

第四章 插值与多项式逼近

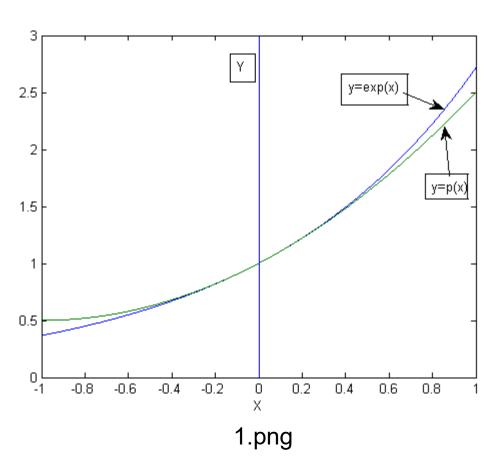
- ■插值
 - □泰勒级数
 - □ Lagrange逼近
 - □ Newton多项式
- 多项式MATLAB命令

4.1问题提出:

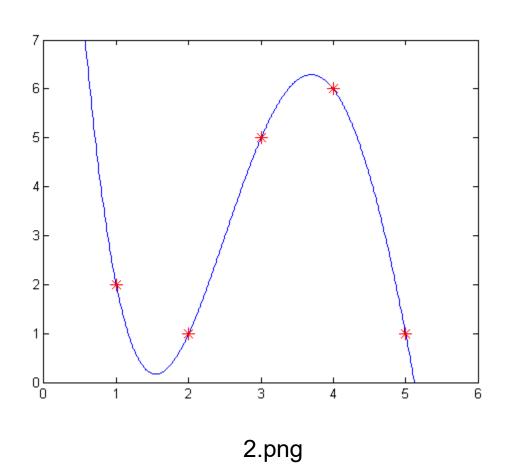
(1) 函数的多项式逼近: sin(x)、cos(x)、e^x

泰勒多项式逼近

fig1.m



(2) 给定平面上多个点,构造多项式,使之过给定点:



定理: 泰勒多项式逼近

设 $f \in C^{N+1}[a,b]$, $x_0 \in [a,b]$, 如果 $x \in [a,b]$, 则有

$$f(x) = P_N(x) + E_N(x)$$

其中用来近似的多项式 P_N(x)

$$f(x) \approx P_N(x) = \sum_{k=0}^{N} \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k$$

误差项

$$E_N(x) = \frac{f^{(N+1)}(c)}{(N+1)!} (x - x_0)^{N+1}$$

c为x和x₀之间的某个值。

误差界:

设 $|\mathbf{x}-\mathbf{x}_0|$ <R,区间宽度为2R,有|误差 $|=|E_N(\mathbf{x})| \le \frac{MR^{N+1}}{(N+1)!}$ M为 $\mathbf{f}^{(N+1)}$ 的上界,当N最大,R最少时,误差最小例: 求逼近多项式 $e^x \approx P_8(x)$ 在区间 $|\mathbf{x}| \le 1.0$ 的误差界。

R=1.0; M=max|f⁽⁹⁾(c)|=e^c ≤ e^{1.0} |误差|= $|E_N(x)| \le \frac{MR^{N+1}}{(N+1)!} = \frac{e^{1.0}1.0^9}{9!} \approx 0.00000749$ N!=prod(1:N)

4.2 插值

构造泰勒多项式需知道 x_0 处的函数f及其导数;设函数y=f(x)在N+1个点 $(x_0,y_0),...,(x_N,y_N)$ 已知,且在区间[a,b]上满足:

 $a \le x_0 < x_1 < \dots < x_N \le b, y_k = f(x_k)$ 可构造经过这N+1个点的N次多项式P(x),

$$P(x) = a_N x^N + a_{N-1} x^{N-1} + \dots + a_1 x + a_0$$

并在整个区间[a,b]用多项式P(x)逼近f(x);

$$x_0 < x < x_N$$
: 内插值: $P(x)$;

x<x₀或x>x_N: 外插值: P(x)

若已知值存在显著误差,chapter5: 曲线拟合

121 agranga

4.3 Lagrange逼近

插值:利用邻近点在已知函数值的加权平均来估计未知函数值;

线性插值:
$$y = P(x) = y0 + (y_1 - y_0) \frac{x - x_0}{x_1 - x_0}$$

Lagrange:

$$y = P_1(x) = y0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0}$$

记力:
$$P_1(x) = \sum_{k=0}^{1} y_k L_{1,k}(x)$$

推广到n+1个点

对给定的n+1个插值点 $x_0, x_1, x_2, \dots, x_n$ 及对应的函数值 $y_0, y_1, y_2, \dots, y_n$,构造的n次Lagrange插值多项式,则对插值区间内任意x的函数值y可通过下式求得:

$$y(x) = \sum_{k=0}^{n} y_k L_{n,k} = \sum_{k=0}^{n} y_k \left(\prod_{\substack{j=0 \ j \neq k}}^{n} \frac{x - x_j}{x_k - x_j} \right)$$

- function [C,L]=lagran(X,Y)
- w=length(X); n=w-1; L=zeros(w,w);
- %Form the Lagrange coefficient polynomials
- for k=1:n+1
- V=1;
- for j=1:n+1
- if k~=j
- V=conv(V,poly(X(j)))/(X(k)-X(j)); end
- end
- L(k,:)=V;
- end
- C=Y*L

$$y(x) = \sum_{k=0}^{n} y_k L_{n,k} = \sum_{k=0}^{n} y_k \left(\prod_{\substack{j=0 \ j \neq k}}^{n} \frac{x - x_j}{x_k - x_j} \right)$$

M

■ 例:给出f(x)=ln(x)的数值表,用Lagrange计算ln(0.54)的近似值。

lagran.m

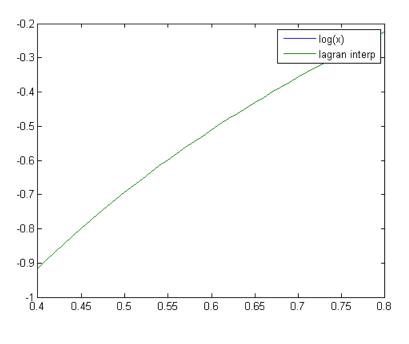
```
y=[-0.916291,-0.693147,-0.510826,-0.356675,-0.223144];
```

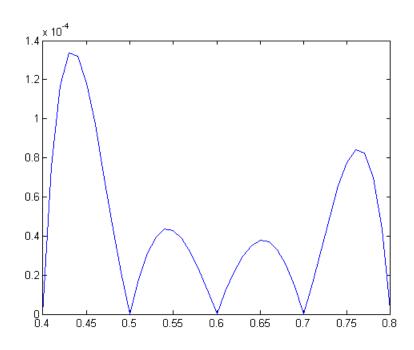
[C,L]=lagran(x,y);

vpa(poly2sym(C),5)%显示多项式

y1=polyval(C,0.54); %求多项式值

Error=abs(y1-log(0.54))





3.png

4.png

м

■误差项和误差界

- 定理(拉格朗日多项式逼近): 设 $f \in C^{N+1}[a,b]$
- 且 $x_0, x_1, x_2, \dots, x_N \in [a,b]$, 如果 $x \in [a,b]$, 则

$$f(x) = P_N(x) + E_N(x)$$

■ 其中逼近多项式 $P_N(x)$: $f(x) \approx P_N(x) = \sum_{k=0}^{N} f(x_k) L_{N,k}(x)$

■ 误差项:

$$E_N(x) = \frac{\prod_{i=0}^{N} (x - x_i) f^{(N+1)}(c)}{(N+1)!}$$

$$c \in [a,b]$$

- м
 - ■定理(等距节点拉格朗日多项式的误差界)
 - ■设f(x)定义在[a,b]上,其中包含等距节点

 - ■则误差项的界分别为

$$|E_1(x)| \le \frac{h^2 M_2}{8}, x \in [x_0, x_1]$$

$$|E_2(x)| \le \frac{h^3 M_3}{9\sqrt{3}}, x \in [x_0, x_2]$$

$$|E_3(x)| \le \frac{h^4 M_4}{24}, x \in [x_0, x_3]$$

$$|E_N(x)| \approx O(h^{N+1})$$

- M
 - 例 考虑[0,1.2]上的y= $f(x)=\cos(x)$,确定拉格 朗日多项式 $P_1(x)$ 、 $P_2(x)$ 、 $P_3(x)$ 在等距节 点时的误差界
 - ■确定导数在区间上的界:

$$|f^{(2)}(x)| = |-\cos(x)| \le |-\cos(0.0)| = 1.0 = M_2$$

$$|f^{(3)}(x)| = |\sin(x)| \le |\sin(1.2)| = 0.932039 = M_3$$

$$|f^{(4)}(x)| = |\cos(x)| \le |\cos(0.0)| = 1.0 = M_4$$

- $P_1(x)$, h=1.2:
- $P_2(x)$, h=0.6:
- $P_3(x)$, h=0.4,

$$|E_1(x)| \le \frac{h^2 M_2}{8} = 0.180000$$

$$|E_2(x)| \le \frac{h^3 M_3}{9\sqrt{3}} = 0.012915$$

$$|E_3(x)| \le \frac{h^4 M_4}{24} = 0.001067$$

多项式MATLAB命令

■ 一个多项式的幂级数形式可表示为:

$$y = c_1 x^n + c_2 x^{n-1} + \dots + c_n x + c_{n+1}$$

■ 也可表为嵌套形式(Horner's method)

$$y = (\cdots((c_1x + c_2)x + c_3)x \cdots + c_n)x + c_{n+1}$$

■或因子形式

$$y = c_1(x - r_1)(x - r_2) \cdots (x - r_n)$$

n阶多项式有n个根,其中包含重根和复根。若多项式所有系数均为实数,则全部复根都将以共轭对的形式出现

■ 多项式系数:在MATLAB里,多项式用行向量表示, 其元素为多项式的系数,并从左至右按降幂排列。

例:
$$y = 2x^3 + x^2 + 4x + 5$$

表示为 $p=[2\ 1\ 4\ 5]$
poly2sym(p)
ans = $2*x^3 + x^2 + 4x + 5$

■ 多项式根:用roots求多项式的零点。

例: r=roots(p)所有零点(解),由一个列向量给出ans = 0.2500 + 1.5612i % the roots of polynomial p 0.2500 - 1.5612i -1.0000

poly: 由零点可得原始多项式的各系数,最高次系数默认为1。

例: poly(2) %ans = 1.0000 -2

conv()

C = conv(A, B)

A和B为两矢量,C矢量即为两多项式乘的系数, 长度为: LENGTH(A)+LENGTH(B)-1

```
r1=poly(2); %r1=[1 -2]
r2=poly([3 4]); %r2=[1 -7 12]
r=conv(r1,r2); %r= [1 -9 26 -24]
poly2sym(r); % x^3 - 9*x^2 + 26*x - 24
```

多项式求值

Y =polyval(P,X), 当 P 为长N+1的矢量,其元素为多 polyval() 形式系数, 求多项式在 X点值.

 $Y = P(1)*X^N + P(2)*X^(N-1) + ... + P(N)*X + P(N+1)$

4.4 牛顿多项式

为了在 $P_1(x)$ 、 $P_2(x)$ 、…, $P_N(x)$ 选择最合适的一个多项式作为逼近,当采用拉格朗日插值多项式时,插值节点若有增减,全部插值基函数 $L_{N,k}(x)(k=0,1,...,N)$ 均要随之变化。

可构造牛顿多项式,具有如下递归关系:

$$P_1(x) = a_0 + a_1(x - x_0)$$

:

$$P_N(x) = P_{N-1}(x) + a_N(x - x_0) \cdots (x - x_{N-1})$$

M

为了找出逼近给定函数f(x)的多项式 $P_1(x)$ 、 $P_2(x)$ 、 ..., $P_N(x)$, 所有系数 $a_0, a_1, \cdots a_N$ 可由 N+1 个插值条件

$$P_N(x_j) = f_j$$
 $(j = 0, 1, \dots, N)$

确定.

当
$$x = x_0$$
 时, $P_N(x_0) = a_0 = f_0$.
当 $x = x_1$ 时, $P_N(x_1) = a_0 + a_1(x_1 - x_0) = f_1$ 推得

$$a_1 = \frac{f_1 - f_0}{x_1 - x_0}.$$

当
$$x=x_2$$
时,由

$$P_N(x_2) = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) = f_2$$

推得

$$a_2 = \frac{\frac{f_2 - f_1}{x_2 - x_1} - \frac{f_1 - f_0}{x_1 - x_0}}{x_2 - x_0}.$$

依此递推可得到 $a_3, \cdots a_N$

定义 $f[x_k] = f(x_k)$

$$f[x_{k-1}, x_k] = \frac{f[x_k] - f[x_{k-1}]}{x_k - x_{k-1}}$$
 为函数 $f(x)$ 的一阶差商.

$$f[x_{k-2}, x_{k-1}, x_k] = \frac{f[x_{k-1}, x_k] - f[x_{k-2}, x_{k-1}]}{x_k - x_{k-2}}$$

称为 f(x) 的二阶差商.

一般地, 称

$$f[x_{k-j}, x_{k-j+1}, \dots x_k] = \frac{f[x_{k-j+1}, \dots, x_k] - f[x_{k-j}, \dots, x_{k-1}]}{x_k - x_{k-j}}$$

为 f(x)的 j 阶差商

差商计算可列表如下(表4-1).

y=f(x) 差商表

$\overline{x_k}$	$f[x_k]$	一阶差商	二阶差商	三阶差商	四阶差商
$\overline{x_0}$	$f[x_0]$				
x_1	$f[x_1]$	$\underline{f[x_0,x_1]}$			
\mathcal{X}_2	$f[x_2]$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$		
X_3	$f[x_3]$	$f[x_2, x_3]$	$\int [x_1, x_2, x_3]$	$f[x_0, x_1, x_2, x_3]$	
X_4	$f[x_4]$	$f[x_3, x_4]$	$\int [x_2, x_3, x_4]$	$f[x_1, x_2, x_3, x_4]$	$f[x_0, x_1, x_2, x_3, x_4]$
:	:	•	:	:	:

定理(牛顿多项式)设 x_0, \dots, x_N 是区间[a,b] 内N+1个不同的数,存在唯一的至多N次的多项式 $P_N(x)$,具有性质

$$f(x_j) = P_N(x_j)$$
 $j = 0, \dots, N$

该多项式的牛顿形式为

$$P_N(x) = a_0 + a_1(x - x_0) + \dots + a_N(x - x_0) \dots (x - x_{N-1})$$

其中系数

$$a_k = f[x_0, \dots, x_k], k = 0, 1, \dots, N$$

为 k 阶差商

牛顿逼近:设用牛顿多项式 $P_N(x)$ 逼近函数f(x),即 $f(x) = P_N(x) + E_N(x)$

如果 $f \in C^{N+1}[a,b]$,则对每个 $x \in [a,b]$,存在 $c \in [a,b]$ 使得

$$E_N(x) = \frac{(x - x_0)(x - x_1) \cdots (x - x_N) f^{(N+1)}(c)}{(N+1)!}$$

м

■ 例 基于5个点 (k,cos(k)),k=0,1,2,3,4;构造 f(x)=cos(x)的差商表,并求出系数a_k和4个牛顿多项式P_k(x),k=1,2,3,4; newton.m

$$P_N(x) = a_0 + a_1(x - x_0) + \dots + a_N(x - x_0) \cdot \dots \cdot (x - x_{N-1})$$

$$f[x_{k-j}, x_{k-j+1}, \dots x_k] = \frac{f[x_{k-j+1}, \dots, x_k] - f[x_{k-j}, \dots, x_{k-1}]}{x_k - x_{k-j}}$$

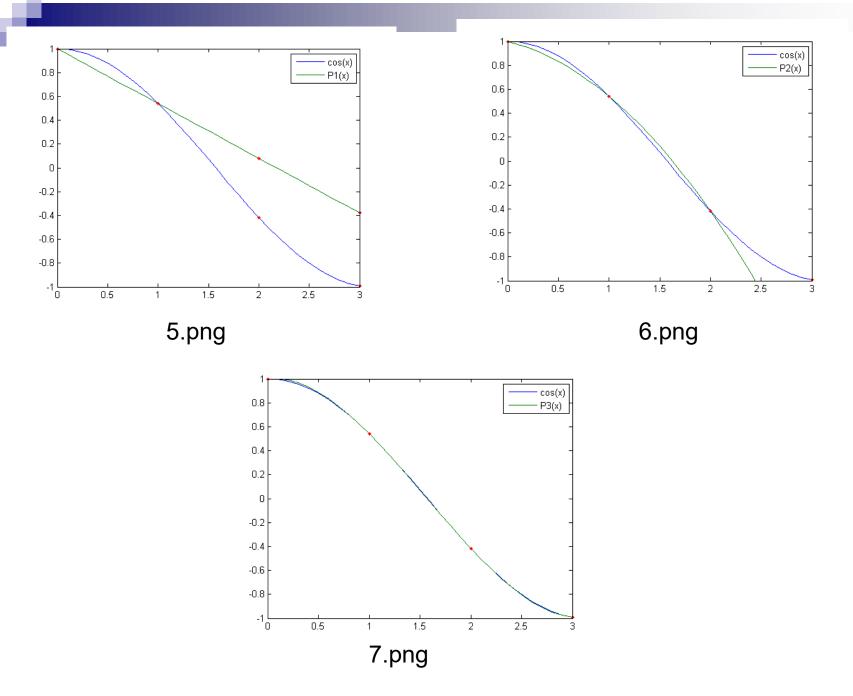
首先根据给定函数表造出差商表.

y=f(x)差商表

$\overline{x_k}$	$f[x_k]$	一阶差商	二阶差商	三阶差商	四阶差商
$x_0 = 0.0$	1.0000000				
$x_1 = 1.0$	0.5403023	<u>-0.4596977</u>			
$x_2 = 2.0$	-0.4161468	-0.9564491	<u>-0.2483757</u>		
$x_3 = 3.0$	-0.9899925	-0.5738457	0.1913017	0.1465592	
$x_4 = 4.0$	-0.6536436	0.3363499	0.4550973	0.0879318	<u>-0.0146568</u>

4个牛顿多项式为:

$$\begin{split} \mathsf{P}_1(\mathsf{x}) &= 1\text{-}0.4596977(\mathsf{x}\text{-}0); \\ \mathsf{P}_2(\mathsf{x}) &= 1\text{-}0.4596977(\mathsf{x}\text{-}0)\text{-}0.2483757(\mathsf{x}\text{-}0)(\mathsf{x}\text{-}1); \\ \mathsf{P}_3(\mathsf{x}) &= 1\text{-}0.4596977(\mathsf{x}\text{-}0)\text{-}0.2483757(\mathsf{x}\text{-}0)(\mathsf{x}\text{-}1) \\ &\quad + 0.1465592(\mathsf{x}\text{-}0)(\mathsf{x}\text{-}1)(\mathsf{x}\text{-}2); \\ \mathsf{P}_4(\mathsf{x}) &= 1\text{-}0.4596977(\mathsf{x}\text{-}0)\text{-}0.2483757(\mathsf{x}\text{-}0)(\mathsf{x}\text{-}1) \\ &\quad + 0.1465592(\mathsf{x}\text{-}0)(\mathsf{x}\text{-}1)(\mathsf{x}\text{-}2) \\ &\quad - 0.0146568(\mathsf{x}\text{-}0)(\mathsf{x}\text{-}1)(\mathsf{x}\text{-}2)(\mathsf{x}\text{-}3); \end{split}$$



■ 在[0,4]间,用P₄(x)逼近f=cos(x)的误差界

$$E_N(x) = \frac{(x - x_0)(x - x_1) \cdots (x - x_N) f^{(N+1)}(c)}{(N+1)!}$$

- $M=\max|f^{(5)}(c)|=|\sin(c)|\leq 1$;

■ 没x=0.5;

$$|E_4(0.5)| = \left| \frac{(0.5-0)(0.5-1)(0.5-2)(0.5-3)(0.5-4)*1}{(4+1)!} \right|$$

= 0.0273

- ■作业:
- 4.1.2 P137 #1
- 4.3.4 P153 #10
- 4.4.3 P161 #7

- ■插值的Matlab实现
- Interp1
- Interp2
- Interp3



yi = interp1(x,y,xi,method)

一维插值函数,

Nearest neighbor interpolation (method = 'nearest').

Linear interpolation (method = 'linear'). (default)

Cubic spline interpolation (method = 'spline').

Cubic interpolation (method = 'pchip' or 'cubic'). These methods are identical. They use the pchip function to perform piecewise cubic Hermite interpolation within the vectors x and y. These methods preserve monotonicity and the shape of the data.

□命令1 interp1

- 功能: 一维数据插值(表格查找)。该命令对数据点之间计算内插值。它找出一元函数f(x)在中间点的数值。其中函数f(x)由所给数据决定。
- 格式1 yi = interp1(x,y,xi)

%返回插值向量yi,每一元素对应于参量xi,同时由向量x与y的内插值决定。参量x指定数据y的点。若y为一矩阵,则按y的每列计算。

■ **例**:对于temp,分别有beta、alpha两组数据与之对应,用分段线性插值法计算当ti=321,440,571时beta、alpha的值。

```
temp=[300,400,500,600]';
beta=1000*[3.33,2.50,2.00,1.67]';
alpha=10000*[0.2128,0.3605,0.5324,0.7190]';
ti=[321,400,571]';
propty=interp1(temp,[beta,alpha],ti);
%propty=interp1(temp,[beta,alpha],ti,' linear');
[ti,propty]
ans =
 1.0e+003 *
   0.3210 3.1557 2.4382
   0.4000 2.5000 3.6050
   0.5710 1.7657 6.6489
```

- 格式2 yi = interp1(y,xi)
 - %假定x=1:N,其中N为向量y的长度,或者为矩阵y的行数。
- **格式3** yi = interp1(x,Y,xi,method) %用指定的算法计算插值:
 - 'nearest':最近邻点插值,直接完成计算;
 - 'linear':线性插值(缺省方式),直接完成计算;
 - 'spline':三次样条函数插值。
 - 'cubic': 分段三次Hermite插值。

其它,如'v5cubic'。

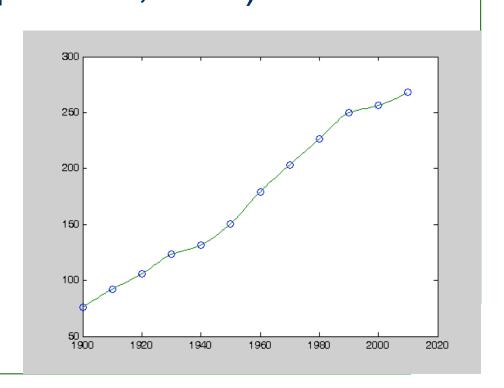
对于超出x范围的xi的分量,使用方

法'nearest'、'linear'、'v5cubic'的插值算法,相应地将返回NaN。对其他的方法,interp1将对超出的分量执行外插值算法。

- yi = interp1(x,y,xi,method,'extrap')
- yi = interp1(x,y,xi,method,extrapval) %确定超出x范围的xi中的分量的外插值extrapval,其值通常取NaN或0。



```
year=1900:10:2010;
product=[74.995,91.972,104.711,123.203,131.669,...
  150.697,179.323,203.212,226.505,249.633,256.344,267.893];
p1995 = interp1(year, product, 1995)
%p1995 = 252.9885
x = 1900:1:2010;
y = interp1(year,...
product,x,'cubic');
plot(year,...
product, 'o', x, y)
```

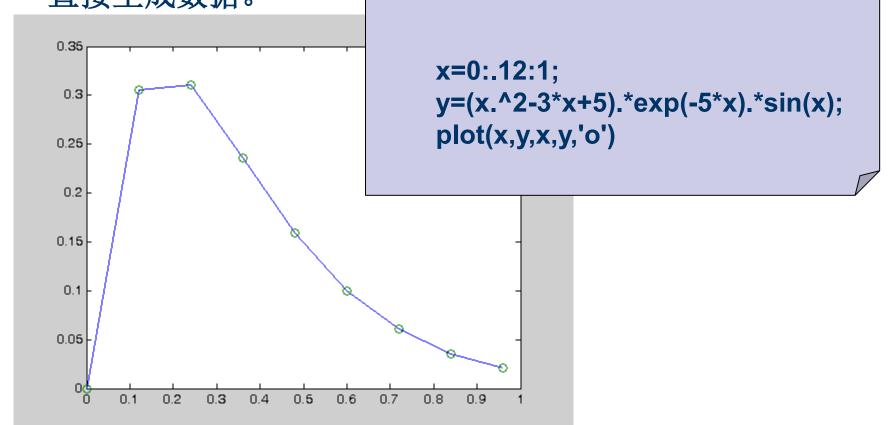


interp1.m

例:已知的数据点来自函数

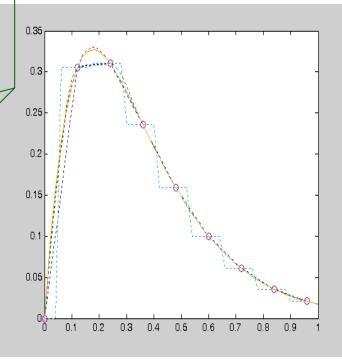
interp1 1.m

$$f(x) = (x^2 - 3x + 5)e^{-5x} \sin x$$



■ 调用interp1()函数:

```
x1=0:.02:1;
y0=(x1.^2-3*x1+5).*exp(-5*x1).*sin(x1);
y1=interp1(x,y,x1);
y2=interp1(x,y,x1,'cubic');
y3=interp1(x,y,x1,'spline');
y4=interp1(x,y,x1,'nearest');
plot(x1,[y1',y2',y3',y4'],':',x,y,'o',x1,y0)
```



•

- ■误差分析
- [max(abs(y0-y1)) max(abs(y0-y2)),
- \blacksquare max(abs(y0-y3)),max(abs(y0-y4))]

ans = 0.0614 0.0177 0.0086 0.1598

□命令2 interp2

- ■功能 二维数据内插值
- 格式1 ZI = interp2(X,Y,Z,XI,YI)

%返回矩阵ZI,其元素包含对应于参量XI与YI (可以是向量、或同型矩阵)的元素。参量X与Y必 须是单调的,且相同的划分格式,就像由命令 meshgrid生成的一样。若Xi与Yi中有在X与Y范围之 外的点,则相应地返回NaN。

- ■**格式2** ZI = interp2(Z,XI,YI) %缺省地,X=1:n、Y=1:m,其中[m,n]=size(Z)。再按第一种情形进行计算。
- ■格式3 ZI = interp2(X,Y,Z,XI,YI,method) %用指定的算法method计算二维插值:
 - 'linear': 双线性插值算法(缺省算法);
 - 'nearest': 最临近插值;
 - 'spline':三次样条插值;
 - 'cubic':双三次插值。

4

■ 例:

interp2_1.m

years=1950:10:1990;

service=10:10:30;

wage = [150.697 199.592 187.625

179.323 194.072 250.287

203.212 179.092 322.767

226.505 153.706 426.730

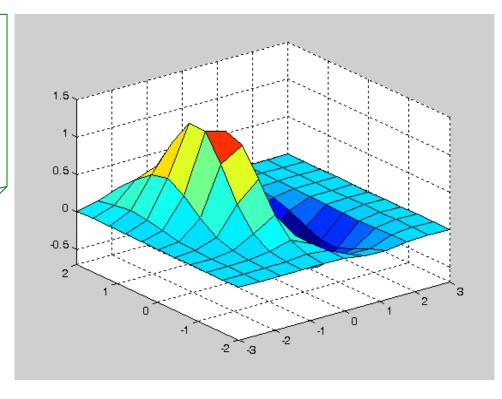
249.633 120.281 598.243];

w = interp2(service, years, wage, 15, 1975)

w = 190.6288

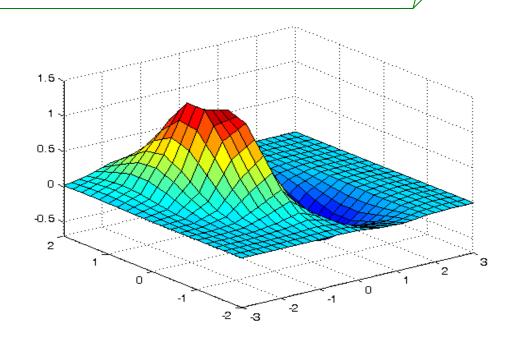
■例: 由 $z = f(x,y) = (x^2 - 2x)e^{-x^2 - y^2 - xy}$ 可计算出一些较稀疏的网格数据,对整个函数曲面进行各种插值拟合,并比较拟合结果,绘制已知数据的网格图。 interp2 2.m

[x,y]=meshgrid(-3:.6:3,-2:.4:2); z=(x.^2-2*x).*exp(-x.^2-y.^2-x.*y); surf(x,y,z), axis([-3,3,-2,2,-0.7,1.5])



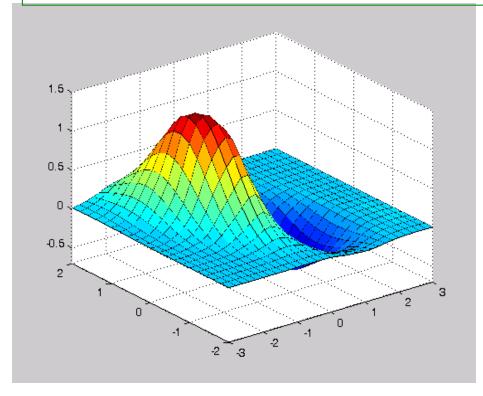
■ 选较密的插值点,用默认的线性插值算 法进行插值

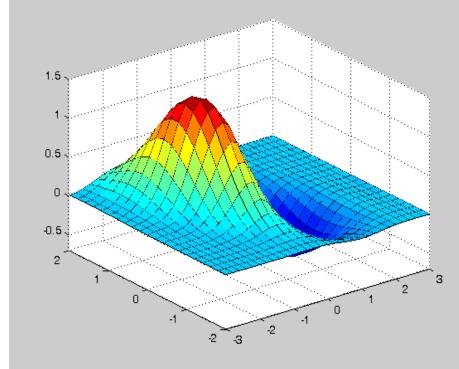
```
[x1,y1]=meshgrid(-3:.2:3,-2:.2:2);
z1=interp2(x,y,z,x1,y1);
surf(x1,y1,z1),axis([-3,3,-2,2,-0.7,1.5])
```



■ 立方和样条插值:

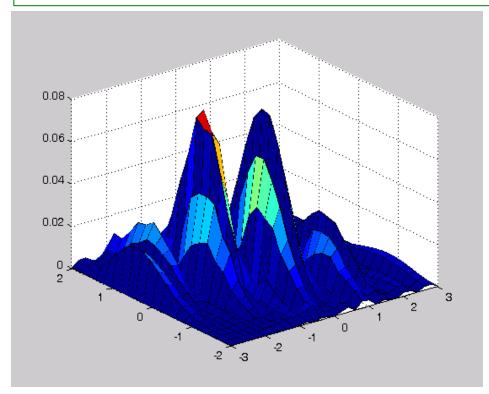
```
z1=interp2(x,y,z,x1,y1,'cubic');
z2=interp2(x,y,z,x1,y1,'spline');
surf(x1,y1,z1),axis([-3,3,-2,2,-0.7,1.5])
figure;surf(x1,y1,z2),axis([-3,3,-2,2,-0.7,1.5])
```

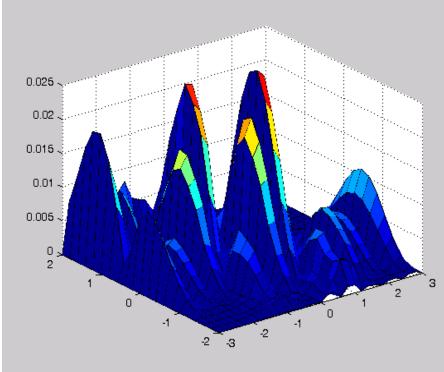




■算法误差的比较

z=(x1.^2-2*x1).*exp(-x1.^2-y1.^2-x1.*y1); surf(x1,y1,abs(z-z1)),axis([-3,3,-2,2,0,0.08]) figure; surf(x1,y1,abs(z-z2)),axis([-3,3,-2,2,0,0.025])





.

- VI = interp3(X,Y,Z,V,XI,YI,ZI,method) 3维插值函数
- Nearest neighbor interpolation (method = 'nearest'). This method chooses the value of the nearest point.
- Trilinear interpolation (method = 'linear'). This method uses piecewise linear interpolation based on the values of the nearest eight points.
- *Tricubic interpolation* (method = 'cubic'). This method uses piecewise cubic interpolation based on the values of the nearest sixty-four points.

М

□命令3 interp3

三维网格生成用meshgrid()函数,调用格式:

[x,y,z]=meshgrid(x1,y1,z1)

其中x1,y1,z1为这三维所需要的分割形式,应以向量形式给出,返回x,y,z为网格的数据生成,均为三维数组。

griddata3() 三维非网格形式的插值拟合

□命令4 interpn

n维网格生成用ndgrid()函数,调用格式:

[x1,x2,...,xn]=ndgrid[v1,v2,...,vn]

griddatan() n维非网格形式的插值拟合

interp3 ()、 interpn()调用格式同interp2()函数一致;

griddata3()、griddatan()调用格式同griddata()函数一致。

■例:通过函数生成一些网格型样本点, 试根据样本点进行插值,并给出最大绝 对误差。

```
[x,y,z]=meshgrid(-1:0.2:1); interp3.m

[x0,y0,z0]=meshgrid(-1:0.05:1);

V= (x.^2.*z+y.^2.*x+z.^2.*y).*cos(x.^2.*y.*z+z.^2.*y.*x);

V0= (x0.^2.*z0+y0.^2.*x0...

+z0.^2.*y0).*cos(x0.^2.*y0.*z0+z0.^2.*y0.*x0);

V1=interp3(x,y,z,V,x0,y0,z0,'spline'); err=V1-V0;

max(err(:))
```