Matlab编程与应用
第六讲

中国科技大学信息学院
陆伟
luwei@ustc.edu.cn

本讲内容

- part1: Matlab多项式运算
- part2: 微分方程数值解法
- part3: 符号计算

Part1:

Matlab多项式运算

Matlab中多项式表示方法：

- n 阶多项式用长度为 n+1的向量表示，
  缺少的幂次项系数为 0！

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

Matlab中表示为向量： $[a_n, a_{n-1}, \ldots, a_1, a_0]$

例：$2x^3 - x^2 + 3$ → [2 -1 0 3]

系数中的0不能省！

- 多项式显示： `poly2sym(p)`

  或者 `poly2str(p,'x')`

多项式加减运算

多项式加减就是其所对应的系数向量的加减运算

- 对于次数相同的多项式，可以直接对其系数向量进行加减运算；
- 如果两个多项式次数不同，则应该把低次多项式中系数不足的高次项用 0 补足，然后进行加减运算。

例：

$$p_1 = 2x^3 - x^2 + 3 \longrightarrow [2, -1, 0, 3]$$

$$p_2 = 2x + 1 \longrightarrow [0, 0, 2, 1]$$

$$p_1 + p_2 = 2x^3 - x^2 + 2x + 4 \longleftarrow [2, -1, 2, 4]$$

多项式加减运算

- 练习：编一个自动补零的多项式加减函数

  p = mypolyadd(a,b)

  输入：a,b为待相加多项式对应的向量（可以是列向量）

  输出：相加后多项式对应的向量。

多项式乘法、除法运算

- 多项式乘法运算：`k = conv(p,q)`

例：计算多项式 $2x^3 - x^2 + 3$ **和** $2x+1$ **的乘积**

```
>> p=[2,-1,0,3];
>> q=[2,1];
>> k=conv(p,q);
```

- 多项式除法运算：`[k,r] = deconv(p,q)`

其中 k 返回的是多项式 p 除以 q 的商，r 是余式。

`[k,r]=deconv(p,q)` **<==>** `p=conv(q,k)+r`

多项式求导

- polyder

```
k=polyder(p)  ： 多项式p的导数；
k=polyder(p,q)： p*q的导数；
[k,d]=polyder(p,q)： p/q的导数，k是分子，d是分母
```

例：已知 $p(x)=2x^3-x^2+3$， $q(x)=2x+1$，

求 $p',(p\cdot q)',(p/q)'$

```
>> k1=polyder([2,-1,0,3]);
>> k2=polyder([2,-1,0,3],[2,1]);
>> [k2,d]=polyder([2,-1,0,3],[2,1]);
```

多项式积分

- **polyint**

k=polyint(p,a) ： 多项式 p 的积分，积分后常数项为a;

k=polyint(p):多项式 p 的积分，积分后常数项为0;

例：先对 $p(x)=2x^3-x^2+3$ **求导，再积分。**

```
>> p = polyder([2,-1,0,3]);
>> k1=polyint(p);
>> k2=polyder(p,3);
```

多项式求根

- **r = roots(p)**

例：求多项式 $p(x)=2x^3-x^2+3$的根

```
>> p = [2 -1 0 3];
>> r = roots(p)
```

- 已知多项式的根，构建对应的多项式

```
>>p1 = poly(r);
```

多项式求值

- 计算多项式在给定点的值
  - 代数多项式求值

`y=polyval(p,x)`：计算多项式 p 在 x 点的值

注：若 x 是向量或矩阵，则采用数组运算 (点运算)！

例：已知 $p(x)=2x^3-x^2+3$，分别取 x=2 和一个 2×2 矩阵，求 $p(x)$ 在 x 处的值

```
>> p=[2,-1,0,3];
>> x=2; y=polyval(p,x)
>> x=[-1, 2;-2,1]; y=polyval(p,x)
```

矩阵多项式求值

```
Y=polyvalm(p,X)     % X 必须是方阵
```

例：已知 $p(x)=2x^3-x^2+3$，则

```
polyvalm(p,A) = 2*A*A*A - A*A + 3*eye(size(A))
polyval(P,A) = 2*A.*A.*A - A.*A + 3*ones(size(A))
```

```
>> p=[2,-1,0,3];
>> x=[-1, 2;-2,1];polyval(p,x)
>> polyvalm(p,x)
```

部分分式展开

```
[r,p,k] = residue(b,a)
```

$$\frac{b(s)}{a(s)} = \frac{b_1 s^m + b_2 s^{m-1} + b_3 s^{m-2} + \ldots + b_{m+1}}{a_1 s^n + a_2 s^{n-1} + a_3 s^{n-2} + \ldots + a_{n+1}}$$

$$\frac{b(s)}{a(s)} = \frac{r_1}{s - p_1} + \frac{r_2}{s - p_2} + \ldots + \frac{r_n}{s - p_n} + k(s)$$

部分分式展开

例：

$$\frac{b(s)}{a(s)} = \frac{5s^3 + 3s^2 - 2s + 7}{-4s^3 + 8s + 3}$$

```
b = [ 5 3 -2 7]
a = [-4 0 8 3]
[r, p, k] = residue(b,a)
```

部分分式展开

例：

$$H(s) = \frac{s^2}{s+1}$$

```
b = [ 1 0 0]
a = [ 1 1]
[r, p, k] = residue(b,a)
```

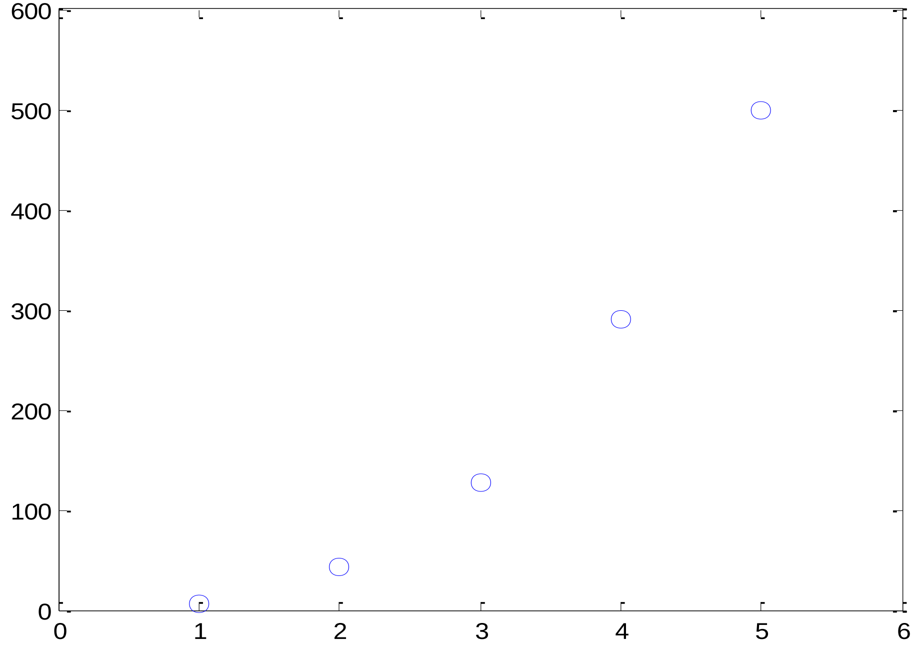
部分分式展开

- 若已知展开后的系数，想要还原原来的多项式，仍然用residue函数

```
[b,a] = residue(r,p,k)
```

多项式数据拟合

- 曲线拟合：用数学函数y=f(x)来表达一组数据的变化规律。
- 若f(x)为多项式，也称为多项式拟合。



多项式数据拟合

- 假定我们要用一阶多项式拟合：$f(x) = a_1x+a_0$
- 问题：如何选择多项式系数a_1,a_2？
- 准则：均方误差最小；
- 方法：最小二乘法。

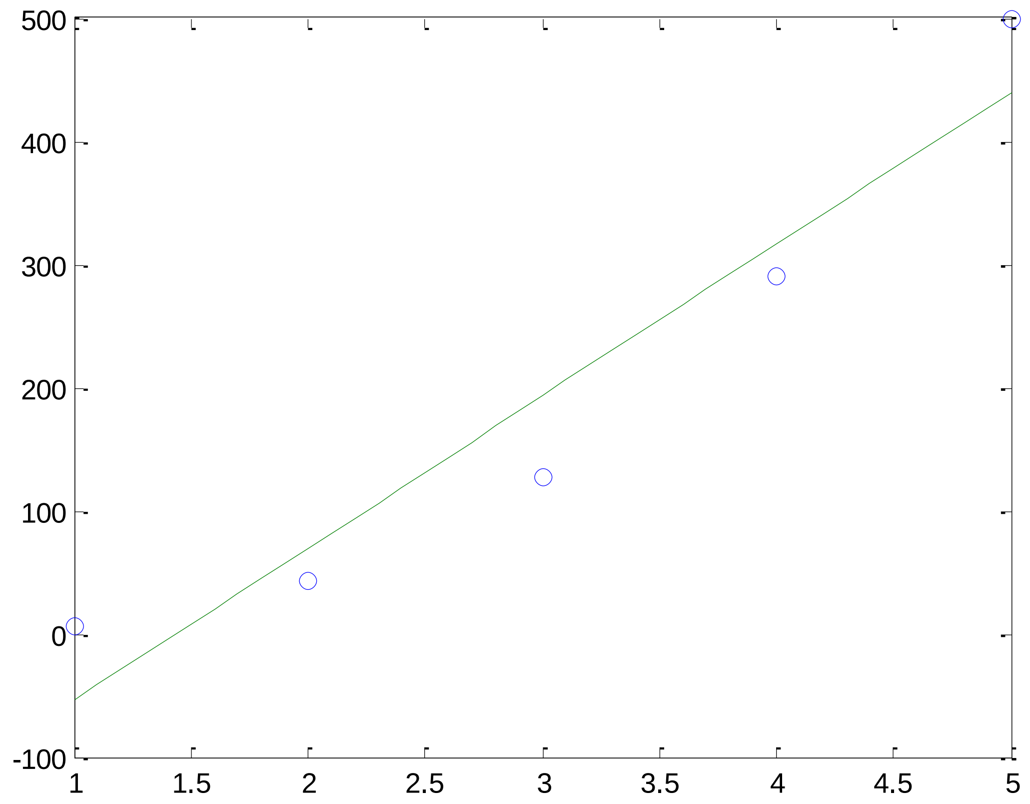
```
p = polyfit(x,y,n)
```

x,y:已知数据点的横纵坐标，
n ： 事先确定的多项式阶次
p：返回的多项式系数

多项式数据拟合

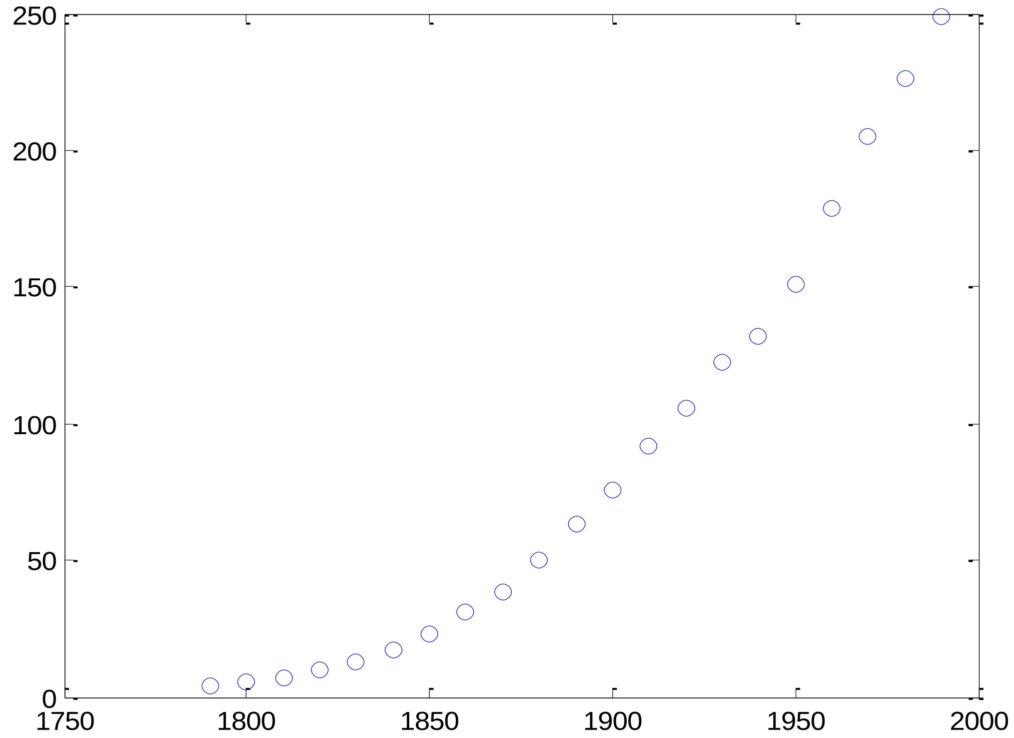
```
x = [1 2 3 4 5];
y = [5.5 43.1 128 290.7 498.4];
p = polyfit(x,y,1);
x2 = 1:0.1:5;
y2 = polyval(p,x2);
plot(x,y,'o',x2,y2);
grid on
yy = polyval(p,x);
err = sum((y-yy).^2);
```

多项式数据拟合



多项式数据拟合

- 美国人口在1790年至1990年（10年一期）的人口数量统计数据。（census.mat）

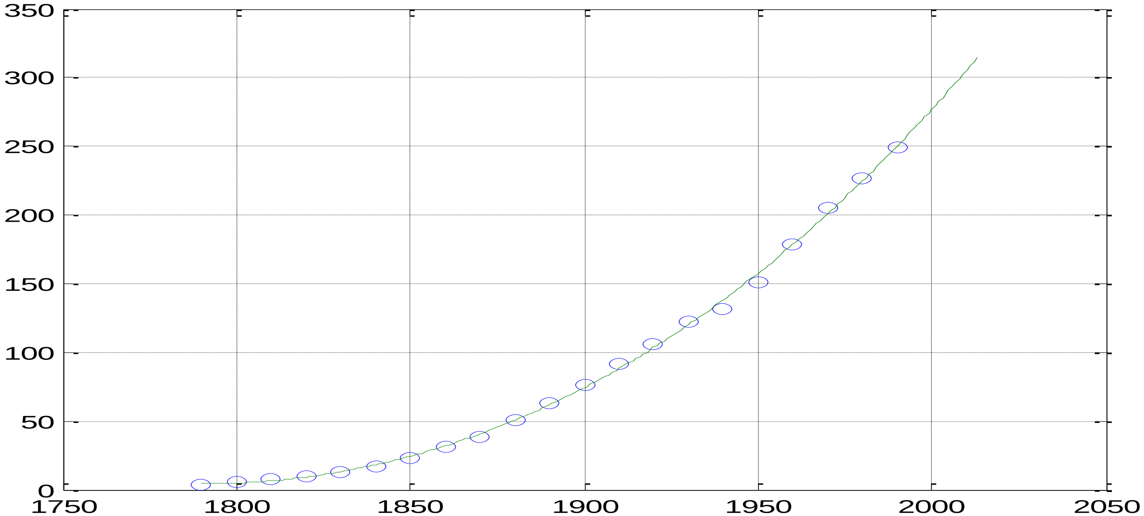


多项式数据拟合

- 希望预测美国在2013年的总人口

```
load census.mat
p3 = polyfit(cdate,pop,3);
cdate2 = 1790:2013;
pop2 = polyval(p3,cdate2);
plot(cdate,pop,'o',cdate2,pop2);
pop2013 = polyval(p3,2013)
pop2013 =

  314.2938
```

多项式数据拟合



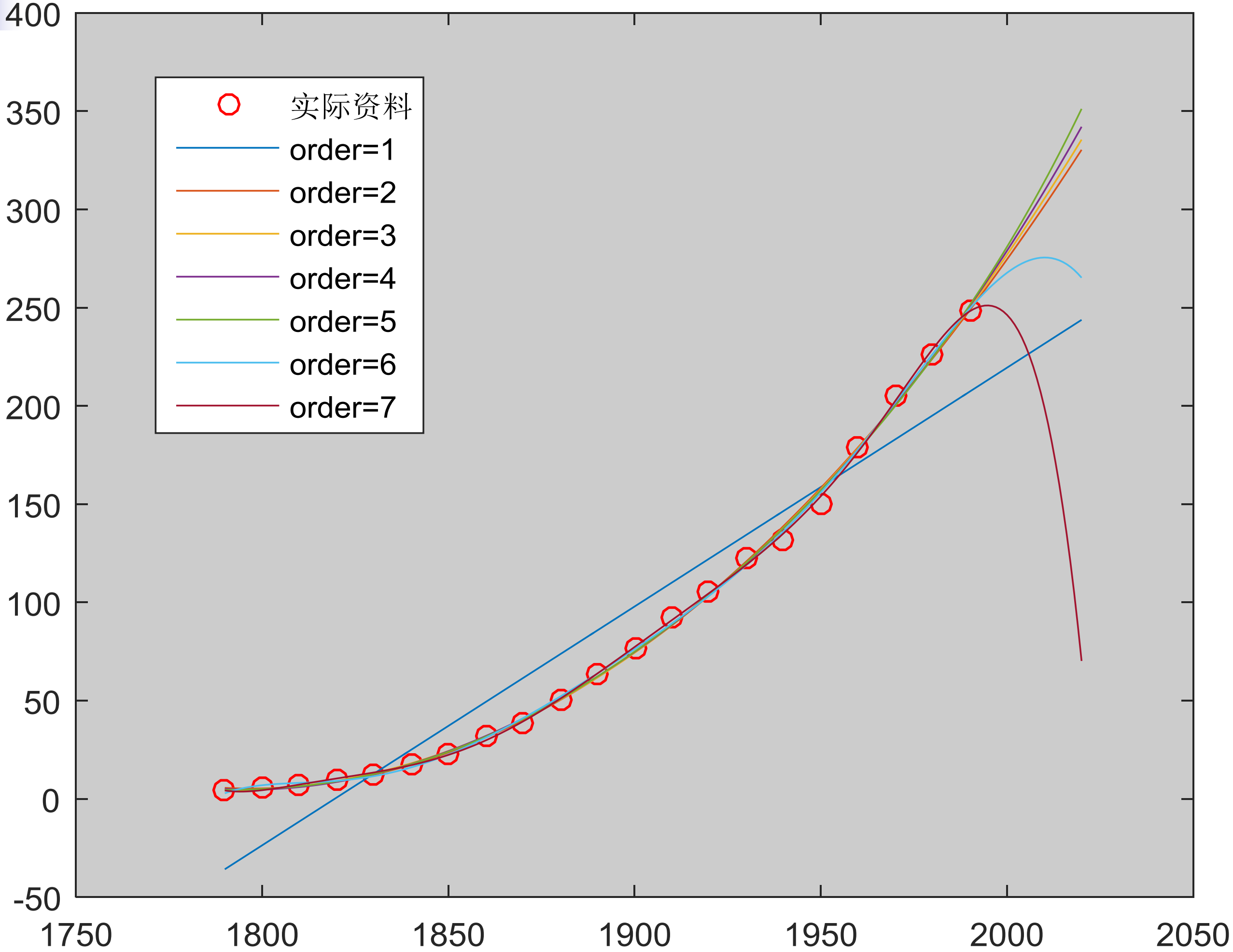
```
>> pop2(end)

ans =     3.263193232696340e+02
```

多项式数据拟合

- 选择不同阶次多项式拟合，预测结果差异很大。

```
load census.mat
cdate2 = min(cdate):(max(cdate)+30);
curve = zeros(7, length(cdate2));
for i = 1:7
curve(i,:)=polyval(polyfit(cdate,pop,i),cdate2);
end
plot(cdate, pop,'o‘),hold on,
plot(cdate2, curve);
legend(‘实际资料’, 'order=1', 'order=2','order=3',
'order=4', 'order=5', 'order=6', 'order=7');
```

多项式数据拟合



实际资料
order=1
order=2
order=3
order=4
order=5
order=6
order=7
400
350
300
250
200
150
100
50
0
-50
1750
1800
1850
1900
1950
2000
2050

Part2:

微分方程数值解法

微分方程数值解法

- Matlab中包含一套完整的微分方程数值解法的算法程序
  - ODE：常微分方程（初值）
  - BVP：边界值问题的常微分方程
  - DDE：延迟微分方程
  - IDE：隐微分方程
  - PDE：偏微分方程
- 其符号工具箱(Symbolic Math Toolbox)提供了求微分方程解析解的算法程序

微分方程数值解法

- 本节讲解Matlab中初值问题常微分方程的数值解法。

  ODE: ordinary differential equations

  Initial Value Problems for ODEs

微分方程数值解法

- 微分方程数值解法介绍--- Euler 折线法

- 考虑一维经典初值问题

$$\frac{dy}{dx} = f(x, y)\ ,\ y(x_0) = y_0\ ,\ x \in [a, b]$$

方程

初值

求值区间

微分方程数值解法

- 基本思想：用差商代替微商

根据 Talyor 公式，$y(x)$ 在点 x_k 处有

$$y(x) = y(x_k) + (x - x_k)y'(x_k) + O(\Delta x^2)$$

$$y(x_{k+1}) = y(x_k) + hy'(x_k) + O(h^2)$$

$$\boxed{h = x_{k+1} - x_k}$$

$$\boxed{\left.\frac{dy}{dx}\right|_{x_k} = \frac{y(x_{k+1}) - y(x_k)}{h} + O(h) \approx \frac{y(x_{k+1}) - y(x_k)}{h}}$$

微分方程数值解法

- **具体步骤：分割求解区间，差商代替微商，解代数方程**
  - **分割求解区间**

    等距剖分：$\boxed{a = x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n = b}$

    步长：$\boxed{h = x_{k+1} - x_k = (b-a)/n, \quad k = 0,1,2,\ldots,n-1}$

  - **差商代替微商**

$$\left.\frac{dy}{dx}\right|_{x_k} \approx \frac{y(x_{k+1}) - y(x_k)}{h} \Rightarrow \boxed{y(x_{k+1}) \approx y(x_k) + h\,y'(x_k)}$$

得方程组：

$$\begin{cases} y_0 = y(x_0) \\ y_{k+1} = y_k + h\,f(x_k, y_k) \\ x_{k+1} = x_k + h \end{cases}$$

微分方程数值解法

例：用 Euler 法解初值问题

$$\begin{cases} \dfrac{dy}{dx} = y + \dfrac{2x}{y^2} & x \in [0,2] \\ y(0) = 1 \end{cases}$$

解：取步长 $h = (2-0)/n = 2/n$，得差分方程

$$\begin{cases} x_0 = 0, \quad y_0 = 1 \\ y_{k+1} = y_k + h f(x_k, y_k) = y_k + h(y_k + 2x_k / y_k) \\ x_{k+1} = x_k + h \end{cases}$$

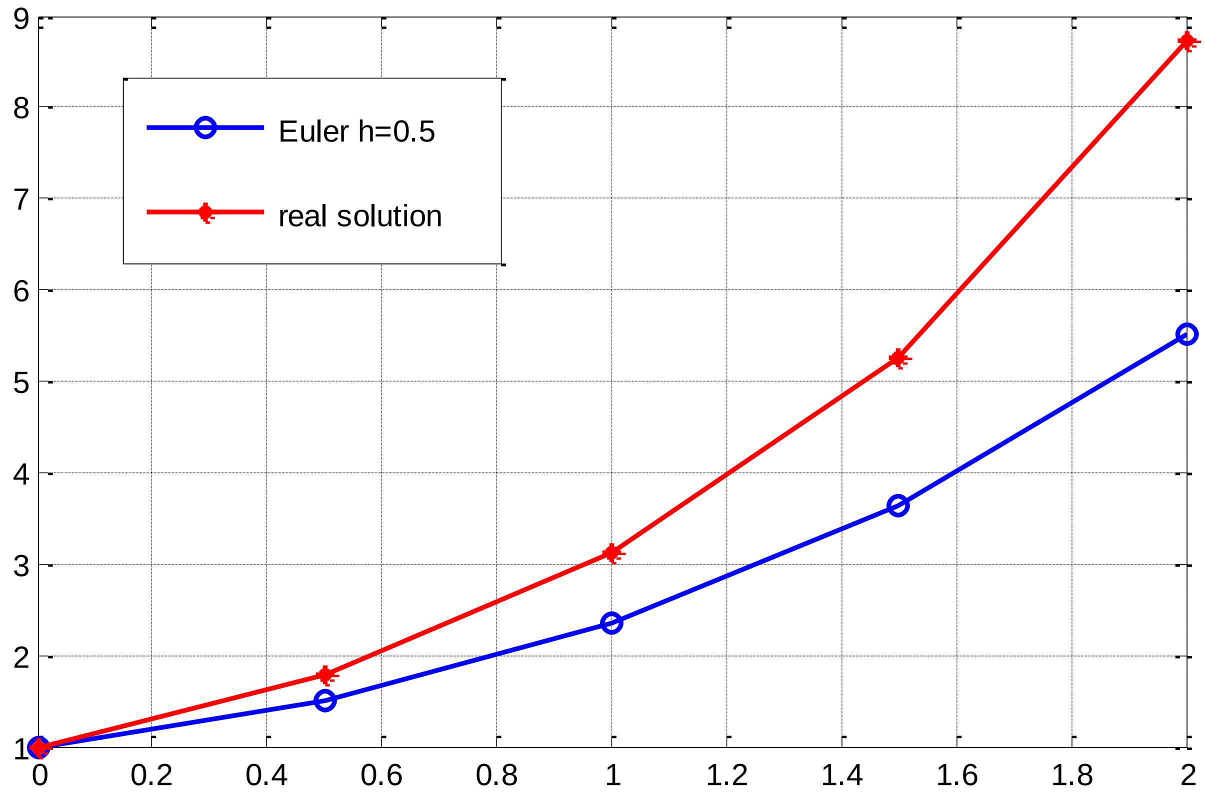
练习：编写**matlab**程序，用**Euler**法解上述微分方程

微分方程数值解法

```
a = 0; %[a b]为积分区间
b = 2;
n = 5; %中间点数
h = (b-a)/(n-1); %步长
x = linspace(a,b,n);
y = zeros(size(x));
y(1) = 1;
for k = 1:n-1
   y(k+1) = y(k) +h *(y(k)+x(k)/(y(k).^2));
end
plot(x,y,'o-');
```

微分方程数值解法

解析解： $$y=\left(\frac{5}{3}e^{3x}-2x-\frac{2}{3}\right)^{1/3}$$

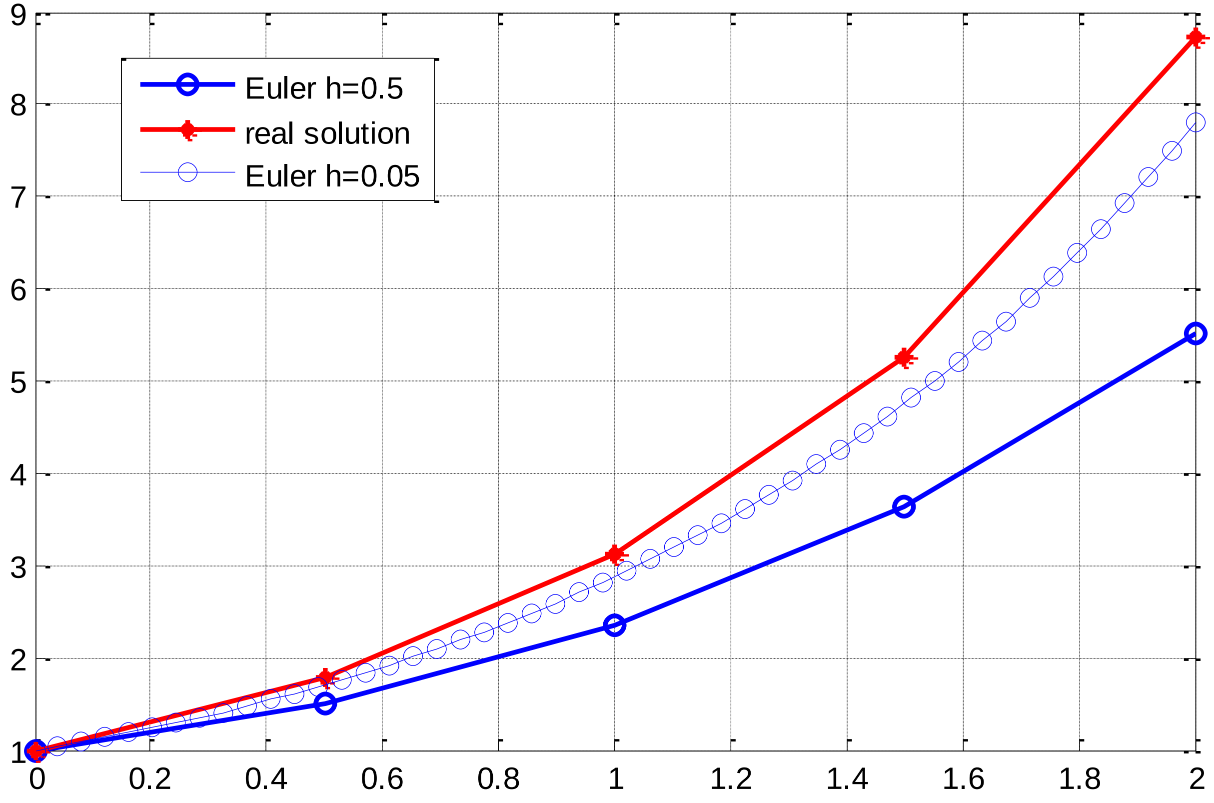


- 为了减小误差，可采用以下方法：
  - 让步长 h 取得更小一些；
  - 改用具有较高精度的数值方法：

    Runge-Kutta (龙格-库塔) 方法

微分方程数值解法

- 步长减小，精度会提高。



微分方程数值解法

四阶R-K方法

$$\begin{cases} y_0 = y(x_0), \quad x_{k+1} = x_k + h \\ y_{k+1} = y_k + h\,(L_1 + 2L_2 + 2L_3 + L_4)/6 \end{cases}$$

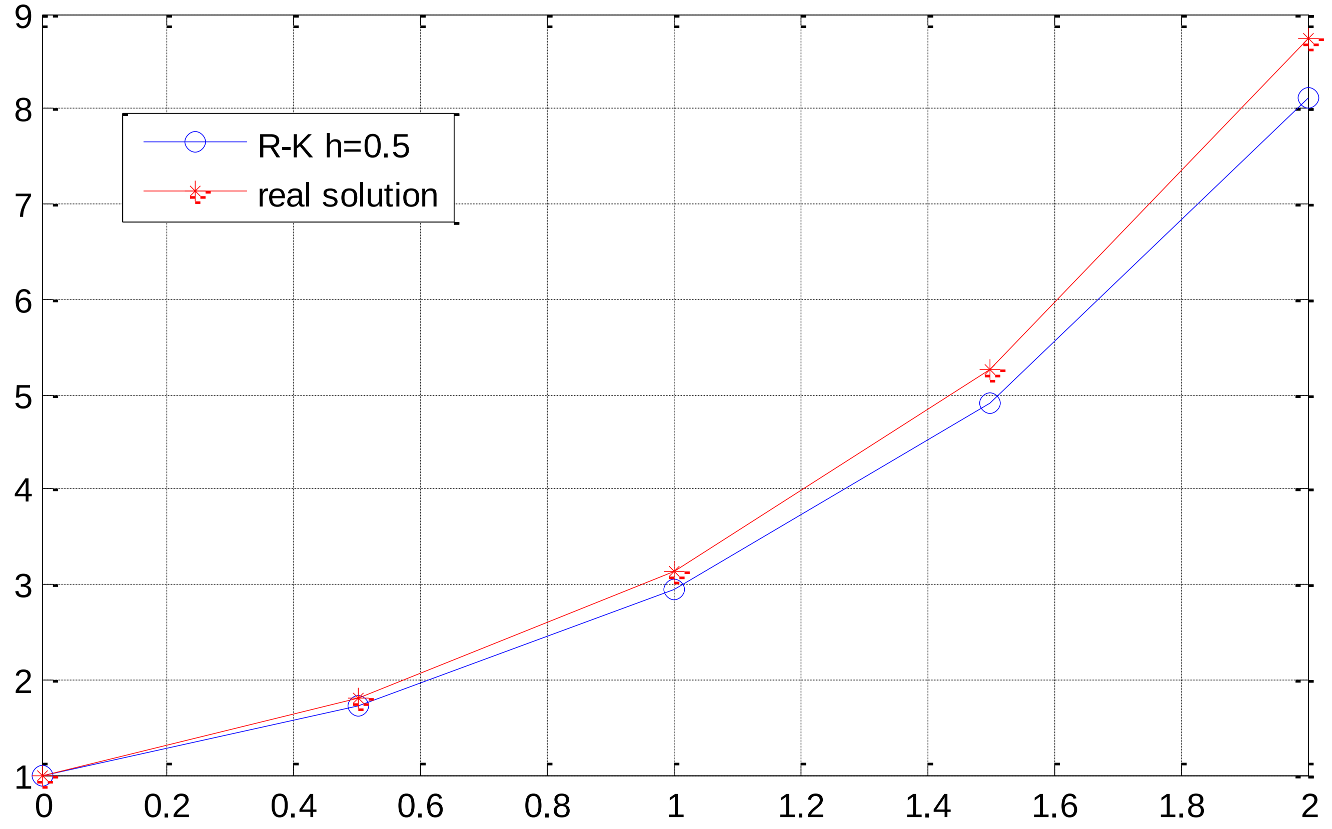
其中

$$\begin{cases} L_1 = f(x_k, y_k) \\ L_2 = f(x_k + h/2, y_k + hL_1/2) \\ L_3 = f(x_k + h/2, y_k + hL_2/2) \\ L_4 = f(x_k + h, y_k + hL_3) \end{cases}$$

微分方程数值解法

```
a = 0;  b = 2; n = 50; h = (b-a)/(n-1);
x = linspace(a,b,n); y = zeros(size(x));
y(1) = 1;
 f = @(u,v) u+v/u^2;
for k = 1:n-1
    L1 = f(y(k),x(k));
    L2 = f(y(k)+h*L1/2, x(k)+h/2);
    L3 = f(y(k)+h*L2/2, x(k)+h/2);
    L4 = f(y(k)+h*L3,   x(k)+h  );
    y(k+1) = y(k) +h *(L1+2*L2+2*L3+L4)/6;
end
```

微分方程数值解法



微分方程数值解法

- 用 **Maltab自带函数** 解初值问题

  - **求解析解：** `dsolve`

  - **求数值解：**

    `ode45`、`ode23`、

    `ode113`、`ode23t`、`ode15s`、

    `ode23s`、`ode23tb`

微分方程数值解法

```
[T,Y] = solver(odefun,tspan,y0)
```

其中 y0 为初值条件，tspan为求解区间；Matlab在数值求解时自动对求解区间进行分割，T (向量) 中返回的是分割点的值(自变量)，Y (向量) 中返回的是解函数在这些分割点上的函数值。

solver 为Matlab的ODE求解器（可以是 ode45、ode23、ode113、ode15s、ode23s、ode23t、ode23tb）

没有一种算法可以有效地解决所有的 ODE 问题，因此 MATLAB 提供了多种ODE求解器，对于不同的ODE，可以调用不同的求解器。

微分方程数值解法

求解器	ODE类型	特点	说明
ode45	非刚性	单步法；4，5 阶 R-K 方法；累计截断误差为 $(\triangle x)^3$	大部分场合的首选方法
ode23	非刚性	单步法；2，3 阶 R-K 方法；累计截断误差为 $(\triangle x)^3$	使用于精度较低的情形
ode113	非刚性	多步法；Adams算法；高低精度均可到 $10^{-3}\sim10^{-6}$	计算时间比 ode45 短
ode23t	适度刚性	采用梯形算法	适度刚性情形
ode15s	刚性	多步法；Gear's 反向数值微分；精度中等	若 ode45 失效时，可尝试使用
ode23s	刚性	单步法；2 阶Rosebrock 算法；低精度	当精度较低时，计算时间比 ode15s 短
ode23tb	刚性	梯形算法；低精度	当精度较低时，计算时间比ode15s短

微分方程数值解法

例： **求初值问题** $\begin{cases} \dfrac{dy}{dx} = -2y + 2x^2 + 2x \\ y(0) = 1 \end{cases}$ **的数值解，求解范围为 [0,0.5]**

```
%fun=inline('-2*y+2*x^2+2*x','x','y');
fun = @(x,y) -2*y+2*x^2+2*x;
[x,y]=ode23(fun,[0,0.5],1);
```

注：也可以在 tspan 中指定对求解区间的分割，如：

```
[x,y]=ode23(fun,[0:0.1:0.5],1); %此时 x=[0:0.1:0.5]
```

微分方程数值解法

如果需求解的问题是高阶常微分方程，则需将其化为一阶常微分方程组，此时需用函数文件来定义该常微分方程组。

例： 求解 Ver der Pol 初值问题
$$\begin{cases} \dfrac{d^2y}{dt^2}-\mu(1-y^2)\dfrac{dy}{dt}+y=0 \\ y(0)=1,\quad y'(0)=0,\quad \mu=7 \end{cases}$$

令 $x_1=y, x_2=\dfrac{dy}{dt}$，**则原方程可化为**

$$\begin{cases} dx_1/dt=x_2 \\ dx_2/dt=\mu(1-x_1^2)x_2-x_1 \\ x_1(0)=1,\quad x_2(0)=0,\quad \mu=7 \end{cases}$$

微分方程数值解法

- 先编写函数文件 verderpol.m

```
function xprime=verderpol(t,x)
global mu;
xprime=[x(2); mu*(1-x(1)^2)*x(2) - x(1)];
```

- 再编写脚本文件 vdpl.m，在命令窗口直接运行该文件。

```
clear;
global mu;mu=7;
y0=[1;0];
[t,x]=ode45('verderpol',[0,40],y0);
plot(t,x(:,1),'r-');
```

Part3:

符号计算

内容

- 符号计算与数值计算
- 符号计算基础
- 积分
- 求导
- 代数方程求解
- 微分方程求解
- 傅立叶变换
- 拉普拉斯变换

符号计算基础

- 符号数学工具箱(Symbolic Math Toolbox)
- 对函数表达式进行微分、积分等运算，求解代数方程、微分方程。结果为相应解析解。
- 先定义符号对象（符号变量、符号表达式），再调用相应符号函数进行推理，得到解析解。

符号对象的建立

- 符号变量定义：sym函数，syms函数
- a = sym(‘a’);
- syms c x t alpha ; %注意变量间不能有逗号！

注意：pi,i,j 不能作为符号变量！

符号对象的建立

- 符号表达式建立：

  1. 利用sym函数直接建立

  (不推荐，以后版本可能不支持)

例：$x^a e^{-x}$

```
f=sym('x^a*exp(-x)'); %注意单引号！
```

2. 使用已定义符号变量组成

```
>> syms  x ,a
>> f = x^a*exp(-x)*sin(2*pi*x)
>> pretty(f)
```

变量替换

- subs(f,x,a)

  将符号表达式f中的符号变量x替换为a

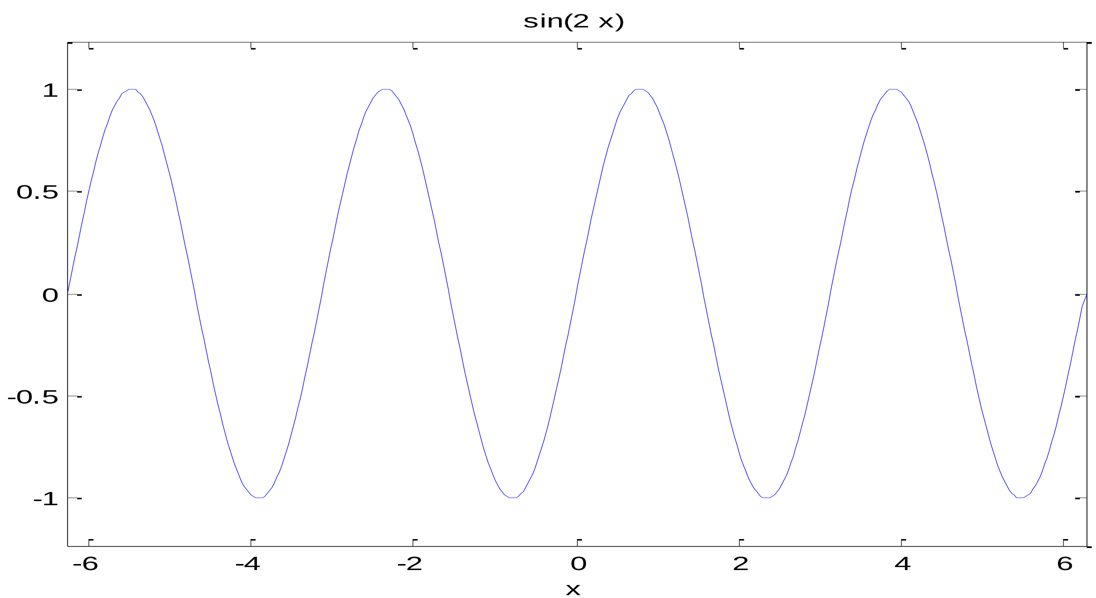
  a可以为 数值/符号变量/符号表达式

```
syms a b c d;
A = [ a b ;c d];
x=det(A);
x1 = subs(x,a,3);
x2 = subs(x,[ b c d] ,[1 7 4]);
x3 = double(x2); %把符号变量变成双精度数
```

函数画图

- ezplot(f)

```
>> syms x
>> ezplot(sin(2*x))  %默认横坐标范围[-2pi,2pi]
```



函数画图

- 画由参数方程x(t),y(t)决定的曲线：

  ezplot(x,y)

```
 syms t
 x = t*sin(5*t);
 y = t*cos(5*t);
ezplot(x,y)
```

函数求导（Differentiation）

- diff(f) 对函数f求导

```
>> syms x
>> f = sin(x^2)
>> df =diff(f)
```

-
```
>> syms x t
>> diff(x*t^2))
>> diff(x*t^2,t)
```

函数求导（Differentiation)

- diff(f) 对函数f求导

- $\frac{df(x)}{dx}$, $f(x) = \sin(5x)$
-

```
>> syms x
>> f = sin(5*x);
>> diff_f = diff(f)
```

函数求导（Differentiation）

- $\frac{d^2 f(t)}{dt^2}, f(t) = \sin(5t)$

```
>> syms t
>> f = sin(5*t);
>> diff_f = diff(f,2) %对f求二阶导数
```

函数求导（Differentiation）

- $\frac{df(x)}{dx}$, $f(x) = \mathrm{e}^{-\mathrm{ax}}\sin(bx)$

```
>> syms a b x
>> f = exp(-a*x)*sin(b*x);
>> diff_f = diff(f)  %对变量x求导

>> subs(f,a,2)     %将变量a替换为2
>> subs(f,[a b],[2 3])

>> diff_f2 = diff(f,a) % 对变量a求导
```

找出符号函数中的符号变量

- symvar函数

```
syms x y a b
f(a, b) = a*x^2/(sin(3*y - b));
symvar(f)
```

不定积分(indefinite integral)

- int(expr) 对函数表达式expr进行不定积分

$$\int (1+t)dt$$

```
>> syms t
>> int_f=int(1+t)
int_f =
(t*(t + 2))/2
```

不定积分(indefinite integral)

- int(expr,v) 对函数表达式expr进行不定积分,指定积分变量为v

$$\int \frac{x}{1+z^2}dx$$

$$\int \frac{x}{1+z^2}dz$$

```
>> syms x z
>> f = x/(1+z^2)
>> int_f1 = int(f)  %默认积分变量为x
>> int_f2 = int(f,z)%指定积分变量为z
```

定积分(definite integral)

- int(expr,a,b) 对函数表达式expr在区间[a，b]求定积分

$$\int_0^6 x dx$$

```
>> syms x
>> f = x;
>> int_f = int(f,0,6)
```

定积分(definite integral)

$$\int_{-6}^{6} x^2 \cos(x)\,dx$$

```
>> syms x
>> f = x^2*cos(x);
>> int_f = int(f,-6,6)
int_f =
24*cos(6) + 68*sin(6)
>> double(int_f)        %转换为双精度数
ans =   4.043833002081825
```

求曲线 e^{-x} 关于 x 轴旋转得到的旋转体在 $1 \leqslant x \leqslant 2$ 内的体积

$$\int_a^b \pi[f(x)]^2 dx$$

```
syms x  a b
f = exp(-x)
intf = int(pi*f^2 , a ,b )
V = subs(intf,[a b],[1 2])
v = double(V);
```

数值积分

q = quad(fun,a,b)　fun:被积函数；
a,b:积分区间
（不推荐）

- >>g = @(x) exp(sin(x));
- >>quad(g,0,10)
- >>quad('exp(sin(t))',0,10);

数值积分

q = integral(fun,xmin,xmax) fun:被积函数;
xmin,xmax :积分区间

例：求积分$f(x) = e^{-x^2}(lnx)^2$ $[0,+\infty]$

- >>f = @(x) exp(-x.^2).*(log(x)).^2;
- >>integral(f,0,+inf)

二重数值积分

```
q = integral2(fun,xmin,xmax,ymin,ymax)
```

例：求积分$f(x)=\frac{1}{\sqrt{x+y}(1+x+y)}$

积分区间：$0 \le x \le 1, 0 \le y \le 1-x$

```
f = @(x,y) 1./(sqrt(x+y).*(1+x+y));
ymax = @(x) 1-x;
integral2(f,0,1,0,ymax);
```

定积分(definite integral)

- 计算卷积：

$$y(t) = x(t) * h(t) = \int_{-\infty}^{+\infty} x(\tau)h(t-\tau)d\tau$$

设 $x(t) = h(t) = e^{-t^2/2}$

```
syms t tau
y = int(exp(-tau^2/2)*exp(-(t-tau)^2/2), -inf,inf)
y = 2^(1/2)*pi^(1/2)*exp(-tau^2/2)        %不对！
y = int(exp(-tau^2/2)*exp(-(t-tau)^2/2),tau,-inf,inf)
y = pi^(1/2)*exp(-t^2/4)                  %OK！
```

定积分(definite integral)

计算卷积：$e^{-t^2/2} * e^{-t^2/2}$

```
>> syms t tau
>> x = @(t)exp(-t^2/2)
>> f =x(tau)*x(t-tau)
>> y = int(f,tau,-inf,inf)
```

定积分(definite integral)

$$x(t) = \cos(\omega_0 t)$$
$$h(t) = e^{-6t}u(t)$$

```
>> syms t tau w0 real
>> f = cos(w0*(t-tau))*exp(-6*(tau))*heaviside(tau)
>> int(f,tau,-inf,inf)

ans =
(6*cos(t*w0) + w0*sin(t*w0))/(w0^2 + 36)
```

- heaviside(x) 阶跃函数

$$u(t) = \begin{cases} 1 & t > 0 \\ 0 & t < 0 \end{cases}$$

- dirac(x) 狄拉克函数

$$\delta(t) = 0 \quad t \neq 0$$

$$\int_{-\infty}^{+\infty} \delta(t) dt = 1$$

多重不定积分

$$\iiint xy^2z^5dxdydz$$

```
>> syms x y z
>> int(int(int(x*y^2*z^5,x),y),z)

ans =

(x^2*y^3*z^6)/36
```

多重定积分

$$\int_1^2 \int_2^4 x^2 y dx dy$$

```
>> syms x y
>> int(int(x^2*y,x,2,4),y,1,2)

ans =

28
```

代数方程求解

- solve(eqn) 求解代数方程eqn

$$x^2 + 3x + 2 = 0$$

```
>>S = solve('x^2+3*x+2') %默认为 ...=0
>> S(1)  %S是一个数组
ans = -2
>> S1 = solve('x^2+3*x+2=0') %结果一样
```

代数方程求解

$$x^2 + 3x + 2 = 0$$

```
>> syms x
>> solve(x^2+3*x+2)  %表达式不加单引号
>> solve(x^2+3*x+2=0) %出错
>> solve(x^2+3*x+2==0) %正确
>> solve(3*x == -x^2-2) %可以
```

代数方程求解

- solve(eqn，var) 指定变量var为未知数

$$ax^2 + bx + c = 0$$

```
>> syms a b c x
>> solve(a*x^2+b*x+c == 0)  %默认变量为x
>> solve(a*x^2+b*x+c == 0，c)  %指定变量为c
```

代数方程求解

$$x^3 + 1 = 0$$

```
>> syms x   %默认x为复数
>> solve(x^3+1 == 0)
>> pretty(ans)
```

```
>> syms x  real  %规定x为实数
>> solve(x^3+1 == 0)   %得到一个实根
```

```
>> syms x  positive  %规定x为正实数
>> solve(x^3+1 == 0)   %答案为空
```

代数方程求解

求解代数方程组：

solve(eqn1,eqn2,…,eqnN,var1,var2,…,varN‵)

2x-3y+4z = 5

y+4z+x = 10

-2z+3x+4y = 0

```
>> syms x y z
>> eq1 = '2*x-3*y+4*z=5‵
>> eq2='y+4*z+x=10‵
>> eq3='-2*z+3*x+4*y=0‵
>> [x,y,z] = solve(eq1,eq2,eq3,x,y,z)
```

常微分方程求解

- S = dsolve(eqn)

例： $\frac{dy}{dt} = ty$

```
>> syms y(t)  %定义一个符号函数
>> dsolve(diff(y)==t*y)
ans =
C2*exp(t^2/2)   %C2 是待定系数，需要初始条件确定
```

```
>>y(t) = dsolve(diff(y) == t*y, y(0) == 2)
y(t) =
2*exp(t^2/2)
```

常微分方程求解

- S = dsolve(eqn)

例： $$\frac{d^2y}{dx^2} = \cos(2*x) - y$$

```
syms y(x)
Dy = diff(y);
y(x) = dsolve(diff(y, 2) == cos(2*x) - y, y(0) == 1, Dy(0) == 0);
y(x) = simplify(y)
y(x) =
1 - (8*sin(x/2)^4)/3
```

傅立叶变换

- *Fourier正变换*

$$F(jw) = \int_{-\infty}^{+\infty} f(t)e^{-jwt}dt$$

fourier(f,trans_var,eval_point)

trans_var时域变量，eval_point频域变量w

- *Fourier逆变换：*

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(jw)e^{jwt}dw$$

ifourier(F,trans_var,eval_point)

trans_var频域变量w，eval_point时域变量

傅立叶变换

求高斯函数$f(x) = e^{-x^2}$的fourier变换

```
>>syms x w
>>f = exp(-x^2);
>>fourier(f)
ans =
pi^(1/2)*exp(-w^2/4)
```

傅立叶变换

求方波信号的fourier变换

```
f = heaviside(t+2)- heaviside(t-2);
ezplot(f,[-3,3]) %画出时域波形

ft = fourier(f)
pretty(ft)
ft2 = simplify(ft)
figure
ezplot(ft)
```

拉普拉斯变换

- *Laplace正变换*

$$F(s) = \int_{0}^{+\infty} f(t)e^{-st}dt$$

laplace(f,trans_var,eval_point)

trans_var默认为t，eval_point默认为s

- *Laplace逆变换：*

$$f(t) = \frac{1}{2\pi j}\int_{\sigma-j\infty}^{\sigma+j\infty} F(s)e^{st}ds$$

ilaplace(F,trans_var,eval_point)

trans_var默认为s，eval_point默认为t

拉普拉斯变换

```
>> syms s t
>> laplace(exp(-t))

ans =

1/(s + 1)
>> ilaplace(F)

ans =
exp(-t)
```

- 注：本讲内容大部分引自华东师范大学课程《数学实验》