第5章 MATLAB数值计算

本章内容

5.1 数据统计分析
5.2 多项式数值运算
5.3 数值微积分
5.4 线性方程组求解
5.5 解常微分方程
5.6 例题讲解

学时： 5

5.1数据统计分析

函数语法	功能
max(A), max(A,[],dim)	返回A各列的最大元素
min(A), min(A,[],dim)	返回A各列的最小元素
sum(A), sum(A, dim)	返回A各列元素之和
mean(A), mean(A, dim)	返回A各列的平均值
prod(A), prod(A, dim)	返回A各列元素的乘积
median(A), median(A, dim)	返回A各列的中间值元素
sort(A), sort(A, dim)	使A的各列元素按递增排序

dim取1或2。默认为1.
为1时，函数对矩阵列向量操作；
为2时，函数对矩阵行向量操作。

5.1数据统计分析

【例1】

- A=[4 8 -9;11 -12 4;-8 0 5;0.6 5 10]
- B=max(A)
- C=min(A)
- D=mean(A)
- E=sum(A)
- F=prod(A)
- G=sort(A)

$$A=\begin{pmatrix} 4 & 8 & -9 \\ 11 & -12 & 4 \\ -8 & 0 & 5 \\ 0.6 & 5 & 10 \end{pmatrix}$$

5.1数据统计分析

1. 标准差，也称均方差，描述一组数据波动的大小。 Matlab用函数std计算。

$$\sigma_1=\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2} \quad \text{或者：} \quad \sigma_2=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2}$$

2. 方差，即标准差的平方。Matlab用函数var计算。
3. 相关系数，描述两组数据之间的相互关系的指标。绝对值越接近1，相关程度越高。Matlab用函数corrcoef计算。

$$r=\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2\sum_{i=1}^{n}(y_i-\bar{y})^2}}$$

4. 协方差，与相关系数类似的指标。Matlab用函数cov计算。

$$c=\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})$$

5.1数据统计分析

函数名称	功能
std(A,flag,dim)	A各列的标准差, flag=0为σ_1， flag=1为σ_2
var(A,flag,dim)	A各列的方差
corrcoef(A)	返回矩阵A的自相关阵，第i行第j列代表A(:,i)和A(:,j)的相关系数。
corrcoef(x,y)	返回2*2方阵，对角线是向量x和y的自相关系数，非对角线元素相等，是向量x和y的相关系数。
cov(A)	返回矩阵A的协方差阵，第i行第j列代表A(:,i)和A(:,j)的协方差。
cov(x， y)	返回2*2方阵，对角线是向量x和y的自协方差，非对角线元素相等，是向量x和y的协方差。

dim取1或2。默认为1。
为1时，函数对矩阵列向量操作；为2时，函数对矩阵行向量操作。

```
X=randn(10000,5);
M=mean(X)    %平均值，接近0
```

```
M = 1×5
    0.0017   -0.0020   -0.0038   -0.0001   -0.0106
```

```
D=std(X)     %标准差，接近1
```

```
D = 1×5
    0.9915    0.9899    0.9995    0.9862    1.0118
```

```
R=corrcoef(X) %相关系数矩阵，接近单位矩阵
```

```
R = 5×5
    1.0000    0.0060   -0.0001    0.0111    0.0005
    0.0060    1.0000   -0.0030   -0.0131   -0.0050
   -0.0001   -0.0030    1.0000   -0.0203   -0.0024
    0.0111   -0.0131   -0.0203    1.0000    0.0122
    0.0005   -0.0050   -0.0024    0.0122    1.0000
```

5.2 多项式数值运算

- 5.2.1 创建多项式
- 5.2.2 多项式求根
- 5.2.3 多项式求值
- 5.2.4 多项式乘法
- 5.2.5 多项式除法
- 5.2.6 多项式求导
- 5.2.7 插值与拟合
- 5.2.8 函数的极值和零点

5.2.1 创建多项式

n次多项式用一个长度为n+1的行向量表示，缺少的幂次项系数为0。如果n次多项式表示为

$$P(x)=a_n x^n+a_{n-1}x^{n-1}+a_{n-2}x^{n-2}+\cdots+a_1 x+a_0$$

则在在matlab中，该多项式表示为向量形式：

$$[a_n, a_{n-1}, a_{n-2}, \cdots, a_1, a_0]$$

在matlab中，多项式的计算转化为系数向量的运算。

- 方法一：用多项式的系数来创建
  - 函数格式： $P=[a_n, a_{n-1}, a_{n-2}, \cdots, a_1, a_0]$
  - 多项式P的系数按高次幂到低次幂排列

5.2.1 创建多项式

- 方法二：创建以向量A（a1,a2,……,an）中的元素为根的多项式

$$P(x)=(x-a_1)(x-a_2)\cdots(x-a_n)$$

- 函数格式：

P=poly(A)

- 【例3】
- A=[1,2,3]
- P=poly(A)

5.2.2 多项式求根

- 函数格式：

  roots(P)

  - 说明：求多项式方程P的根，根以列向量的形式给出。
- 【例4】
- A1=[1,2,3];
- P1=poly(A1);
- A2=roots(P1)

5.2.2 多项式求根

- 【例6】求如下方程的全部根。

$$2x^5 - 3x^3 + 71x^2 - 9x + 13 = 0$$

- x =
- -3.4914
- 1.6863 + 2.6947i
- 1.6863 - 2.6947i
- 0.0594 + 0.4251i
- 0.0594 - 0.4251i

```
p=[2,0,-3,71,-9,13];
x=roots(p)
```

5.2.2 多项式求根

- 【例7】分别用solve和roots两个命令，解方程：x^2+2*x+3=0

- syms x
- f2=x^2+2*x+3;
- x2=solve(f2)

```
x2 =

 - 2^(1/2)*1i - 1
 - 1 + 2^(1/2)*1i


ans =

  -1.0000 + 1.4142i
  -1.0000 - 1.4142i
```

- roots(1:3)

5.2.2 多项式求根

- 【例8】分别用solve和roots两个命令，解方程：2*x^5 - 3*x^3+ 71*x^2 - 9*x + 13=0
- syms x
- f1=2*x^5 - 3*x^3+ 71*x^2 - 9*x + 13;
- x1=solve(f1)
- roots([2,0,-3,71,-9,13])

```
x1 =

root(z^5 - (3*z^3)/2 + (71*z^2)/2 - (9*z)/2 + 13/2, z, 1)
root(z^5 - (3*z^3)/2 + (71*z^2)/2 - (9*z)/2 + 13/2, z, 2)
root(z^5 - (3*z^3)/2 + (71*z^2)/2 - (9*z)/2 + 13/2, z, 3)
root(z^5 - (3*z^3)/2 + (71*z^2)/2 - (9*z)/2 + 13/2, z, 4)
root(z^5 - (3*z^3)/2 + (71*z^2)/2 - (9*z)/2 + 13/2, z, 5)
```

```
ans =

 -3.4914 + 0.0000i
  1.6863 + 2.6947i
  1.6863 - 2.6947i
  0.0594 + 0.4251i
  0.0594 - 0.4251i
```

5.2.2 多项式求根

- 比较solve和roots这两个命令

1. solve是符号解，roots是数值解

2. roots只能解多项式方程；

solve可解别的方程和方程组，例如exp(-x)-x^2+3=0。

3. 对于5次和5次以上的一元函数，有时无法用solve求根，这时可以使用roots命令求解。

5.2.3 多项式求值

- 函数格式：

polyval(P，x)

  - 说明：

当x为标量时，求多项式P在该点的值；

当x为向量或矩阵时，则对x中的每个元素求多项式的值。

5.2.3 多项式求值

【例9】计算y=3x*x+2x+1,当x=5,7,9,10时y的值。

用数值计算polyval和符号计算subs两种方法。

- p = [3,2,1];
- polyval(p,[5,7,9,10])

```
ans =

    86   162   262   321
```

5.2.3 多项式求值

【例9】计算$y=3x*x+2x+1$,当x=5,7,9,10时y的值。

用数值计算polyval和符号计算subs两种方法。

- syms x
- y=3*x^2+2*x+1;
- subs(y, x, [5,7,9,10])　　%替代函数

```
ans =

    86   162   262   321
```

5.2.3 多项式求值

【练】在MATLAB中建立多项式 $f(x)=4x^3-3x^2+2x-5$，并求出$f(x)=0$时的根及$x=3$、$x=3.6$的值

- P=[4,-3,2,-5];
- x=roots(P)
-
- x=[3,3.6];
- f=polyval(P,x)

```
x =

   1.2007 + 0.0000i
  -0.2253 + 0.9951i
  -0.2253 - 0.9951i


f =

   82.0000  149.9440
```

5.2.4 多项式乘法

- 函数格式：

  conv(P1,P2)

  - 说明：P1多项式与P2多项式相乘

- 【例11】预测以下代码中A3的结果

```
A1=[1,2,3];
P1=poly(A1);
A2=[4,5,6];
P2=poly(A2);
P3=conv(P1,P2);
A3=roots(P3)
```

5.2.5 多项式除法

- 函数格式：

[Q,r]=deconv(P1,P2)

- 说明：P1多项式与P2多项式相除。Q为商，r为余数

即有P1=conv(P2,Q)+r

- 【例12】求多项式 x^4+8x^3-10除以 $2x^2-x+3$
- P1=[1, 8, 0,0, -10];
- P2=[2,-1,3];
- [Q,r]=deconv(P1,P2)

```
Q = 1×3
    0.5000    4.2500    1.3750

r = 1×5
         0         0         0  -11.3750  -14.1250
```

5.2.6 多项式求导

- 函数格式：

  p=polyder(P)　　%多项式P的导函数

  p=polyder(P,Q)　　%多项式P*Q的导函数

  [p,q] = polyder(P,Q)　%多项式P/Q的导函数,导函数的分子存入p，分母存入q。

- 【例13 】
- P1=[1 1 1 1];
- P2=polyder(P1)

```
P2 = 1×3

     3     2     1
```

5.2.6 多项式求导

- 【例14】求有理分式的导数

$$f(x)=\frac{3x^5+5x^4-8x^2+x-5}{10x^{10}+5x^9+6x^6+7x^3-x^2-100}$$

- P=[3,5,0,-8,1,-5];
- Q=[10,5,0,0,6,0,0,7,-1,0,-100];
- [p,q]=polyder(P,Q)

5.2.6 多项式求导

- 【练】已知

$$f(x)=3x^5-5x^4+2x^3-7x^2+5x+6$$

$$g(x)=3x^2+5x-3$$

- 求f(x)+g(x)、f(x)-g(x) 、 f(x)×g(x)、f(x)/g(x)，以及f(x)×g(x)的导数、f(x)/g(x)的导数

5.2.6 多项式求导

- 【练】
- f=[3,-5,2,-7,5,6];
- g=[3,5,-3];
- g1=[0,0,0,g];
- f+g1　　%求f(x)+g(x)
- f-g1　　%求f(x)-g(x)
- conv(f,g)　　%求f(x)*g(x)
- [Q,r]=deconv(f,g) %求f(x)/g(x)，商Q，余r
- P=polyder(f,g)　　% f(x)×g(x)的导数、
- [R,s]=polyder(f,g) % f(x)/g(x)的导数

5.2.7 插值与拟合

引言：

在工程测量和科学实验中得到的数据通常都是离散的。例如，测量得n个点的数据分别为

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$$

这些数据点反映了一个函数关系 $y = f(x)$ 。然而，它的解析式是未知的。

插值的目的：根据这些已知数据，利用插值方法，得到这些离散点以外的其他点的数值。

曲线拟合的目的：根据这些已知数据，构造一个简单的函数 $y = g(x)$ 来逼近 $y = f(x)$ ，放弃在采样点使两个函数值完全相同的要求。

5.2.7 插值与拟合

（1）一维数据插值

函数：Zq=interp1（X，Y，Xq，‘method’）

说明：X是向量，表示采样点；
Y是采样点上的样本值，与X等长；
Xq表示插值点；
Zq是根据相应的插值方法得到的的插值结果。

注意：以上X必须为单调的升序或者降序排列。

5.2.7 插值与拟合

参数名称	说明	特点
nearest	邻近点插值法。根据已知两点间的插值点与这两点之间的位置远近插值。当插值点距离前点近时，取前点的值，否则取后点的值	速度最快，但平滑性差
linear 默认值	线性插值。把相邻的数据点用直线连接，按所生成的曲线进行插值，是默认的插值方法	占有的内存较邻近点插值方法多，运算时间也稍长，与邻近点插值不同，其结果是连续的，但在顶点处的斜率会改变
spline	三次样条插值。用已知数据求出样条函数后，按照样条函数插值	运算时间长，但内存的占有较立方插值方法要少，三次样条插值的平滑性很好，但如果输入的数据不一致或数据点过近，可能出现很差的插值结果
pchip	三次埃尔米特插值。用已知数据构造出三次多项式进行插值	需要较多的内存和运算时间，平滑性很好

5.2.7 插值与拟合

【例16】美国人口数据预测

- 美国1900年到1990年期间每10年进行的人口统计数据为

[75.995 91.972 105.711 123.203 131.669...

150.697 179.323 203.212 226.505 249.633]（单位：万人）。

试用不同的插值方法，预测美国1900~1990年期间每年的人口数量，并画图比较统计数据和各插值方法得到的预测数据。

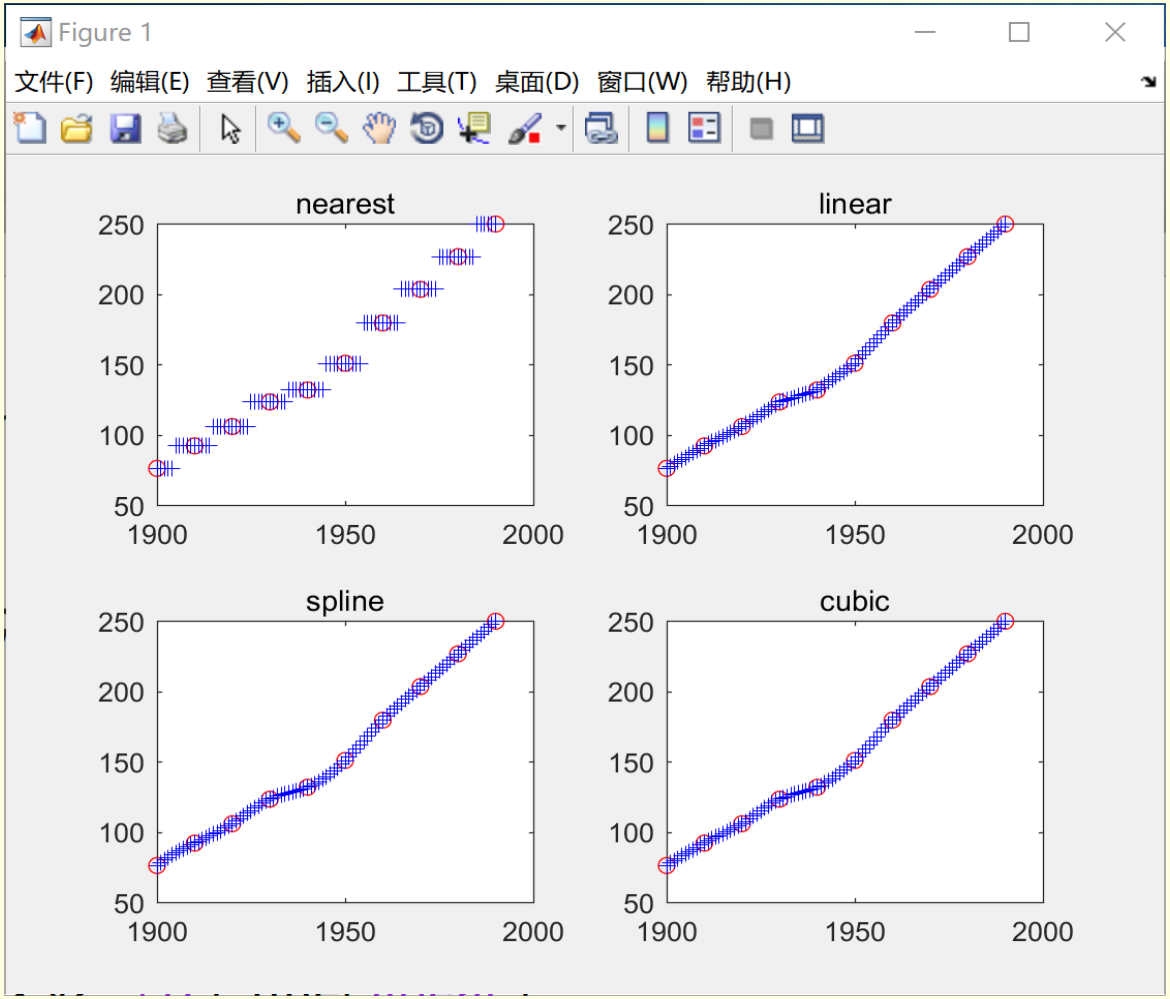
5.2.7 插值与拟合

```
t = 1900:10:1990;
p = [75.995  91.972  105.711  123.203  131.669...
     150.697  179.323  203.212  226.505  249.633];

x = 1900:1:1990;
y1 = interp1(t,p,x,'nearest');
y2 = interp1(t,p,x,'linear');
y3 = interp1(t,p,x,'spline');
y4 = interp1(t,p,x, 'pchip');
 subplot(2,2,1);  plot(t,p,'or',x,y1,'+b'); title('nearest')
subplot(2,2,2);  plot(t,p,'or',x,y2,'+b'); title('linear')
subplot(2,2,3);  plot(t,p,'or',x,y3,'+b'); title('spline')
subplot(2,2,4);  plot(t,p,'or',x,y4,'+b'); title('pchip')
```

5.2.7 插值与拟合

【例16】美国人口数据预测



5.2.7 插值与拟合

【例17】某检测参数f随时间t的采样结果如表，用数据插值法计算t=0:1:65时的f值，并画出相关图像。

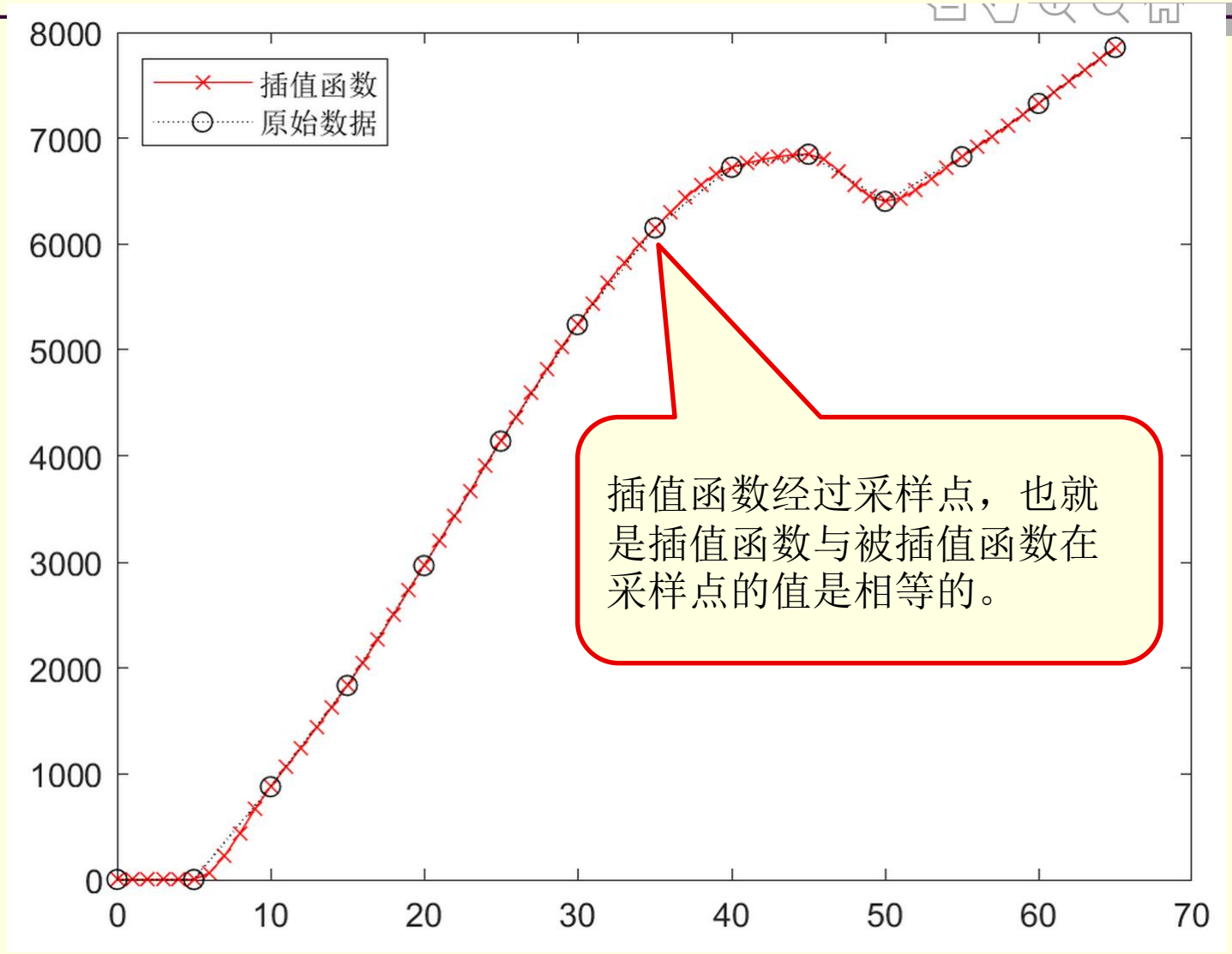
t	0	5	10	15	20	25	30
f	3.1025	2.256	879.5	1835.9	2968.8	4136.2	5237.9
t	35	40	45	50	55	60	65
f	6152.7	6725.3	6848.3	6403.5	6824.7	7328.5	7857.6

5.2.7 插值与拟合

- 【例17】

```
T=0:5:65;
X=0:1:65;
F=[3.2015,2.2560,879.5,1835.9,2968.8,4136.2,5237.9,6152.7,6725.3,
6848.3,6403.5,6824.7,7328.5,7857.6];
F1=interp1(T,F,X, 'pchip');
plot(X,F1,'-xr',T,F,':ok')
legend('插值函数','原始数据','Location', 'northwest');
```

5.2.7 插值与拟合


插值函数
原始数据
8000
7000
6000
5000
4000
3000
2000
1000
0
0
10
20
30
40
50
60
70
插值函数经过采样点，也就是插值函数与被插值函数在采样点的值是相等的。

5.2.7 插值与拟合

（2）二维数据插值是对双变量函数同时做插值。

函数：Zq=interp2(X，Y，Z，Xq，Yq，‘method’)

说明：X、Y分别存储采样点的平面坐标；
Z是与采样点对应的样本值；
Xq、Yq分别存储插值点的平面坐标；
Zq是根据相应的插值方法得到的插值结果。

5.2.7 插值与拟合

- 【例18】

1950到1990年工龄10、20、30年工作人员劳动报酬如下：

服务年限 年份	10	20	30
1950	150.697	169.592	187.652
1960	179.323	195.072	250.287
1970	203.212	239.092	322.767
1980	226.505	273.706	426.730
1990	249.633	370.281	598.243

计算1975年时15年工龄的工作人员平均工资。

5.2.7 插值与拟合

- x=[1950 1960 1970 1980 1990];
- y=[10 20 30];
- [X,Y]=meshgrid(x,y);　%产生采样点网格坐标
- Z=[150.697 179.323 203.212 226.505 249.633;169.592 195.072 239.092 273.706 370.281;187.652 250.287 322.767 426.730 598.243];
- xq=1975;
- yq=15;
- zq=interp2(X,Y,Z,xq,yq,'linear')

```
zq = 235.6288
```

5.2.7 插值与拟合

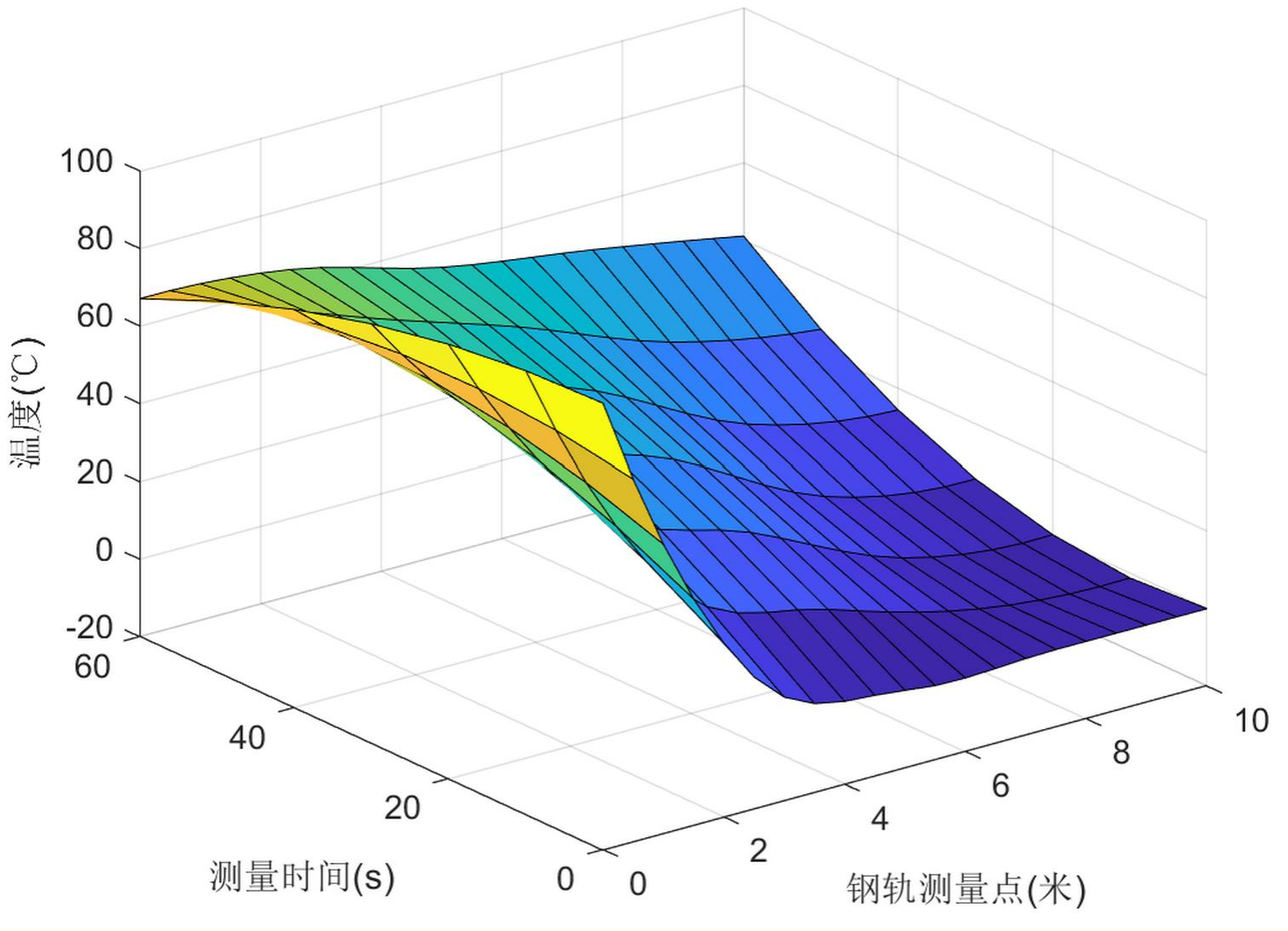
- 【例18】对一根长10米的钢轨进行热源的温度传播测试。　用x表示测量点(米)，用h表示测量时间(秒)，用T表示测得各点的温度(℃)，测量结果如表所示, 用3次多项式插值求出在一分钟内每隔10秒、钢轨每隔0.5米处的温度 。

T　x h	0	2.5	5	7.5	10
0	95	14	0	0	0
30	88	48	32	12	6
60	67	64	54	48	41

5.2.7 插值与拟合

```
x=0:2.5:10;
h=0:30:60;
[X,H]=meshgrid(x,h);    %产生采样点网格坐标
T=[95,14,0,0,0;88,48,32,12,6;67,64,54,48,41];
xi=0:0.5:10;
hi=0:10:60;
[Xi,Hi]=meshgrid(xi,hi);    %产生插值点网格坐标
Ti=interp2(X,H,T,Xi,Hi,'cubic');  %插值
surf(Xi,Hi,Ti);
xlabel('钢轨测量点(米)');
ylabel('测量时间(s)');
zlabel('温度(℃)');
```

5.2.7 插值与拟合



5.2.7 插值与拟合

b. 曲线拟合

放弃在采样点使两个函数值完全相同的要求，用polyfit函数来求最小二乘拟合多项式的系数。

函数：[P，S]=polyfit(X，Y，n)

说明：X是采样点；

Y是采样点上的函数值；

n是多项式的次数；

P是一个长度为n+1的向量，代表n次多项式；

S是在采样点的误差向量。

5.2.7 插值与拟合

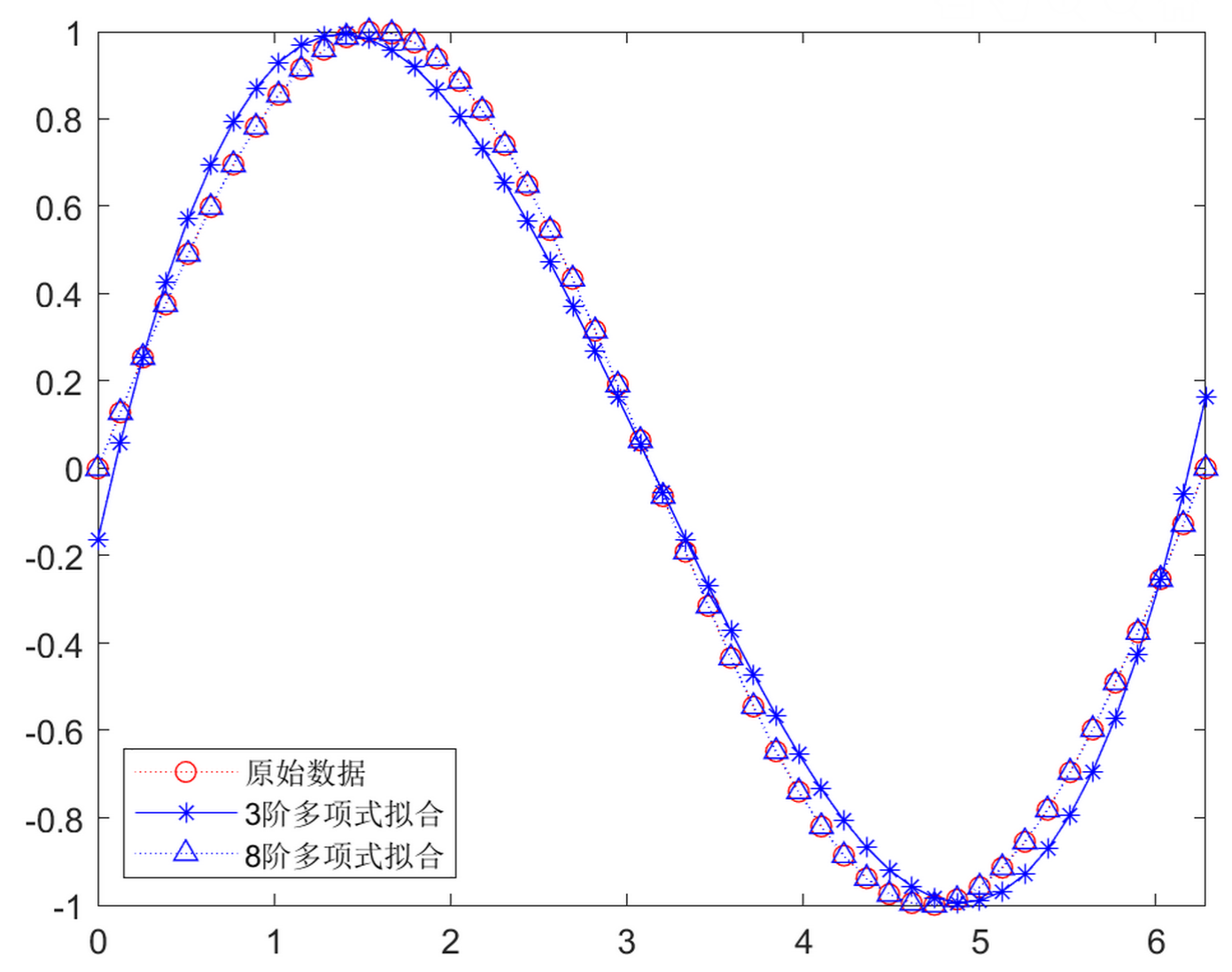
【例20】：用3次多项式和8次多项式在区间[0，2π]内逼近函数$\sin x$

5.2.7 插值与拟合

方法一：

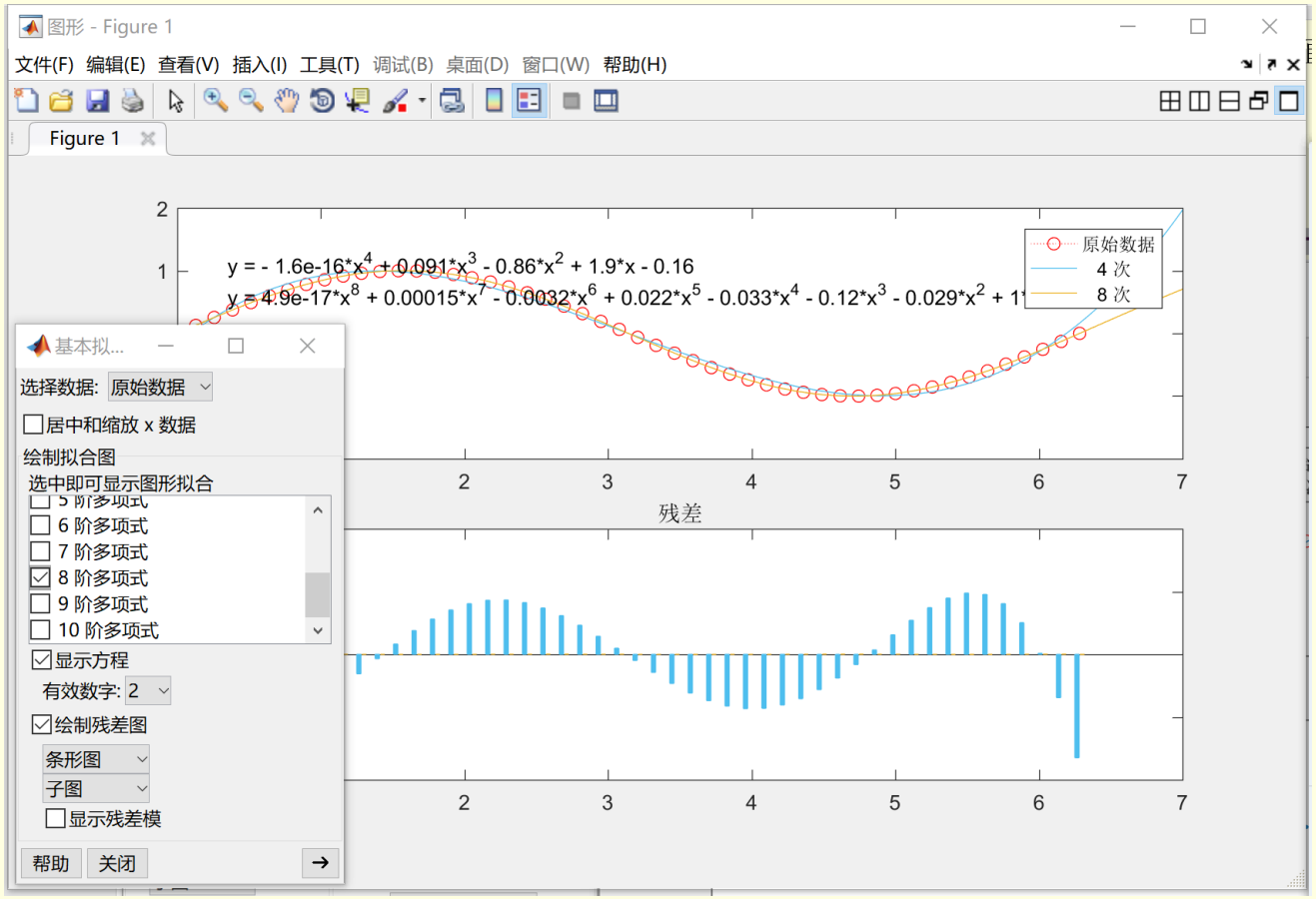
- X=linspace(0,2*pi,50);
- Y=sin(X);
- P3=polyfit(X,Y,3);　%得到3次多项式的系数
- Y3=polyval(P3,X);　%离散点X上的3次多项式取值
- P8=polyfit(X,Y,8);　%得到8次多项式的系数
- Y8=polyval(P8,X);
- plot(X,Y,':ro',X,Y3,'-b*',X,Y8,':^b')
- axis([0,2*pi,-1,1])　　%坐标轴范围
- legend('原始数据','3阶多项式拟合','8阶多项式拟合','Location', ‘southwest');

5.2.7 插值与拟合



5.2.7 插值与拟合

- 方法二：曲线拟合图形用户接口
- Figure窗口点击工具-基本拟合



5.2.8 函数的极值和零点

函数名称	函数格式	说明
函数极小值	x=fminbnd('fun',a,b)	fun为待求极值的单变量函数，a、b为求极值的区间。x为函数极值点，y为极值点的函数值 Find minimum band
	[x,y]=fminbnd('fun',a,b)	
函数零点	x=fzero('fun',a)	寻找a附近的零点，或者寻找区间[a b]上的零点。 x为函数零点，y为零点的函数值。若没有零点，则返回Nan（非数） Find zero
	x=fzero('fun',[a b])	
	[x,y]=fzero('fun',a)	
	[x,y]=fzero('fun',[a b])	

5.2.8 函数的极值和零点

【例21】求函数$f(x)=x^3-2x+1$在$x=[-1\ \ 1]$之间的极小值和零点

- [x,y]=fminbnd('x^3-2*x+1',-1,1)
- [x,y]=fzero('x^3-2*x+1',[-1,1])
- [x,y]=fminbnd('-(x^3-2*x+1)',-1,1)

5.2.8 函数的极值和零点

【例22】用fzero求解单变量非线性方程

$$\frac{1}{(x+4)^2+1}+\frac{1}{(x-4)^2+1}=\frac{1}{2}$$

```
xzero =

   -5.0246

xzero =

   -2.9587

xzero =

    2.9587

xzero =

    5.0246
```

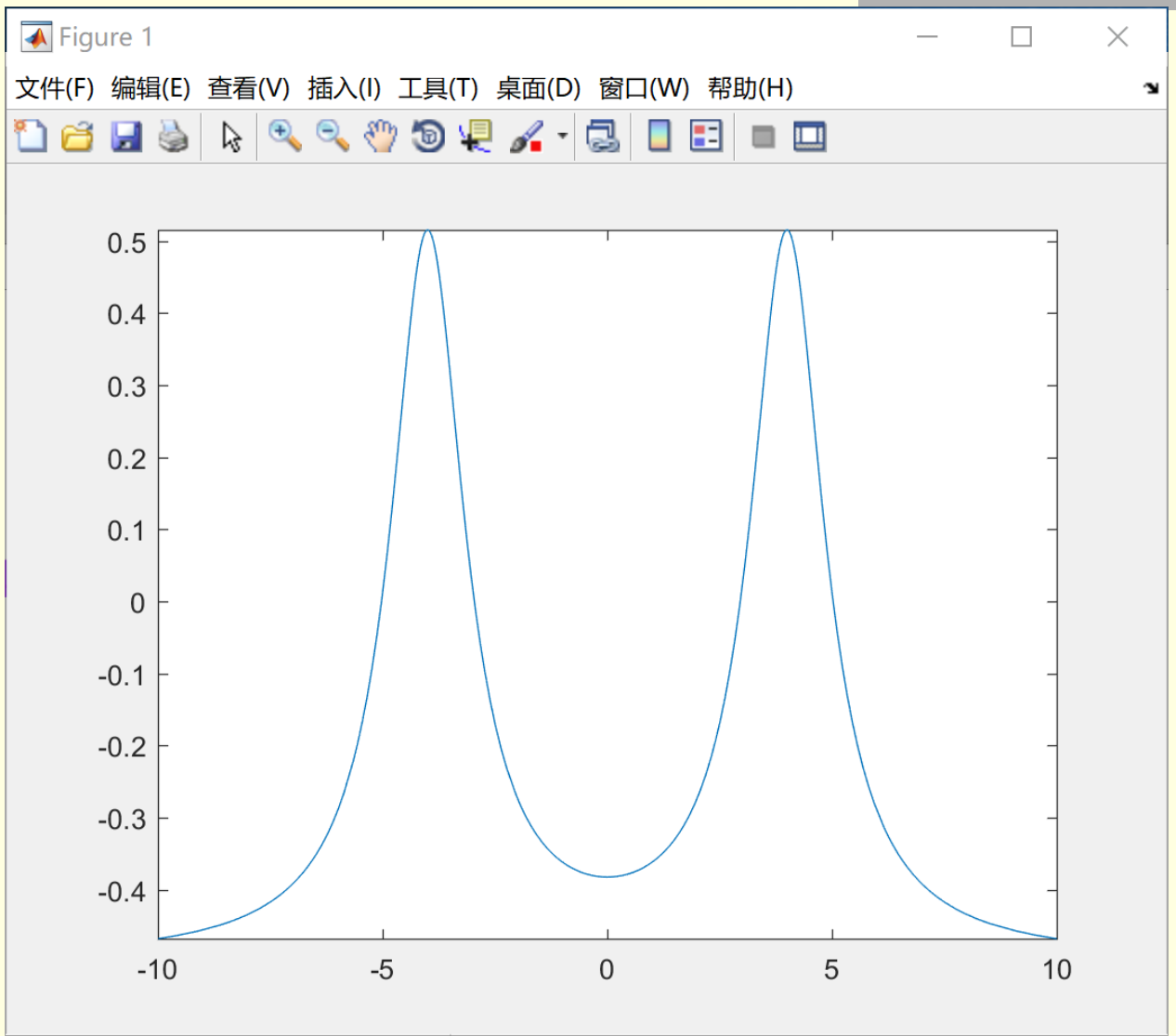
5.2.8 函数的极值和零点

【例22】用fzero求解方程 $\frac{1}{(x+4)^2+1}+\frac{1}{(x-4)^2+1}=\frac{1}{2}$

- syms x
- f = 1/((x+4)^2+1)+1/((x-4)^2+1)-1/2;
- fplot(f,[-10,10])
-
- y='1./((x+4).^2+1)+1./((x-4).^2+1)-1/2';
- xzero=fzero(y,-5)
- xzero=fzero(y,-3)
- xzero=fzero(y,3)
- xzero=fzero(y,5)

5.2.8 函数的极值和零点

【例22】



5.3 数值微积分

- 5.3.1 数值微分
- 5.3.1 数值积分

5.3.1 数值微分

在数学中，函数 $f(x)$ 在点 $x = x_0$ 的导数是通过极限定义的。

$$f'(x_0) = \lim_{\Delta x \to 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$

如果一个函数是以离散数值的形式给出的，那么就无法用极限运算方法求得导数，更无法用求导数方法计算函数在某点处的导数值。

在科学实验和生产实践中，有时要根据已知数据点推算某点处的一阶或高阶导数，这时就要使用数值微分方法。

5.3.1 数值微分

函数的导数依然是一个函数。

高等数学关注 $g(x)=f'(x)$的函数形式和性质。

数值微分关注怎样计算 $g(x)$ 在多个离散点 $X=(x_1,x_2,.....,x_n)$

的近似值 $G=(g_1,g_2,.....,g_n)$ 以及得到的近似值有多大误差。

$f(x)$在点x处以$h(h>0)$为步长的向前差分、向前差商：

向前差分$\Delta f(x)=f(x+h)-f(x)$

向前差商$\dfrac{\Delta f(x)}{h}=\dfrac{f(x)-f(x-h)}{h}$

当步长h足够小时，$f'(x)\approx\dfrac{\Delta f(x)}{h}$

5.3.1 数值微分的实现

在MATLAB中，没有直接提供求数值导数的函数。有两种方法计算数值导数。

方法1：用diff函数计算向前差分，将向前差商作为导数。

函数格式1：DX=diff(X)　　%向量X的向前差分，DX(i)=X(i+1)-X(i)

函数格式2：DX=diff(X,n)　%向量X的n阶向前差分。

% diff(X,2)=diff(diff(X))。

方法2：用polyfit将已知数据在一定范围内拟合为高次多项式f，再对拟合多项式求导数(polyder)，得到函数值

方法3：用符号求导diff（s，x，n）先求出f'(x),再代入点求解导数值

5.3.1 数值微分

【例23】求函数f(x)的导数

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

- 试用不同的方法
- 1.直接求数值导数(diff)
- 2.先进行多项式拟合(polyfit)，再对拟合多项式求导数(polyder)
- 3.对符号函数求导数(diff)

5.3.1 数值微分

【例23】求函数f(x)的导数

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

- %直接对f(x)求数值导数
- x=-3:0.1:3;
- x1=-3:0.1:3.1;
- f1=sqrt(x1.^3+2*x1.^2-x1+12)+(x1+5).^(1/6)+5*x1+2;
- dx=diff(f1)/0.1;
- plot(x,dx,'-ob');
- hold on

5.3.1 数值微分

【例23】求函数f(x)的数值导数

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

- %先进行多项式拟合，再对拟合多项式求导数
- x=-3:0.1:3;
- f=sqrt(x.^3+2*x.^2-x+12)+(x+5).^(1/6)+5*x+2;
- p=polyfit(x,f,5)　　%用5次多项式p拟合f(x)
- dp=polyder(p);　　%对拟合多项式p求导数dp
- dpx=polyval(dp,x);　　%求dp在假设点的函数值
- plot(x,dpx,'-*r');

5.3.1 数值微分

【例23】求函数f(x)的导数

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

- %利用符号导数
- syms s ;
- f2=sqrt(s^3+2*s^2-s+12)+(s+5)^(1/6)+5*s+2;
- f3=diff(f2,s);
- fplot(f3,[-3,3],'-^k')
- legend('直接数值导数','多项式拟合后求导',...
- '符号导数','Location', 'northwest');
  - hold off

5.3.1 数值微分

【例23】

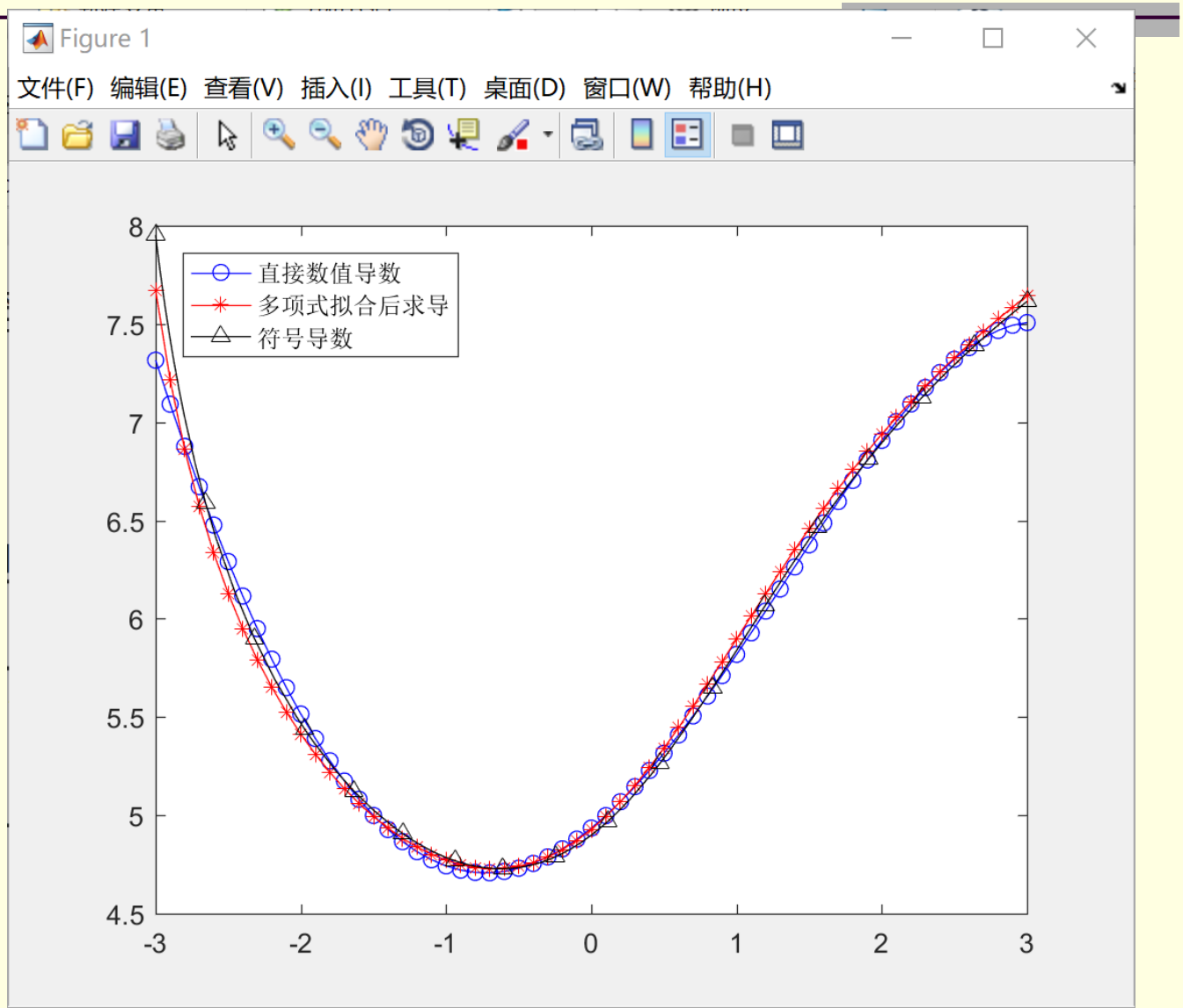


Figure 1
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
直接数值导数
多项式拟合后求导
符号导数
8
7.5
7
6.5
6
5.5
5
4.5
-3
-2
-1
0
1
2
3

5.3.2 数值积分

对于积分 $I=\int_a^b f(x)dx$，只要找到被积函数的原函数 $F(x)$ 便可以根据Newton-Leibniz公式，得到：

$$I=\int_a^b f(x)dx=F(b)-F(a)$$

但以下两种情况：

1. 被积函数的原函数找不到或比较复杂；
2. 被积函数是以一组离散数据的形式给出。

需要用数值积分方法求解。

5.3.2 数值积分

在数学中，定积分是求解一个函数在给定区间内的面积或曲线长度的方法。

数值积分是对定积分的数值近似计算，基本思想都是将整个积分区间分成若干个子区间，计算每个小区间上函数值的加权平均来求得，这样求定积分问题就分解为求和问题。

常见的数值积分方法包括矩形法、梯形法、辛普森(Simpson) 法、牛顿－柯特斯(Newton-Cotes)法等。

5.3.2 数值积分

MATLAB 提供了多种数值积分函数,这些函数使用不同的数值积分方法。

- **Q= integral(f,a,b)** **%一重定积分**
- **Q= quadgk(f,a,b)** **%一重定积分 高斯-勒让德积分法**
- **Q= trapz(x,y)** **%用于计算离散数据的一维数值积分**
- **Q = integral2(f,xmin,xmax,ymin,ymax)** **%二重定积分**
- **Q = integral3(f,xmin,xmax,ymin,ymax,zmin,zmax)**

f是用于计算被积函数的句柄或函数名称

a和b是积分区间的上下限

5.3.2 数值积分

在科学实验和生产实践中遇到的定积分问题通常被积函数的原函数找不到或比较复杂，这时需要用数值积分方法。

格式： **Q= integral(f,a,b)**

其中：f - 被积函数的函数句柄；

a 积分下限；b 积分上限

函数句柄的创建：

方式①：@函数名

```
fun1 = @sin;
```

方式②：@(参数列表)函数表达式

```
fun2 = @(x, y)x.^2 + y.^2;
```

方式③：str2func函数

语法：str2func('函数名')

```
fun3 = str2func('cos');
```

5.3.2 数值积分

- 【例24】求函数的数值积分

$$I = \int_0^1 e^{-x^2} dx$$

- **fun=@(x)exp(-x.^2);**
- **I=integral(fun,0,1)**

5.3.2 数值积分

quadgk()函数：自适应Gauss-Kronrod数值积分，适用于高精度和震荡数值积分、广义积分甚至是复数积分，支持无穷区间，可以处理端点处的适度奇异性。此函数还可以解决沿分段性路径的围道积分。

调用格式：

Q= quadgk(f,a,b,param1,val1,param2,val2,…)

其中：f - 被积函数的函数句柄；

a 积分下限；b 积分上限

parami,vali: 相关属性名及其属性值。具体看帮助

注：**1.被积函数可以剧烈震荡**

2.积分限可以是-inf或inf，**但必须快速衰减；**

3.被积函数在端点可以有奇点，如果区间内部有奇点，将以奇点区间划分成多个，也就是说奇点只能出现在端点上.

5.3.2 数值积分

- 【例25】计算无穷积分

$$I = \int_0^{\infty} e^{-x^2} dx$$

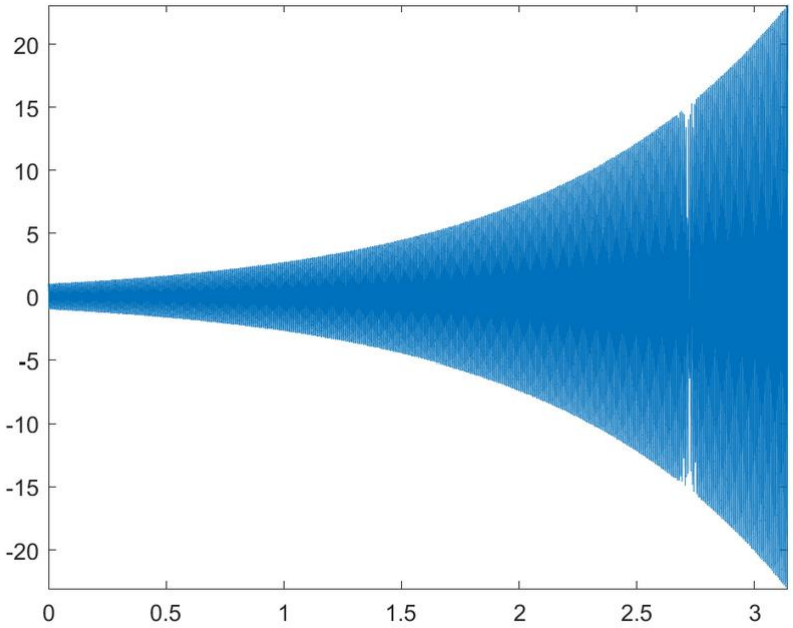
```
I = 0.8862
```

```
fun=@(x)exp(-x.^2);
I=integral(fun,0,inf)
I=quadgk(fun,0,inf)
```

5.3.2 数值积分

- 【例25】计算震荡积分

$$I = \int_0^{\pi} e^x \cos(1000x)dx$$



- **fun=@(x)exp(x).*cos(1**
- fplot(fun,[0,pi])
- **I=integral(fun,0,pi)**
- **I=quadgk(fun,0,pi,'MaxIntervalCount',1000)**

5.3.2 数值积分

- 【例25】计算复数积分

$$I=\int_{2}^{6-j5} e^{-x^2-jx}\sin\left[(7+j2)x\right]dx$$

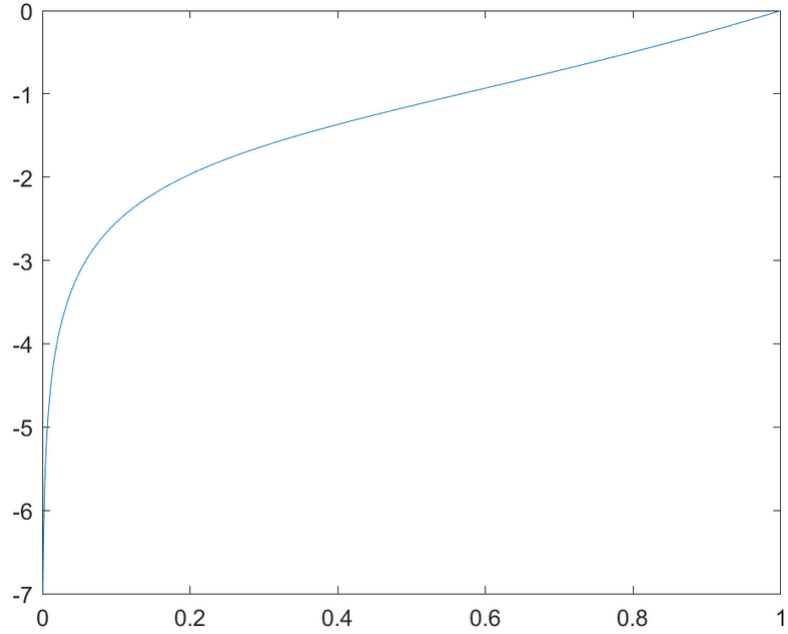
```
I = -0.9245 + 25.7921i
```

- fun=@(x)exp(-x.^2-x*j).*sin((7+2*j)*x);
- I=integral(fun,2,6-5j)
- I=quadgk(fun, 2,6-5j)

5.3.2 数值积分

- 【例25】计算有奇点积分

$$I = \int_0^1 e^x \ln(x)\, dx$$



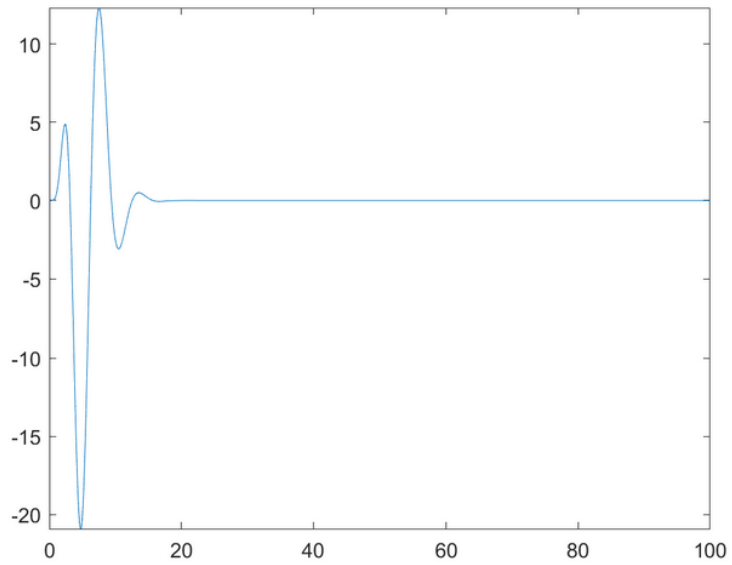

```
I = -1.3179
```

- fun=@(x)exp(x).*log(x);　%奇点只能出现在端点上
- fplot(fun,[0,1])
- I=integral(fun,0,1)
- I=quadgk(fun,0,1)

5.3.2 数值积分

【例25】计算半无限震荡积分

$$I=\int_0^{\infty} x^5 e^{-x}\sin(x)\,dx$$

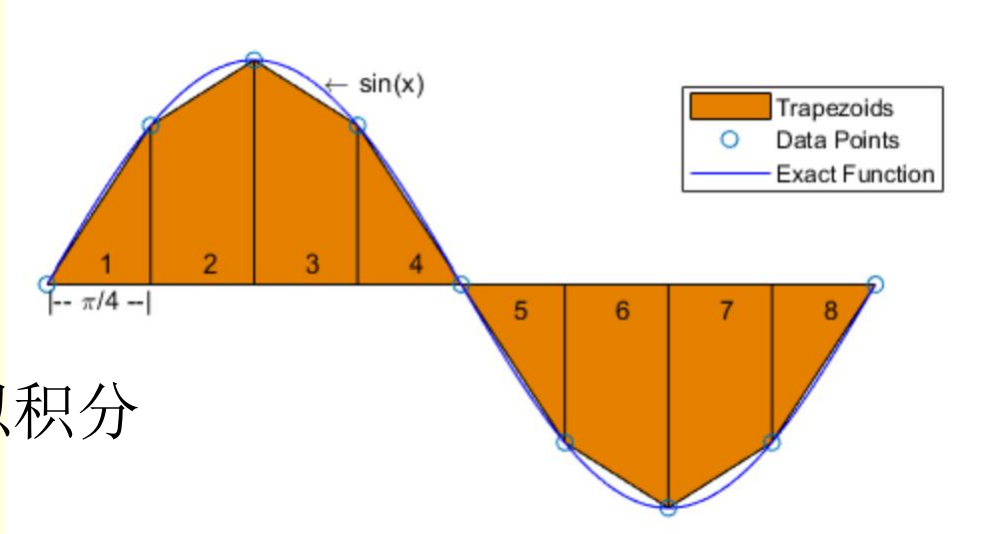



```
I = -15.0000
```

- fun=@(x)x.^5.*exp(-x).*sin(x);
- fplot(fun,[0,100]) %半无限震荡积分必须快速收敛
- I=integral(fun,0,inf)
- I=quadgk(fun,0,inf)

5.3.2 数值积分

trapz()函数对数值数据、而不是函数表达式求积分.
在已知函数表达式的情况下，可以改用 integral、integral2 或 integral3。



调用格式：

Q=trapz(Y)

%通过梯形法计算 Y 的近似积分

Q = trapz(X,Y)

%用梯形法计算y在x点的积分。