wsel = w(idx,1);
end
```

nlbvpfdm4T_hyper.m

```
function w = nlbvpfdm4T_hyper(tspan,h,bv,TOL,w0,c) % with hyper-param
n = (tspan(2)-tspan(1))/h-1;
w1 = w0-DF_hyper(w0,n,h,c)\F_hyper(w0,n,h,bv,c);
while norm(w0-w1)>TOL
    w0 = w1;
    w1 = w0-DF_hyper(w0,n,h,c)\F_hyper(w0,n,h,bv,c);
end
w = w1;
w = [bv(1);w;bv(2)];
end
```

F_hyper.m

```
function Fv = F_hyper(w,n,h,bv,c) % with hyper-param c
Fv = zeros(n,1);
for i = 1:n
    if i == 1
        Fv(i) = bv(1)-(2+c*h^2)*w(i)+c*h^2*w(i)^2+w(i+1);
    elseif i == n
        Fv(i) = w(i-1)-(2+c*h^2)*w(i)+c*h^2*w(i)^2+bv(2);
    else
        Fv(i) = w(i-1)-(2+c*h^2)*w(i)+c*h^2*w(i)^2+w(i+1);
    end
end
end
```

DF_hyper.m

```
function DFv = DF_hyper(w,n,h,c) % with hyper-param c
d0 = zeros(1,n);
for i = 1:n
    d0(i) = 2*c*h^2*w(i)-(2+c*h^2);
end
DFv = diag(d0)+diag(ones(1,n-1),1)+diag(ones(1,n-1),-1);
end
```

mybisect.m

```
function r = mybisect(f,a,b,TOL)
c = (a+b)/2;
while abs(b-a)>TOL
    c = (a+b)/2;
```

```
        if f(c)==0 || abs(f(c))<=TOL
                break;
        elseif f(a)*f(c)<0
                b = c;
        else
                a = c;
        end
end
r = (a+b)/2;
end
```

第五题
示例 1：
homework8_5.m

```
clear
clc

h1=0.1;
t1=0:h1:1;
n1=length(t1)-2;
alpha1=@(i) 4/3*h1+2/h1+4*h1^3*(1/5+(i-1)/2+(i-1)^2/3 + 1/30+i/6+i^2/3);
beta1=@(i) 1/3*h1-1/h1+4*h1^3*(1/20+i/6+i^2/6);

A1=zeros(n1);
A1(1,1:2)=[alpha1(1),beta1(1)];
A1(end,end-1:end)=[beta1(n1-1),alpha1(n1)];
for K=2:n1-1
    A1(K,K-1:K+1)=[beta1(K-1),alpha1(K),beta1(K)];
end

b1=zeros(n1,1);
b1(1)=-1*beta1(0);
b1(end)=-exp(1)*beta1(n1);
c1=A1\b1;
c1=transpose([1;c1;exp(1)]);

h2=0.05;
t2=0:h2:1;
n2=length(t2)-2;
alpha2=@(i) 4/3*h2+2/h2+4*h2^3*(1/5+(i-1)/2+(i-1)^2/3 + 1/30+i/6+i^2/3);
beta2=@(i) 1/3*h2-1/h2+4*h2^3*(1/20+i/6+i^2/6);

A2=zeros(n2);
A2(1,1:2)=[alpha2(1),beta2(1)];
A2(end,end-1:end)=[beta2(n2-1),alpha2(n2)];
for K=2:n2-1
    A2(K,K-1:K+1)=[beta2(K-1),alpha2(K),beta2(K)];
end

b2=zeros(n2,1);
b2(1)=-1*beta2(0);
b2(end)=-exp(1)*beta2(n2);
c2=A2\b2;
c2=transpose([1;c2;exp(1)]);
```

```
f=@(t) exp(t.^2);
u=linspace(0,1);
plot(t1,c1,'-*','MarkerSize',2)
hold on
plot(t2,c2,'-o','MarkerSize',2)
plot(u,f(u))
hold off
xlabel('t')
ylabel('y')
title('solution  节点')
legend('n=9','n=19','解析解','Location','NorthWest')

figure
plot(t1,abs(c1-f(t1)),'-*','MarkerSize',4)
hold on
plot(t2,abs(c2-f(t2)),'-o','MarkerSize',4)
hold off
xlabel('t')
ylabel('误差')
title('误差图  节点')
legend('n=9','n=19','Location','NorthWest')

figure
u=linspace(0,1);
v1=cal(t1,c1,u);
v2=cal(t2,c2,u);
plot(u,v1,'-*','MarkerSize',2)
hold on
plot(u,v2,'-o','MarkerSize',2)
plot(u,f(u))
hold off
xlabel('t')
ylabel('y')
title('solution  区间')
legend('n=9','n=19','解析解  区间','Location','NorthWest')

figure
plot(u,v1-f(u),'-*','MarkerSize',2)
hold on
plot(u,v2-f(u),'-*','MarkerSize',2)
hold off
xlabel('t')
ylabel('误差')
title('误差图  区间')
```

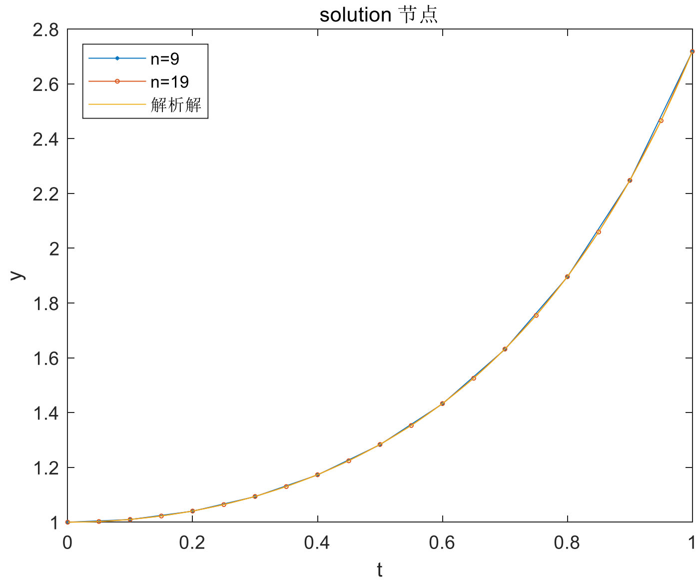
```
legend('n=9','n=19','Location','NorthWest')

function v=cal(t,c,u)
    % 根据系数向量 c 和节点向量 t 计算函数值
    % 利用了数组的逻辑索引法
    delta=diff(t);

    t1=t(2:end);
    K1=length(t1)*ones(size(u));
    c1=c(1:end-1);
    for i=length(t1):-1:1
        K1(t1(i)>u)=i;
    end
    s1=c1(K1).*(t1(K1)-u)./delta(K1);

    K2=ones(size(u));
    t2=t(1:end-1);
    c2=c(2:end);
    for j=1:length(t2)
        K2(t2(j)<=u)=j;
    end
    s2=c2(K2).*(u-t2(K2))./delta(K2);
    v=s1+s2;
end
```

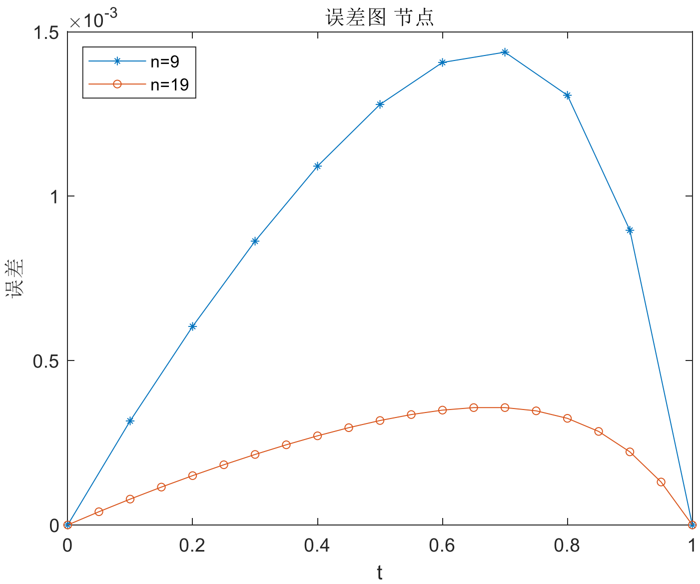
有限元方法得到了

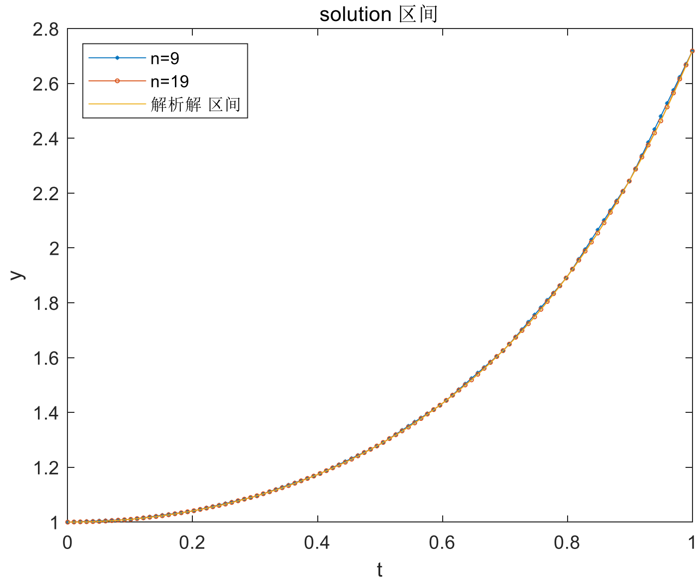
$$S(t)=\sum_{i=0}^{n+1} c_i \phi_i(t)$$

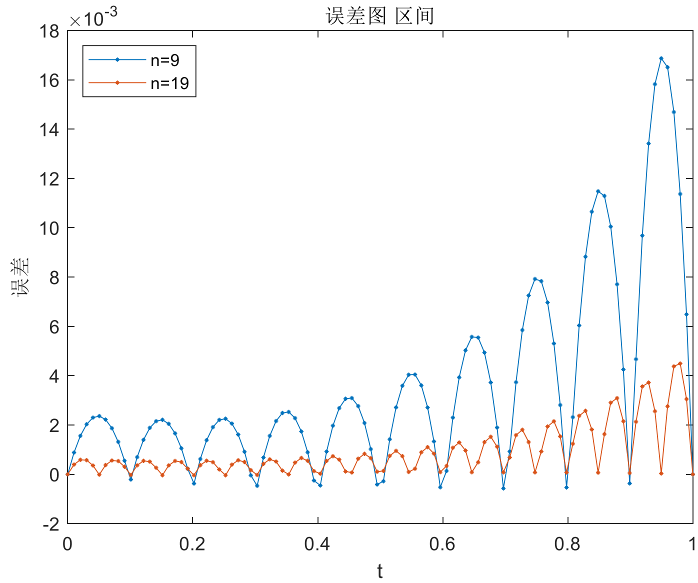
实际上这是对整个区间的表达式，因此，题目中要求的每幅图我都会给出两种，一种是只画出节点的图，即画出 y_i 与 t_i，另一种是在整个区间上计算函数值得到的图。

执行 homework8_5.m，得到如下结果：


solution 节点
n=9
n=19
解析解
y
t
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0
0.2
0.4
0.6
0.8
1


误差图 节点
$\times 10^{-3}$
n=9
n=19
误差
t
1.5
1
0.5
0
0
0.2
0.4
0.6
0.8
1





观察两组误差图，可以得到这样的结论：有限元方法的误差，节点处的误差较小，节点之间的误差较大;对于所有节点来说,靠近两端的节点处误差较小,靠近中心的节点处误差较大。

示例 2：

建立数学表达式。

这里的推导确实难了一点，容易错。

$$0=\int_0^1 \phi_i(t) f\left(t, \Sigma c_j \phi_j(t), \Sigma c_j \phi_j'(t)\right) dt+\int_0^1 \phi'_i(t) \Sigma c_j \phi_j'(t) dt, \qquad for\ i=1, \ldots, n$$

$$\rightarrow 0=\int_0^1 \phi_i(t)(2+4\mathrm{t}^2) \Sigma c_j \phi_j(t) dt+\int_0^1 \phi'_i(t) \Sigma c_j \phi_j'(t) dt$$

$$=\Sigma c_j\left(\int_0^1 \phi_i(t)(2+4\mathrm{t}^2) \phi_j(t) dt+\int_0^1 \phi'_i(t) \phi_j'(t) dt\right)$$

$$=\Sigma c_j\left(2 \int_0^1 \phi_i(t) \phi_j(t) dt+4 \int_0^1 t^2 \phi_i(t) \phi_j(t) dt+\int_0^1 \phi'_i(t) \phi_j'(t) dt\right)$$

$$\int_0^1 \phi_i(t) \phi_{i+1}(t) dt=\frac{2}{3} h$$

$$\int_0^1 \phi_i(t) \phi_i(t) dt=\frac{1}{6} h$$

$$\int_0^1 \phi'_i(t) \phi'_{i+1}(t) dt=\frac{2}{h}$$

$$\int_0^1 \phi'_i(t) \phi'_i(t) dt=-\frac{1}{h}$$

$$B(i, h)=\int_0^1 t^2 \phi_i(t) \phi_{i+1}(t) dt=\int_{ih}^{(i+1)h} t^2 \frac{(i+1)h-t}{h} \frac{t-ih}{h} dt$$

$$=\frac{1}{h^2} \int_{ih}^{(i+1)h} t^2\left(-t^2+(2i+1)h+i(i+1)h^2\right) dt$$

$$=\frac{1}{h^2}\left(-\frac{1}{5} t^5+\frac{(2i+1)h}{4} t^4-\frac{i(i+1)h^2}{3} t^3\right)\Big|_{ih}^{(i+1)h}$$

$$=h^3\left(-\frac{(i+1)^5-i^5}{5}+\frac{(2i+1)\left((i+1)^4-i^4\right)}{4}-\frac{i(i+1)\left((i+1)^3-i^3\right)}{3}\right)$$

$$A(i, h)=\int_0^1 t^2\left(\phi_i(t)\right)^2 dt=\int_{(i-1)h}^{ih} t^2\left(\frac{t-(i-1)h}{h}\right)^2 dt+\int_{ih}^{(i+1)h} t^2\left(\frac{(i+1)h-t}{h}\right)^2 dt$$

$$=\frac{1}{h^2}\left(\int_{(i-1)h}^{ih} t^2\left(t^2-2(i-1)ht+(i-1)^2 h^2\right) dt\right.$$

$$\left.+\int_{ih}^{(i+1)h} t^2\left(t^2-2(i+1)th+(i+1)^2 h^2\right) dt\right)$$

$$=\frac{1}{h^2}\left(\left(\frac{1}{5} t^5-\frac{2(i-1)h}{4} t^4+\frac{(i-1)^2 h^2}{3} t^3\right)\Big|_{(i-1)h}^{ih}\right.$$

$$\left.+\left(\frac{1}{5} t^5-\frac{2(i+1)h}{4} t^4+\frac{(i+1)^2 h^2}{3} t^3\right)\Big|_{ih}^{(i+1)h}\right)$$

$$=h^3\left(\left(\frac{i^5-(i-1)^5}{5}-\frac{2(i-1)\left(i^4-(i-1)^4\right)}{4}+\frac{(i-1)^2\left(i^3-(i-1)^3\right)}{3}\right)\right.$$

$$\left.+\left(\frac{(i+1)^5-i^5}{5}-\frac{2(i+1)\left((i-1)^4-i^4\right)}{4}+\frac{(i-1)^2\left((i-1)^3-i^3\right)}{3}\right)\right)$$

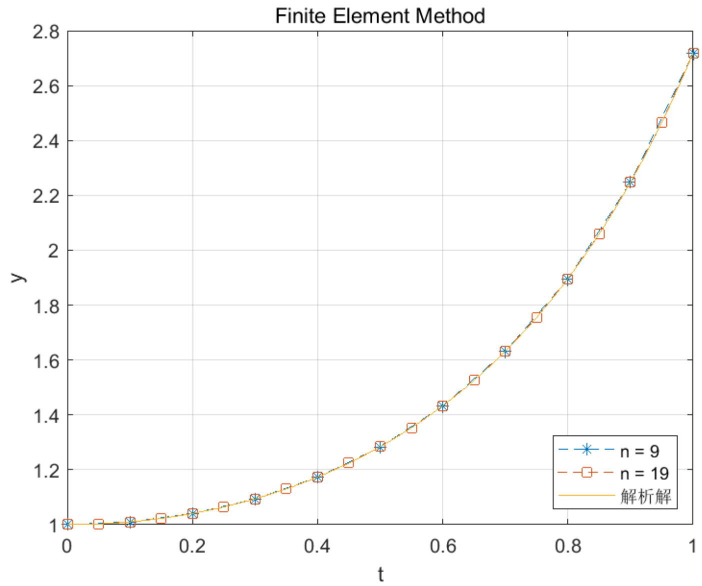
则

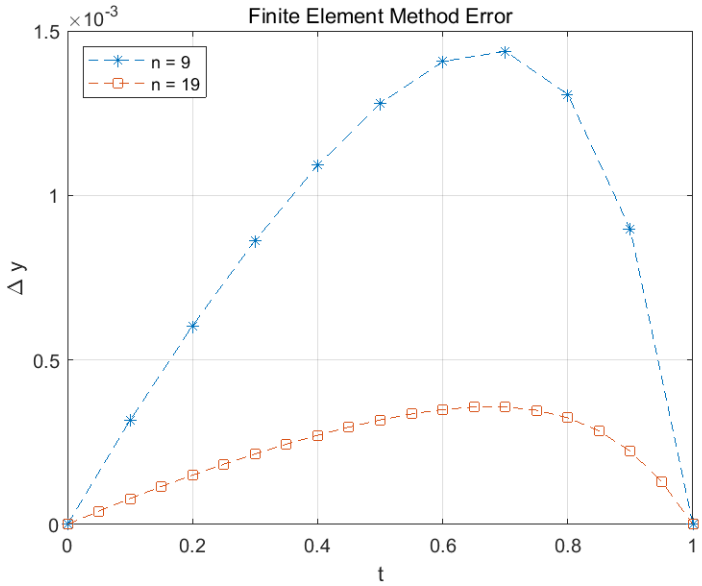
$$0=\Sigma c_j\left(2\int_0^1 \phi_i(t)\phi_j(t)dt+4\int_0^1 t^2\phi_i(t)\phi_j(t)dt+\int_0^1 \phi'_i(t)\phi'_j(t)dt\right)$$

$$=\left(\frac{h}{3}+4B(i-1,h)-\frac{1}{h}\right)\mathrm{c}_{\mathrm{i-1}}+\left(\frac{4}{3}h+4A(i,h)+\frac{2}{h}\right)c_i+\left(\frac{h}{3}+4B(i,h)-\frac{1}{h}\right)\mathrm{c}_{\mathrm{i+1}}$$

$$\triangleq \beta(i-1,h)c_{i-1}+\alpha(i,h)c_i+\beta(i,h)c_{i+1}$$

根据上述数学描述进行计算得到结果如下图所示。





hw8_5FE.m（输出系数矩阵的函数）

```
function [F,J,t] = hw8_5FEM(ya,yb,alpha,beta,a,b,n)
t = linspace(a,b,n+2); %定义自变量数组
h = (b-a)/(n+1); %定义步长
F = zeros(n,1);
J = zeros(n,n); %为系数矩阵开辟空间
```

```
for i = 1:1:n
    if i == 1
        F(i,1) = -1*ya*beta(i-1,h);
        J(i,i) = alpha(i,h);
        J(i,i+1) = beta(i,h);
    elseif i == n
        F(i,1) = -1*yb*beta(i,h);
        J(i,i) = alpha(i,h);
        J(i,i-1) = beta(i-1,h);
    else
        J(i,i) = alpha(i,h);
        J(i,i+1) = beta(i,h);
        J(i,i-1) = beta(i-1,h);
    end %为每个非 0 元素赋值
end
end
```

hw8_5.m

```
a = 0;
b = 1; %定义区间左右端点
ya = 1;
yb = exp(1); %定义端点函数值
n1 = 9;
n2 = 19; %定义步数
alpha = @(n,h) 4/3*h+2/h+...
                4*h^3*((n^5-(n-1)^5)/5-(2*(n-1)*(n^4-(n-1)^4))/4+((n-1)^2*(n^3-(n-1)^3))/3+...
                ((n+1)^5-n^5)/5-(2*(n+1)*((n+1)^4-n^4))/4+((n+1)^2*((n+1)^3-n^3))/3); %定义
alpha 系数函数
beta = @(n,h) h/3-1/h+...
               4*h^3*(-1*((n+1)^5-n^5)/5+((2*n+1)*((n+1)^4-n^4))/4-(n*(n+1)*((n+1)^3-
n^3))/3); %定义 beta 系数函数
[F1,J1,t1] = hw8_5FEM(ya,yb,alpha,beta,a,b,n1); %计算系数矩阵
c1t = J1\F1;
c1 = [ya,c1t',yb]'; %带上端点值为因变量数组
[F2,J2,t2] = hw8_5FEM(ya,yb,alpha,beta,a,b,n2);
c2t = J2\F2;
c2 = [ya,c2t',yb]'; %同上述操作
t3 = linspace(0,1,101);
yt3 = exp(t3.^2); %计算解析解
figure
plot(t1,c1,'*--','color',[0 0.4470 0.7410],'LineWidth',0.5);
hold on
plot(t2,c2,'s--','color',[0.8500 0.3250 0.0980],'LineWidth',0.5);
hold on
```

```
plot(t3,yt3,'color',[0.9290 0.6940 0.1250],'LineWidth',0.25);
hold on
grid on
legend('n = 9','n = 19','解析解','Location','Southeast');
xlabel('t');
ylabel('y');
title(' Finite Element Method '); %作图
figure
plot(t1,abs(c1'-exp(t1.^2)),'*--','color',[0 0.4470 0.7410],'LineWidth',0.5);
hold on
plot(t2,abs(c2'-exp(t2.^2)),'s--','color',[0.8500 0.3250 0.0980],'LineWidth',0.5);
hold on
grid on
legend('n = 9','n = 19','Location','Northwest');
xlabel('t');
ylabel('\Delta y');
title(' Finite Element Method Error '); %作图
```

第六题

示例 1：

homework8_6.m

这一题完成情况很好，大家都做得很好，赞！

```
clear
clc

h=0.1;
t=0:h:1;
n=length(t);

A=zeros(n);
for i=1:n
    if(i==1)
        for j=1:n
            A(1,j)=(0)^(j-1);
        end
    elseif(i==n)
        for j=1:n
            A(n,j)=1^(j-1);
        end
    else
        for j=1:n
            A(i,j)=(j-1)*(j-2)*t(i)^(j-3)+2*(j-1)*t(i).^(j-2)-3*t(i)^(j-1);
    end
    end
end

b=zeros(n,1);
b(1)=exp(3);
b(n)=1;
c=fliplr(transpose(A\b));
y=polyval(c,t);
plot(t,y)
hold on
bvpfcn=@(x,y) [y(2);3*y(1)-2*y(2)];
bcfcn=@(ya,yb) [ya(1)-exp(3);yb(1)-1];
guess=@(x) [-x;1];
h=1e-3;
tmesh = linspace(0,1,1/h+1);
solinit = bvpinit(tmesh, guess);
sol = bvp5c(bvpfcn, bcfcn, solinit);
Y=sol.y(1,:);
plot(sol.x,Y,'--')
xlabel('t')
```

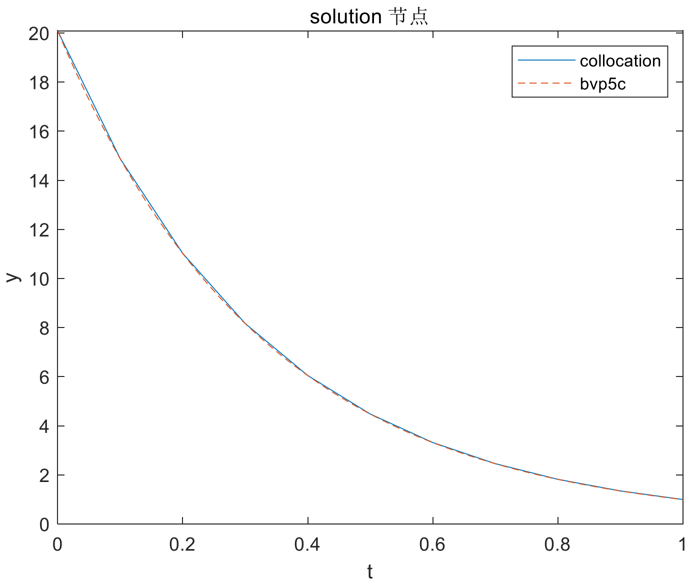
```
ylabel('y')
title('solution  节点')
legend('collocation','bvp5c')
figure
hold on
u=linspace(0,1);
v=polyval(c,u);
plot(u,v);
plot(sol.x,Y,'--')
xlabel('t')
ylabel('y')
title('solution  区间')
legend('collocation','bvp5c')
```

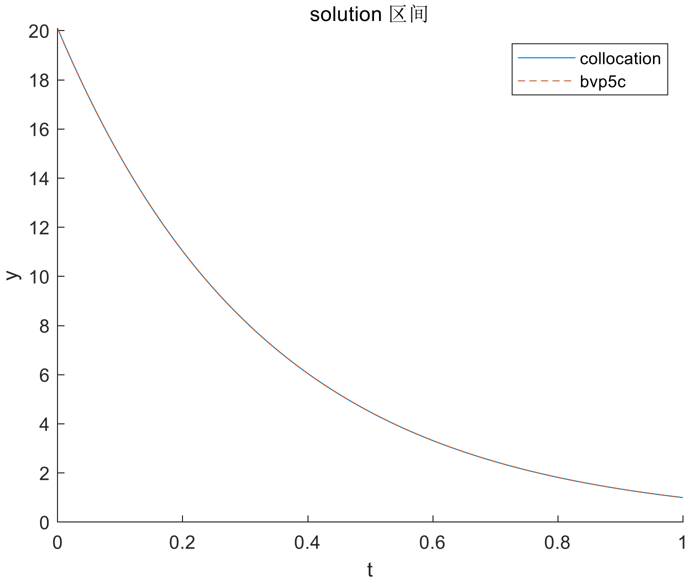
Collocation Method 得到了

$$y(t) = \sum_{j=1}^{n} c_j \phi_j(t) = \sum_{j-1}^{n} c_j t^{j-1}$$

实际上这是对整个区间的表达式，因此，题目中要求的图我会给出两种，一种是只画出节点的图，即画出 y_i 与 t_i，另一种是在整个区间上计算函数值得到的图。

执行 homework8_6.m，得到如下结果：





从图上看，两种方法得到的结果非常接近。

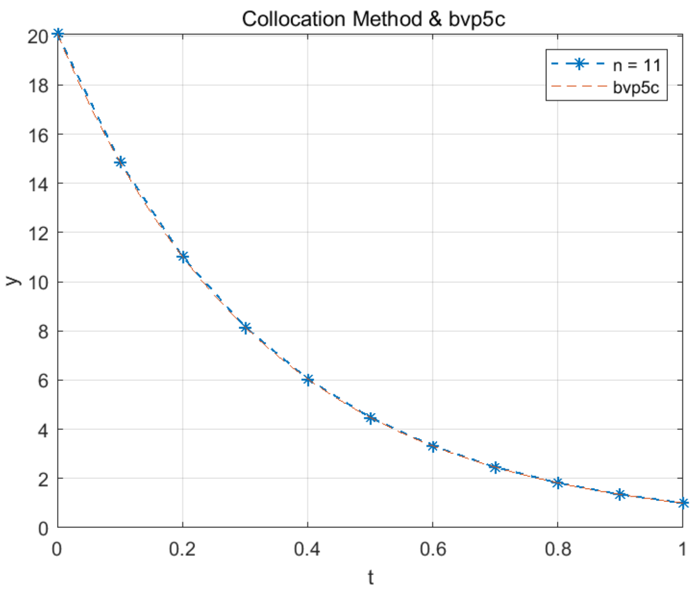
示例 2：
建立数学描述。

$$i=1: \Sigma c_j a^{j-1}=y(a)$$
$$i=n: \Sigma c_j b^{j-1}=y(b)$$
$$\Sigma(j-1)(j-2)c_j t^{j-3}-f\left(t, \Sigma c_j t^{j-1}, \Sigma c_j (j-1) t^{j-2}\right)=0$$
$$\rightarrow \Sigma(j-1)(j-2)c_j t^{j-3}-\left(3 \Sigma c_j t^{j-1}-2 \Sigma(j-1) c_j t^{j-2}\right)=0$$
$$\rightarrow \Sigma\left((j-1)(j-2) t^{j-3}-3 t^{j-1}+2(j-1) t^{j-2}\right) c_j=0$$

对于每个点t_i：

$$\Sigma\left((j-1)(j-2) t_i^{j-3}-3 t_i^{j-1}+2(j-1) t_i^{j-2}\right) c_j=0$$

根据上述数学描述计算得到结果如下图所示。



hw8_6CM.m（输出系数矩阵的函数）

```
function [F,J,t] = hw8_6CM(ya,yb,a,b,n)
t = (linspace(a,b,n))'; %定义自变量数组
F = zeros(n,1);
J = zeros(n,n); %开辟系数矩阵空间
for i = 1:1:n
    if i == 1
        F(i,1) = ya;
        for j = 1:1:n
            J(i,j) = a^(j-1);
        end
    elseif i == n
        F(i,1) = yb;
        for j = 1:1:n
            J(i,j) = b^(j-1);
        end
```

```
        else
            for j = 1:1:n
                J(i,j) = -3*(t(i))^(j-1)+2*(j-1)*(t(i))^(j-2)+(j-1)*(j-2)*(t(i))^(j-3);
            end
        end %根据数学描述为每一个元素赋值
    end
end
```

hw8_6.m

```
n = 11; %定义步数
a = 0;
b = 1;
ya = exp(3);
yb = 1; %定义边界条件
[F,J,t1] = hw8_6CM(ya,yb,a,b,n); %计算系数矩阵
c1 = J\F; %计算系数
yt1 = zeros(n,1);
for i = 1:1:n
    yt1(:,1) = yt1(:,1)+c1(i)*(t1).^(i-1);
end %计算因变量的值
ydt = @(t,y) [y(2);3*y(1)-2*y(2)];
res = @(ya,yb) [ya(1)-exp(3);yb(1)-1];
g = @(x) [0;0];
solinit = bvpinit(0:0.01:1,g);
sol = bvp5c(ydt,res,solinit); %bvp5c 求解
figure
plot(t1,yt1,'*--','color',[0 0.4470 0.7410],'LineWidth',1);
hold on
plot(sol.x(1,:),sol.y(1,:),'k--','color',[0.8500 0.3250 0.0980],'LineWidth',0.5);
hold on
grid on
legend('n = 11','bvp5c','Location','Northeast');
xlabel('t');
ylabel('y');
title('    Collocation Method & bvp5c '); %作图
```