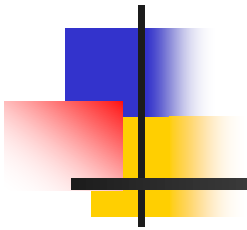


Matlab数值计算

第7讲

6.1 多项式(polynomial)

- 多项式的MATLAB表达
  - 多项式由一个行向量表示
    - 该向量元素是该多项式的系数
    - 且按降幂次序排列

如：多项式$x^4-12x^3+25x+116$由行向量：

p=[1 -12 0 25 116]表示。

注意，必须包括具有零系数的项。

- 求解多项式的根?

roots指令

6.1 多项式(polynomial)

■ 举例：求解多项式$x^4-12x^3+25x+116$的根

```
>>p=[1  -12  0  25  116]
p =
    1   -12    0   25   116
>>r=roots(p)
r =
  11.7473
   2.7028
  -1.2251 + 1.4672i
  -1.2251 - 1.4672i
```

◆ MATLAB按惯例规定，多项式是行向量，根是列向量

6.1 多项式(polynomial)

- 已知多项式的根，求对应的多项式？
  - 能！
  - 使用poly指令
- 举例：由上例所得的根求其多项式

```
>> pp=poly(r)
pp =
    1.0000  -12.0000   -0.0000   25.0000  116.0000
```

即：$x^4-12x^3+25x+116$

6.1 多项式(polynomial)

- 多项式的乘法(conv指令)
- 举例： 多项式$a(x)=x^3+2x^2+3x+4$

  和$b(x)=x^3+4x^2+9x+16$的乘积。

```
>> a=[1  2  3  4] ;  b=[1  4  9  16];
>> c=conv(a, b)
c =
    1    6   20   50   75   84   64
```

  - 两个以上的多项式的乘法需要重复使用conv.

6.1 多项式(polynomial)

- 多项式加法: MATLAB没有提供进行加法运算的函数。
- 如果两个多项式向量大小相同，标准的数组加法有效。

把多项式a(x)与上面给出的b(x)相加。

```
>> d=a+b
d =
     2     6    12    20
```

结果: $d(x)= 2x^3+6x^2+12x+20$

- 当两个多项式阶次不同，低阶的多项式必须用首零填补，使其与高阶多项式有同样的阶次。

考虑上面多项式c和d相加:

```
>>  e=c+[0  0  0  d]
e =
     1     6    20    52    81    96    84
```

结果: $d(x)= x^6+6x^5+20x^4+52x^3+81x^2+96x+84$

6.1 多项式(polynomial)

- 问题：编写一个多项式加法运算的函数文件
- 明确需求
  - 定义一个函数，需要有两个输入参数（比如：p1、p2），一个输出参数p_out。
    - p1、p2表示两个待计算的多项式。
    - p_out表示两个多项式的求和结果
  - 函数的内部处理
    - 如p1、p2两参数大小相等，则直接相加: p_out=p1+p2
    - 如p1、p2两参数大小不等：
      - If length(p1)>length(p2)
        - P2前面要补0元素，使p1、p2两参数大小相等
      - 否则
        - P1前面要补0元素，使p1、p2两参数大小相等

6.1 多项式(polynomial)

- 函数文件的编写

```
function p_out=poly_sum(p1, p2)
% calculate the sum of two polynomials

if length(p1)==length(p2)
   p_out=p1+p2;
elseif length(p1)>length(p2)
   p2=[zeros(1,length(p1)-length(p2)), p2];
   p_out=p1+p2;
else
   p1=[zeros(1,length(p2)-length(p1)), p1];
   p_out=p1+p2;
end
```

6.1 多项式(polynomial)

- 多项式的除法(deconv)
- 举例说明:

$c(x)=x^6+6x^5+20x^4+50x^3+75x^2+84x+64$

除以$b(x)= x^3+4x^2+9x+16$

```
>> c=[1    6    20    50    75    84    64];
>> b=[1  4  9  16];
>>  [q , r]=deconv(c , b)
q =
    1    2    3    4
r =
    0    0    0    0    0    0    0
```

6.1 多项式(polynomial)

- 多项式的导数 (polyder)
- 举例： 求$b(x)= x^3+4x^2+9x+16$的导数。

```
>> b=[1    4    9    16];
>> d=polyder(b)
d =
    3    8    9
```

结果为： $3x^2+8x+9$

6.1 多项式(polynomial)

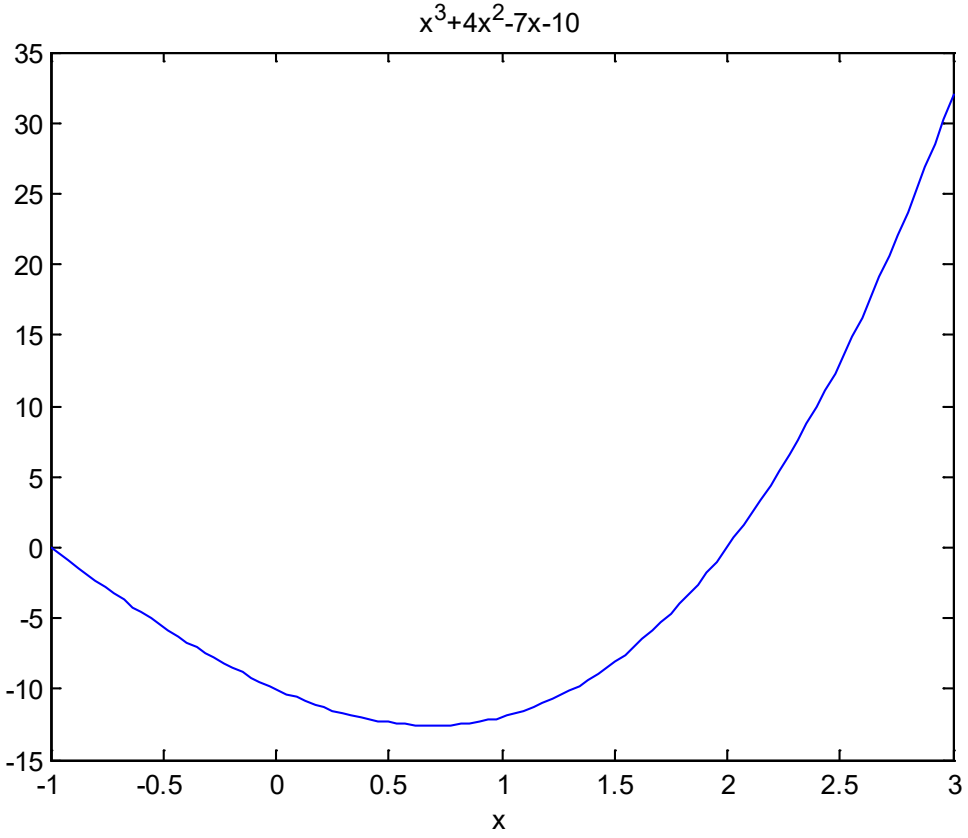
- 多项式的估值(polyval)
- 举例： 绘制$p(x) = x^3 + 4x^2 - 7x - 10$在[-1, 3]段上的曲线。

```
x=linspace(-1, 3);  % choose 100 data points between -1and 3.
p=[1  4  -7  -10];
v=polyval(p, x);

plot(x, v);
title('x^{3}+4x^{2}-7x-10');
xlabel('x')
```

6.1 多项式(polynomial)

$p(x) = x^3 + 4x^2 - 7x - 10$在[-1, 3]段上的曲线:



6.2 函数的数值导数

- 导数定义为：

$$\frac{dy}{dx}=\lim_{h\to 0}\frac{f(x+h)-f(x)}{(x+h)-(x)}$$

- 则y=f(x)的导数可近似为：

$$\frac{dy}{dx}\approx\frac{f(x+h)-f(x)}{(x+h)-(x)}$$

这里h>0

它是y的有限差分除以x的有限差分。

- MATLAB中没有直接提供数值导数的函数，只有计算向前差分的函数diff，其调用格式为：

DX = diff(X)　计算向量X的向前差分

DX = diff(X，n)　计算向量X的n阶向前差分

例题

设x由[0,2π]间均匀分布的10个点组成，求sinx的1-3阶差分。

命令如下：

```
X = linspace(0,2*pi,10);
Y = sin(X);
DY = diff(Y)
D2Y = diff(Y,2)
D3Y = diff(Y,3)
```

```
DY =
    0.6428    0.3420   -0.1188   -0.5240   -0.6840   -0.5240   -0.1188    0.3420    0.6428
D2Y =
   -0.3008   -0.4608   -0.4052   -0.1600    0.1600    0.4052    0.4608    0.3008
D3Y =
   -0.1600    0.0556    0.2452    0.3201    0.2452    0.0556   -0.1600
```

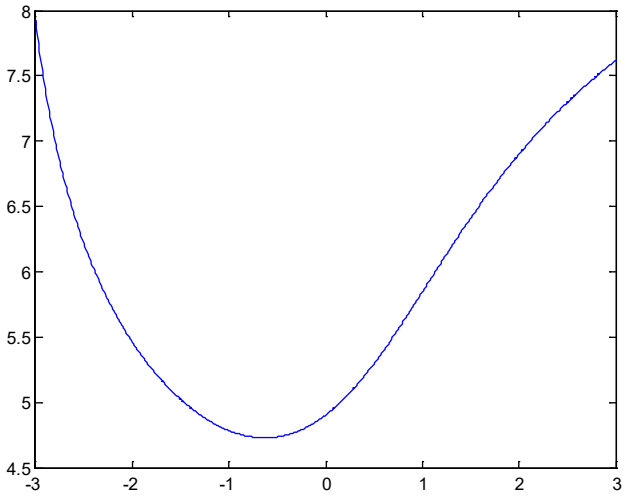
6.2 函数的数值导数（续）

例：设 $f(x)=\sqrt{x^3+2x^2-x+12}+\sqrt[6]{x+5}+5x+2$

在[-3,3]区间内以0.01为步长求数值导数。并画出导函数图像。

程序如下：

```
f = inline('sqrt(x.^3+2*x.^2-x+12)+(x+5).^(1/6)+5*x+2'); %内联函数
x = -3:0.01:3;
dx = diff(f([x,3.01]))/0.01;  %根据定义式求导数
plot(x,dx)
```



6.3 数值积分

一元函数的数值积分

- 常用积分指令：**quad**和**quadl**。
- 具体调用格式如下：

```
q = quadl(fun,a,b)
q = quadl(fun,a,b,tol)
q = quadl(fun,a,b,tol,trace)
[q,fcnt] = quadl(fun,a,b,...)
```

  - 输入量fun为被积函数的句柄。
  - 输入量a, b分别是积分的下限、和上限，都必须是确定的数值;
  - 前3个输入参数是调用积分指令所必须的，其他可以缺省;
  - 输入量tol是一个标量，控制绝对误差;
  - 输入量trace为非0值时，将随积分的进程逐点画出被积分函数;
  - 输出参数fcnt返回函数的执行次数。

Note：quad的调用格式与quadl相同

6.3 数值积分（续）

- 举例：求定积分 $I=\int_0^1 e^{-x^2}dx$

MATLAB指令quad和quadl求积分

```
>>fun=inline('exp(-x.*x)','x');  %必须采用数组乘符号.*
>>Isim=quad(fun,0,1),  I8=quadl(fun,0,1)
```

```
Isim =  0.7468
I8 =  0.7468
```

6.3 数值积分（续）

- 举例：求解定积分 $I=\int_0^1 \sqrt{\ln\frac{1}{x}}dx$

用quad指令求积分

```
>>ff=inline('sqrt(log(1./x))','x');
>>Isim=quad(ff,0,1)
Warning: Divide by zero.
> In inlineeval at 13
In inline.subsref at 25
In quad at 63
Isim = 0.8862
```

6.4 元素排序

Matlab中对向量X排序的函数是sort(X), 函数返回一个对X中的元素按升序排列的新向量。

sort函数也可以对矩阵A的各列（或行）重新排序，其调用格式为：

[Y,I] = sort(A,dim)

dim=1,按列排序；dim=2,按行排序，Y是排序后的矩阵，I记录Y中的元素在A中的位置。

例：对下列矩阵做各种排序。

$$A=\begin{bmatrix} 1 & -8 & 5 \\ 4 & 12 & 6 \\ 13 & 7 & -13 \end{bmatrix}$$

命令如下：

```
A = [1,-8,5;4,12,6;13,7,-13];
 sort(A)
 ans =
 1    -8   -13
 4     7     5
13    12     6
-sort(-A,2)  %对A的每行按降序排列
ans =
5      1    -8
12     6     4
13     7   -13
```

$$A=\begin{bmatrix} 1 & -8 & 5 \\ 4 & 12 & 6 \\ 13 & 7 & -13 \end{bmatrix}$$

6.5 数据插值

在工程测量和科学实验中，所得到的数据通常是离散的，要得到这些离散点以外的其他点的数值，就需要根据已知的数据进行插值。插值函数一般由线性函数、多项式、样条函数或这些函数的分段函数充当。

一维数据插值：被插值函数有一个单变量。

采用的方法有：线性方法、最近方法、三次样条和三次插值。

在Matlab中实现这些插值的函数是interp1，其调用格式如下：

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，允许的取值为：

6.5 数据插值

（1） ‘linear’：线性插值。默认的插值方式。它是把插值点靠近的两个数据点用直线连接，然后在直线上选取对应插值点的数据。

（2） ‘nearest’：最近点插值。根据已知插值点与已知数据点的远近程度进行插值。插值点优先选择较近的数据点进行插值。

（3） ‘cubic’：3次多项式插值。根据已知数据求出一个3次多项式，然后根据该多项式进行插值。

（4） ‘spline’：3次样条插值。指在每个分段内构造一个3次多项式，使其满足插值条件外，在各节点处具有光滑的条件。

例：给出概率积分数据表如下，用不同的插值方法计算f(0.472)。

x	0.46	0.47	0.48	0.49
f(x)	0.4846555	0.4937542	0.5027498	0.5116683

命令如下：

```
x = 0.46:0.01:0.49;
f = [0.4846555,0.4937542,0.5027498,0.5116683];
format long
interp1(x,f,0.472)
ans =
 0.49555332000000
interp1(x,f,0.472,'nearest')
ans =
0.49375420000000
interp1(x,f,0.472,'spline')
ans =
0.49556073600000
interp1(x,f,0.472,'cubic')
ans =
0.49556111971206
```

其中，3次样条和3次多项式的插值结果优于最近点插值方法和线性插值方法，但插值方法的好坏依赖于被插值函数，没有一种对所有函数都是最好的插值方法。

6.6 曲线拟合

数值插值要求逼近函数在采样点与被逼近函数相等，但由于测量误差，所获得的数据不一定准确，如果强求逼近显然不够合理。

曲线拟合不要求逼近函数通过各采样点，但要尽量的接近这些点，使误差在某种意义上达到最小。

曲线拟合的实现：

在matlab中，用**polyfit**函数来求得最小二乘拟合多项式的系数，再用**polyval**函数按所得的多项式计算所给出点上的函数近似值。

polyfit函数的调用格式为：

[P,S] = polyfit(X,Y,m)

函数根据采样点X和采样点函数值Y，产生一个m次多项式P及其在采样点的误差向量S。其中X、Y是两个等长的向量，P是一个长度为m+1的向量，P的元素是多项式系数。

polyval函数的功能是按多项式的系数计算x点多项式的值。

例：用一个三次多项式在区间[0,2π]内逼近函数sinx。

在给定区间内，均匀的选择20个采样点，并计算采样点的函数值然后利用3次多项式逼近。

命令如下：

```
x = linspace(0,2*pi,20);
y = sin(x);
p = polyfit(x,y,3)
y1 = polyval(p,x)
plot(x,y,':o',x,y1,'-*')
legend('sin(x)','fit')
```