第 6 讲　插值模型

司守奎
烟台市，海军航空大学
Email：sishoukui@163.com

插值，通俗地说，就是已知若干点的函数值插入计算一些未知点的函数值。插值在工程技术和数据处理等领域都有广泛的应用。

6.1　一维插值方法

一维插值问题如下：已知$n+1$个结点(x_i, y_i)，$i=0,1,\cdots,n$，且x_i互不相同，求任意插值点x^*处的插值y^*。

求解一维插值问题的思想如下：假设这$n+1$个结点是由未知的连续函数$f(x)$产生的，即$f(x_i)=y_i$（$i=0,1,\cdots,n$），构造相对简单的函数$g(x)$来逼近$f(x)$，使得$g(x)$经过这$n+1$个结点，即满足$g(x_i)=y_i$（$i=0,1,\cdots,n$），然后用$g(x)$计算插值点x^*处的插值，即$y^*=g(x^*)$。

下面，我们介绍三种一维插值方法：多项式插值，分段线性插值，三次样条插值。

Matlab 工具箱要求x_i（$i=0,1,\cdots,n$）是单调的，不失一般性，不妨设$x_0<x_1<\cdots<x_n$。

6.1.1　多项式插值

已知结点(x_i, y_i)，$i=0,1,\cdots,n$，并且$x_0,x_1,\cdots,x_n$是互异的实数，构造至多n次插值多项式

$$p_n(x)=a_1x^n+a_2x^{n-1}+\cdots+a_nx+a_{n+1} \tag{1}$$

满足

$$p_n(x_i)=y_i,\quad i=0,1,\cdots,n, \tag{2}$$

然后用$p_n(x)$计算插值点x^*处的插值$y^*=p_n(x^*)$。

n次多项式（1）有$n+1$个待定系数，由插值条件（2）恰好给出$n+1$个方程

$$\begin{cases} a_1x_0^n+a_2x_0^{n-1}+\cdots+a_nx_0+a_{n+1}=y_0 \\ a_1x_1^n+a_2x_1^{n-1}+\cdots+a_nx_1+a_{n+1}=y_1 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ a_1x_n^n+a_2x_n^{n-1}+\cdots+a_nx_n+a_{n+1}=y_n \end{cases} \tag{3}$$

记此方程组的系数矩阵为A，则

$$\det(A)=\begin{vmatrix} x_0^n & x_0^{n-1} & \cdots & x_0 & 1 \\ x_1^n & x_1^{n-1} & \cdots & x_1 & 1 \\ \vdots & \vdots & & \vdots & \vdots \\ x_n^n & x_n^{n-1} & \cdots & x_n & 1 \end{vmatrix}$$

是范德蒙特(Vandermonde)行列式。当$x_0,x_1,\cdots,x_n$互不相同时，此行列式值不为零。因此方程组（3）有唯一解。这表明，只要$n+1$个节点互不相同，满足插值要求（2）的插值多项式（1）是唯一的。

构造插值多项式的常用方法有拉格朗日法和牛顿法两种，由插值多项式的唯一性，无论用什么插值方法，求得的插值多项式都是一样的。在此介绍拉格朗日法。

实际上比较方便的作法不是解方程组（3）求待定系数，而是先构造一组基函数

$$l_i(x)=\frac{(x-x_0)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)}$$

$$=\prod_{\substack{j=0\\j\neq i}}^{n}\frac{x-x_j}{x_i-x_j}\quad,\qquad(i=0,1,\cdots,n)$$

$l_i(x)$ 是 n 次多项式，满足

$$l_i(x_j)=\begin{cases}0 & j\neq i\\1 & j=i\end{cases}$$

令

$$L_n(x)=\sum_{i=0}^{n}y_il_i(x)=\sum_{i=0}^{n}y_i\left(\prod_{\substack{j=0\\j\neq i}}^{n}\frac{x-x_j}{x_i-x_j}\right)\qquad（4）$$

上式称为 n 次 Lagrange 插值多项式，由方程组（3）解的唯一性，$n+1$ 个节点的 n 次 Lagrange 插值多项式存在唯一。

由于使用多项式插值一般最高次数不超过 3 次，拉格朗日插值和牛顿插值只有理论上的意义，实际应用价值不大，Matlab 工具箱中没有拉格朗日插值和牛顿插值的函数。

我们可以编写如下拉格朗日插值的函数：

```
function y=mylagrange(x0,y0,x);
n=length(x0);m=length(x);
for i=1:m
    z=x(i);
    s=0.0;
    for k=1:n
        p=1.0;
        for j=1:n
            if j~=k
                p=p*(z-x0(j))/(x0(k)-x0(j));
            end
        end
        s=p*y0(k)+s;
    end
    y(i)=s;
end
```

用 Lagrange 插值多项式 $L_n(x)$ 近似 $f(x)(a\le x\le b)$，虽然随着节点个数的增加，$L_n(x)$ 的次数 n 变大，多数情况下误差 $|L_n(x)-f(x)|$ 会变小。但是 n 增大时，$L_n(x)$ 的光滑性变坏，有时会出现很大的振荡。理论上，当 $n\to\infty$，在 $[a,b]$ 内并不能保证 $L_n(x)$ 处处收敛于 $f(x)$。Runge 给出了一个有名的例子：

$$f(x)=\frac{1}{1+x^2},\quad x\in[-5,5],$$

对于较大的 $|x|$，随着 n 的增大，$L_n(x)$ 振荡越来越大，事实上可以证明，仅当 $|x|\le 3.63$ 时，才有 $\lim_{n\to\infty}L_n(x)=f(x)$，而在此区间外，$L_n(x)$ 是发散的。

例 1（Runge 现象）　在区间$[-5,5]$上分别等间距地取 $f(x)=\dfrac{1}{1+x^2}$ 的 11 个点和 16 个点，做出对应的 10 次和 15 次拉格朗日插值多项式，比较插值多项式与原来函数曲线的差异。

求插值函数及画图的 Matlab 程序如下：

```
clc, clear
fx=@(x)1./(1+x.^2);   %定义原来函数的匿名函数
x1=linspace(-5,5,11); x2=linspace(-5,5,16);
y1=fx(x1); y2=fx(x2); %计算离散点的函数值
p1=polyfit(x1,y1,10); p2=polyfit(x2,y2,15); %拟合多项式
fplot(fx,[-5,5],'k.-'), hold on
fplot(@(x)polyval(p1,x),[-5,5],'r>-')
fplot(@(x)polyval(p2,x),[-5,5],'gP-')
legend('原来函数','10 次插值多项式','15 次插值多项式','Location','best')
```

原来函数与插值函数的比较见图 1。

说明：在上面的计算中，我们借用了多项式拟合函数 polyfit，给定$n+1$个观测点拟合n多项式（待定$n+1$个系数），拟合的多项式刚好过已知数据点，拟合和插值在这里是等价的。

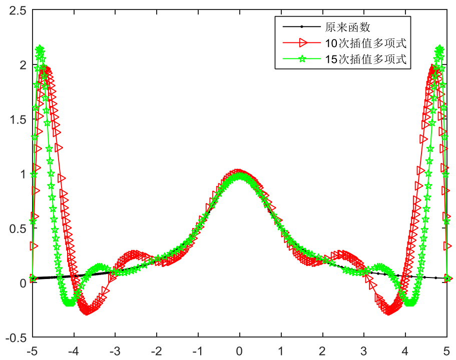


图 1　原来函数与插值函数的比较

高次插值多项式的这些缺陷，促使人们转而寻求简单的低次多项式插值。

6.1.1　分段线性插值

简单地说，将每两个相邻的节点用直线连起来，如此形成的一条折线就是分段线性插值函数，记作 $I_n(x)$，它满足 $I_n(x_i)=y_i$，且 $I_n(x)$ 在每个小区间$[x_i,x_{i+1}]$上是线性函数$(i=0,1,\cdots,n-1)$。

$I_n(x)$ 可以表示为

$$I_n(x)=\sum_{i=0}^{n} y_i l_i(x)$$

$$l_i(x)=\begin{cases}\dfrac{x-x_{i-1}}{x_i-x_{i-1}}, & x\in[x_{i-1},x_i]\ (i=0\text{时舍去）}\\ \dfrac{x-x_{i+1}}{x_i-x_{i+1}}, & x\in[x_i,x_{i+1}]\ \text{（}i=n\text{时舍去）}\\ 0, & \text{其它}\end{cases}$$

$I_n(x)$ 有良好的收敛性，即对于 $x\in[a,b]$有，

$$\lim_{n\to\infty} I_n(x)=f(x),$$

用 $I_n(x)$ 计算 x 点的插值时，只用到 x 左右的两个节点，计算量与节点个数 n 无关。但 n 越大，分段越多，插值误差越小。实际上用函数表作插值计算时，分段线性插值就足够了，如

数学、物理中用的特殊函数表，数理统计中用的概率分布表等。

6.1.3　样条插值

许多工程技术中提出的计算问题对插值函数的光滑性有较高要求，如飞机的机翼外形，内燃机的进、排气门的凸轮曲线，都要求曲线具有较高的光滑程度，不仅要连续，而且要有连续的曲率，这就导致了样条插值的产生。

1.样条函数的概念

所谓样条（Spline）本来是工程设计中使用的一种绘图工具，它是富有弹性的细木条或细金属条。绘图员利用它把一些已知点连接成一条光滑曲线（称为样条曲线），并使连接点处有连续的曲率。三次样条插值就是由此抽象出来的。

数学上将具有一定光滑性的分段多项式称为样条函数。具体地说，给定区间$[a,b]$的一个分划

$$\Delta:\quad a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b,$$

如果函数$S(x)$满足：

（i）在每个小区间$[x_i, x_{i-1}](i = 0,1,\cdots,n-1)$上$S(x)$是$m$次多项式；

（ii）$S(x)$在$[a,b]$上具有$m-1$阶连续导数。

则称$S(x)$为关于分划Δ的m次样条函数，其图形为m次样条曲线。

显然，折线是一次样条曲线。

2.三次样条插值

利用样条函数进行插值，即取插值函数为样条函数，称为样条插值。例如分段线性插值是一次样条插值。我们只介绍三次样条插值，即对于未知函数$y = f(x)$在区间$[a,b]$上的$n+1$个节点

$$a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b$$

上的值$y_i(i = 0,1,\cdots,n)$，求插值函数$S(x)$，使得

（i）$S(x_i) = y_i \quad (i = 0,1,\cdots,n)$；　　（5）

（ii）在每个小区间$[x_j, x_{j+1}](j = 0,1,\cdots,n-1)$上$S(x)$是三次多项式，记为$S_j(x)$；

（iii）$S(x)$在$[a,b]$上二阶连续可微。

函数$S(x)$称为$f(x)$的三次样条插值函数。

由条件（ii），不妨将$S(x)$记为

$$S(x) = \{S_j(x), x \in [x_j, x_{j+1}],\ j = 0,1,\cdots n-1\}$$

$$S_j(x) = a_j x^3 + b_j x^2 + c_j x + d_j$$

其中a_j, b_j, c_j, d_j为待定系数，共$4n$个。由条件（iii）

$$\begin{cases} S_j(x_{j+1}) = S_{j+1}(x_{j+1}) \\ S'_j(x_{j+1}) = S'_{j+1}(x_{j+1}) \qquad j = 0,1,\cdots,n-1 \\ S''_j(x_{j+1}) = S''_{j+1}(x_{j+1}) \end{cases} \tag{6}$$

容易看出，（5）、（6）式共含有$4n-2$个方程，为确定$S(x)$的$4n$个待定参数，尚需再给出2个条件。

常用的三次样条函数的边界条件有3种类型：

（i）$S'(a) = y'_0, S'(b) = y'_n$。由这种边界条件建立的样条插值函数称为$f(x)$的完备三次样条插值函数。

特别地，$y'_0 = y'_n = 0$时，样条曲线在端点处呈水平状态。

如果$f'(x)$不知道，我们可以要求$S'(x)$与$f'(x)$在端点处近似相等。这时以x_0, x_1, x_2, x_3为节点作一个三次 Newton 插值多项式$N_a(x)$，以$x_n, x_{n-1}, x_{n-2}, x_{n-3}$作一个三次 Newton 插值多项式$N_b(x)$，要求

$$S'(a)=N'_a(a), S'(b)=N'_b(b).$$

由这种边界条件建立的三次样条称为 $f(x)$ 的 Lagrange 三次样条插值函数。

（ii）$S''(a)=y''_0, S''(b)=y''_n$。特别地 $y''_0=y''_n=0$ 时，称为自然边界条件。

（iii）$S'(a+0)=S'(b-0), S''(a+0)=S''(b-0)$，此条件称为周期条件。

对于二维插值和三维插值的思想和数学理论，我们就不介绍了，下面主要介绍一下 Matlab 插值工具箱的使用。

6.2 Matlab 一维插值函数

所有的插值方法要求 x0 是单调的。

6.2.1 interp1 函数

Matlab 中一维插值函数 interp1 的调用格式有两种。

第一种调用格式为：

vq=interp1(x0,y0,xq,method,extrapolation)

其中 x0 为已知的插值节点，y0 是对应于 x0 的函数值，xq 是欲求函数值的节点坐标，返回值 vq 是求得的节点 xq 处的函数值，method 指定插值的方法，默认为线性插值。其值常用的有：

'nearest' 最近邻插值；

'linear' 线性插值；

'spline' 三次样条插值，函数是二次光滑的；

'cubic' 立方插值，函数是一次光滑的。

extrapolation 是外插策略。

在未来版本中，下面调用格式

interp1(...,'cubic')

pp=interp1(...,'pp')

可能会被移去。

6.2.2 griddedInterpolant 函数

一维插值的调用格式为：

F=griddedInterpolant(x,v, method,extrapolation)

计算对应的函数值的使用格式为 Vq=F(Xq)。

n 维插值的调用格式为：

F=griddedInterpolant(x1,x2,…xn,v, method,extrapolation)

计算对应的函数值的使用格式为 Vq=F(xq1,xq1,…,xqn)。

6.2.3 三次样条插值

三次样条插值还可以函数 csape，csape 的返回值是 pp 形式，要求插值点的函数值，必须调用函数 ppval(或 fnval)。

pp=csape(x0,y0)使用默认的边界条件，即 Lagrange 边界条件。

pp=csape(x0,y0,conds,valconds)中的 conds 指定插值的边界条件，其值可为

'complete' 边界为一阶导数，一阶导数的值在 valconds 参数中给出，若忽略 valconds 参数，则按缺省情况处理。

'not-a-knot' 非扭结条件。

'periodic' 周期条件。

'second' 边界为二阶导数，二阶导数的值在 valconds 参数中给出，若忽略 valconds 参数，二阶导数的缺省值为[0, 0]。

'variational' 设置边界的二阶导数值为[0,0]。

对于一些特殊的边界条件，可以通过 conds 的一个 1×2 矩阵来表示，conds 元素的取值为 0，1，2。

conds(i)=j 的含义是给定端点 i 的 j 阶导数，即 conds 的第一个元素表示左边界的条件，第二个元素表示右边界的条件，conds=[2,1]表示左边界是二阶导数，右边界是一阶导数，对应的值由 valconds 给出。

详细情况请使用帮助 doc csape。

利用 pp 结构的返回值，还可以计算返回值函数的导数和积分，命令分别为 fnder，fnint，这两个函数的返回值还是 pp 结构。

pp 结构相关函数的功能总结于表 1。

表 1 pp 结构的相关函数

调用格式	函数功能
pp1=csape(x0,y0)	计算插值函数
pp2=fnder(pp1)	计算 pp1 对应函数的导数，返回值 pp2 也是 pp 结构
pp3=fnint(pp1)	计算 pp1 对应函数的积分，返回值 pp3 也是 pp 结构
y=ppval(pp1,x)	计算 pp1 对应的分段多项式在 x 点的取值
y=fnval(pp1,x)	计算 pp1 对应的函数在 x 点的取值

6.2.4 举例

例 6.2 已知速度曲线 $v(t)$ 上的 4 个数据点如表 2 所示。用三次样条函数求速度函数，并求位移 $S=\int_0^3 v(t)dt$。

表 2 速度的 4 个观测值

t	0	1	2	3
$v(t)$	1	3	4	6

```
clc, clear
t0=0:3; v0=[1 3 4 6];
pp=csape(t0,v0)   %计算三次样条插值函数的另一种调用方式
xishu=pp.coefs   %显示分段三次多项式的系数
s1=quadl(@(t)fnval(pp,t),0,3)   %求数值积分
s2=integral(@(t)ppval(pp,t),0,3)   %求数值积分，用于对比
pp2=fnint(pp)   %求插值函数的积分，即原函数，返回值也是 pp 结构
s3=fnval(pp2,3)-fnval(pp2,0) %计算原函数取值的差，用于对比
f=griddedInterpolant(t0,v0,'spline') %计算三次样条插值函数
s4=quadl(@(t)f(t),0,3)
pp3=interp1(t0,v0,'spline','pp') %在未来的版本，该调用方法将被移除
xishu2=pp3.coefs
```

求出的速度三次样条函数为

$$v(t)=\begin{cases}0.3333t^3-1.5t^2+3.1667t+1, & t\in[0,1],\\ 0.3333(t-1)^3-0.5(t-1)^2+1.1667(t-1)+3, & t\in[1,2],\\ 0.3333(t-2)^3+0.5(t-2)^2+1.1667(t-2)+4, & t\in[2,3].\end{cases}$$

用四种方法计算的位移结果都是一样的，位移 $S=\int_0^3 v(t)dt=10.5$。

例 6.3 已知欧洲一个国家的地图，为了算出它的国土面积和边界长度，首先对地图作如下测量：以由西向东方向为 x 轴，由南向北方向为 y 轴，选择方便的原点，并将从最西边界点到最东边界点在 x 轴上的区间适当地分为若干段，在每个分点的 y 方向测出南边界点和北边界点的 y 坐标 y_1 和 y_2，这样就得到了表 3 的数据（单位：mm）。

表 3 某国国土地图边界测量值（单位：mm）

x	7.0	10.5	13.0	17.5	34.0	40.5	44.5	48.0	56.0
y_1	44	45	47	50	50	38	30	30	34
y_2	44	59	70	72	93	100	110	110	110

x	61.0	68.5	76.5	80.5	91.0	96.0	101.0	104.0	106.5
y_1	36	34	41	45	46	43	37	33	28
y_2	117	118	116	118	118	121	124	121	121
x	111.5	118.0	123.5	136.5	142.0	146.0	150.0	157.0	158.0
y_1	32	65	55	54	52	50	66	66	68
y_2	121	122	116	83	81	82	86	85	68

根据地图的比例我们知道18mm相当于40km，试由测量数据计算该国国土的近似面积和边界的近似长度，并与国土面积的精确值41288km^{2}比较。

解　该地区的示意图见图 2。

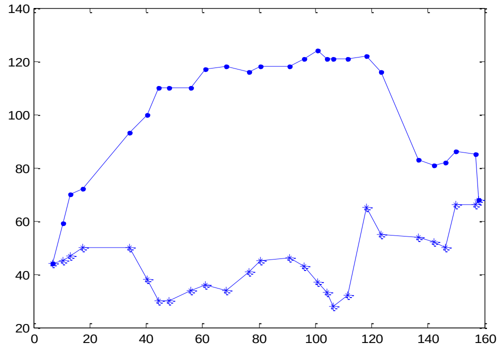


图 2　区域边界示意图

若区域的下边界和上边界曲线的方程分别为 $y_1 = y_1(x)$， $y_2 = y_2(x)$， $a \le x \le b$，则该地区的边界线长为

$$\int_a^b \sqrt{1+y_1'(x)^2}dx + \int_a^b \sqrt{1+y_2'(x)^2}dx,$$

计算时用数值积分即可。

计算该区域的面积，可以把该区域看成是上、下两个边界为曲边的曲边四边形，则区域的面积

$$S = \int_a^b (y_2(x) - y_1(x))dx.$$

计算相应的数值积分就可求出面积。

为了提高计算的精度，可以把上、下边界曲线分别进行三次样条插值，利用三次样条函数计算相应的弧长和曲边四边形的面积。

利用三次样条插值计算时，得到边界长度的近似值为 1161km，区域面积的近似值为 42476km^2，与其准确值 41288km^2 只相差 2.8768%。

计算的Matlab程序如下

```
clc, clear
a=[7.0  10.5   13.0   17.5   34.0   40.5   44.5   48.0   56.0
44  45  47  50  50  38  30  30  34
44  59  70  72  93  100 110 110 110
61.0    68.5   76.5   80.5   91.0   96.0   101.0  104.0  106.5
36  34  41  45  46  43  37  33  28
117 118 116 118 118 121 124 121 121
111.5   118.0  123.5  136.5  142.0  146.0  150.0  157.0  158.0
32  65  55  54  52  50  66  66  68
121 122 116 83  81  82  86  85  68];
x0=a(1:3:end,:); x0=x0'; x0=x0(:); %提取点的横坐标
```

```
y1=a(2:3:end,:); y1=y1'; y1=y1(:); %提出下边界的纵坐标
y2=a(3:3:end,:); y2=y2'; y2=y2(:); %提出上边界的纵坐标
pp1=csape(x0,y1); pp2=csape(x0,y2); %计算三次样条插值函数
dp1=fnder(pp1); dp2=fnder(pp2); %求三次样条插值函数的导数
L1=quad(@(x)sqrt(1+fnval(dp1,x).^2)+sqrt(1+fnval(dp2,x).^2),x0(1),x0(end))
L2=L1/18*40  %换算成边界的实际长度
S1=quad(@(x)fnval(pp2,x)-fnval(pp1,x),x0(1),x0(end))  %计算地图上面积
S2=S1/18^2*1600
delta=(S2-41288)/41288*100
```

注：为了熟悉一些Matlab离散点数值积分的命令，下面我们给出根据表3的数据直接进行数值积分计算边界长度和国土面积的程序

```
clc, clear
a=[7.0  10.5    13.0    17.5    34.0    40.5    44.5    48.0    56.0
44  45  47  50  50  38  30  30  34
44  59  70  72  93  100 110 110 110
61.0    68.5    76.5    80.5    91.0    96.0    101.0   104.0   106.5
36  34  41  45  46  43  37  33  28
117 118 116 118 118 121 124 121 121
111.5   118.0   123.5   136.5   142.0   146.0   150.0   157.0   158.0
32  65  55  54  52  50  66  66  68
121 122 116 83  81  82  86  85  68];
x0=a(1:3:end,:); x0=x0'; x0=x0(:); %提取点的横坐标
y1=a(2:3:end,:); y1=y1'; y1=y1(:); %提出下边界的纵坐标
y2=a(3:3:end,:); y2=y2'; y2=y2(:);
plot(x0,y1,'*-') %画下边界曲线
hold on
plot(x0,y2,'.-') %画上边界曲线
L1=trapz(x0,sqrt(1+gradient(y1,x0).^2)) %计算下边界的长度
L2=trapz(x0,sqrt(1+gradient(y2,x0).^2)) %计算上边界的长度
L=L1+L2; %计算地图上边界的长度
LL=L/18*40 %计算实际的边界长度
S=trapz(x0,y2-y1); %计算地图上的近似面积
SS=S/18^2*1600
```

6.3 Matlab 二维插值函数

6.3.1 网格数据的插值

已知$m\times n$个节点：(x_i,y_j,z_{ij})（$i=1,2,\cdots,m; j=1,2,\cdots,n$），且$x_1<\cdots<x_m$；$y_1<\cdots<y_n$。求点$(x,y)$处的插值$z$。

Matlab 中有一些计算二维插值的命令。如

z=interp2(x0,y0,z0,x,y,'method')

其中 x0，y0 分别为m维和n维向量，表示节点，z0 为$n\times m$维矩阵，表示节点值，x，y 为一维数组，表示插值点，x 与 y 应是方向不同的向量，即一个是行向量，另一个是列向量，z 为矩阵，它的行数为 y 的维数，列数为 x 的维数，表示得到的插值，'method'的用法同上面的一维插值。

如果是三次样条插值，可以使用命令

pp=csape({x0,y0},z0,conds,valconds)，z=fnval(pp,{x,y})

其中 x0，y0 分别为m维和n维向量，z0 为$m\times n$维矩阵，z为矩阵，它的行数为 x 的维数，列数为 y 的维数，表示得到的插值，具体使用方法同一维插值。

例 6.4（续例 1.22）　已知平面区域$0\le x\le 1400$，$0\le y\le 1200$的高程数据见表 4（单

位：m)。

表 4　高程数据表

1200	1350	1370	1390	1400	1410	960	940	880	800	690	570	430	290	210	150
1100	1370	1390	1410	1430	1440	1140	1110	1050	950	820	690	540	380	300	210
1000	1380	1410	1430	1450	1470	1320	1280	1200	1080	940	780	620	460	370	350
900	1420	1430	1450	1480	1500	1550	1510	1430	1300	1200	980	850	750	550	500
800	1430	1450	1460	1500	1550	1600	1550	1600	1600	1600	1550	1500	1500	1550	1550
700	950	1190	1370	1500	1200	1100	1550	1600	1550	1380	1070	900	1050	1150	1200
600	910	1090	1270	1500	1200	1100	1350	1450	1200	1150	1010	880	1000	1050	1100
500	880	1060	1230	1390	1500	1500	1400	900	1100	1060	950	870	900	936	950
400	830	980	1180	1320	1450	1420	400	1300	700	900	850	810	380	780	750
300	740	880	1080	1130	1250	1280	1230	1040	900	500	700	780	750	650	550
200	650	760	880	970	1020	1050	1020	830	800	700	300	500	550	480	350
100	510	620	730	800	850	870	850	780	720	650	500	200	300	350	320
0	370	470	550	600	670	690	670	620	580	450	400	300	100	150	250
y/x	0	100	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400

（1）用双线性插值，求 x， y 方向的间隔都为 10 的节点处的高程值，并求该区域的最大高程和最小高程，并画出三维表面图。

（2）用双三次样条插值，求该区域的最大高程和最小高程，并画出三维表面图。

（3）用双线性插值，求该区域表面面积的近似值。

（4）用双三次样条插值，求该区域表面面积的近似值。

双线性插值的 Matlab 程序如下：

```
clc, clear
z0=load('ditu.txt');
x0=0:100:1400; y0=1200:-100:0;
x=0:10:1400; y=[1200:-10:0]';
z=interp2(x0,y0,z0,x,y);
max_z=max(max(z)), min_z=min(min(z))
surf(x,y,z)
m=length(x); n=length(y); s=0;
for i=1:m-1
    for j=1:n-1
        p1=[x(i),y(j),z(j,i)];
        p2=[x(i+1),y(j),z(j,i+1)];
        p3=[x(i+1),y(j+1),z(j+1,i+1)];
        p4=[x(i),y(j+1),z(j+1,i)];
        p12=norm(p1-p2); p23=norm(p3-p2); p13=norm(p3-p1);
        p14=norm(p4-p1); p34=norm(p4-p3);
        z1=(p12+p23+p13)/2;s1=sqrt(z1*(z1-p12)*(z1-p23)*(z1-p13));
        z2=(p13+p14+p34)/2; s2=sqrt(z2*(z2-p13)*(z2-p14)*(z2-p34));
        s=s+s1+s2;
    end
end
s %显示面积的计算值
```

双三次样条插值的 Matlab 程序如下：

```
clc, clear
z0=load('ditu.txt');
x0=0:100:1400; y0=1200:-100:0;
x=0:10:1400; y=[1200:-10:0]';
z1=interp2(x0,y0,z0,x,y,'spline'); %双三次样条插值
max_z1=max(max(z1)), min_z1=min(min(z1))
```

```
pp=csape({x0,y0},z0'); %双三次样条插值,注意这里 z0 要转置
z2=fnval(pp,{x,y}); z2=z2'; %为了和 z1 的矩阵维数一致
max_z2=max(max(z2)), min_z2=min(min(z2))
subplot(121),surf(x,y,z1), title('interp2 双三次样条插值')
subplot(122), surf(x,y,z2),title('csape 双三次样条插值')
m=length(x); n=length(y); z=z1; s=0; %计算 interp2 插值的面积,计算 csape 插值的面积，把z1 换成 z2 即可
for i=1:m-1
    for j=1:n-1
        p1=[x(i),y(j),z(j,i)];
        p2=[x(i+1),y(j),z(j,i+1)];
        p3=[x(i+1),y(j+1),z(j+1,i+1)];
        p4=[x(i),y(j+1),z(j+1,i)];
        p12=norm(p1-p2); p23=norm(p3-p2); p13=norm(p3-p1);
        p14=norm(p4-p1); p34=norm(p4-p3);
        z1=(p12+p23+p13)/2;s1=sqrt(z1*(z1-p12)*(z1-p23)*(z1-p13));
        z2=(p13+p14+p34)/2; s2=sqrt(z2*(z2-p13)*(z2-p14)*(z2-p34));
        s=s+s1+s2;
    end
end
s %显示面积的计算值
```

用双线性插值求得的最大高程为 1600，最小高程为 100。

用双三次样条插值函数 interp2 计算得到的最大高程为 1649.6，最小高程为 87.5817。

用双三次样条插值函数 csape 计算得到的最大高程为 1649.6，最小高程为 89.1392。

同样都是双三次样条插值，但最小高程的计算有差异，可能使用的边界条件有差异，csape 命令使用的是拉格朗日边界条件。

两种双三次样条插值结果的三维表面图见图 3。

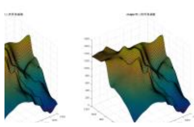


图 3　两种双三次样条插值函数的图形比较

6.3.2　散点插值

已知 n 个节点 $(x_i, y_i, z_i)(i = 1,2,\cdots,n)$，求点 (x, y) 处的插值 z 。

对上述问题，Matlab 中提供了插值函数 griddata 和 scatteredInterpolant。

1.函数 griddata

函数 griddata 的调用格式为

ZI = griddata(x,y,z,XI,YI)

其中 x、y、z 均为 n 维向量，指明所给数据点的横坐标、纵坐标和竖坐标。向量 XI、YI 是给定的网格点的横坐标和纵坐标，返回值 ZI 为网格（XI，YI）处的函数值。XI 与 YI 应是方向不同的向量，即一个是行向量，另一个是列向量。

2.函数 scatteredInterpolant

函数 scatteredInterpolant 的调用格式为

Fz=scatteredInterpolant(x0,y0,z0,Method,ExtrapolationMethod);

其中返回值 Fz 是结构数组，相当于给出了插值函数的表达式；x0,y0,z0 分别为已知 n 个点的 x, y, z 坐标；Method 是插值方法；ExtrapolationMethod 是区域外部节点的外插方法，ExtrapolationMethod 的值可以取为'linear'和'nearest'，默认值为'linear'。

要计算插值点（x,y）处的值，调用函数 Fz 即可，

z=Fz(x,y)

就求出所要求的插值。

3.例题

例 6. 5 在某海域测得一些点（x, y）处的水深 z 由表 5 给出，在 x, y 的最大范围所确定的矩形区域内画出海底曲面的图形，并求最深点的坐标及深度。要求分别用如下两种函数插值：

（1）用函数 griddata 插值；

（2）用函数 scatteredInterpolant 插值。

表 5 海底高程数据

x	129	140	103.5	88	185.5	195	105	157.5	107.5	77	81	162	162	117.5
y	7.5	141.5	23	147	22.5	137.5	85.5	−6.5	−81	3	56.5	−66.5	84	−33.5
z	4	8	6	8	6	8	8	9	9	8	8	9	4	9

计算及画图的 Matlab 程序如下：

```
clc, clear
x0=[129,140,103.5,88,185.5,195,105,157.5,107.5,77,81,162,162,117.5];
y0=[7.5,141.5,23,147,22.5,137.5,85.5,-6.5,-81,3,56.5,-66.5,84,-33.5];
z0=-[4,8,6,8,6,8,8,9,9,8,8,9,4,9];
xm0=minmax(x0)   %求 x0 的最小值和最大值
ym0=minmax(y0)   %求 y0 的最小值和最大值
xi=xm0(1):xm0(2); %插值节点的 x 坐标取值
yi=ym0(1):ym0(2); %插值节点的 y 坐标取值
zi11=griddata(x0,y0,z0,xi,yi','natural'); %自然插值
zi12=griddata(x0,y0,z0,xi,yi','nearest'); %最近点插值
zi1=zi11;   % natural 插值与最近点插值的混合插值的初始值
zi1(isnan(zi11))=zi12(isnan(zi11));   %把自然插值中的不确定值换成最近点插值的结果
min_z1=min(min(zi1))   %求矩阵中的最小值
[i1,j1]=find(zi1==min_z1)
xm1=xi(j1), ym1=yi(i1) %求最深处的 x 坐标和 y 坐标
subplot(1,2,1), mesh(xi,yi,zi1), title('griddata 插值')
Fz=scatteredInterpolant(x0',y0',z0','natural','nearest')
[xi,yi]=meshgrid(xi,yi); %化成网格数据
zi2=Fz(xi,yi); %计算插值点的函数值
min_z2=min(min(zi2))   %求最小值
[i2,j2]=find(zi2==min_z2)   %求最小值的 x,y 坐标的地址
xm2=xi(j2), ym2=yi(i2)   %求最深处的 x 坐标和 y 坐标
subplot(122), mesh(xi,yi,zi2), title('scatteredInterpolant 插值')
```

调用两个不同函数插值求得的最大深度都是 9 米。用 griddata 函数计算在点 $(123,-43)$ 和点 $(146,-42)$ 处达到最大深度；用 scatteredInterpolant 函数计算在点 $(77,-43)$ 和点 $(77,-42)$ 处达到最大深度。

两种插值结果所得到的三维网格图见图 4。

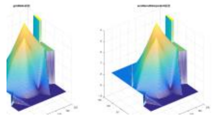


图 4　两种不同插值函数计算结果的图形比较

6.4　Matlab 三维插值函数

Matlab 的三维网格节点的插值有 interp3。三维散点插值也使用函数 griddata 和 scatteredInterpolant。下面我们举一个三维散点插值的例子，该题目是 2015 年西北工业大学校赛的 A 题。

例 6.6　电阻率数据插值加密及成像问题。

在实际问题中，由于技术与测量成本等原因，我们不能对一个物体进行密集测量，而只能等间隔的选取部分点进行测量。而实际中我们却需要更多位置的数据，这时候就需要采用数学的方法，根据已知位置的数据计算所需位置的数据。

data5.txt 为某空间三维体电阻率数据文件，每一行为一个数据点，第一列为 x 坐标，第二列为 y 坐标，第三列为 z 坐标，第四列为对应坐标点的电阻率值。

这个数据文件中给出的坐标网格大小是 10m*10m*10m，要求通过插值加密后获得网格大小为 1m*1m*1m 的三维电阻率数据，必须保证插值后的电阻率数据极值（包含最大值和最小值）与插值前的电阻率数据极值相等，并且极值出现的坐标位置相同。另外，必须保证插值后的电阻率数据三维成像结果与插值前的电阻率数据三维成像结果形态基本一致，只是前者像素更高。

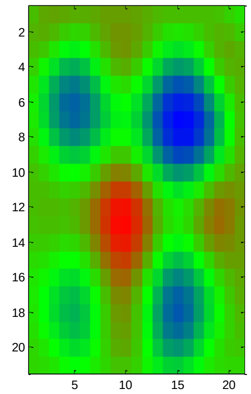
请你解决如下问题：

(1) 请给出符合条件的两种计算方法，并给出相应数学公式，证明你的方法插值后的电阻率数据极值与位置不变。采用你给出的两种方法，分别计算出空间某点(45.8,-32.7,68.2)处的电阻率数值。

(2) 利用（1）中你给出的两种方法，分别计算出网格大小为1m*1m*1m的三维电阻率数据，要求给出你的计算流程，并对两种方法的计算量或计算复杂性进行评估。同时给出原网格数据及你采用两种方法加密网格后数据的平均值与标准差。

(3) 对加密网格后的直观效果，可采用颜色图展示。颜色图就是用不同的颜色来表示每个像素。如Z=40对应的原图像为图5，采用某种方法加密后得到的颜色图像如图6，从中可以看出像素提高了。对每一幅需要对比显示效果的图，请将最小值置为纯蓝色（RGB为(0,0,255)），中间值置为纯绿色（RGB为(0,255,0)），最大值置为纯红色（RGB为(255,0,0)），中间数值采用过渡的颜色，可自行设计。采用这种方法，请对切片Z=0,50分别给出原数据，两种方法加密网格后数据的颜色对比图。另外分别对切片X=82，Y=47，Z=88 ，请给出两种加密方法得到数据的颜色图。

（4）对不同的插值加密方法的效果，你能否给出定量的指标。请对你给出的指标，分别计算出你采用的两种不同加密方法得到数据的指标值，并给出你的评价。



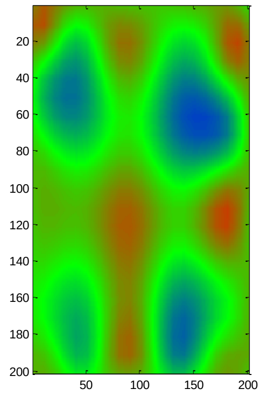


图5　原数据成像　　　图6　插值加密后数据成像

下面我们给出（1）和（2）的计算，并画出图 5 的效果图。

1.前两问的 Matlab 程序

```
clc, clear
a=load('data3D_.txt');
x0=a(:,1); y0=a(:,2); z0=a(:,3); r0=a(:,4);
r01=griddata(x0,y0,z0,r0,45.8,-32.7,68.2)   %线性插值的定点值计算
rxyz=scatteredInterpolant(x0,y0,z0,r0);   %计算线性插值的插值函数
r02=rxyz(45.8,-32.7,68.2)   %另一种计算定点的插值
x=-100:100; y=-100:100; z=0:100;
[x,y,z]=meshgrid(x,y,z); %数据网络化
r1=griddata(x0,y0,z0,r0,x,y,z); %线性插值
r1=r1(:); %把插值结果展开成列向量
mu1=mean(r1),   s1=std(r1)   %计算均值和标准差
r2=griddata(x0,y0,z0,r0,x,y,z,'natural'); %“Triangulation-based natural neighbor interpolation”插值
r2=r2(:); mu2=mean(r2),   s2=std(r2)   %计算均值和标准差
```

2.第三问的思想及画图程序

记 $z=40$ 处电阻率的最小值，中间值和最大值分别记为 r_1,r_2,r_3。

对每一幅需要对比显示效果的图，题目中要求将最小值置为纯蓝色（RGB 为(0,0,255)），中间值置为纯绿色（RGB 为(0,255,0)），最大值置为纯红色（RGB 为(255,0,0)），中间数值采用过渡的颜色，可自行设计。这里实际上也需要对三种颜色的像素值进行插值。

我们设计的电阻率与 R，G，B 像素值之间的对应关系为分段线性关系，取电阻 r 与红、绿、蓝像素 $R(r)$、$G(r)$、$B(r)$ 的函数关系为：

$$R(r)=\begin{cases}0, & r_1<r<r_2,\\ \dfrac{255}{r_3-r_2}(r-r_2), & r_2<r<r_3.\end{cases}$$

$$G(r)=\begin{cases}\dfrac{255}{r_2-r_1}(r-r_1), & r_1<r<r_2,\\ \dfrac{255}{r_3-r_2}(r_3-r), & r_2<r<r_3.\end{cases}$$

$$B(r)=\begin{cases}\dfrac{255}{r_2-r_1}(r_2-r), & r_1<r<r_2,\\ 0, & r_2<r<r_3.\end{cases}$$

计算的 Matlab 程序如下：

```
clc, clear
a=load('data3D_.txt');
x0=a(:,1); y0=a(:,2); z0=a(:,3); r0=a(:,4);
xb=-100:10:100; yb=-100:10:100; %初始的网格划分
n=length(xb)
ind=find(z0==40);
x40=x0(ind), y40=y0(ind)   %提出 z=40 对应的 x,y 坐标
r40=r0(ind)   %提出 z=40 的电阻率
r1=min(r40), r2=median(r40), r3=max(r40) %z=40 时电阻率的最小，中间，最大值
b(n,n,1)=0; b(n,n,2)=0; b(n,n,3)=0;   %电阻率对应的 RGB 矩阵的初始化
Rr=@(r)255/(r3-r2)*(r-r2).*(r>r2 & r<r3); %定义红色像素值的匿名函数
Gr=@(r)255/(r2-r1)*(r-r1).*(r>r1 & r<r2)+255/(r3-r2)*(r3-r).*(r>r2 & r<r3); %定义绿色像素值的匿名函数
Br=@(r)255/(r2-r1)*(r2-r).*(r>r1 & r<r2);
Rr0=Rr(r40)   %红色像素的取值
Gr0=Gr(r40)   %绿色像素的取值
Br0=Br(r40)   %蓝色像素的取值
for k=1:length(ind)
    i=find(xb==x0(ind(k))); j=find(yb==y0(ind(k)));
    b(i,j,1)=Rr0(k); b(i,j,2)=Gr0(k); b(i,j,3)=Br0(k);
end
b=uint8(b);
image(b)   %image 显示图片带坐标刻度，imshow 显示图片不带坐标轴
imwrite(b,'xibei.jpg') %把 3 维矩阵，即彩色图片写到 jpg 文件。
```

画出的 $z=40$ 时的电阻率对应的彩色图片的效果见图 7。

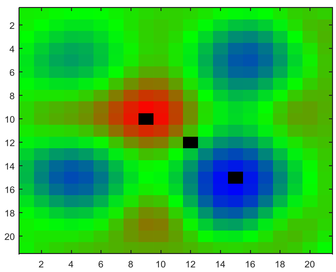


图 7　z=40 时的电阻率显示效果图

习题 6

6.1　在区间上 $[0,10]$ 上等间距取 1000 个点 $x_i(i=1,2,\cdots,1000)$，计算在这些点 x_i 处函数 $g(x)=\dfrac{(3x^2+4x+6)\sin x}{x^2+8x+6}$ 的函数值 y_i，利用观测点 (x_i,y_i)（$i=1,2,\cdots,1000$），求三次样条插值函数 $\hat{g}(x)$，画出插值函数 $\hat{g}(x)$ 的图形，并求积分 $\int_0^{10} g(x)dx$ 和 $\int_0^{10} \hat{g}(x)dx$。

6.2　附件 1：区域高程数据.xlsx 给出了某区域 43.65×58.2（km）的高程数据，求该区域地表面积的近似值。

6.3　已知当温度为 $T=[700,720,740,760,780]$ 时，过热蒸汽体积的变化为 $V=[0.0977, 0.1218, 0.1406, 0.01551, 0.1664]$，分别采用线性插值和三次样条插值求解 $T=750$，7 时的体积变化，并在一个图形界面中画出线性插值函数和三次样条插值函数。