信陽師範學院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

第5章 微分方程(组)数值解

讲授人：牛言涛

日期：2020年3月5日

目录
CONTENTS

常微分方程边值问题

- 前面讨论的ode系列函数只能用来求解初值问题，但是在实际中经常可以遇到一些边值问题。譬如热传导问题，初值时候的热源状态已知，一定时间后温度达到均匀。再比如弦振动问题，弦两端端点的位置是固定的。像这种知道自变量在前后两端时系统状态的问题被称为边值问题，可以使用下面方程来描述：

$$f(t, y, y') = 0,$$

定解条件：从 $y(0)=a$，$y(t_0)=b$，$y'(0)=c$，$y'(t_0)=d$ 中两端点0和t_0的两个表达式中各选一个组成定界条件 。MATLAB中提供了bvp4c（4阶）和bvp5c（5阶）求解边值问题 。

常微分方程边值问题

- 函数格式：sol = bvp4c(odefun,bcfun,solinit,options)
  - odefun：待求解的函数句柄
  - bcfun：函数边值条件的函数句柄，一般形式为res = bcfun(ya,yb,...)，表示y分别在a和b处的值，
  - solinit：一个结构体，为该方程解的初始估计值。
  - options：可选参数，用于指定积分算法，该参数为一个结构体，可以通过函数 bvpset 创建。
- solinit = bvpinit(x,v,parameters)：生成bvp4c调用指令所必须的“解猜测网”。
- sxint = deval(sol,xint)：计算微分方程积分区间内任何一点的解值。

常微分方程边值问题

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

- solinit = bvpinit(x,v,parameters)
  - x指定边界区间[a,b]上的初始网络，通常是等距排列的(1×M)一维数组（注意：使x(1)=a, x(end)=b，格点要单调排列）
  - v是对解的初始猜测
  - solint是“解猜测网Mesh”，它是一个结构体，带如下两个域：
    - solint.x表示初始网格有序节点的(1×M)一维数组，并且，solint.x(1)一定是a，solint.x(end)一定是b，M不易取得过大，一般为10数量级。
    - solint.y表示网格上微分方程解的猜测值的二维数组，solint.y(:,i)表示节点处solint.x(i)的解的猜测值。

常微分方程边值问题——案例分析

例：求解 $y'' = 2y'\cos t - y\sin 4t - \cos 3t$ 边值问题在区间t = [0,4]上的解：

边值条件：$y(0) = 1, y(4) = 2$

令 $y_1(t) = y(t), y_2(t) = y_1'(t)$ 得

$$\begin{cases} y_1' = y_2 \\ y_2' = 2y_2\cos t - y_1\sin 4t - \cos 3t \end{cases}$$

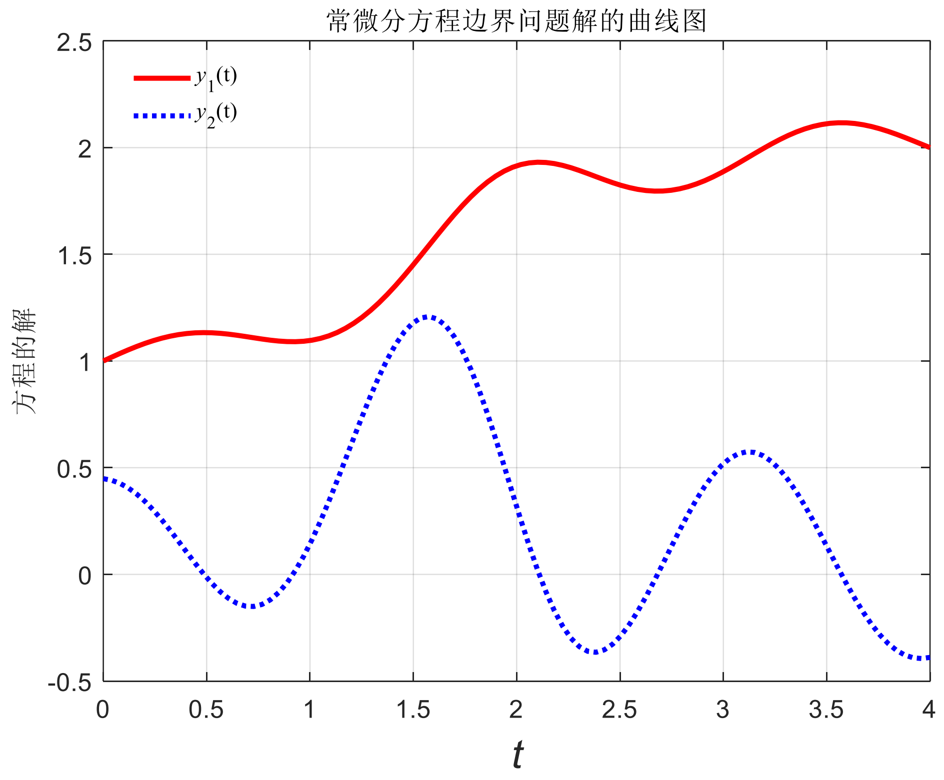
```
function dydx = odefun(t,y)
  %微分方程组函数M文件
  dydx = [y(2);
          2*y(2)*cos(t)-y(1)*sin(4*t)-cos(3*t)];
end
```

```
function yinit = initfun(t)
% 对y初值的估计函数，由于y1(0) = 1,y1(4) = 2；所以挑选一个满足上述条件的函数，这里选择的是1+t/4来作为对y1(t)的估计，从而其导数1/4作为对y2(t)的估计
    yinit = [  1+t/4; 1/4 ];
end
```

```
function res = bcfun(ya,yb)
  %边值条件M文件
  res = [ya(1)-1;
         yb(1)-2];
end
```

常微分方程边值问题——案例分析

```
solinit = bvpinit(linspace(0,4,10),@initfun);%由bvpinit生成的初始化网格
sol = bvp4c(@odefun,@bcfun,solinit);%调用bvp4c求解,也可以换成bvp5c
tint = linspace(0,4);  %默认生成100等分序列
Stint = deval(sol,tint);%根据得到的sol利用deval函数求出[0,4]区间内更多的解
plot(tint,Stint(1,:),'r-','linewidth',2);
grid on, hold on
plot(tint,Stint(2,:),'b:','linewidth',2);
L = legend('{\ity}_1(t)','{\ity}_2(t)','Location','best');
set(L,'fontname','Times New Roman');
xlabel('\itt','fontsize',16); ylabel('方程的解');
title('常微分方程边界问题解的曲线图')
```



常微分方程边值问题——案例分析

使用 bvpset 打开求解器统计信息的显示，并指定误差容限。此外，为了提高效率，指定分析 Jacobian 矩阵：

$$J=\frac{\partial f_i}{\partial y}=\begin{bmatrix}\dfrac{\partial f_1}{\partial y_1} & \dfrac{\partial f_1}{\partial y_2}\\ \dfrac{\partial f_2}{\partial y_1} & \dfrac{\partial f_2}{\partial y_2}\end{bmatrix}=\begin{bmatrix}0 & 1\\ -\sin 4t & 2\cos t\end{bmatrix}$$

```
function dfdy = jac(t,y)
%雅可比矩阵定义
dfdy = [0          1
        -sin(4*t)    2*cos(t)];
end
```

```
opts = bvpset('FJacobian',@jac,'RelTol',1e-4,'AbsTol',1e-10,'Stats','on');
solinit = bvpinit(linspace(0,4,100),@initfun);   %由bvpinit生成的初始化网格
sol4 = bvp4c(@odefun,@bcfun,solinit,opts);  %调用bvp4c求解
```

解是在一个 96 点网格上获得的。最大残差为 9.965e-05。

存在对 ODE 函数的 1169 个调用。存在对 BC 函数的 19 个调用。

```
sol5 = bvp5c(@odefun,@bcfun,solinit,opts)   %调用bvp5c
```

解是在一个 100 点网格上获得的。最大误差为 9.975e-06。

存在对 ODE 函数的 894 个调用。存在对 BC 函数的 9 个调用。

常微分方程边值问题——案例分析

例：求解常微分方程 $y''+\frac{2}{x}y'+\frac{1}{x^4}y=0$ 边值问题，求解区间 $\left[\frac{1}{3\pi},1\right]$

$$\begin{cases} y_1'=y_2 \\ y_2'=-\frac{2}{x}y_2-\frac{1}{x^4}y_1 \end{cases},\ y\left(\frac{1}{3\pi}\right)=0,\ y(1)=\sin 1$$

雅可比矩阵 $J=\frac{\partial f_i}{\partial y}=\begin{bmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} \end{bmatrix}=\begin{bmatrix} 0 & 1 \\ -\frac{1}{x^4} & -\frac{2}{x} \end{bmatrix}$

该方程组的精确解 $\begin{cases} y_1=\sin\left(\frac{1}{x}\right) \\ y_2=-\frac{2}{x^2}\cos\left(\frac{1}{x}\right) \end{cases}$

```
function dydx = bvpfcn(x,y)  %微分方程组
dydx = [y(2)
        -2*y(2)/x - y(1)/x^4];
end
function res = bcfcn(ya,yb)  %边值条件函数
res = [ya(1)
    yb(1)-sin(1)];
end
function dfdy = jac(x,y)  %雅可比矩阵定义
dfdy = [0     1
    -1/x^4 -2/x];
end
```

常微分方程边值问题——案例分析

opts = bvpset('FJacobian',@jac,'RelTol',0.1,'AbsTol',0.1,'Stats','on');

%使用 bvpinit 创建解的初始估计值。指定一个常量函数作为初始估计值，初始网格包含区间 [1/3π,1] 中的 10 个点。

xmesh = linspace(1/(3*pi), 1, 10);

solinit = bvpinit(xmesh, [1; 1]);

sol4c = bvp4c(@bvpfcn, @bcfcn, solinit, opts);

解是在一个 9 点网格上获得的。

最大残差为 9.794e-02。

存在对 ODE 函数的 157 个调用。

存在对 BC 函数的 28 个调用。

sol5c = bvp5c(@bvpfcn, @bcfcn, solinit, opts);

解是在一个 11 点网格上获得的。

最大误差为 6.742e-02。

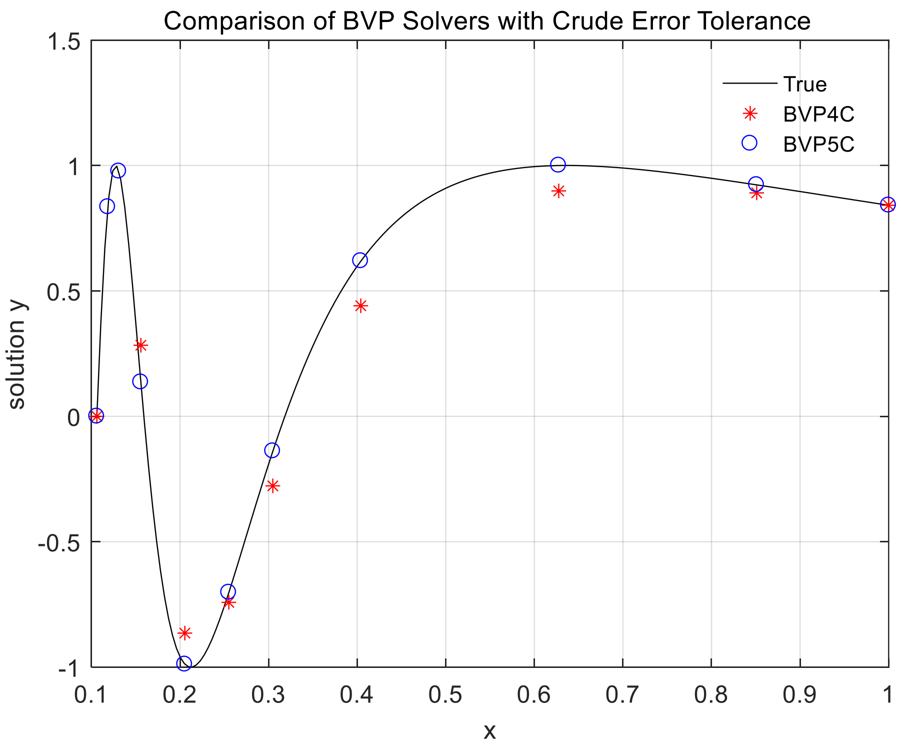
存在对 ODE 函数的 244 个调用。

存在对 BC 函数的 29 个调用。

xplot = linspace(1/(3*pi),1,200);

yplot = [sin(1./xplot); -cos(1./xplot)./xplot.^2];

plot(xplot,yplot(1,:),'k',sol4c.x,sol4c.y(1,:),'r*',sol5c.x,sol5c.y(1,:),'bo')



绘图验证了 bvp5c 直接控制计算中的真误差，而 bvp4c 仅间接控制计算中的真误差。在更严格的误差容限下，求解器之间的这种差异没有这么明显。

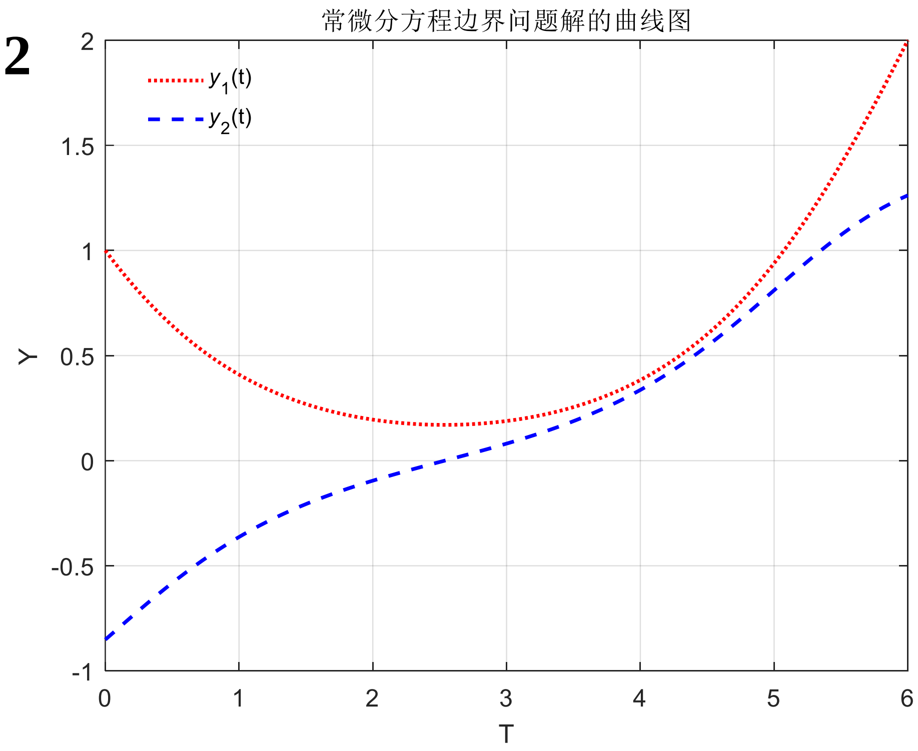
常微分方程边值问题——案例分析

信阳师范学院
数学与统计学院
SCHOOL OF MATHEMATICS AND STATISTICS

例：求解下列边值问题在区间t = [0,6]上的解：

$$\begin{cases} y_1{}' = y_2 \\ y_2{}' = \cos y_2 \sin y_1 \end{cases}，边值条件：y_1(0) = 1, y_1(6) = 2$$

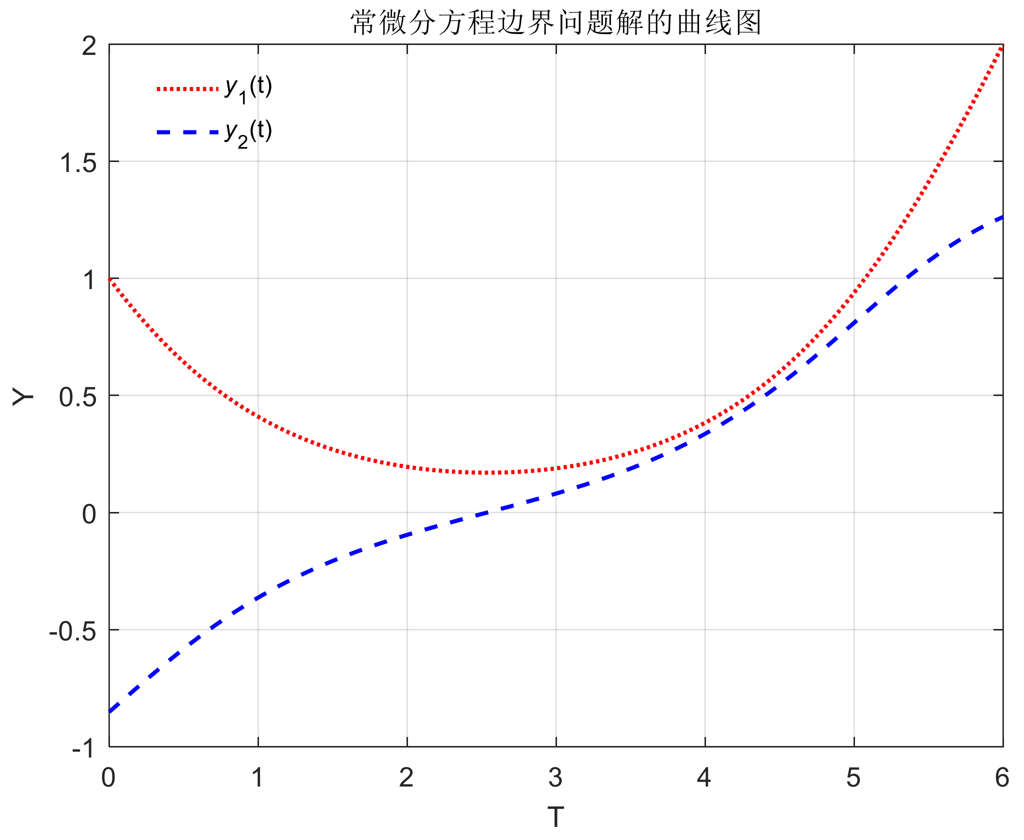
```
function yinit = initfun(t)  %解猜测网
    yinit = [ 1+t/6;  1/6 ];
end
function res = bcfun(ya,yb)  %边值条件
    res = [ ya(1)-1;  yb(1)-2];
end
function dydx = odefun(t,y)
      %微分方程函数
      dydx = [ y(2); cos(y(2))*sin(y(1)) ];
end
```




```
solinit = bvpinit(linspace(0,6,15),@initfun);%由bvpinit生成的初始化网格
sol = bvp4c(@odefun,@bcfun,solinit);%调用bvp4c求解,也可以换成bvp5c
```

常微分方程边值问题——案例分析


```
solinit = bvpinit(linspace(0,6,15),@initfun);%由bvpinit生成的初始化网格
sol = bvp4c(@odefun,@bcfun,solinit);%调用bvp4c求解,也可以换成bvp5c
sol =
  包含以下字段的 struct:
    solver: 'bvp4c'
         x: [1×19 double]
         y: [2×19 double]
        yp: [2×19 double]
     stats: [1×1 struct]
ti = linspace(0,6,100);yi = deval(sol,ti);
plot(ti,yi(1,:),'r:', ti,yi(2,:),'b--','LineWidth',1.5)
grid on
title('常微分方程边界问题解的曲线图'); xlabel('T');ylabel('Y')
legend('{\ity}_1(t)','{\ity}_2(t)','Location','best'); legend('boxoff')
```

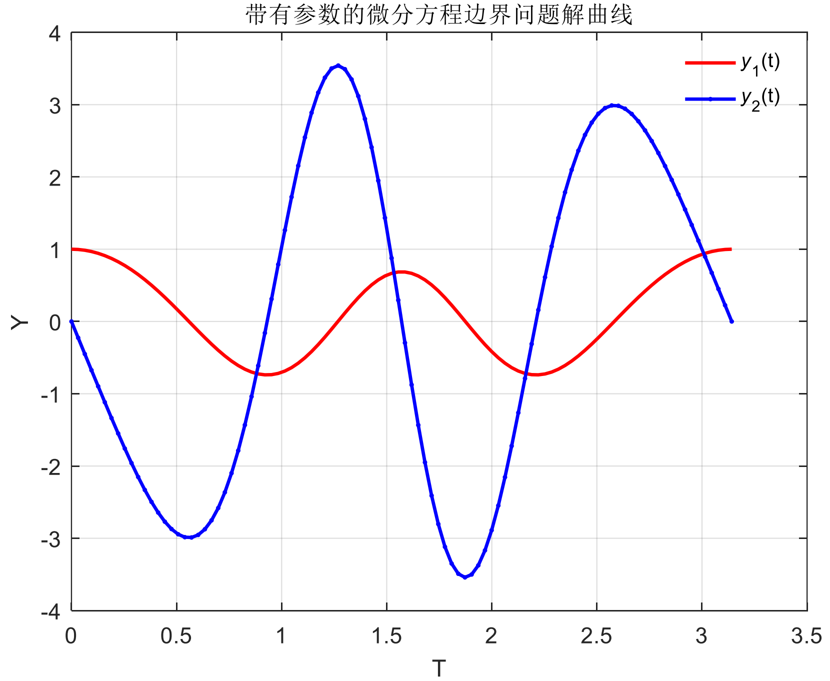


常微分方程边值问题——案例分析

例：求解下列边值问题在区间$x = [0,\pi]$上的解：

$$\begin{cases} y_1' = y_2 \\ y_2' = -(\lambda - 2q\cos 2x)y_1 \end{cases}，边值条件：y_1(0)=1, y_2(0)=0, y_2(\pi)=0$$

```
function yinit = mat4init(x)  %解猜测网
     yinit = [ cos(4*x); -4*sin(4*x) ];
end
function res = mat4bc(ya,yb,lambda) %边界条件
     res = [ya(1)-1; ya(2); yb(2)];
end
function dydx = mat4ode(x,y,lambda)  %方程函数文件
     q = 5;
     dydx = [ y(2);  -(lambda - 2*q*cos(2*x))*y(1) ];
end
```




```
lambda = 15;  %对lambda的猜测值
solinit = bvpinit(linspace(0,pi,100),@mat4init,lambda);
sol = bvp5c(@mat4ode,@mat4bc,solinit);%调用bvp5c求解
```


01
IDEA
02
LEARNING
03
CREATIVE
INSPIRATION
EDUCATION
BUSINESS
PHOTO BOOK

感谢聆听