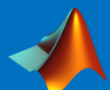


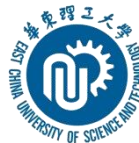
## 第6章 常微分方程数值解

---

- 常微分方程数值求解方法
- MATLAB求解初值问题函数：ode\*\*



# 上讲内容



- quadl函数的使用方法

匿名函数表示被积函数:  $\text{Int}=\text{quadl}(\text{AnonymousF},a,b)$

子函数表示被积函数:  $\text{Int}=\text{quadl}(@\text{fun},a,b)$

被积函数支持向量化运算（加点的运算符！）

- 列表型函数的积分方法

step1: 采用插值或拟合生成近似的被积函数

$\text{pp}=\text{pchip}(x,y)$

step2: quadl函数积分

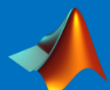
$\text{Int}=\text{quadl}(@\text{fun},a,b,[],[],\text{pp})$

$\text{function } y=\text{fun}(x,\text{pp})$



# 微分方程在化工模型中的应用

- 间歇反应器的计算
- 活塞流反应器的计算
- 全混流反应器的动态模拟
- 定态一维热传导问题
- 逆流壁冷式固定床反应器一维模型
- 固定床反应器的分散模型
- 各种传递过程



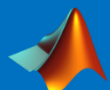
# 微分方程的定义

$$\frac{d^2 s}{dt^2} = -mg \quad (6-1)$$

$$\text{若设: } t_0 = 0 \text{ 时 } s = 0, s' = 1 \quad (6-2)$$

$$\text{或设: } \begin{cases} t_0 = 0 \text{ 时} & s = 0 \\ t_1 = 1 \text{ 时} & s = 0 \end{cases} \quad (6-3)$$

- 只有一个自变量的微分方程为常微分方程(ODE), 含有多个自变量的微分方程为偏微分方程(PDE);
- 方程中未知函数导数的最高阶数称为方程的阶;
- (6-2),(6-3)是(6-1)的定解条件, 可以获得微分方程的一个特解, 通常数值计算针对微分方程的特解问题。



# 常微分方程求解问题分类

## 初值问题:

- 定解附加条件在自变量的一端

- 一般形式为:

$$\begin{cases} y' = f(x, y) \\ y(a) = y_0 \end{cases}$$

- 初值问题的数值解法一般采用**步进法**，如Runge-Kutta法

## 边值问题:

- 在自变量两端均给定附加条件

- 一般形式:

$$\begin{cases} y' = f(x, y) \\ y(a) = y_1, y(b) = y_2 \end{cases}$$

- 边值问题可能有解、也可能无解，可能有唯一解、也可能有无数解
- 边值问题有迭加法、打靶法、松弛法等基本解法



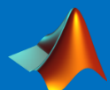
# 初值问题的数值解方法

$$\begin{cases} y' = f(t, y) \\ y(a) = y_0 \end{cases}$$

## ● 数值解即求方程在指定节点上的近似值

在数值解法中，首先把求解区间 $[t_0, t_f]$ 插入一系列分点 $t_i$ ，使 $t_0 < t_1 < \dots < t_i < \dots < t_n = t_f$ ；记 $h_i = t_{i+1} - t_i$ ， $h_i$ 称为步长。数值解即求在指定点上的函数值，结果以表格或图形的方式表示。

## ● 可将连续的首值问题离散化为离散方程求解



$$\begin{cases} y' = f(t, y) \\ y(a) = y_0 \end{cases}$$

微分法

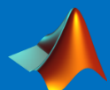
$$y_i' = \lim_{x_{i+1} \rightarrow x_i} \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \approx \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = f(x_i, y_i)$$

积分法

$$y_{i+1} - y_i = \int_{x_i}^{x_{i+1}} f(x, y) dx$$

泰勒级数法

$$y_{i+1} - y_i = hy_i' + \frac{h^2}{2!} y_i'' + \dots$$



# 欧拉法

$$\begin{cases} y' = f(t, y) \\ y(a) = y_0 \end{cases}$$



$$y(t_{i+1}) = y(t_i) + \int_{t_i}^{t_{i+1}} f(t, y) dt$$



$f(t, y)$  近似地看成  
是常数  $f(t_i, y_i)$

$$y(t_{i+1}) \approx y(t_i) + f[t_i, y(t_i)] \cdot (t_{i+1} - t_i) = y(t_i) + h \cdot f[t_i, y(t_i)]$$

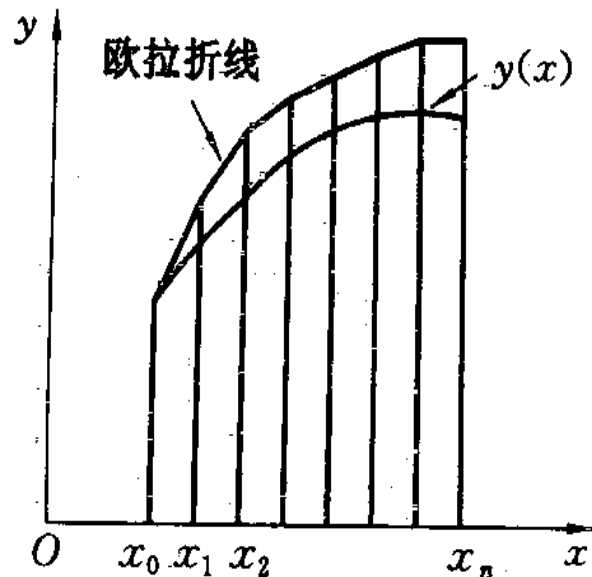


$i=0$

$$y(t_1) = y(t_0) + h \cdot f[t_0, y(t_0)]$$

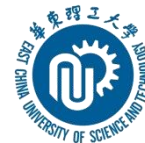
误差:

$$\delta(t_{n+1}) = \frac{h^2}{2} y''(t_n) + O(h^3)$$



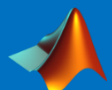


# 改进的欧拉法



$$\begin{cases} y' = f(t, y) \\ y(a) = y_0 \end{cases} \longrightarrow y(t_{i+1}) = y(t_i) + \int_{t_i}^{t_{i+1}} f(t, y) dt$$
$$\downarrow \int_{t_i}^{t_{i+1}} f[x, y(x)] dx \approx \frac{h}{2} [f(t_i, y_i) + f(t_{i+1}, y_{i+1})]$$
$$y_{i+1} = y_i + \frac{h}{2} [f(t_i, y_i) + f(t_{i+1}, y_{i+1})]$$

?  $y_{i+1}$  如何求取



# 预测 - 校正法

$$\begin{cases} y' = f(t, y) \\ y(a) = y_0 \end{cases} \quad \longrightarrow \quad y_{i+1} = y_i + \int_{t_i}^{x_{i+1}} f(x, y) dx$$

预测

$$y_{i+1}^* = y_i + hf(x_i, y_i)$$

校正

$$y_{i+1} = y_i + \frac{h}{2} [f(x_i, y_i) + f(x_{i+1}, y_{i+1}^*)]$$
$$(i = 0, 1, 2, \dots, n-1)$$

$$K_1 = f(t_i, y_i)$$

$$K_2 = f(t_i + h, y_i + hK_1)$$

$$y_{i+1} = y_i + h(K_1 + K_2)/2$$

$$t_{i+1} = t_i + h$$

# 龙格-库塔法

$$\begin{cases} y' = f(t, y) \\ y(a) = y_0 \end{cases} \quad \longrightarrow$$

$$y_{i+1} = y_i + h[\alpha_1 f(t_i, y(t_i)) + \alpha_2 f(t_i + \lambda_2, y(t_i) + \mu_2 h) + \cdots + \alpha_r f(t_i + \lambda_r, y(t_i) + \mu_r h)]$$

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

$$k_1 = f(t_i, y_i),$$

$$k_2 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_1),$$

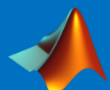
$$k_3 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_2),$$

$$k_4 = f(t_i + h, y_i + hk_3).$$

四阶的龙格-库塔法

如何确定解的误差？如何确定积分步长？

不同精度的算法  
联合使用



# 阿达姆斯法（多步法）

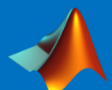
$$\begin{cases} y' = f(t, y) \\ y(a) = y_0 \end{cases} \longrightarrow y(t_{i+1}) \approx y(t_i) + \int_{t_i}^{t_{i+1}} P_r(t, y_{i+1}, y, y_{i-1}, \dots, y_{i-r}) dt$$



利用已知数据点构造一个  
多项式，代入上式积分

显式 
$$y_{i+1} = y_i + \frac{h}{24} (55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3})$$

隐式 
$$y_{i+1} = y_i + \frac{h}{24} (9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2})$$



# MATLAB求解初值问题方法

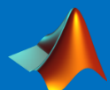
1. 将待求解转化为标准形式，并“翻译”成MATLAB可以理解的语言，即编写一个函数文件表示微分方程



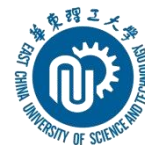
2. 选择合适的解算指令求解问题



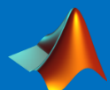
3. 根据求解问题的要求，设置解算指令的调用格式



# MATLAB求解初值问题函数

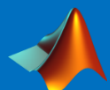


函数分类	函数
求解函数	ode45, ode23, ode78, ode89, ode113(非刚性方程) ode15s, ode23t, ode23s, ode23tb(刚性方程) ode15i(全隐式方程)
求解选项	odeset, odeget
求解输出	odeplot, odephas2, odephas3, odeprint
扩展函数	deval, odextend



# 常微分方程在MATLAB中的表示

- 常将待求解的微分方程表示为一个MATLAB函数，作为整个求解程序的一个子函数
- 描述微分方程的函数声明语句如下：
  - $\text{dy}=\text{odefun}(\text{t},\text{y})$
- 输入变量至少包括：自变量 $\text{t}$ 和因变量 $\text{y}$ ；
- 输出变量 $\text{dy}$ 表示 $\text{y}$ 的一阶导数表达式，对于常微分方程组， $\text{dy}$ 必须是一个列向量；
- 可以向ode文件中传递参数，数目不受限制



# odefile的编写

求解初值问题：

$$\begin{cases} y' = y - \frac{2x}{y} \\ y(0) = 1 \end{cases} \quad (0 \leq x \leq 1) \quad \begin{cases} y' = f(x, y) \\ y(a) = y_0 \end{cases}$$

自变量在前，因变量在后

```
function f=fun(x,y)
f=y-2*x/y;
```

输出变量为因变量导数的表达式

$$\begin{cases} y' = y + y^2 \\ y(0) = 1 \end{cases} \quad (0 \leq x \leq 1) \quad \longrightarrow$$

```
function f=fun(x,y)
f=y+y^2;
```



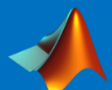


# 常微分方程组odefile的编写

常微分方程组与单个常微分方程表示方法相同，只需在编写函数文件时将各变量的导函数作为一个**列向量**输出。

$$\begin{cases} y_1' = 0.04(1 - y_1) - (1 - y_2)y_1 + 0.0001(1 - y_2)^2 \\ y_2' = -10^4 y_1' + 3000(1 - y_2)^2 \\ y_1(0) = 0, y_2(0) = 1, 0 \leq x \leq 100 \end{cases}$$

```
function f=fun(x,y)
dy1dx = 0.04*(1-y(1))-(1-y(2)).*y(1)+0.0001*(1-y(2)).^2;
dy2dx = -1e4*dy1dx + 3000*(1-y(2)).^2;
f = [dy1dx; dy2dx];
```



# 高阶微分方程

$$\begin{cases} y^{(m)} = f(x, y, y', \dots, y^{(m-1)}) \\ y(x_0) = y_0, y'(x_0) = y_1, \dots, y^{(m-1)}(x_0) = y_{m-1} \end{cases}$$

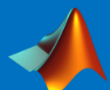
引入新变量  $y_1 = y, y_2 = y', \dots, y_m = y^{(m-1)}$

原方程变换为一阶微分  
方程组：

$$\begin{cases} y_1' = y_2 \\ y_2' = y_3 \\ \vdots \\ y_{m-1}' = y_m \\ y_m' = f(x, y_1, y_2, \dots, y_m) \end{cases}$$

初始条件：

$$y_1(x_0) = y(x_0) = y_0, y_2(x_0) = y'(x_0) = y_1, \dots, y_m(x_0) = y^{(m-1)}(x_0) = y_{m-1}$$



# 高阶微分方程odefile的编写

$$y'' + a(t)(y')^2 + b(t)y = e^t \cos 2\pi t$$

$$a(t) = -e^{-t} + \cos 2\pi t e^{-2t}, b(t) = \cos(2\pi t)$$

方程系数非线性



可在odefile中定义

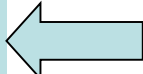
方程高阶，非标准形式



变量代换：令 $y_1=y$ ； $y_2=y'$

则原方程等价于：

```
function f=fun(t,y)
a=-exp(-t)+cos(2*pi*t)*exp(-2*t);
b=cos(2*pi*t);
f=[y(2);
    -a*y(2)^2-b*y(1)+exp(t)*b];
```


$$\begin{cases} y_1' = y_2 \\ y_2' = -ay_2^2 - by_1 + e^t \cos 2\pi t \end{cases}$$



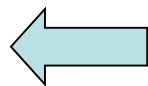
# 高阶微分方程odefile的编写

$$\begin{cases} y_1' = x * y_2' + y_1 \\ y_2'' = y_1' + \sin(x) * y_2 \end{cases}$$

变量代换：令 $y(1)=y_1$ ； $y(2)=y_2$ ； $y(3)=y_2'$

则原方程等价于：

```
function f=Deq6_4(x,y)
f=zeros(3,1);
f(1)=x*y(3)+y(1);
f(2)=y(3);
f(3)=f(1)+sin(x)*y(2);
```

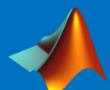


$$\begin{cases} y(1)' = x * y(3) + y(1) \\ y(2)' = y(3) \\ y(3)' = y(1)' + \sin(x) * y(2) \end{cases}$$



# 常微分方程初值问题解算指令比较

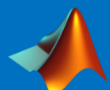
解算指令	算法	精度
ode45	四五阶Runge-Kutta法	较高
ode23	二三阶Runge-Kutta法	低
ode113	可变阶Adams-Bashforth-Moulton法	
ode15s	基于数值差分的可变阶方法 (BDFs, Gear)	低~中
ode23s	二阶改进的Rosenbrock法	低
ode23t	使用梯形规则	适中
ode23tb	TR-BDF2 (隐式Runge-Kutta法)	低



# 求解函数的使用方法

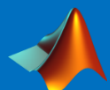
所有求解函数的使用方法一样，以ode45为例：

1.  $[T,Y]=ode45(@fun, TSPAN,Y0)$
2.  $[T,Y]=ode45(@fun, TSPAN,Y0,options)$
3.  $[T,Y]=ode45(@fun, TSPAN,Y0,options,P1,P2,...)$
4.  $sol=ode45(@fun, TSPAN,Y0,options,P1,P2,...)$
5.  $ode45(@fun, TSPAN,Y0,options,P1,P2,...)$
6.  $[T,Y,TE,YE,IE]=ode45(@fun, TSPAN,Y0,options,P1,P2,...)$



# 求解函数的输入变量

- 在最简单的格式下，ode45需要有三个输入变量：@fun，TSPAN和Y0
- @fun为表示待求解微分方程子函数的函数句柄；
- TSPAN表示待求解方程的求解区间；
  - 当TSPAN仅有两个元素时，这两个元素分别表示求解区间的起始与终止时刻；
  - 当TSPAN含有三个及以上元素时，TSPAN（1）和TSPAN（end）表示求解区间的起始与终止时刻，中间各元素为指定求解函数必须求解的时刻；
  - 求解函数的积分步长与TSPAN中含有几个元素无关，TSPAN元素个数的不同仅改变了求解函数的输出值



# 求解函数的输入变量

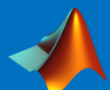
- $Y0$ 为初始条件，它也是一个向量，其元素的个数与表示微分方程的子函数返回值长度相同
- `options`用于设置一些可选的参数值，以提高求解质量和精度，当求解结果不理想或对结果有特殊要求时，可更改这些参数值；
- $P1, P2, \dots$ 的作用是传递附加参数 $P1, P2, \dots$ 到表示微分方程的子函数。当`options`缺省时，应在相应位置保留`[]`，以便正确传递参数。





# 求解函数的输出变量

- MATLAB微分方程求解函数的输出变量可以由0，1，2或5个，最常用的是2个输出变量，如调用格式的1-3
- 输出变量T为返回求解节点的列向量；当TSPAN中仅含有两个元素时，T的元素个数为求解函数自行选择；当TSPAN含有三个及以上元素时，T中元素的个数与TSPAN相同；
- 输出变量Y为返回的各因变量的值；它是一个矩阵，矩阵的列数与待求解的因变量数相同；行数与T相同，每行元素为对应T所在行时刻的值；



# 求解函数的输出变量

- 输出变量可以仅有一个，如调用格式4。当待求解方程是通过函数句柄（不能是匿名函数名）传递给求解函数时，它将是一个结构体，其中包括三个域，`sol.x`是MATLAB自行选择求解节点；每一列的`sol.y(:,i)`是与`sol.x(i)`对应的解；`sol.solver`是求解函数的函数名；如果求解函数的控制选项中定义了事件（参见6.3.8节），则`sol`中还含有与之相关的三个域。`sol`与`deval`函数配合使用可以获得求解区间内任意时刻的函数值；
- `deval`的调用格式是`sxint = deval(sol,xint)`，`sol`为求解函数返回的结构体，`xint`为需要计算函数值的节点，返回值`sxint`为计算所得函数值；
- 输出变量也可以没有，如调用格式5。此时求解函数将调用`odeplot`函数，绘制因变量与自变量的关系图；



# ode解算指令的使用

求解初值问题：

$$\begin{cases} y' = y - \frac{2x}{y} \\ y(0) = 1 \end{cases} \quad (0 \leq x \leq 1)$$

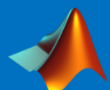
```
function Cha6demo5
```

```
y0=1;
```

```
[x1,y1]=ode45(@Deq6_1,[0,1],y0)
```

```
function dydx=Deq6_1(x,y)
```

```
dydx=y-2*x/y;
```



# ode解算指令的使用

利用图形直观的表示计算结果：

这是较为常用的求解方法！

利用plot命令

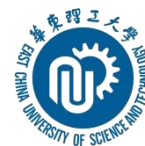
```
function Cha6demo5_A
y0=1;
[x1,y1]=ode45(@Deq6_1,[0,1],y0);
plot(x1,y1,'b-o')
function dydx=Deq6_1(x,y)
dydx=y-2*x/y;
```

利用ode自  
带输出函数

```
function Cha6demo5_B
y0=1;
ode45(@Deq6_1,[0,1],y0)
function dydx=Deq6_1(x,y)
dydx=y-2*x/y;
```



# ode解算指令的使用



如何返回指定点的值:

请注意其它后处理过程的实现方法

利用关系运算

```
function Cha6demo5_C
y0=1;
t=0:0.1:1;
[x1,y1]=ode45(@Deq6_1,t,y0);
%查找与x=0.5对应的y值
yout=y1(x1==0.5)
function dydx=Deq6_1(x,y)
dydx=y-2*x/y;
```

利用deval函数

```
function Cha6demo5_D
y0=1;
sol=ode45(@Deq6_1,[0,1],y0);
yout=deval(sol,0.5)
function dydx=Deq6_1(x,y)
dydx=y-2*x/y;
```



# 例题

在三个串联的CSTR反应器中，发生简单的一级不可逆反应：

$A \rightarrow B$ ，已知初始条件：

进料初始浓度， $CA0=1.8 \text{ kmol/m}^3$ ，

三釜内初始浓度， $CA10=0.4 \text{ kmol/m}^3$ ， $CA20=0.2 \text{ kmol/m}^3$ ，

$CA30=0.1 \text{ kmol/m}^3$ ，

动力学参数： $k=0.5 \text{ min}^{-1}$ ， $\tau=2 \text{ min}$

求解在10分钟内三个反应器中组分A浓度随时间的变化规律。

模型：

$$\frac{dC_{A1}}{dt} = \frac{C_{A0} - C_{A1}}{\tau} - kC_{A1}$$

$$\frac{dC_{A2}}{dt} = \frac{C_{A1} - C_{A2}}{\tau} - kC_{A2}$$

$$\frac{dC_{A3}}{dt} = \frac{C_{A2} - C_{A3}}{\tau} - kC_{A3}$$



# 例题



```
function Cha6demo6
CA10=0.4; CA20=0.2; CA30=0.1;
stoptime=10;
[t,y]=ode45(@CSTR,[0 stoptime],[CA10 CA20 CA30]);
plot(t,y(:,1),'k--',t,y(:,2),'b:',t,y(:,3),'r-')
```

.....

% -----

```
function dCdt=CSTR(t,y)
CA0=1.8; k=0.5; tau=2;
CA1=y(1); CA2=y(2); CA3=y(3);
dCA1dt=(CA0-CA1)/tau-k*CA1;
dCA2dt=(CA1-CA2)/tau-k*CA2;
dCA3dt=(CA2-CA3)/tau-k*CA3;
dCdt = [dCA1dt; dCA2dt; dCA3dt];
```

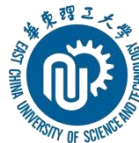
$$\frac{dC_{A1}}{dt} = \frac{C_{A0} - C_{A1}}{\tau} - kC_{A1}$$

$$\frac{dC_{A2}}{dt} = \frac{C_{A1} - C_{A2}}{\tau} - kC_{A2}$$

$$\frac{dC_{A3}}{dt} = \frac{C_{A2} - C_{A3}}{\tau} - kC_{A3}$$



# 参数传递

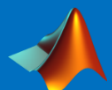


```
function Cha6demo6
CA0=1.8; k=0.5; tau=2;
CA10=0.4; CA20=0.2; CA30=0.1; stoptime=10;
[t,y]=ode45(@CSTR,[0 stoptime],[CA10 CA20
CA30],[],CA0,k,tau);
plot(t,y(:,1),'k--',t,y(:,2),'b:',t,y(:,3),'r-')
```

.....

% -----

```
function dCdt=CSTR(t,y,CA0,k,tau)
CA1=y(1); CA2=y(2); CA3=y(3);
dCA1dt=(CA0-CA1)/tau-k*CA1;
dCA2dt=(CA1-CA2)/tau-k*CA2;
dCA3dt=(CA2-CA3)/tau-k*CA3;
dCdt = [dCA1dt; dCA2dt; dCA3dt];
```





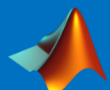
# ode解算指令的选择(1)

## 1.根据常微分方程要求的求解精度与速度要求

求解初值问题：

$$\begin{cases} y' = y - \frac{2x}{y} \\ y(0) = 1 \end{cases} \quad (0 \leq x \leq 1)$$

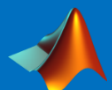
比较ode45和ode23的求解精度和速度



# ode45和ode23的比较

```
function Cha6demo10
format long
y0=1;
tic,[x1,y1]=ode45(@fun,[0,1],y0);t_ode45=toc
tic,[x2,y2]=ode23(@fun,[0,1],y0);t_ode23=toc
plot(x1,y1,'b-o',x2,y2,'m-*'),
xlabel('x'),ylabel('y'),
legend('ODE45','ODE23','location','Northwest')
disp('Comparative Results at x=1:');
fprintf('\nODE45\t\t\t y=%.8f\nODE23\t\t\t y=%.8f\nPrecise Result=%.8f\n'...
        ,y1(end),y2(end),1.7320508)
function f=fun(x,y)
f=y-2*x/y;
```

ODE45	y=1.73205082
ODE23	y=1.73215488
Precise Result	=1.73205080



# ode解算指令的选择(2)

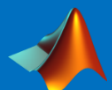
## 2. 根据常微分方程组是否为刚性方程

- 如果在一个过程中的快变子过程与慢变子过程变化速率相差非常大，在数学上称这种过程具有“刚性”
- 刚性方程在化学工程和自动控制领域的模型中比较常见。

$$\frac{d}{dt} \begin{bmatrix} y_1(t) \\ y_2(t) \\ \dots \\ y_n(t) \end{bmatrix} = \begin{bmatrix} f_1(t, y_1, \dots, y_n) \\ f_2(t, y_1, \dots, y_n) \\ \dots \\ f_n(t, y_1, \dots, y_n) \end{bmatrix}$$

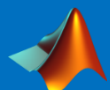
刚性比

$$SR = \frac{\max_{1 \leq i \leq n} (\text{Re } \lambda_i)}{\min_{1 \leq i \leq n} (\text{Re } \lambda_i)}$$



## ode解算指令的选择(2)

- 常微分方程组数值积分的稳定步长受模值最大的特征值控制，即受快变量分量约束，特征值大则允许步长小；而过程趋于稳定的时间又由模值最小的特征值控制，特征值小则积分到稳定的时间则长。
- MATLAB提供了不同种类的刚性方程求解指令：`ode15s` `ode23s` `ode23t` `ode23tb`，可根据实际情况选用。一般使用`ode15s`较多。



# 刚性常微分方程组求解

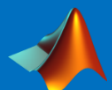
$$\begin{cases} y_1' = 0.04(1 - y_1) - (1 - y_2)y_1 + 0.0001(1 - y_2)^2 \\ y_2' = -10^4 y_1' + 3000(1 - y_2)^2 \\ y_1(0) = 0, y_2(0) = 1, 0 \leq x \leq 100 \end{cases}$$

```
function Cha6demoStiff
figure
ode23s(@fun,[0,100],[0;1])
figure,
ode45(@fun,[0,100],[0;1])
function f=fun(x,y)
dy1dx = 0.04*(1-y(1))-(1-y(2)).*y(1)+0.0001*(1-y(2)).^2;
dy2dx = -1e4*dy1dx + 3000*(1-y(2)).^2;
f = [dy1dx; dy2dx];
```



# 解算指令的options选项

1. RelTol—相对误差，它应用于解向量的所有分量。在每一步积分过程中，第*i*个分量误差 $e(i)$ 满足： $e(i) \leq \max(\text{RelTol} * \text{abs}(y(i)), \text{AbsTol}(i))$ 。
2. AbsTol—绝对误差，若是实数，则应用于解向量的所有分量，若是向量，则它的每一个元素应用于对应位置解向量元素。
3. OutputFcn—可调用的输出函数名。每一步计算完后，这个函数将被调用输出结果，可以选择的值为：odeplot, odephas2, odephas3, odeprint。
4. OutputSel—输出序列选择。指定解向量的哪个分量被传递给OutputFcn。
5. MaxSetp—步长上界，缺省值为求解区间的1/10。
6. InitialStep—初始步长，缺省时自动设置。
7. Events事件记录，取‘on’时将返回事件记录；
8. 采用odeset改变原有选项的值



# 解算指令的图形输出

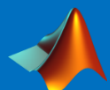
1. 在无输出变量时，将调用默认的odeplot输出解的图形，此图形以表示T与Y关系，因此当odeplot绘制曲线的条数等于Y的个数；
2. 除了以odeplot形式输出外，还可以以odephas2，和odephas3的形式输出解向量的二维和三维相平面图；相平面输出描述的是求解变量Y之间的关系。
3. 采用以下语句options=odeset('outputfcn','odephas2')可以将输出方法改变为相平面输出；
4. 采用以下语句options=odeset('outputfcn','odeprint')输出求解过程每一步的解；
5. 采用以下语句options=odeset('outputsel',[A,B,...])可以将A，B等指定的y的分量输出；



# 解算指令的图形输出

例题6，要求输出CA2和CA3浓度的关系

```
function Cha6demo6_3  
CA10=0.4; CA20=0.2; CA30=0.1;  
stoptime=10;  
opt=odeset('OutputFcn','odephas2','OutputSel',[2,3])  
[t,y]=ode45(@CSTR,[0 stoptime],[CA10 CA20 CA30],opt);  
title('The relationship between CA2 and CA3')  
xlabel('Time (min)')  
ylabel('Concentration')
```

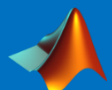




# 解算指令的误差控制

可以采用odeset函数将求解函数的RelTol和AbsTol选项修改至更小的值以获得更高的求解精度。

```
opt=odeset('RelTol' 1e-10)
```



# 解算指令的误差控制

已知Apollo卫星的运动轨迹(x,y)满足以下方程

$$\begin{cases} \frac{d^2x}{dt^2} = 2\frac{dy}{dt} + x - \frac{\mu^*(x+\mu)}{r_1^3} - \frac{\mu(x-\mu^*)}{r_2^3} \\ \frac{d^2y}{dt^2} = -2\frac{dx}{dt} + y - \frac{\mu^*y}{r_1^3} - \frac{\mu y}{r_2^3} \end{cases}$$

其中,  $\mu=1/82.45$ ,  $\mu^*=1-\mu$ ,  $r_1=\sqrt{(x+\mu)^2+y^2}$ ,  $r_2=\sqrt{(x-\mu^*)^2+y^2}$

初始值:  $x(0)=1.2$ ,  $x'(0)=0$ ,  $y(0)=0$ ,  $y'(0)=-1.04935751$

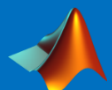
试按以下要求, 在 $t=[0\ 20]$ 的范围内求解上述方程

1. 使用ode45函数, 采用默认设置, 输出x,y的数值并绘制运动轨迹x~y的关系;
2. 更改求解函数的相对精度为 $1e-8$ 再次求解, 比较两次求解是否相同

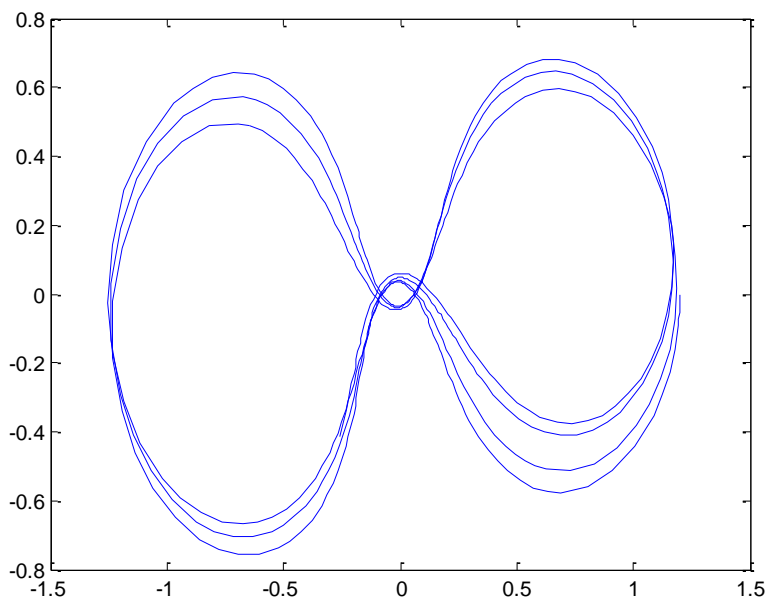


# 解算指令的误差控制

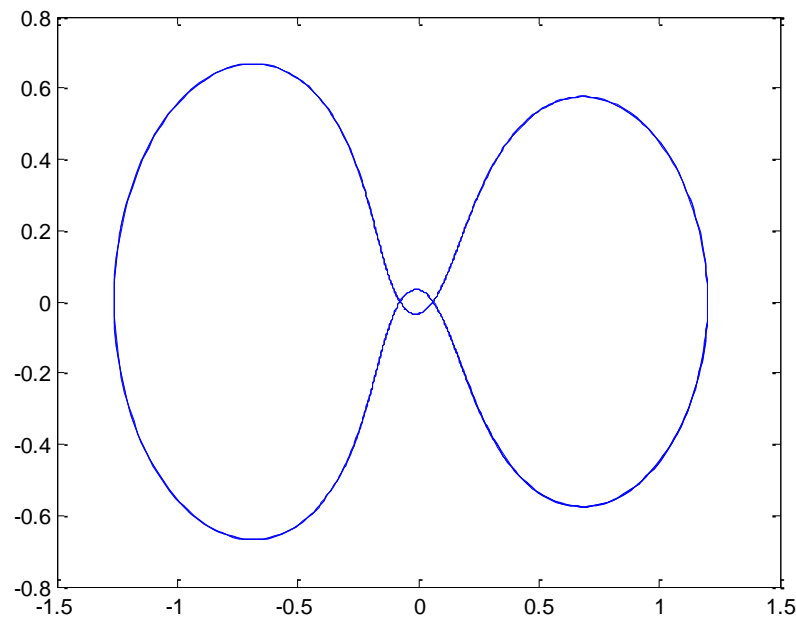
```
function SolveApollo
x0=[1.2 0 0 -1.04935751];
[t,y]=ode45(@ApolloEq,[0,20],x0);
plot(y(:,1),y(:,3))
opt=odeset('RelTol',1e-8)
[t2,y2]=ode45(@ApolloEq,[0,20],x0,opt);
figure
plot(y2(:,1),y2(:,3))
%-----
function f=ApolloEq(t,x)
miu=1/82.45;miu2=1-miu;
r1=sqrt((x(1)+miu)^2+x(3)^2);
r2=sqrt((x(1)-miu2)^2+x(3)^2);
f=zeros(4,1);
f(1)=x(2);
f(2)=2*x(4)+x(1)-miu2*(x(1)+miu)/r1^3-miu*(x(1)-miu2)/r2^3;
f(3)=x(4);
f(4)=-2*x(2)+x(3)-miu2*x(3)/r1^3-miu*x(3)/r2^3;
```



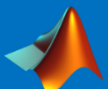
# 解算指令的误差控制



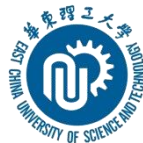
低精度时的求解结果



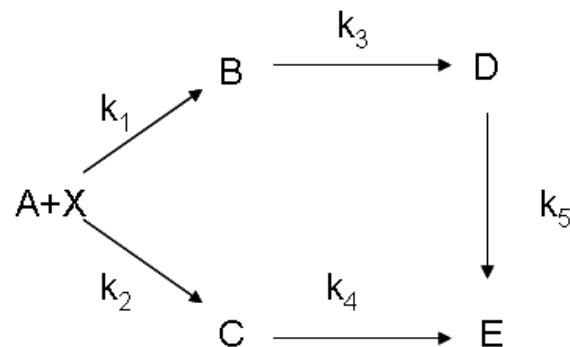
高精度时的求解结果



# 例题



在间歇反应器中进行液相反应制备产物B，反应网络如图所示。反应可在180~260°C的温度范围内进行，反应物X大量过剩，而C、D和E为副产物。各反应均为一级动力学关系： $r = -kC$ ，式中  $k = k_0 \cdot \exp(-E_a/RT)$



已知：

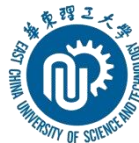
$k_{01} = 5.78052 \times 10^{10}$ ,  $k_{02} = 3.92317 \times 10^{12}$ ,  $k_{03} = 1.64254 \times 10^4$ ,  $k_{04} = 6.264 \times 10^8$ ,  $E_{a1} = 124670$ ,  $E_{a2} = 150386$ ,  $E_{a3} = 77954$ ,  $E_{a4} = 111528$ ,  $k_5 = 2.1667 \times 10^{-4}$ 。

初始浓度： $C_A = 1 \text{ kmol/m}^3$ ，其余物质浓度为0。已知使产物B收率最大的最优反应温度为224.6°C

试计算1)在最优反应温度下各组分浓度随时间的动态变化；  
2)最优反应时间；3)输出产物D对反应物浓度A的关系图。



# 例题



数学模型

$$\frac{dC_A}{dt} = -(k_1 + k_2)C_A$$

$$\frac{dC_B}{dt} = k_1C_A - k_3C_B$$

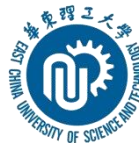
$$\frac{dC_C}{dt} = k_2C_A - k_4C_C$$

$$\frac{dC_D}{dt} = k_3C_B - k_5C_D$$

$$\frac{dC_E}{dt} = k_4C_C + k_5C_D$$



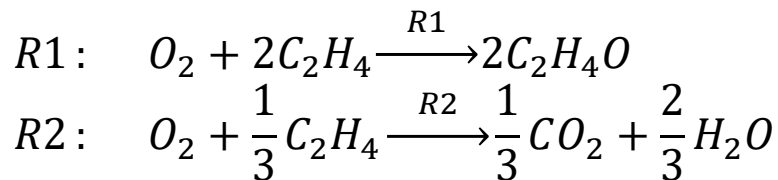
# 例题



```
function Cha6demo4
T = 224.6 + 273.15; R = 8.31434;
k0 = [5.78052E+10  3.92317E+12  1.64254E+4  6.264E+8];
Ea = [124670  150386  77954  111528];
C0 = [1  0  0  0  0]; tspan = [0 1e4];
opt=odeset('outputfcn','odephas2','outputsel',[1;4])
[t,C] = ode45(@MassEquations, tspan, C0,opt,k0,Ea,R,T)
plot(t,C(:,1),'r-',t,C(:,2),'k:',t,C(:,3),'b-.',t,C(:,4),'k--');
xlabel('Time (s)');ylabel('Concentration (kmol/m^3)');
legend('A','B','C','D')
CBmax = max(C(:,2));
yBmax = CBmax/C0(1)
index = find(C(:,2)==CBmax);
t_opt = t(index)           % t_opt: the optimum batch time, s
% -----
function dCdt = MassEquations(t,C,k0,Ea,R,T)
k = k0.*exp(-Ea/(R*T)); k(5) = 2.16667E-04;
rA = -(k(1)+k(2))*C(1); rB = k(1)*C(1)-k(3)*C(2); rC = k(2)*C(1)-
k(4)*C(3);
rD = k(3)*C(2)-k(5)*C(4); rE = k(4)*C(3)+k(5)*C(4);
dCdt = [rA; rB; rC; rD; rE];
```



# 固定床反应器一维拟均相模型求解



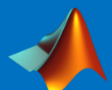
$$R_1 = 810k_1C_{O_2}, \quad k_1 = 35.2 \exp(-59860/R/T)$$

$$R_2 = 2430k_2C_{O_2}, \quad k_2 = 24700 \exp(-89791/R/T)$$

采用一维拟均相模型求解产物环氧乙烷浓度和反应温度沿反应器管长的变化。反应器模型如下：

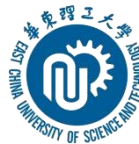
$$\left\{ \begin{array}{l} u_s \frac{dC_{C_2H_4}}{dz} = -\left(2R_1 + \frac{1}{3}R_2\right) \\ u_s \frac{dC_{O_2}}{dz} = -(R_1 + R_2) \\ u_s \frac{dC_{C_2H_4O}}{dz} = 2R_1 \\ u_s \rho_f C_p \frac{dT}{dz} = (-\Delta H_1 R_1 - \Delta H_2 R_2) - \frac{4U_w}{d_t} (T - T_w) \end{array} \right.$$

求解所需参数：反应器管径，  
 $dt=0.0508$ ；反应器管长， $L=12$ ；反应  
 气体表观流速， $us=1.3$ ；流体密度，  
 $pf=6.06$ ；流体热容， $Cp=1160$ ；反应  
 器管壁温度， $Tw=498$ ；反应器的总传  
 热系数， $Uw=270$ ；R1反应的反应热，  
 $210000$ ；R2反应的反应热， $473000$ 。  
 反应器初始条件：反应器入口温度，  
 $498$ ；氧气入口浓度， $14$ ，乙烯入口浓  
 度， $224$ ，进口气体中不含任何反应产  
 物。





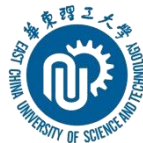
# 例题



```
function EOModelA
%modeling Ethylene oxide reactor by using 1-D pseudo-
homogeneous model
C0=[224 14 0 498];
L=12;%m
[L,C]=ode15s(@modelA,[0 L],C0);
plot(L,C(:,3),'Linewidth',2);
xlabel('Bed Length [m]','fontsize',16)
ylabel('EO Concentration [mol/m^3]','fontsize',16)
set(gca,'FontSize',16)
figure
plot(L,C(:,4),'Linewidth',2)
xlabel('Bed Length [m]','fontsize',16)
ylabel('Bed Temperature [K]','fontsize',16)
set(gca,'FontSize',16)
```



# 例题



```
function dCT=modelA(z,C)
CEH=C(1);CO2=C(2);CEO=C(3);T=C(4);
us=1.3;%m/s
dt=0.0508;%m
rhoulf=6.06;%kg/m3;
cp=1160;%J/kg/K;
Tw=498;%K
Uw=270;%W/m2/K
dH1=-210000;%J/mol;
dH2=-473000;%J/mol
R=8.314;
k1=35.2*exp(-59860/R/T);
k2=24700*exp(-89791/R/T);
R1=810*k1*CO2;
R2=2430*k2*CO2;
dCT=zeros(4,1);
dCT(1)=-(2*R1+1/3*R2)/us;
dCT(2)=-(R1+R2)/us;
dCT(3)=2*R1/us;
dCT(4)=( (-dH1*R1-dH2*R2) -4*Uw*(T-Tw)/dt)/(us*rhouf*cp);
```

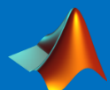


# 事件与求解区间

1. 对于操作型问题的微分方程，求解区间实际隐含在实际的操作和设备参数中；对于设计型问题的微分方程，则求解区间未知；对于设计型问题的求解可以通过定义options选项中的事件（Events）来求解；
2. 所谓的事件是指待用户自定义函数到达、离开或通过零点。当求解函数监测到这些事件发生时，可以选择终止求解或仅记录这些事件发生时的时刻，并输出事件发生时的自变量和因变量值
3. 可以通过如下语句定义求解函数的事件选项：

**opt=odeset('Events',@Events) ;**

其中@Events为用户自定义事件函数的句柄。

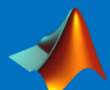


# 事件函数的定义

- 自定义的事件函数应具有以下声明语句：

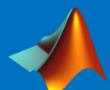
```
function [value, isterminal, direction]=events(t,y)
```

- 其中的输出变量value, isterminal和direction都是向量，其第i个元素的值对应于第i个事件函数。
- value(i)是第i个事件函数的值；
- 当value(i)为零时，isterminal(i) = 1表示终止求解，如不终止求解则等于0；
- 默认direction(i)=0表示所有的零点都为事件发生点，direction(i)=+1表示仅事件函数增加过程中的零点为事件发生点，direction(i)=-1则为函数递减过程中的零点为事件发生点；

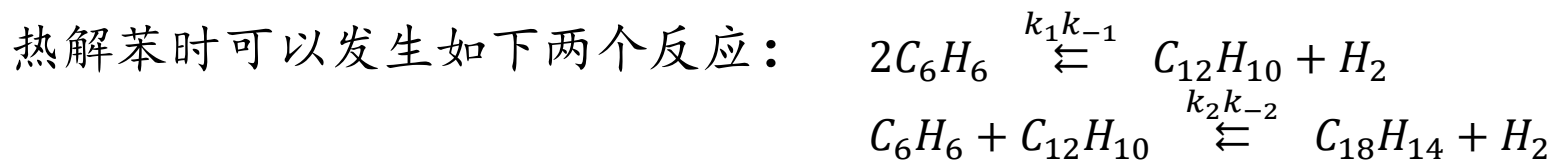
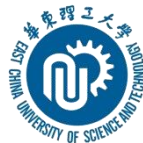


# 事件函数的定义

- 当定义了事件函数且事件发生时，求解函数可以返回返回3个附加输出变量：事件发生的时刻、事件发生时的函数值、求解函数探测到事件发生的类型。
- 在调用求解函数使用`[T,Y,TE,YE,IE] = solver(odefun,tspan,y0,options)`的格式时，TE、YE和IE就是以上三个附加输出变量。
- 当使用`sol = solver(odefun,tspan,y0,options)`时，3个附加变量分别为`sol.xe`、`sol.ye`和`sol.ie`。



# 例题10



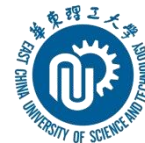
此时反应物浓度随时间变化的规律满足以下微分方程组：

$$\begin{cases} dC_B/dt = -2 * r1 - r2 \\ dC_D/dt = r1 - r2 \\ dC_T/dt = r2 \\ dC_H/dt = r1 + r2 \end{cases} \quad \begin{cases} r1 = k_1 \left( C_B^2 - \frac{C_D C_H}{K_1} \right) \\ r2 = k_2 \left( C_B C_D - \frac{C_T C_H}{K_2} \right) \end{cases}$$

$k_1 = 7e5 \text{ L}/(\text{mol} \cdot \text{h})$ ,  $k_2 = 4e5 \text{ L}/(\text{mol} \cdot \text{h})$ ,  $K_1 = 0.31$ ,  $K_2 = 0.48$ ,  $C_B$ 的初始值为 $0.0117 \text{ mol/L}$ , 其余浓度初始值为 $0$ 。试计算以上条件下, 苯转化率为 $50\%$ 时所需的反应时间。



# 例题10



```
function Cha6demo10
```

```
%事件使用示例
```

```
global CB0
```

```
CB0=0.0117;C0=[CB0,0 0 0];
```

```
opt=odeset('Events',@stoptime);
```

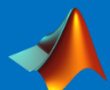
```
[tout Cout]=ode45(@C6H6Pyro,[0 1000],C0,opt);
```

```
plot(tout,Cout)
```

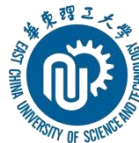
```
tend=tout(end) %返回时间节点的最后一个即为终止时间
```

```
CBin=CB0
```

```
CBout=Cout(end,1)
```



# 例题10



```
%----- (续上页) -----  
function dC=C6H6Pyro(t,C)  
k1=7e5;k2=4e5;K1=0.31;K2=0.48;  
r1=k1*(C(1)^2-C(2)*C(4)/K1);  
r2=k2*(C(1)*C(2)-C(3)*C(4)/K2);  
dC=zeros(4,1);  
dC(1)=-2*r1-r2;dC(2)=r1-r2;dC(3)=r2;dC(4)=r1+r2;  
%-----  
function [value Termin Direct]=stoptime(t,C)  
%事件函数  
global CB0  
value=(CB0-C(1))/CB0-0.5; %事件函数的值  
Termin=1; %事件函数值为0时终止求解  
Direct=0;
```



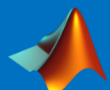


# 练习

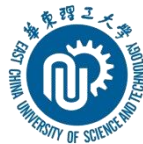
编写一个名为exer的函数求解常微分方程，将t和y的关系以图形方式输出。

$$\frac{d^2y}{dt^2} + y = 1 - \frac{t^2}{\pi}, \quad t \in [-2, 7]$$

初始条件：  $y(-2)=-5, y'(-2)=5$



# 作业



公共邮箱下载文档：work11.pdf，直接打印  
、完成后上交

