

第7章 **MATLAB**多项式计算

7.1 多项式

7.1.1 多项式的表示

Matlab中用按降幂排列的多项式系数组成的行向量表示多项式,

如: $p(x)=x^3-2x-5$ 被表示为:

$p = [1 \ 0 \ -2 \ -5];$

1. 多项式的加减运算

2. 多项式乘法运算

函数**conv(P1,P2)**用于求多项式**P1**和**P2**的乘积。
这里，**P1**、**P2**是两个多项式系数向量。

例7-1 求多项式 x^7+8x^3-10 与多项式 $2x^2-x+3$ 的乘积。

3. 多项式除法

函数 $[Q,r]=\text{deconv}(P1,P2)$ 用于对多项式 $P1$ 和 $P2$ 作除法运算。

Q : 多项式 $P1$ 除以 $P2$ 的商式,

r : $P1$ 除以 $P2$ 的余式。

Q 和 r 仍是多项式系数向量。

deconv 是 conv 的逆函数, 即有 $P1=\text{conv}(P2,Q)+r$ 。

例7-2 求多项式 x^7+8x^3-10 除以多项式 $2x^2-x+3$ 的结果。

7.1.2 多项式的导数

对多项式求导数的函数是：

p=polyder(A): 求多项式A的导函数

p=polyder(A,B): 求A与B积的导函数

[p,q]=polyder(A,B): 求A/B的导函数，导函数的分子存入p，分母存入q。

上述函数中，参数A,B是多项式的向量表示，结果p,q也是多项式的向量表示。

例7-3 求有理分式的导数。

命令如下：

P=[1];

Q=[1,0,5];

[p,q]=polyder(P,Q)

7.1.3 多项式的求值

两种求多项式值的函数：

polyval(P,x)

polyvalm(P,x)

两者的区别在于前者是代数多项式求值，而后者是矩阵多项式求值。

1. 代数多项式求值

polyval函数用来求代数多项式的值，

调用格式： $Y = \text{polyval}(P, x)$

若 x 为一数值，则求多项式在该点的值；若 x 为向量或矩阵，则对向量或矩阵中的每个元素求其多项式的值。

例7-7 已知多项式 $x^4 + 8x^3 - 10$ ，分别取 $x = 1.2$ 和一个 2×3 矩阵为自变量计算该多项式的值。

2. 矩阵多项式求值

polyvalm函数要求x为方阵，它以方阵为自变量求多项式的值。
设A为方阵，P代表多项式 x^3-5x^2+8 ，那么

polyval(P,A)的含义是：

$$A.*A.*A-5*A.*A+8*ones(size(A))$$

而**polyvalm(P,A)**的含义是：

$$A*A*A-5*A*A+8*eye(size(A))$$

例7-5 仍以多项式 x^7+8x^3-10 为例，取一个 2×2 矩阵为自变量分别用**polyval**和**polyvalm**计算该多项式的值。

7.1.7 多项式求根

n 次多项式具有 n 个根，当然这些根可能是实根，也可能含有若干对共轭复根。**MATLAB**提供的**roots**函数用于求多项式的全部根，其调用格式为：

$x = \text{roots}(P)$

其中 **P** 为多项式的系数向量，求得的根赋给向量 **x** ，即 **$x(1), x(2), \dots, x(n)$** 分别代表多项式的 n 个根。

注意：根被储存为列向量

例7-6 求多项式 x^4+8x^3-10 的根。

命令如下：

A=[1,8,0,0,-10];

x=roots(A)

若已知多项式的全部根，则可以用**poly**函数建立起该多项式，其调用格式为：

P=poly(x)

若**x**为具有**n**个元素的向量，则**poly(x)**建立以**x**为其根的多项式，且将该多项式的系数赋给向量**P**。

例7-7 已知 $f(x)$

(1) 计算 $f(x)=0$ 的全部根。

(2) 由方程 $f(x)=0$ 的根构造一个多项式 $g(x)$ ，并与 $f(x)$ 进行对比。

命令如下：

```
P=[3,0,7,-5,-7.2,5];
```

```
X=roots(P)           %求方程 $f(x)=0$ 的根
```

```
G=poly(X)           %求多项式 $g(x)$ 
```

7.1.5 多项式拟合

`p=polyfit(x,y,n)`:用最小二乘法对已知数据 **`x,y`** 进行拟合, 以求得 **`n`** 阶多项式的系数向量。

例7-8

```
x=linspace(0,2*pi,100);
```

```
y=sin(x);
```

```
t=polyfit(x,y,6);
```

```
y1=polyval(t,x);
```

```
plot(x,y,'ro',x,y1,'b-')
```

7.2 数据插值

7.2.1 一维数据插值

在MATLAB中，实现这些插值的函数是**interp1**，其调用格式为：**Y1=interp1(X,Y,X1,'method')**

函数根据**X,Y**的值，计算函数在**X1**处的值。**X,Y**是两个等长的已知向量，分别描述采样点和样本值，**X1**是一个向量或标量，描述欲插值的点，**Y1**是一个与**X1**等长的插值结果。

method是插值方法，允许的取值有

- A、nearest** 寻找最近数据点,由其得出函数值;
- B、linear** 线性插值(该函数的默认方法);
- C、spline** 样条插值,数据点处光滑--左导等于右导;
- D、cubic** 三次插值

例7-9 某观测站测得某日7:00时至18:00时之间每隔2小时的室内外温度(°C)，用3次样条插值分别求得该日室内外7:30至17:30时之间每隔2小时各点的近似温度(°C)。

设时间变量**h**为一行向量，温度变量**t**为一个两列矩阵，其中第一列存放室内温度，第二列储存室外温度。命令如下：

```
h =7:2:18;
```

```
t=[18,20,22,25,30,28,27,28; 17,18,19,27,27,23,23,25]';
```

```
XI =7.5:2:17.5
```

```
YI=interp1(h,t,XI,'spline')      %用3次样条插值计算
```


7.2.2 二维数据插值

在MATLAB中，提供了解决二维插值问题的函数 **interp2**，其调用格式为：

Z1=interp2(X,Y,Z,X1,Y1,'method')

其中**X,Y**是两个向量，分别描述两个参数的采样点，**Z**是与参数采样点对应的函数值，**X1,Y1**是两个向量或标量，描述欲插值的点。**Z1**是根据相应的插值方法得到的插值结果。

A、nearest 寻找最近数据点,由其得出函数值;

B、linear 二维线性插值

C、cubic 二维三次插值

例7-10 某实验对一根长10米的钢轨进行热源的温度传播测试。用**x**表示测量点**0:2.5:10**(米)，用**h**表示测量时间**0:30:60**(秒)，用**T**表示测试所得各点的温度(°C)。试用线性插值求出在一分钟内每隔**20**秒、钢轨每隔**1**米处的温度**TI**。

命令如下：

```
x=0:2.5:10;
```

```
h=[0:30:60]';
```

```
T=[95,17,0,0,0;88,78,32,12,7;77,77,57,78,71];
```

```
xi=[0:10];
```

```
hi=[0:20:70]';
```

```
TI=interp2(x,h,T,xi,hi)
```