CHAPTER 3 第3章

MATLAB的数值计算

MATLAB提供了许多用于处理数值运算的函数，用户可以方便地实现数值计算（如多项式的运算、方程的数值解等）。

3.1 多项式的创建与运算

3.1.1 多项式的描述与创建

1. 多项式的描述

在MATLAB环境下，多项式的表达式

$$p(x)=a_0x^n+a_1x^{n-1}+\cdots+a_{n-1}x+a_n$$

可以转换成向量的形式来描述，即 $\boldsymbol{p}=[a_0 \quad a_1 \quad \cdots \quad a_{n-1} \quad a_n]$。

向量最右边的元素表示多项式的0阶系数，向左数依次表示多项式的1阶系数，2阶系数，3阶系数……也就是说，MATLAB按降幂形式排列多项式的系数。例如，多项式表达式 $6x^4+3x^3+4x^2+2x+7$，在MATLAB环境下用向量描述为[6 3 4 2 7]。

2. 多项式的创建

创建多项式的方法有许多种。

(1) 方法一：直接输入法

在MATLAB的命令窗口中直接输入多项式的系数向量，然后利用转换函数poly2sym将多项式由系数向量描述转换为多项式表达式的形式。注意，缺少的各项在向量中以“0”代替。

【例3-1】 在MATLAB环境下采用直接输入法，创建多项式 x^3+3x^2-2x+5。

程序设计为：

```
>>p=[1 3-2 5];          %多项式的系数向量
>>poly2sym(p)           %生成多项式的表达式
```

运行结果为：

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

(2) 方法二：特征多项式输入法

由函数 poly 提取 n 阶特征多项式系数向量，该系数向量的第一个元素必须为 1，然后用函数 poly2sym 将其转换为特征多项式的表达式。

【例 3-2】 已知矩阵 $\boldsymbol{M}=\begin{pmatrix}2 & 1 & 3\\ 4 & 2 & 1\\ 1 & 2 & 3\end{pmatrix}$，求其特征多项式的表达式。

程序设计为：

```
>>M=[2 1 3;4 2 1;1 2 3];
>>P=poly(M)                          %生成特征多项式系数向量
```

运行结果为：

```
P=
    1.0000   -7.0000    7.0000  -15.0000
>>poly2sym(P)                       %生成特征多项式的表达式
```

运行结果为：

```
ans=
    x^3-7*x^2+7*x-15
```

(3) 方法三：由根向量创建多项式

也可以利用给定的多项式的根向量来创建多项式，同样由函数 poly 实现。

【例 3-3】 已知多项式的根向量为[－0.5　－0.2＋0.3i　－0.2－0.3i]，求其对应的多项式表达式。

程序设计为：

```
>>r=[-0.5 -0.2+0.3i -0.2-0.3i];
>>p=poly(r)
```

运行结果为：

```
p=
    1.0000    0.9000    0.3300    0.0650
>>poly2sym(p)
```

运行结果为：

```
ans=
    x^3+9/10*x^2+33/100*x+13/200
```

3.1.2 多项式的运算

1. 求多项式的值

求多项式的值的方法有两种。一种是按数组运算规则计算，由函数 polyval 实现；另一种是按矩阵的运算规则计算，由函数 polyvalm 来实现。

【例 3-4】 求多项式 $4x^2-2x+10$ 在 5、6 和 8 处的值。

程序设计为：

```
>>p=[4-2 10];
>>x=[5 6 8];
>>polyval(p,x)
```

运行结果为：

```
ans=
   100   142   250
```

【例 3-5】 求多项式 $4x^2-2x+10$ 对于矩阵 $\begin{pmatrix}3 & 5\\7 & 2\end{pmatrix}$ 的值。

程序设计为：

```
>>p=[4 -2 10];
>>x=[3 5;7 2];
>>polyvalm(p,x)
```

运行结果为：

```
ans=
   180    90
   126   162
```

2. 求多项式的根

求多项式的根，即求多项式为零时的值。在 MATLAB 环境下求根时有两种方法：一种是直接调用根函数 roots 来求解多项式的根的方法；另外一种方法是先求多项式的伴随矩阵及特征值，再求多项式的根。这里仅介绍前面一种方法。

【例 3-6】 求多项式 $x^5+5x^4-6x^3+10x^2+3x+20$ 的根。

程序设计为：

```
>>a=[1 5 -6 10 3 20];
>>r=roots(a)
```

运行结果为：

```
r=
  -6.2232
   1.1158+1.1853i
   1.1158-1.1853i
  -0.5042+0.9791i
  -0.5042-0.9791i
```

3. 多项式的加、减运算

多项式的加、减运算是多项式对应元素的加、减运算。多项式的阶数可以不同，但在

进行多项式定义时，应当在低阶多项式的前面补充“0”，使其阶数相等，否则，不能进行加、减运算。

【例 3-7】 对两个多项式 x^4+3x^3-2x+3 和 x^2+3x+4 进行加、减运算。

程序设计为：

```
>>a=[1 3 0 -2 3];
>>b=[0 0 1 3 4];
>>c=a+b
```

运行结果为：

```
c=
     1     3     1     1     7
>>poly2sym(c)
ans=
    x^4+3*x^3+x^2+x+7
```

4. 多项式的乘、除运算

多项式的乘法运算由函数 conv 实现；多项式的除法运算由函数 deconv 实现。

【例 3-8】 计算两个多项式 x^3+5x^2-2x+1 和 $x+5$ 的乘积。

程序设计为：

```
>>a=[1 5 -2 1];
>>b=[1 5];
>>c=conv(a,b)
```

运行结果为：

```
c=
     1    10    23    -9     5
>>poly2sym(c)
ans=
    x^4+10*x^3+23*x^2-9*x+5
```

【例 3-9】 多项式 x^3+5x^2-2x+1 与 $x+5$ 的除法运算。

程序设计为：

```
>>a=[1 5 -2 1];              %被除数多项式
>>b=[1 5];                   %除数多项式
>>[q r]=deconv(a,b)          %q表示除法运算的商,r表示除法运算的余数
```

运行结果为：

```
q=
     1     0    -2
r=
     0     0     0    11
```

```
>>poly2sym(q)                %商的多项式表达式
ans=
    x^2-2
```

5. 多项式的微积分

多项式的微分由函数 polyder 实现;多项式的积分由函数 polyint 实现。

【例 3-10】 求多项式 $3x^4-2x^3+5x^2-10x+6$ 的一阶导数。

程序设计为:

```
>p=[3 -2 5 -10 6];
>>polyder(p)
```

运行结果为:

```
ans=
    12    -6    10   -10
>>poly2sym(ans)
ans=
    12*x^3-6*x^2+10*x-10
```

【例 3-11】 求多项式 $12x^3-8x^2+6x-5$ 的积分。

程序设计为:

```
>>p=[12 -8 6 -5];
>>polyint(p)
```

运行结果为:

```
ans=
    3.0000   -2.6667    3.0000   -5.0000         0
>>poly2sym(ans)
ans=
    3*x^4-8/3*x^3+3*x^2-5*x
```

6. 多项式的部分分式展开

在线性系统的傅里叶变换、拉普拉斯变换和 Z 变换中,经常要用到多项式部分分式展开的情况。一个多项式可以描述成如下部分分式的情况:

$$\frac{\mathrm{num}(x)}{\mathrm{den}(x)}=\frac{r_1}{x+p_1}+\frac{r_2}{x+p_2}+\cdots+\frac{r_n}{x+p_n}+k(x)$$

式中,$p_1,p_2,\cdots,p_n$ 称为极点;$r_1,r_2,\cdots,r_n$ 称为留数;$k(x)$称为常数项。

MATLAB 提供了一些进行多项式部分分式展开运算的函数。其中,常用的函数为 residue,它的调用格式为:

```
[r,p,k]=residue(num,den)
```

式中,num 和 den 分别表示多项式的分子和分母多项式的系数向量;r、p、k 分别为留数、极点和常数项。

同时，可以将部分分式展开转换为原多项式表达式 num(x)和 den(x)系数向量。调用的函数格式为：

```
[num,den]=residue(r,p,k)
```

【例 3-12】 求多项式$\frac{\text{num}(x)}{\text{den}(x)}=\frac{10(x+3)}{(x+1)(x^2+x+3)}$的部分分式展开。

程序设计为：

```
>>num=[10 30];
>>den=[1 2 4 3];
>>[r,p,k]=residue(num,den)
```

运行结果为：

```
r=
  -3.3333-4.0202i
  -3.3333+4.0202i
   6.6667
p=
  -0.5000+1.6583i
  -0.5000-1.6583i
  -1.0000
k=
     []
```

其中，“k =[]”表示没有常数项。

【例 3-13】 求多项式$\frac{\text{num}(x)}{\text{den}(x)}=\frac{5x^3+3x^2-2x+7}{-4x^3+8x+3}$的部分分式展开。

程序设计为：

```
>>num=[5 3 -2 7];
>>den=[-4 0 8 3];
>>[r,p,k]=residue(num,den)
```

运行结果为：

```
r=
   -1.4167
   -0.6653
    1.3320
p=
    1.5737
   -1.1644
   -0.4093
k=
   -1.2500
```

【例 3-14】 求多项式$\frac{\text{num}(x)}{\text{den}(x)}=\frac{10x+20}{(x+1)(x^2+x+3)}$的部分分式展开。

程序设计为：

```
>>num=[10 20];
>>den=[1 2 4 3];
>>[r,p,k]=residue(num,den)
```

运行结果为：

```
r=
  -1.6667-3.5176i
  -1.6667+3.5176i
   3.3333
p=
  -0.5000+1.6583i
  -0.5000-1.6583i
  -1.0000
k=
     []
```

3.2 线性方程求解

MATLAB提供了求解方程的一些函数，包括代数方程和微分方程，可以方便用户解决实际问题。

3.2.1 代数方程及代数方程组的求解

在MATLAB中，使用solve函数求代数方程，其调用格式为：

```
[x1,x2,…,xn]=solve('eqn1','eqn2',…,'eqnn')
```

表示对n个未知变量$x_1,x_2,\cdots,x_n$求解n个方程。

【例 3-15】 求一元二次方程$ax^2+bx+c=0$的根。

在命令窗口中输入如下命令：

```
>>solve('a*x^2+b*x+c','x')
```

则有：

```
ans=
  [ 1/2/a*(-b+(b^2-4*a*c)^(1/2))]
  [ 1/2/a*(-b-(b^2-4*a*c)^(1/2))]
```

或者在命令窗口中输入命令：

```
>>[x]=solve('a*x^2+b*x+c')
```

显示结果为：

```
x=
  [ 1/2/a*(-b+(b^2-4*a*c)^(1/2))]
  [ 1/2/a*(-b-(b^2-4*a*c)^(1/2))]
```

此外，还可以利用 solve 命令求解若干个代数方程的解，即代数方程组的解。

【例 3-16】 求代数方程组 $\begin{cases} x+y+z=2b \\ 2x+y+2z=2b \\ 2x+2y+z=5b \end{cases}$ 的解。

在命令窗口中输入如下命令：

```
>>eqn1='x+y+z=2*b';
>>eqn2='2*x+y+2*z=2*b';
>>eqn3='2*x+2*y+z=5*b';
>>[x,y,z]=solve(eqn1,eqn2,eqn3)
```

结果显示为：

```
x=
    b
y=
    2*b
z=
    -b
```

3.2.2 微分方程及微分方程组的求解

在 MATLAB 中，使用 dsolve 函数求解常微分方程，其调用格式为：

```
[y1,y2,…]=dsolve('eqn1','eqn2',…)
```

式中，输入量 eqn 可以是微分方程，也可以是求解微分方程的初始条件，还可以声明独立变量。一般情况下，前面的是微分方程，中间的是初始条件，最后的是独立变量声明。

在表达微分方程时，n 阶导数表示为 Dny。例如，$\frac{dy}{dt}$、$\frac{dx}{dt}$、$\frac{d^2y}{dt^2}$和$\frac{d^3y}{dt^3}$分别表示为 Dy、Dx、D2y 和 D3y。

【例 3-17】 求解 $\frac{dy}{dx}=x^2+2x$。

程序设计及运行结果为：

```
>>y=dsolve('Dy=x^2+2*x','x')                    %求解微分方程,x 为声明的独立变量
y=
    1/3*x^3+x^2+C1                              %C1 为任意实常量
```

【例 3-18】 求二阶微分方程 $\frac{d^2y}{dt^2}-8\frac{dy}{dt}+6y=0$ 的解。初始条件为 $y(0)=0$，$\left.\frac{dy}{dx}\right|_{x=0}=1$。

程序设计及运行结果为：

```
>>y=dsolve('D2y-8*Dy+6y=0','x')                    %求二阶微分方程通解
y=
    3/4*x+C1+C2*exp(8*x)
>>y=dsolve('D2y-8*Dy+6y=0','y(0)=0,Dy(0)=1','x')   %求二阶微分方程特解
y=
    3/4*x-1/32+1/32*exp(8*x)
```

【例 3-19】 求一阶微分方程组$\begin{cases}\dfrac{dx}{dt}=3x+4y\\ \dfrac{dy}{dt}=-4x+3y\end{cases}$的解。初始条件为 $x(0)=0, y(0)=1$。

程序设计及运行结果为：

```
>>eqn1='Dx=3*x+4*y';
>>eqn2='Dy=-4*x+3*y';
>>[x,y]=dsolve(eqn1,eqn2,'x(0)=0,y(0)=1')
x=
    exp(3*t)*sin(4*t)
y=
    exp(3*t)*cos(4*t)
```

3.3 曲线拟合与插值

3.3.1 曲线拟合

在许多实际应用工程中，常常只能测得一些分散的数据点。为了能从这些分散的数据点中找出它们内在的变化规律，人们依据这些分散的数据点，运用最小二乘法、多项式或其他的已知函数等方法来生成一个新的多项式或者函数来逼近这些已知的数据点，这一过程称为曲线拟合。

由于 MATLAB 提供了强大的计算功能和绘图功能，使得用户可以很方便地进行曲线拟合，并绘制出曲线拟合图。

曲线拟合涉及两个基本问题：最佳拟合意味着什么？应该用什么样的曲线？由于可以用许多不同的方法定义最佳拟合，并存在无穷数目的曲线，而当最佳拟合被解释为在数据点的最小误差平方和，且所用的曲线限定为多项式时，那么曲线拟合是相当简捷的，数学上称为最小二乘法曲线拟合。最小二乘法曲线拟合是最常用的曲线拟合方法。

MATLAB 提供了 polyfit 函数来求解最小二乘法曲线拟合问题。它的调用格式为：

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

其含义为：用最小二乘法对所给定数据 x、y 进行 n 阶多项式拟合，其返回值是一个多项

式系数的行向量。

【例 3-20】 已知某实验数据如表 3-1 所示，利用 polyfit 函数求其最小二乘法拟合曲线。

表 3-1 实验数据

x	1	2	3	4	5	6	7	8	9	10
y	−0.47	1.98	3.3	6.2	7.1	7.3	7.7	10	9.5	9.3

为了采用 polyfit 函数，必须给出函数所需要的数据 x 和 y，以及期望的最佳拟合数据的多项式的阶数 n。如果选择 $n=1$，会得到最简单的线性近似，通常称为线性回归。

程序设计为：

```
>>x=[1 2 3 4 5 6 7 8 9 10];
>>y=[-0.47 1.98 3.3 6.2 7.1 7.3 7.7 10 9.5 9.3];
>>a=polyfit(x,y,1)
```

运行结果为：

```
a=
    1.0835    0.2320
```

为了查看拟合的多项式与原来数据的拟合程度，在命令窗口中继续输入如下命令，并按 Enter 键确认：

```
>>x1=1:0.05:10;
>>y1=a(1)*x1+a(2);
>>plot(x,y,'*',x1,y1,'-')
```

运行结果如图 3-1 所示。

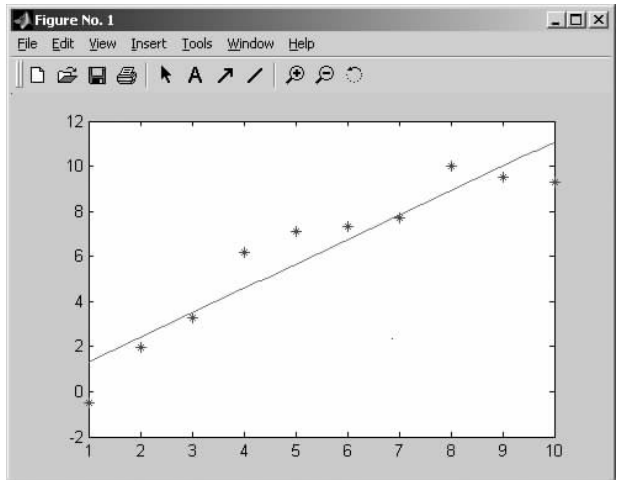

图 3-1 线性回归拟合曲线

从图 3-1 所示中可以看出，拟合曲线与原数据点相比，拟合误差比较大，精度不高。为此，增加多项式的阶数 n 值。若选择 $n=2$，在命令窗口中继续输入如下命令，并按