



符号计算是指在运算时，无须事先对变量进行赋值，而是将所有得到的结果以标准的符号形式表示。符号计算是以符号对象和符号表达式作为运算对象的表示方法，在运算过程中得到的解析值是一个精确的数学表达式，它不会受到计算误差累积问题的影响，其计算命令较为简单，相比数值计算的结果减少了占用资源。多项式是在符号表示的基础上建立的表达式，使用 MATLAB 动态系统建模、仿真与分析时，常使用多项式表示。自动控制系统模型描述中的传递函数均使用多项式描述，在此基础上对系统再进行仿真分析。

3.1 MATLAB 中的符号运算

符号运算与数值运算的区别是：数值运算中必须先对变量赋值才能参与运算。符号运算无须事先对变量赋值，运算结果以标准的符号形式表示，即数值运算矩阵变量中不允许有未定义的变量，而符号运算中可以含有未定义的符号变量。

3.1.1 符号变量与符号表达式

在数学表达式中，一般习惯于使用排在字母表中前面的字母作为变量的系数，而排在后面的字母表示变量。例如：

$$f(x)=ax^2+bx+c$$

表达式中的 a, b, c 通常被认为是常数，用作变量的系数；而将 x 看作自变量。MATLAB 提供了 `sym()` 和 `syms()` 两个建立符号对象的函数。

1. Sym()函数用法

`Sym()` 函数用于建立单个符号对象，语法格式：

符号对象名=`sym`(A)

其中：

(1) A 为符号对象。它可以是数值常量、数值矩阵或数值表达式，不加单引号为一个符

号常量；加单引号时，该符号对象为一个符号变量。

例如：

```
x=sym(a); y=sym('x'); z=sym('y');    %a 为符号常量，x、y 为符号变量
```

(2) 当需要定义多个字符变量时，可添加需要变量的行数、列数，以生成多行多列的字符变量矩阵，方法见【实战练习 3-2】。

(3) 使用 sym() 函数能够将数据的值进行精确的保留，不必担心计算的误差等问题。

例如：计算 $1/123456 \times (4/5)$ 时，分别使用下面两种方法，其结果精度不同。

```
d1=1/123456*(4/5)
d2=1/sym(123456)*(4/5)
```

结果分别为：

```
d1 =    6.4800e-06
d2 =    1/154320
```

2. syms 函数用法

syms 函数用于一次定义多个符号变量，语法格式：

```
syms 符号变量名 1 符号变量名 2...符号变量名 n
```

其中：变量名不能加单引号，相互之间用空格隔开。

例如：要同时定义四个符号变量 a 、 b 、 c 、 d ，则可以输入如下命令：

```
syms a b c d;    或 syms('a','b','c','d')
```

说明：

(1) syms() 函数可以替换 sym() 函数，定义符号变量的类型时可使用：syms x positive; 限定 x 为正数，若要取消该限定，可用命令 syms x clear;

(2) syms 建立符号矩阵时，可在符号后添加行值和列值，使用方法见【实战练习 3-1】。

3. 符号及操作

例如：

定义符号变量并建立符号表达式，命令如下：

```
x=sym('x'); y=sym('y'); z=sym('z');
a=[1,3,5]; b=[3,7,9]; c=[11,12,13];
Y=a*x+b*y+c*z
```

结果：

```
Y = [ x + 3*y + 11*z, 3*x + 7*y + 12*z, 5*x + 9*y + 13*z]
```

4. 查找符号变量函数 symvar()

语法格式：

```
symvar(f,n)    %f 为符号表达式，n 为查找的符号变量个数，默认为查找所有符号变量
```

例如：

```
syms x y z a
f=2*x-3*y+z^2+5*a
symvar(f)
```

%查找所有符号变量

```
symvar(f,1)           %查找一个符号变量
symvar(f,3)           %查找三个符号变量
```

结果:

```
[ a, x, y, z]
x
[ x, y, z]
```

【实战练习 3-1】建立符号矩阵

建立 3×4 的符号矩阵 A 和 2×3 的符号矩阵 B 。

编程代码如下:

```
A=sym('a',[3 4])
syms B [2,3]
```

结果:

```
A =[a1_1, a1_2, a1_3, a1_4]
    [a2_1, a2_2, a2_3, a2_4]
    [a3_1, a3_2, a3_3, a3_4]
B =[B1_1, B1_2, B1_3]
    [B2_1, B2_2, B2_3]
```

5. 符号变量与符号表达式

语法格式:

```
F='符号表达式'
```

例如: $F = \sin(x) + 5x$

说明: F 为符号变量名, $\sin(x) + 5x$ 为符号表达式, $'$ 为符号标识, 符号表达式一定要用单引号引起来 MATLAB 才能识别。引号内容既可以是符号表达式, 也可以是符号方程。

例如:

```
f1='a*x^2+b*x+c'      %二次三项式
f2='a*x^2+b*x+c=0'    %方程
f3='Dy+y^2=1'         %微分方程
```

符号表达式或符号方程可以赋给符号变量, 以后调用方便, 也可以不赋给符号变量而直接参与计算。

6. 符号矩阵的创建

语法格式:

```
A=sym(' [      ]')
```

说明: 符号矩阵输入内容同数值矩阵, 必须使用 `sym` 命令定义, 且需用单引号 (`' '`) 标识, 若定义数值矩阵必须是数值, 否则不能识别, 例如:

$A = [1, 2; 3, 4]$ 可以, 但 $A = [a, b; c, d]$ 则出错。应使用如下命令:

```
A = sym(' [a,b;c,d]')
```

结果：

```
A = [ a, b]
     [ c, d]
```

符号矩阵每一行的两端都有方括号，这是与 MATLAB 数值矩阵的一个重要区别。若用字符串直接创建矩阵，需保证同一列中各元素字符串有相同的维度。可以用 `syms` 先定义 `a,b,c,d` 为符号变量再建立符号矩阵，方法是：

```
syms a b c d
A=[a,b;c,d]
```

也可以使用：

```
A = ['[a, b]'; '[c, d]']
```

3.1.2 符号基本运算

因为符号运算不需进行数值运算，不会出现误差，因此符号运算是非常准确的。符号运算可以得出完全封闭解或任意精度的数值解，但符号运算的时间比数值运算速度慢。

1. 符号算术运算

符号运算包括算术运算、关系运算、逻辑运算和因式分解运算等，算术运算与四则运算的数值运算一样，用 `+`、`-`、`*`、`/`、`^` 运算符实现，其运算结果依然是一个符号表达式；关系运算仅列出相应的关系表达式；逻辑运算结果还是逻辑值。

【实战练习 3-2】符号算术运算

定义符号表达式，进行算术与关系运算。

编程代码如下：

```
syms x y;
g1=x^2+2*x+1; %定义符号表达式
g2=3*x^2+7*x+10;
G1=g1+g2
G2=g1-g2
G3=g1.*g2
G4=g1/g2
G5=g1>=g2
```

结果：

```
G1 = 4*x^2 + 9*x + 11
G2 = - 2*x^2 - 5*x - 9
G3 = (x^2 + 2*x + 1)*(3*x^2 + 7*x + 10)
G4 = (x^2 + 2*x + 1)/(3*x^2 + 7*x + 10)
G5 = 3*x^2 + 7*x + 10 <= x^2 + 2*x + 1
```

2. 符号关系运算与逻辑运算

(1) 符号关系运算。

符号关系运算与数值关系运算一样，包括 6 种运算符：`<`、`<=`、`>`、`>=`、`==`、`~=`。

也可以使用对应的 6 个函数来实现：lt()、le()、gt()、ge()、eq()、ne()。

例如：

```
syms a b c
a > b < c
```

结果：

```
ans = (b < a) < c
```

若参与运算的是符号，其结果是一个符号关系表达式；若参与运算的是符号矩阵，其结果是由符号关系表达式组成的矩阵。

在进行符号对象运算前，可用 assume() 函数对符号对象设置值域，语法格式：

```
assume(条件) 或 assume(表达式,集合)           %设置表达式属于集合中，使用方法见
                                                    %【实战练习 3-4】
```

(2) 符号逻辑运算。

常用的符号逻辑运算有 3 种：& (与)、| (或)、~ (非)。

4 个逻辑运算函数分别为：and(a,b)、or(a,b)、not(a) 和 xor(a,b)。

说明：符号逻辑运算的结果还是逻辑符号表达式，使用方法见【实战练习 3-3】。

【实战练习 3-3】 符号的关系运算与逻辑运算

已知符号 x ，先使用 assume() 函数进行判断，如果 a 是一个正数，用 assume($a > 0$) 进行判断，则结果是 a ；如果 a 是一个负数，用 assume($a < 0$) 进行判断，则结果是 $-a$ 。

编程代码如下：

```
syms x;
assume(x < 0);
y1 = abs(x) == x
assume(x, 'positive')
y2 = abs(x) == x
y3 = x > 0 & x < 10
y4 = or(x < 0, x > 10)
```

结果：

```
y1 = -x == x
y2 = x == x
y3 = 0 < x & x < 10
y4 = x < 0 | 10 < x
```

说明：因为 $x < 0$ ，所以 abs(x) 的值 $y1$ 为 $-x$ ； x 为整数，所以 abs(x) 的值 $y2$ 为 x 。

3. 符号替换函数 subs()

使用符号定义了一个符号函数后，subs() 函数可将符号函数中的一些符号替换成其他符号或者其他数值类型。

语法格式：

```
subs(S, old, new)
```

其中：符号表达式 S 可利用 new 中的符号或数值替换 old 中的符号。

例如：

```
f=x^2+y^2
f1=subs(f,y,3)
```

结果：

```
f1=x^2+9
```

【实战练习 3-4】符号表达式的替换

根据符号表达式 $f(x,y)$ 和 f ，分别对下面两个多项式进行符号、数值和多数值替换操作。

$$f(x,y) = \frac{ax^2 + by^2}{c^2}, \quad f = ax^2 + by + c$$

编程代码如下：

```
syms a b c x y m %定义符号运算变量
fxy=(a*x^2+b*y^2)/c^2 %生成符号函数
fxy1=subs(fxy,[a,b,c],[1,2,3]) %数值替换符号
f = a * x^2 + b * y + c; %原表达式
f1 = subs(f, [x y], [sin(x) log(y)]) %符号替换符号
f2 = subs(f, [a b], [2 3]) %数值替换符号
f3 = subs(f, a, 1:4) %多数值替换符号
```

结果：

```
fxy = (a*x^2 + b*y^2)/c^2
fxy1 = x^2/9 + (2*y^2)/9
f1 = a*sin(x)^2 + c + b*log(y)
f2 = 2*x^2 + c + 3*y
f3 = [x^2 + c + b*y, 2*x^2 + c + b*y, 3*x^2 + c + b*y, 4*x^2 + c + b*y]
```

3.2 多项式表示

MATLAB 语言把多项式表达成一个行向量，该向量中的元素是按多项式降幂排列的。

例如：

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

多项式输入时，缺少项的向量系数的相应位置用零补足，在 MATLAB 中建立方法为：

$$P = [a_n + a_{n-1} + \cdots + a_0]$$

3.2.1 直接建立多项式

由于 MATLAB 自动将向量元素按降幂顺序分配给各系数，所有输入按照由高项系数到低项系数依次输入并直接建立多项式，若中间有缺项需要添加零，例如：对于多项式

$$P(x) = x^5 + 3x^4 + 11x^3 + 25x + 36$$

$$P = [1 \ 3 \ 11 \ 0 \ 25 \ 36]$$

3.2.2 使用函数建立多项式

1. 使用sym2poly()函数建立多项式系数

sym2poly()函数将符号多项式转换为多项式系数，由高次项系数到低次项系数依次输出。

例如：

```
syms x;  
sym2poly(7*x^3 + 3*x^2 - 5*x - 18)
```

结果为：

```
7 3 -5 -18
```

2. 使用poly2sym()函数建立符号多项式

poly2sym()函数把多项式系数转换为符号多项式。

例如：

```
sym x;  
poly2sym([3 5 4],x);
```

结果：

```
ans = 3*x^2+5*x+4
```

【实战练习 3-5】符号与多项式的转换

使用 sym2poly()与 poly2sym()函数完成符号与多项式的转换。

(1) 生成表达式： $a = x \sin\left(\frac{\pi}{6}\right) + ex^2 + \ln 100 \cdot x^3 + 1$ 并计算；

(2) 生成 3 次项的符号表达式 c ；

(3) 给定系数【1, -1, -3】生成符号表达式 d 。

编程代码如下：

```
syms x c1 c2 c3 c4  
a=sym2poly(sin(pi/6)*x + exp(1)*x^2+log(100)*x^3+1)  
b = [c1 c2,c3,c4]  
c = poly2sym(b)  
d = poly2sym([-1,1,-30])
```

结果：

```
a = 4.6052 2.7183 0.5000 1.0000  
b = [c1, c2, c3, c4]  
c = c1*x^3 + c2*x^2 + c3*x + c4  
d = - x^2 + x - 30
```

3.3 多项式运算

多项式是一个式子，可以包含数字、变量和运算符号，是代数学中的基础概念。多项式运算除多项式加、减、乘、除运算外，还包括多项式提取、分解、展开、合并同类项等操作。

3.3.1 多项式加、减运算

多项式加、减运算就是其相同幂次系数向量的加、减运算，幂次相同的多项式直接对向量系数进行加、减运算，若两个多项式幂次不同，在低次多项式中不足的高次多项式用零补足，再进行加、减运算。

【实战练习 3-6】多项式加、减运算的应用

已知 $p1=3x^3+5x^2+7$ ， $p2=2x^2+5x+3$ ，请计算： $P1=p1+p2$ 和 $P2=p1-p2$ 并转换为符号表达式。

编程代码如下：

```
p1=[3 5 0 7]; p2=[0 2 5 3];
P1=p1+p2;
P2= p1-p2;
p3=poly2sym(P1)
p4=poly2sym(P2)
```

结果：

```
p3 = 3*x^3 + 7*x^2 + 5*x + 10
p4 = 3*x^3 + 3*x^2 - 5*x + 4
```

加、减运算直接加入加、减运算符即可完成。

3.3.2 多项式乘、除运算

多项式乘、除运算是一个复杂的过程，使用 MATLAB 提供的函数就异常简单，语法格式：

```
k=conv(p,q)           %多项式相乘
[q,r]=deconv(a,b)      %多项式相除，q 为多项式的商，r 为多项式的余数
```

【实战练习 3-7】多项式的乘、除运算的应用

已知 $a=6x^4+2x^3+3x^2+12$ ， $b=3x^2+2x+5$ ，求 $c=a*b$ ， $d=a/b$ 。

编程代码如下：

```
a=[6 2 3 0 12]; b=[3 2 5];
c=conv(a,b);
[d,r]=deconv(a,b);
C=poly2sym(c)
d=poly2sym(d)
r=poly2sym(r)
```

结果：

```
C=18*x^6+18*x^5+43*x^4+16*x^3+51*x^2+24*x+60
d=2*x^2-(2*x)/3-17/9
r=(64*x)/9+193/9
```

说明：多项式相乘就是两个代表多项式的行向量的卷积，两个以上多项式相乘，conv 命令使用嵌套。

例如：计算表达式 $a \times b \times c$ 时，命令为 `conv(conv(a,b),c)`。

3.3.3 多项式分解、展开与提取

1. 多项式分解与展开

语法格式：

```
F=factor(f)           %对 f 多项式进行因式分解，也可用于正整数的分解
F=expand(f)           %对多项式 f 展开
F=collect(f)           %对于多项式 f 中相同变量且幂次相同项合并系数，即合并同类项
F=collect(f,v)         %按变量 v 进行合并同类项
```

2. 提取多项式的分子和分母

如果符号表达式是一个有理分式或可以展开为有理分式，可利用 `numden()` 函数提取符号表达式中的分子或者分母，语法格式：

```
[n,d]=numden(s)        %n 表示分子，d 表示分母
```

【实战练习 3-8】多项式分解与展开的应用

对 f 多项式及常数 y 进行分解并展开显示。

编程代码如下：

```
syms x;
f=x^9-1; f=factor(f)
y = 2025;
y1=factor(y)
y2 = factor(sym(y))
y3=poly2sym(y2)
```

结果：

```
f=[x-1,x^2+x+1,x^6+x^3+1]
y1=3      3      3      3      5      5
y2=[3,3,3,3,5,5]
y3=3*x^5+3*x^4+3*x^3+3*x^2+5*x+5
```

【实战练习 3-9】提取多项式的分子和分母

已知多项式 g_1 、 g_2 ，提取 g_1 和 g_2 多项式的分子和分母。

编程代码如下：

```
syms x y;
g1=x^2+2*x+1; g2=3*x^2+7*x+10; G=g1/g2;
[n,d]=numden(G)
```

结果：

```
n = x^2 + 2*x + 1
d = 3*x^2 + 7*x + 10
```

【实战练习 3-10】提取与展开三角函数多项式

展开三角函数多项式，并对展开的多项式提取其系数及变量。

编程代码如下：

```
syms x y z;
f = sin(2*x)+cos(2*y) ;
f1 = expand(f)
f0 = (z+1)^8; f2 = expand(f0)
[p,x1]=coeffs(f2,'z')
```

结果：

```
f1 = 2*cos(x)*sin(x) + 2*cos(y)^2 - 1
f2 = z^8 + 8*z^7 + 28*z^6 + 56*z^5 + 70*z^4 + 56*z^3 + 28*z^2 + 8*z + 1
p = [ 1, 8, 28, 56, 70, 56, 28, 8, 1]
x1= [ z^8, z^7, z^6, z^5, z^4, z^3, z^2, z, 1]
```

【实战练习 3-11】合并同类项的应用

已知 g_1 、 g_2 的符号表达式，先计算它们的积和商，再合并同类项。

编程代码如下：

```
g1=sym('x^2+2*x+1'); g2=sym('x+1');
G1=g1*g2
G2=g1/g2
R1=collect(G1) %按符号合并同类项
R2=collect(G2)
```

结果：

```
G1 = (x + 1)*(x^2 + 2*x + 1)
G2 = (x^2 + 2*x + 1)/(x + 1)
R1 = x^3 + 3*x^2 + 3*x + 1
R2 = x + 1
```

说明：针对符号乘、除运算可以使用 `collect()` 函数合并结果。

3.3.4 多项式化简

多项式化简语法格式：

```
simplify(S); %对表达式 S 进行化简
```

【实战练习 3-12】化简并计算简单多项式

化简下列表达式：

$$w1 = \frac{a^4}{(a-b)(a-c)} + \frac{b^4}{(b-c)(b-a)} + \frac{c^4}{(c-a)(c-b)}$$

$$w2 = 2\sin(x)\cos(x)$$

编程代码如下：

```
syms a b c x;
w1= a^4/((a-b)*(a-c))+b^4/((b-c)*(b-a))+c^4/((c-a)*(c-b));
w2=2*sin(x)*cos(x);
w11=simplify(w1)
w22=simplify(w2)
```

结果：

```
w11=a^2 + a*b + a*c + b^2 + b*c + c^2
w22=sin(2*x)
```

【实战练习 3-13】化简并计算复杂多项式

化简下列多项式：

$$f1(x) = e^{c \cdot \log \sqrt{a+b}}$$

$$f2(x) = \sqrt[3]{\frac{1}{x^3} + \frac{6}{x^2} + \frac{12}{x} + 8}$$

$$f3(x) = \sin^2(x) + \cos^2(x)$$

编程代码如下：

```
syms a b c x;
f1= exp(c*log(sqrt(a+b))); f2=(1/x^3+6/x^2+12/x+8)^(1/3);
f3 = sin(x)^2 + cos(x)^2 ;
y1 = simplify(f1);y2 = simplify(f2) ;y3= simplify(f3)
```

结果：

```
y1 = (a + b)^(c/2)
y2 = ((2*x + 1)^3/x^3)^(1/3)
y3 = 1
```

3.3.5 表达式之间的转换

符号表达式与数值表达式之间的转换函数包括：

- (1) 利用函数 `sym()` 可以将数值表达式转换成符号表达式。
- (2) 使用 `eval()` 函数可以将符号表达式转换成数值表达式并计算。

例如：将符号表达式转换成数值表达式并计算。

```
s='sin(pi/4)+(1+sqrt(5))/2 '
b=eval(s)
```

结果：

```
s = sin(pi/4)+(1+sqrt(5))/2
b = 2.3251
```

【实战练习 3-14】多项式替换的应用

设 $w1=((a+b)(a-b))^2$ ，在 $w1$ 表达式中，使用 3 替换字符 a ，使用 1、2 分别替换字符 a 、 b 。

编程代码如下：

```
syms a b;
w1 = ((a+b)*(a-b))^2
w2 = subs(w1,a,3)
w3 = subs(w1,[a,b],[1,2])
```

结果：

```
w1 = (a + b)^2*(a - b)^2
w2 = (b - 3)^2*(b + 3)^2
w3 = 9
```

3.3.6 复合函数与反函数

若 $f=f(x)$, 自变量又是 x 的函数 $g=g(y)$, 求 f 对 y 的过程称为复合函数。

若 $y=f(x)$, 自变量又是 x 的函数 $g=g(y)$, 对于 $g(f(x))$ 都有唯一确定的 x , 则计算 $g(f(x))=x$ 称为反函数。

语法格式:

```
compose(f,g)           %返回 f=f(x), g=g(y), 复合函数 f(g(y)) 的运算。
g=finverse(f)          %f 为符号函数表达式, g 为反函数, x 为单变量
```

【实战练习 3-15】复合函数及反函数的应用

已知: $f=\cos(x/t)$; $y=\sin(y/g)$; 求 $f(y)$ 及反函数。

编程代码如下:

```
syms x y g t;
f=cos(x/t); y=sin(y/g);
x1=compose(f,y)
f=log(x);
x2=finverse(f)
```

结果:

```
x1 = cos(sin(y/u)/t)
x2 = exp(x)
```

3.3.7 分数多项式通分

分数多项式通分的语法格式:

```
[N,D]=numden(f)           %f 为分数多项式
```

说明: 对分数多项式 f 通分, N 为通分后的分子, D 为通分后的分母。

【实战练习 3-16】分数多项式通分计算的应用

已知: $f(x)=\frac{x+3}{y+2}+\frac{y-5}{x^2+1}$, 完成通分计算。

编程代码如下:

```
syms x y;
f = (x+3)/(y+2)+(y-5)/(x^2+1);
[N,D]=numden(f)
```

结果:

```
N = x^3 + 3*x^2 + x + y^2 - 3*y - 7
D = (x^2 + 1)*(y + 2)
```

3.4 多项式求解（多项式方程求解）

多项式与多项式方程的区别是：令多项式为零就是多项式方程，因此求多项式的解也是求多项式方程的解。MATLAB 中多项式方程求解主要使用 `roots()` 函数，其返回值包含了多项式参数的所有实数和虚数解；使用 `solve()` 函数主要用于求解多项式方程符号解；使用 `polyval()` 函数用于求多项式数值解。本节包括高次方程数值解、符号方程求解、微分方程求解及数据拟合。

3.4.1 多项式的特征值（多项式的解）

求多项式的解的语法格式：

```
r=roots(p)
```

使用命令 `roots` 可以求出多项式等于零的根，即该命令可以用于求高次方程的解。根用列向量表示。

3.4.2 特征多项式系数

求多项式系数的语法格式：

```
p=poly(r)
```

若已知多项式等于零的解，可使用 `poly` 命令求出相应多项式系数。

说明：

- (1) 特征多项式一定是 $n+1$ 维的。
- (2) 特征多项式第一个元素一定是 1。
- (3) 若要生成实系数多项式，则根中的复数必定对应共轭，生成的多项式向量包含很小的虚部时可用 `real` 命令将其过滤掉。

【实战练习 3-17】求多项式的解及多项式系数

求方程 $x^4-12x^3+25x-16=0$ 的解，并根据解构造多项式系数。

编程代码如下：

```
p=[1 -12 0 25 -16];  
r=roots(p)  
r1=real(r)  
p=poly(r)
```

结果：

```
r = 11.8311 + 0.0000i  
    -1.6035 + 0.0000i  
     0.8862 + 0.2408i  
     0.8862 - 0.2408i  
r1= 11.8311  
    -1.6035
```

```

0.8862
0.8862
p = 1.0000 -12.0000 0.0000 25.0000 -16.0000

```

3.4.3 方程与方程组的符号解

求方程与方程组的符号解的语法格式：

```
solve(eq1, [x, y])
```

%使用 solve() 函数时，需要先定义变量为符号变量。

其中：eq1 是方程或方程组，[x,y]参数是未知数的符号变量。solve()函数返回的是一个结构体数组，每个元素对应一个符号解。

【实战练习 3-18】求方程及方程组的数值解

已知：一元三次方程 $x^3+x^2+x+1=0$ 和二元方程组

$$\begin{cases} x+y=1 \\ 2x-y=4 \end{cases}$$

分别求数值解。

编程代码如下：

(1) 求一元方程解：

```

syms x
f=x^3+x^2+x+1==0;
x=solve(f,x)

```

或

```

syms x
f1=x^3+x^2+x+1;
x=solve(f1)

```

也可以使用句柄求解：

```

f = @(x) x.^3 + x.^2 + x + 1;
x1 = solve(f,x)

```

结果：

```

x = -1
    -1i
    1i

```

(2) 求二元方程解：

```

syms x y
[x,y]=solve(x+y==1,2*x-y==4,x,y)

```

或

```

syms x y
[x,y]=solve(x+y-1,2*x-y-4)

```

结果：

```

x = 5/3
y = -2/3

```

【实战练习 3-19】求微分方程的符号解

已知范德瓦尔斯气体的三个临界参量：

$$\left(P + \frac{a}{V^2}\right)(V-b) = RT, \quad \left(\frac{\partial P}{\partial V}\right)_T = 0, \quad \left(\frac{\partial^2 P}{\partial V^2}\right)_T = 0$$

求 P_c, V_c, T_c 的值。

编程代码如下：

```
clc;clear all;
syms a b R T V
P = R*T/(V-b) - a/V^2
D1 = diff(P,V)
D2 = diff(D1,V)
S = solve(D1,D2,V,T)
Pc = subs(P,[V,T],[S.V,S.T])
Vc = S.V
Tc = S.T
```

结果：

```
Pc = a/(27*b^2)
Vc = 3*b
Tc = (8*a)/(27*R*b)
```

【实战练习 3-20】求方程组的符号解

已知 $\begin{cases} ax-by=1 \\ ax+by=5 \end{cases}$ ，求方程组的符号解。

a,b,x,y 均为符号运算量。在符号运算前，应先将 a,b,x,y 定义为符号运算量。

编程代码如下：

```
syms a b x y;
[x,y]=solve(a*x-b*y-1,a*x+b*y-5,x,y) %以 a,b 为符号常数，x,y 为符号变量
```

结果：

```
x =3/a
y =2/b
```

【实战练习 3-21】求复数乘积多项式的解

已知一个复数表达式为： $z=x+i*y$ ，试求其共轭复数，并求该表达式与其共轭复数乘积的多项式。

为了使乘积表达式 x^2+y^2 非负，这里把变量 x 和 y 定义为实数。

编程代码如下：

```
x=sym('x','real'); %x 为实数
y=sym('y','real'); %y 为实数
z=x+i*y; %定义复数表达式
z1=conj(z) %求共轭复数
z2=expand(z*conj(z)) %求表达式与其共轭复数乘积的多项式
[a,b]=solve(z2)
```

结果：

```
z1 =x - y*1i
z2 =x^2 + y^2
a= 0
b= 0
```

说明：若要去掉 x 的属性，可以使用下面的语句：

```
x = sym('x', 'unreal') %表示将 x 创建为纯格式的符号变量
```

3.4.4 多项式数值解

在 MATLAB 中，多项式在某个参数下的求值用 `polyval()` 函数，其功能是求多项式在给定参数下的值。

语法格式：

```
polyval(p,n) %返回多项式 p 在 n 点的值
```

利用多项式求值函数可以求得多项式在某一点的值。

例如：求方程 $y = -0.0602x^2 + 1.7020x + 0.3096$ 在 x 等于 0, 0.2, ..., 5 时的函数值。

```
p = [-0.0602, 1.702, 0.3096];
xi = linspace(0, 1, 6);
z = polyval(p, xi)
```

结果：

```
z = 0.3096 0.6476 0.9808 1.3091 1.6327 1.9514
```

说明： p 代表的多项式在 xi 的每一个元素处的值， x 可以是一个向量或矩阵。

【实战练习 3-22】求多项式指定的数值解

求多项式 $p = 3x^4 + 8x^3 + 18x^2 + 16x + 15$ 在 $x = [1, 2, 3, 4, 5]$ 的解。

编程代码如下：

```
p1=[3,8,18,16,15];
x=[ 1,2,3,4,5]
p=polyval(p1,x)
```

```
结果：P= 60 231 684 1647 34203
```

3.4.5 多项式拟合

多项式拟合是一种数学技术，实际应用中可以用来拟合多种类型的数据，例如统计数据、科学数据和经济数据等。多项式拟合的基本原理是使用一组多项式函数来拟合一组数据，以便更好地理解数据变化趋势。

语法格式：

```
y=polyfit(x,y,n) %拟合唯一确定 n 阶多项式的系数
```

其中： n 表示多项式最高阶数， x 、 y 为将要拟合的数据，它是用数组的方式输入，输出参数 y 为拟合多项式 $y = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$ ，共 $n+1$ 个系数。 $n=1$ 为拟合直线；`polyfit` 只适合形如 $y = a[k] * x^k + a[k-1] * x^{(k-1)} + \cdots + a[1] * x + a[0]$ 的完全一元多项式数据拟合。

【实战练习 3-23】温度值的直线拟合

使用多项式拟合函数拟合深圳 2023 年 3 月 18 日一天从 12 点到 19 点温度为直线。

时间：12 点—19 点每隔 1 个小时采集一次温度如下。

温度：25℃，24℃，24℃，25℃，24℃，23℃，23℃，22℃。

编程代码如下：

```
clc;
x=12:19;
y=[25,24,24,25,24,23,23,22];
a=polyfit(x,y,1)
y1=polyval(a,x);
plot(x,y,'bp',x,y1,'r-');
title("温度直线拟合");
```

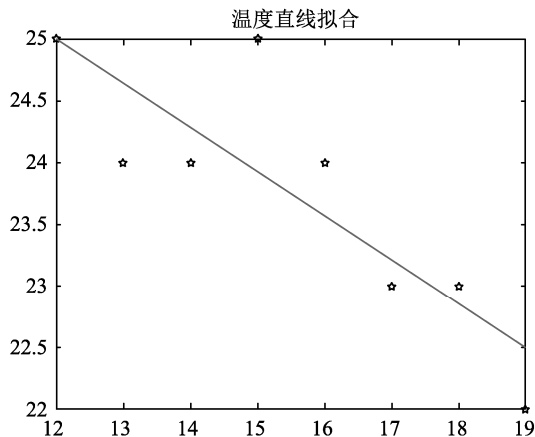


图 3.1 多项式拟合直线

多项式拟合直线如图 3.1 所示。

【实战练习 3-24】求多项式的 2 阶和 4 阶拟合

对 2 阶多项式 x^2+x-30 分别进行 2 阶和 4 阶拟合。

编程代码如下：

```
x=-5:0.5:5;
R=sqrt(2*x.^2);
y=x.^2-x-30;
a=polyfit(x,y,2); %2 阶拟合
b=polyfit(x,y,4); %4 阶拟合
y1=polyval(a,x);
y2=polyval(b,x);
subplot(1,2,1);plot(x,y,'bp',x,y1,'r-');
title("多项式 x^2-x-30 的 2 阶拟合");
subplot(1,2,2);plot(x,y,'bp',x,y2,'r-');
title("多项式 x^2-x-30 的 4 阶拟合");
```

2 阶和 4 阶多项式拟合曲线如图 3.2 所示。

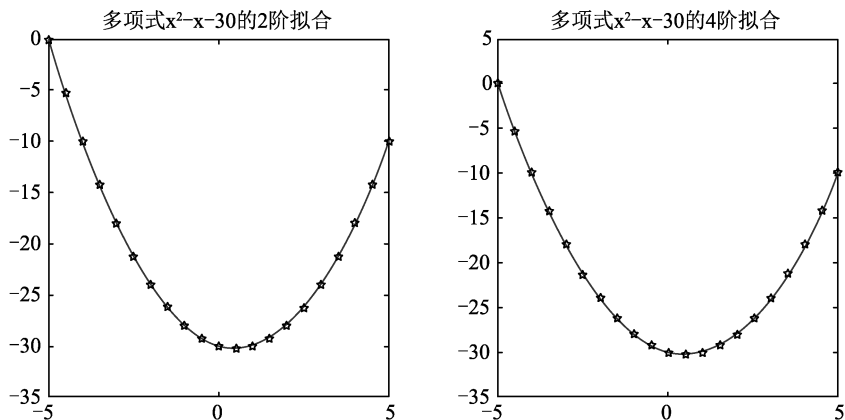


图 3.2 2 阶和 4 阶多项式拟合结果曲线

【实战练习 3-25】求多项式的曲线拟合

设数组 $y = [-0.447 \ 1.978 \ 3.28 \ 6.16 \ 7.08 \ 7.34 \ 7.66 \ 9.56 \ 9.48 \ 9.30 \ 11.2]$ ，在横坐标 0~1 对 y 进行 2 阶多项式拟合，并画图表示拟合结果。

编程代码如下：

```
y = [-0.447 1.978 3.28 6.16 7.08 7.34 7.66 9.56 9.48 9.30 11.2];
x = 0:0.1:1;
y1 = polyfit(x, y, 2);           %进行 2 阶拟合
z = polyval(y1, x);              %计算 y1 为系数向量对 x 点的函数值
plot(x, y, 'r*', x, z, 'b-') ;    % “*” 为原数据点，曲线为拟合结果
```

结果：

2 阶多项式拟合曲线如图 3.3 所示。

plot()函数使用方法见 6.1 节。

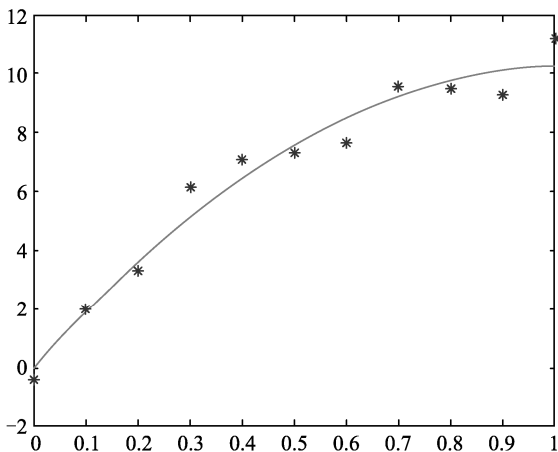


图 3.3 2 阶多项式拟合结果曲线

3.5 多项式求导

多项式求导是在数学中常见的一种操作，它可以计算出多项式的导函数。计算多项式的导函数时，需要对每一项分别进行求导，并根据求导规则将各项相加即可得到导函数。由于多项式是连续函数，它是平滑的，其导数必定也是多项式。

3.5.1 多项式直接求导数

多项式直接求导数的语法格式：

```
k = polyder(p) :           %返回多项式 p 的 1 阶导数系数
k = polyder(p, q) :        %返回多项式 p 与 q 乘积的 1 阶导数系数
[k, d] = polyder(p, q) :    %返回 p/q 的导数，k 是分子，d 是分母
```

【实战练习 3-26】多项式求导的应用

已知 $p1=3x^3+5x^2+7$, $p2=2x^2+5x+3$, 求 $p1$ 导数、 $p1$ 与 $p2$ 乘积和 $p1$ 与 $p2$ 商的导数。

编程代码如下:

```
clc;
p1=[3 5 0 7]; p2=[0 2 5 3];
pp1=polyder(p1)
pp2=polyder(p1,p2) %等价 pp2=polyder(conv(p1,p2))
disp("多项式 p1 的导数");
poly2sym(pp1)
disp("多项式 p1 与 p2 乘积的导数");
poly2sym(pp2)
```

结果:

```
pp1 =      9      10      0
pp2 =     30     100     102     58     35
多项式 p1 的导数
ans=    9*x^2 + 10*x
多项式 p1 与 p2 乘积的导数
ans=   30*x^4 + 100*x^3 + 102*x^2 + 58*x + 35
```

3.5.2 插值、拟合多项式并求导

当已知函数在一些离散点上的函数值时, 该函数可用插值或拟合多项式来近似, 然后对多项式进行微分求得导数, 函数求导主要是研究函数值随自变量变化而变化的趋势。

语法格式:

```
polyfit(x-xn,y,length(x)-1); %xn 表示 x 某一点的值
```

【实战练习 3-27】插值、拟合多项式并求导数

通过实验测试, 得到某数据集为:

X : 0~5

Y : 0.3200 0.5720 0.7012 0.8132 0.9013 0.9616

要求: 计算 $x=5$ 处的各阶导数。

编程代码如下:

```
clc;clear;
x=0:5;
y=[0.3200 0.5720 0.7012 0.8132 0.9013 0.9616]; %原数据点
xn=3; %指定坐标点
L=length(x); %得到数据长度
dd=polyfit(x-xn,y,L-1); %进行多项式拟合, (x-xn,y) 为坐标, (L-1) 为长度
fact=[1];
for k=1:L-1;
    fact=[factorial(k),fact]; %进行微分求导数
end
```

```
diffder=dd.*fact           %计算拟合点
plot(x,y,"b*",x,diffder);  % “*” 号为原数据点，曲线为拟合多项式导数
```

结果为：

```
diffder =    0.1151    0.0028   -0.0341   -0.0241    0.1048    0.8132
```

原数据点与拟合多项式求导结果如图 3.4 所示。

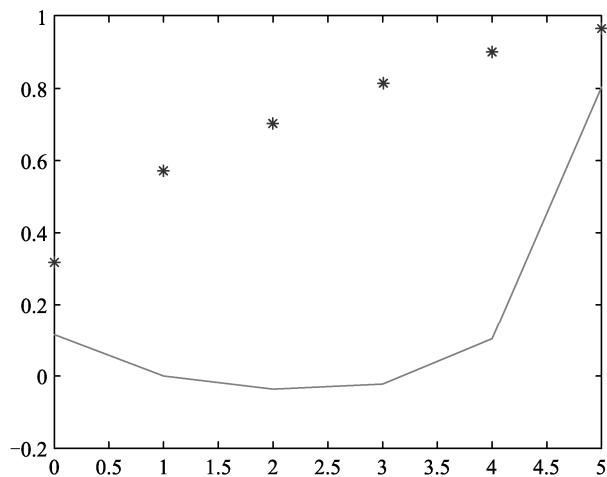


图 3.4 原数据点与拟合多项式求导结果