第五章 最小二乘法与曲线拟合

科学实验，统计分析，获得大量数据

x_i	x_0	x_1	x_2	$\cdots$	x_n
y_i	y_0	y_1	y_2	$\cdots$	y_n

(n很大)

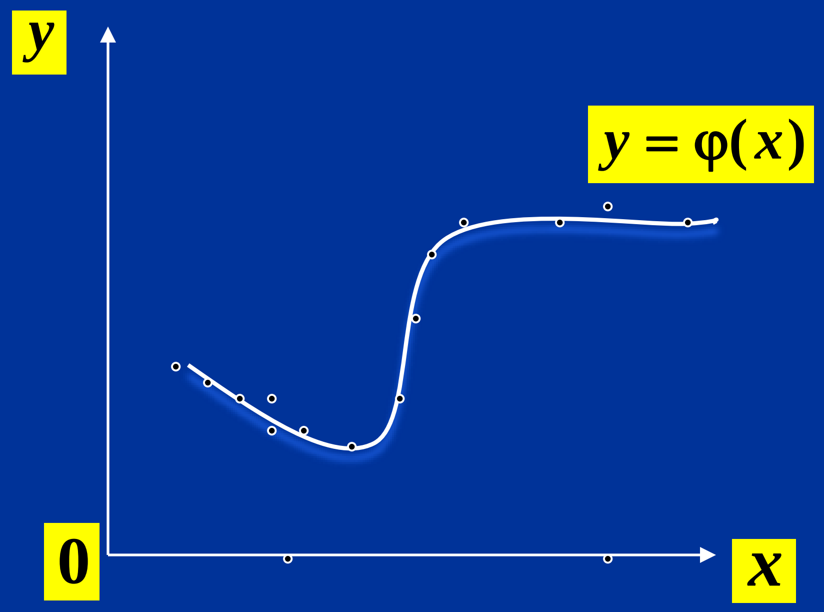
确定y与x之间的近似表达式

方法一——→插值。几何上，插值曲线经过所有点

方法二——→曲线拟合。求一连续曲线$y=\varphi(x)$，使得

误差$Q=\sum_{i=0}^{n}[\varphi(x_i)-y_i]^2$达到最小或$\tilde{Q}=\max\limits_{0\le i\le n}\left|\varphi(x_i)-y_i\right|$达最小。

本章讲述第一种方法，称为最小二乘原理，求得的函数称为拟和函数或经验公式。



当数据量特别大时一般不用插值法。这是因为数据量很大时所求插值曲线中的未知参数就很多，而且数据量很大时，多项式插值会出现高次插值（效果不理想）或分段低次插值（精度不高）；另外，测量数据本身往往就有误差，所以，使插值曲线刻意经过这些点也不必要。

而曲线拟合是，首先根据物理规律或描点画草图确定一条用来拟合的函数曲线形式，也可选择低次多项式形式（所含参数比较少），然后按最小二乘法求出该曲线，它未必经过所有已知点，但它能反映出数据的基本趋势，且误差最小，效果比较好。

5.1　最小二乘原理

定义1：设$x_1, x_2, \cdots, x_n$为互不相同的点,$\varphi_0(x), \varphi_1(x), \cdots, \varphi_m(x)$是$m+1$个已知函数.如果存在不全为零的常数$c_0, c_1, \cdots, c_m$使得

$$c_0\varphi_0(x_j)+c_1\varphi_1(x_j)+\cdots+c_m\varphi_m(x_j)=0,\ j=1,2,\cdots,n$$

则称$\varphi_0(x), \varphi_1(x), \cdots, \varphi_m(x)$(关于点$x_1, x_2, \cdots, x_n$)是线性相关的,否则称为线性无关的.

定义2：给定数据$(x_j, y_j), j=1,2,\cdots,n$.假设拟合函数的形式为$p(x)=a_0\varphi_0(x)+a_1\varphi_1(x)+\cdots+a_m\varphi_m(x)$ (5.1)

这里$\{\varphi_k(x)\}_{k=0}^{m}$为已知的线性无关函数.求系数

$a_0*, a_1*, \cdots, a_m*$使得

$$\varphi(a_0, a_1, \cdots, a_m)$$

$$=\sum_{j=1}^{n}[p(x_j)-y_j]^2=\sum_{j=0}^{n}[\sum_{k=0}^{m}a_k\varphi_k(x_j)-y_j]^2 \qquad (5.2)$$

取最小值.称$p(x)=\sum_{k=0}^{m}a_k*\varphi_k(x_j)$为拟合函数或经验公式. (5.3)

如果$\varphi_k(x)=x^k (k=0,1,\cdots,m)$,则称式(5.3)为$m$次最小二乘拟合多项式.

用多元函数求极值的方法求(5.2)式的最小点和最小值，即将求偏导数，得到方程组如下：

$$\frac{\partial\varphi}{\partial a_k}=2\sum_{j=1}^{n}\left(\sum_{i=0}^{m}a_i^*\varphi_i(x_j)-y_j\right)\varphi_k(x_j)=0,$$

$k=0,1,\cdots,m,$即

$$\sum_{i=0}^{m}\left[\sum_{j=1}^{n}\varphi_i(x_j)\varphi_k(x_j)\right]a_i^*=\sum_{j=1}^{n}y_j\varphi_k(x_j).(5.4)$$

求出a_i^*,代入式(5.3)，即可得到所要求的拟合

函数$p^*(x)=\sum_{k=0}^{m}a_k^*\varphi_k(x).$

简化方程组的系数矩阵表示形式,引入内积。

点集函数：$u=(u(x_1),u(x_2),\cdots,u(x_n))^{\mathrm{T}}$,

$$v=(v(x_1),v(x_2),\cdots,v(x_n))^{\mathrm{T}},$$

$$w=(w(x_1),w(x_2),\cdots,w(x_n))^{\mathrm{T}},$$

u与v的内积:$(u,v)=\sum_{j=1}^{n}u(x_j)v(x_j)$.

性质:(1)非负性:$(u,u)\geq 0;(u,u)=0\Leftrightarrow u(x_j)=0$

(2)对称性:$(u,v)=(v,u)$;

(3)分配律:$(\alpha u+\beta v,w)=\alpha(u,w)+\beta(v,w)$,$\alpha$，$\beta\in R$.

利用内积改写后的正规方程组

$$\begin{bmatrix} (\varphi_0,\varphi_0) & (\varphi_0,\varphi_1) & \cdots & (\varphi_0,\varphi_m) \\ (\varphi_1,\varphi_0) & (\varphi_1,\varphi_1) & \cdots & (\varphi_1,\varphi_m) \\ \vdots & \vdots & \vdots & \vdots \\ (\varphi_m,\varphi_0) & (\varphi_m,\varphi_1) & \cdots & (\varphi_m,\varphi_m) \end{bmatrix} \begin{bmatrix} a_0^* \\ a_1^* \\ \vdots \\ a_m^* \end{bmatrix} = \begin{bmatrix} (y,\varphi_0) \\ (y,\varphi_1) \\ \vdots \\ (y,\varphi_m) \end{bmatrix}, \text{其中}$$

$$\varphi_0 = \begin{bmatrix} \varphi_0(x_1) \\ \varphi_0(x_2) \\ \vdots \\ \varphi_0(x_n) \end{bmatrix}, \varphi_1 = \begin{bmatrix} \varphi_1(x_1) \\ \varphi_1(x_2) \\ \vdots \\ \varphi_1(x_n) \end{bmatrix}, \cdots, \varphi_m = \begin{bmatrix} \varphi_m(x_1) \\ \varphi_m(x_2) \\ \vdots \\ \varphi_m(x_n) \end{bmatrix}, y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

例1 设从某一实验中测得2个变量x和y的一组数据如表所示，求一代数多项式曲线，使其最好地拟合这组数据。

i	1	2	3	4	5	6	7	8	9
x_i	1	3	4	5	6	7	8	9	10
y_i	10	5	4	2	1	1	2	3	4

解：画图得知所给数据大致在一条抛物线上。

故设拟合曲线方程为：$f(x)=a+bx+cx^2$,

即取$m=2,\varphi_0(x)=1,\varphi_1(x)=x,\varphi_2(x)=x^2.$

$$\begin{bmatrix}(\varphi_0,\varphi_0) & (\varphi_0,\varphi_1) & (\varphi_0,\varphi_2)\\(\varphi_1,\varphi_0) & (\varphi_1,\varphi_1) & (\varphi_1,\varphi_2)\\(\varphi_2,\varphi_0) & (\varphi_2,\varphi_1) & (\varphi_2,\varphi_2)\end{bmatrix}\begin{bmatrix}a\\b\\c\end{bmatrix}=\begin{bmatrix}(y,\varphi_0)\\(y,\varphi_1)\\(y,\varphi_2)\end{bmatrix}$$

计算:$(\varphi_0,\varphi_0)=\sum_{i=1}^{9}1\times1=9;(\varphi_1,\varphi_1)=\sum_{i=1}^{9}x_i^2=381;$

$$(\varphi_2,\varphi_2)=\sum_{i=1}^{9}x_i^4=25317;(\varphi_1,\varphi_0)=(\varphi_0,\varphi_1)=\sum_{i=1}^{9}x_i=53;$$

$$(\varphi_2,\varphi_0)=(\varphi_0,\varphi_2)=\sum_{i=1}^{9}x_i^2=381;$$

$$(\varphi_1,\varphi_2)=(\varphi_2,\varphi_1)=\sum_{i=1}^{9}x_i^3=3017;(y,\varphi_0)=\sum_{i=1}^{9}y_i=32;$$

$$(y,\varphi_1)=\sum_{i=1}^{9}y_i\,x_i=147;(y,\varphi_2)=\sum_{i=1}^{9}y_i x_i^2=1025.$$

把数据代入方程组，得

$$\begin{bmatrix} 9 & 53 & 381 \\ 53 & 381 & 3017 \\ 381 & 3017 & 25317 \end{bmatrix}\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 32 \\ 147 \\ 1025 \end{bmatrix},$$

应用列主元*Gauss*消去法解得

$$a = 13.4609, b = -3.60585, c = 0.267616$$

因而$f(x) = 13.4609 + 3.60585x + 0.267616x^2.$

用Matlab实现最小二乘拟合

在Matlab程序编辑器中输入：

```
function p=nafit(x,y,m)  % x,y为数据；m为拟合多项式次数
A=zeros(m+1,m+1);  % zeros()生成零矩阵
for i=0:m
    for j=0:m
        A(i+1,j+1)=sum(x.^(i+j));
    end
    b(i+1)=sum(x.^i.*y);
end
a=A\b';
p=fliplr(a');  % fliplr()为矩阵左右翻转
```

E:\Matlab6.5\work\nafit.m

File Edit View Text Debug Breakpoints Web Window Help

```
function p=nafit(x,y,m)  % x,y为数据；m为拟合多项式次数
A=zeros(m+1,m+1);
for i=0:m
    for j=0:m
        A(i+1,j+1)=sum(x.^(i+j));
    end
    b(i+1)=sum(x.^i.*y);
end
a=A\b';
p=fliplr(a');  % fliplr()为矩阵左右翻转
```

最小二乘

拟合多项式

结果如右图：

Command Window

```
>> x=[1 3 4 5 6 7 8 9 10];y=[10 5 4 2 1 1 2 3 4];
>> nafit(x,y,2)

ans =

    0.2676   -3.6053   13.4597

>>
```

例2：求一形如$P(x)=Ae^{Mx}$的经验公式，使得拟合以下数据。

x_i	1	2	3	4
P_i	7	11	17	27

解：所求拟合公式为指数函数，对其两边取自然对数，得到$\ln P=\ln A+Mx$.从而函数变换表为：

x_i	1	2	3	4
$y_i=\ln P_i$	1.95	2.40	2.83	3.30

(*)

记$y=\ln P, a_0=\ln A, a_1=M$，则有$y=a_0+a_1x$

于是原问题转化为求 (*) 表的一次拟合多项式。

此时，$m=1, \varphi_0(x)=1, \varphi_1(x)=x.$

正规方程组为：$\begin{cases} 4a_0+10a_1=10.48 \end{cases}$

注意：函数表达，要按题目原来的函数表达

求得$a_0=1.50, a_1=0.448,$

于是$y=1.50+0.448x$.又$P=e^y$,

故$P(x)=e^{1.50+0.448x}=4.48e^{0.448x}$.

用最小二乘法解决实际问题的过程：

- （1）确定函数类；
- （2）根据原则求取最小二乘解，亦即确定其系数。

5.2　超定方程组（矛盾方程组）

设有线性方程组Ax=b,

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m1} & \cdots & a_{mn} \end{bmatrix}, b=\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, x=\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

当m>n时，称为超定方程组，一般来说，方程组无解.

即$\delta_i=\sum_{j=1}^{n} a_{ij}x_j-b_i, i=1,2,\cdots,m$不会全为零.但是可以求一组

$x^*=(x_1^*,x_2^*,\cdots,x_n^*,)$,使得$\varphi(x_1,x_2,\cdots,x_n)=\sum_{i=1}^{m}\delta_i^2$取最小值。

利用第一节中求多元函数极值的方法,可得方程组的形式为$A^{\mathrm{T}}Ax = A^{\mathrm{T}}b$.如果$A$是列满秩的,则方程组存在唯一解。而且该解使得$\varphi(x_1, x_2, \cdots, x_n)$取得最小值。我们把方程组的解称为超定方程组的最小二乘解。

例3: 用最小二乘法求下列超定方程组的近似解。

$$\begin{cases} 2x_1 - x_2 = 1 \\ 8x_1 + 4x_2 = 0 \\ 2x_1 + x_2 = 1 \\ 7x_1 - x_2 = 8 \\ 4x_1 = 3 \end{cases}$$

- 练习：P128， 习题5: 6

基本要求:

1.掌握曲线拟合的定义和几何意义;

2.掌握最小二乘法求超定方程组的最小二乘解。

作业：见课堂在线

第六章　数值积分与数值微分

6.1 问题的提出

求积分　　$I = \int_a^b f(x)dx$

问题：１．积不出．如$\int_a^b \sin x^2 dx, \int_a^b \frac{\sin t}{t} dt$　等．

２．难积．当$f(x)$复杂时，不好使用$N-L$公式．

３．表格函数

x_i	x_0	x_1	$\cdots$	x_n
$f(x_i)$	$f(x_0)$	$f(x_1)$	$\cdots$	$f(x_n)$

希望：用近似，简单有效方法求$I \to$数值积分公式

一般形式为$\int_a^b f(x)dx \approx \sum_{k=0}^{n} A_k f(x_k)$(计算简单)

其中x_k为求积节点，A_k为求积系数，同时都与$f(x)$无关。

$R[f] = \int_a^b f(x)dx - \sum_{k=0}^{n} A_k f(x_k)$叫截断误差（余项）

想法的合理性？？---积分中值定理及积分的定义。

6.2 插值型求积公式

6.2.1.插值型求积公式

已知$(x_k, f(x_k))(k=0,1,2,\cdots,n)$,则可得$f(x)$的插值多项式为

$$L_n(x)=\sum_{k=0}^{n} f(x_k)l_k(x),$$ 于是可以取

$$I_n(f)=\int_a^b L_n(x)dx=\int_a^b \sum_{k=0}^{n} f(x_k)l_k(x)dx=\sum_{k=0}^{n}[\int_a^b l_k(x)]dx\, f(x_k)$$

作为$I(f)=\int_a^b f(x)dx$的近似值。

定义.插值型求积公式

设有计算$I(f)=\int_a^b f(x)dx$的求积公式

$$L_n(x)=\sum_{k=0}^{n} A_k f(x_k), \qquad (6.1)$$

如其求积系数

$$A_k=\int_a^b l_k(x)dx(k=0,1,\cdots,n),$$

则称此求积公式为插值型求积公式。

6.2.2 梯形公式、辛卜生公式和柯特斯公式

梯形公式（两点插值型求积公式）：

当$n=1$时，若取$x_0=a, x_1=b$,则有

$$A_0=\int_a^b l_0(x)dx=\frac{1}{2}(b-a)$$

$$A_1=\int_a^b l_1(x)dx=\frac{1}{2}(b-a)$$

$$\int_a^b f(x)dx\approx\frac{b-a}{2}[f(a)+f(b)]=T(f) \qquad (6.2)$$

三点$Simpson$公式：若取$x_0=a, x_1=\frac{b+a}{2}, x_2=b$，得：

$$S(f)=\frac{b-a}{6}[f(a)+4f(\frac{b+a}{2})+f(b)] \qquad (6.3)$$

五点柯特斯（Cotes）公式：

取五个等距节点$x_i=a+ih(0\le i\le 4; h=\frac{b-a}{4})$，得：

$$C(f)=\frac{b-a}{90}[7f(x_0)+32f(x_1)+12f(x_2)+32f(x_3)+7f(x_4)]$$

6.2.3.求积公式的代数精度 (可用来衡量求积公式精 度)

定义6.2：如果一个求积公式$I_n(f)=\sum_{k=0}^{n}A_k f(x_k)$ (6.4)

对于次数$\le m$的多项式均能准确成立,但至少对一个$m+1$次多项式不准确成立,则称该求积公式具有m次代数精度.

定理6.1:求积公式$I_n(f)=\sum_{k=0}^{n}A_k f(x_k)$至少具有$n$次代数精度的充要条件是该公式是插值型的,

即$A_k=\int_a^b l_k(x)dx(k=0,1,\cdots,n)$.

[证明]: 由于$n+1$个节点的插值型求积公式的截断误差为:

$$R(f)=\int_a^b [f(x)-L_n(x)]dx=\int_a^b \frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x)dx,$$

从而其代数精度至少为n; 反过来, 如果(6.4)的代数精度为n,

则它对$l_i(x)=\prod_{\substack{j=0\\ j\neq i}}^{n}\frac{x-x_j}{x_i-x_j}$是精确成立的,

即有$\int_a^b l_i(x)dx=I_n(l_i)=\sum_{k=0}^{n}A_k l_i(x_k)=A_i$

于是, $A_i=\int_a^b l_i(x)dx \quad i=0,1,\cdots,n$

即(6.4)式是插值型的。

定理6.2：求积公式(6.4)具有m次代数精度$\Leftrightarrow$该公式对$f(x)=1, x, x^2, \cdots, x^m$精确成立，而对$f(x)=x^{m+1}$不精确成立.

例1 考察下面求积公式的代数精度：

$$\int_{-1}^{1} f(x)dx \approx \frac{1}{2}[f(-1)+2f(0)+f(1)].$$

解：当$f(x)=1$时，左边$=\int_{-1}^{1} 1dx = 2$，右边$=2$；

当$f(x)=x$时，左边$=\int_{-1}^{1} xdx = 0$，右边$=0$；

当$f(x)=x^2$时，左边$=\int_{-1}^{1} x^2dx = \frac{2}{3}$，右边$=1$；

所以次求积公式具有一次代数精度。

命题：*Simpson*公式具有三次代数精度.

$$(S(f)=\frac{b-a}{6}[f(a)+4f(\frac{a+b}{2})+f(b)])$$

证明：首先，*Simpson*公式是插值型的,

所以至少具有二次代数 精度.

当$f(x)=x^3$时，有$I(f)=\int_a^b x^3dx=\frac{1}{4}(b^4-a^4)$

$$S(f)=\frac{b-a}{6}[a^3+4(\frac{a+b}{2})^3+b^3]=\frac{1}{4}(b^4-a^4).$$

当$f(x)=x^4$时，$I(f)=\int_a^b x^4 dx=\frac{1}{4}(b^5-a^5)$

$$S(f)=\frac{b-a}{6}[a^4+4(\frac{a+b}{2})^4+b^4],$$

两者不等(可以通过比较b^5的系数)，故具有三次代数精度.

练习习题6-4，有没有什么规律？？

6.2.4.截断误差

梯形公式的截断误差：

$$R_T(f) = I(f) - T(f) = \int_a^b \frac{f''(\xi)}{2}(x-a)(x-b)dx$$

由于当$x \in (a,b)$时（$(x-a)(x-b) < 0$,

应用积分中值定理的推广（积分第一中值定理）有

$$R_T(f) = \frac{f''(\eta)}{2}\int_a^b (x-a)(x-b)dx = -\frac{(b-a)^3}{12} f''(\eta) \quad \eta \in (a,b)$$

积分中值定理？

推广（积分第一中值定理）？

积分中值定理

如果函数$f(x)$在闭区间$[a,b]$上连续，则在积分区间$[a,b]$上至少存在一个点ξ，使下式成立：

$$\int_a^b f(x)dx = f(\xi)(b-a)，\text{其中} a \le \xi \le b.$$

积分中值定理推广（第一定理）

如果函数$f(x)$，$g(x)$在闭区间$[a,b]$上连续，且$g(x)$在$[a,b]$上不变号，则在积分区间(a,b)上至少存在一个点ξ，使下式成立：

$$\int_a^b f(x)g(x)dx = f(\xi)\int_a^b g(x)dx$$

*Simpson*公式和*Cotes*公式呢？

$$R_S(f)=I(f)-S(f)=-\frac{b-a}{180}\left(\frac{b-a}{2}\right)^4 f^{(4)}(\eta)\qquad \eta\in(a,b)$$

$$R_C(f)=I(f)-C(f)=-\frac{2(b-a)}{945}\left(\frac{b-a}{4}\right)^6 f^{(6)}(\eta)\qquad \eta\in(a,b)$$

例：给定求积结点$-h, 0, h$，试确定系数A_{-1}, A_0, A_1，使

$$\int_{-2h}^{2h} f(x)dx \approx A_{-1} f(-h) + A_0 f(0) + A_1 f(h)$$

具尽可能高代数精度，并指出代数精度是多少.

解：令公式对$f = 1, x, x^2$精确成立，则

$$\int_{-2h}^{2h} 1 \cdot dx = A_{-1} + A_0 + A_1$$

$$\int_{-2h}^{2h} x \cdot dx = -hA_{-1} + hA_1$$

$$\int_{-2h}^{2h} x^2 \cdot dx = (-h)^2 A_{-1} + h^2 A_1$$

解得 $A_0 = -\frac{4}{3}h \qquad A_{-1} = A_1 = \frac{8}{3}h$

令$f = x^3$，左=右；令$f = x^4$，左$\neq$右.

故代数精度为3.

基本要求:

1. 掌握插值型求积公式;

2. 掌握常用的求积公式即两点梯型公式及三点Simpson公式及其误差;

3. 熟悉求积公式的代数精确度.

6.3复化求积公式

由梯形公式、Simpson公式及Cotes公式的截断误差表达式可以看出其截断误差依赖于求积区间的长度.若求积区间的长度是小量，则截断误差是求积区间长度的高阶小量。

从而，为了提高求积精度,可把积分区间分为若干个小区间，将求积公式写成这些小区间上的积分之和，然后对每个小区间上的积分应用Newton- Cotes公式,并把每个小区间上的结果累加,所得到的求积公式称为复化求积公式。

将区间n等分，记$h=\frac{b-a}{n}$, $x_i=a+ih(0\le i\le n)$，于是

$$I(f)=\sum_{k=0}^{n-1}\int_{x_k}^{x_{k+1}}f(x)dx,\text{并记}I_k(f)=\int_{x_k}^{x_{k+1}}f(x)dx.$$

6.3.1. 复化梯形公式

将区间n等分,记$h=\dfrac{b-a}{n}, x_i=a+ih(0\le i\le n)$,于是

$$I(f)=\sum_{k=0}^{n-1}\int_{x_k}^{x_{k+1}} f(x)dx, \text{并记} I_k(f)=\int_{x_k}^{x_{k+1}} f(x)dx.$$

在每个小区间上使用梯形公式有

$$\int_a^b f(x)dx \approx \frac{h}{2}[f(x_0)+f(x_1)]+\frac{h}{2}[f(x_1)+f(x_2)]$$

$$+\cdots+\frac{h}{2}[f(x_{n-1})+f(x_n)]$$

$$\int_a^b f(x)dx \approx \frac{h}{2}[f(a)+2\sum_{k=0}^{N-1} f(x_k)+f(b)]$$

记$T_n(f)=\frac{h}{2}[f(a)+2\sum_{k=1}^{n-1}f(x_k)+f(b)]$

复化梯形公式的截断误差为：

$$I(f)-T_n(f)=\sum_{k=0}^{n-1}\int_{x_k}^{x_{k+1}}f(x)dx-\sum_{k=0}^{n-1}\frac{h}{2}[f(x_k)+f(x_{k+1})]$$

$$=\sum_{k=0}^{n-1}\left\{\int_{x_k}^{x_{k+1}}f(x)dx-\frac{h}{2}[f(x_k)+f(x_{k+1})]\right\}$$

$$=\sum_{k=0}^{n-1}\left[-\frac{h^3}{12}f''(\eta_k)\right]=-\frac{h^3}{12}\sum_{k=0}^{n-1}f''(\eta_k)=-\frac{nh^3}{12}\sum_{k=0}^{n-1}\frac{1}{n}f''(\eta_k)$$

$$=-\frac{b-a}{12}f''(\eta)h^2$$

6.3.2.复化*Simpson*公式

记$x_{k+\frac{1}{2}}=\dfrac{1}{2}(x_k+x_{k+1})$,在每个区间应用*Simpson*公式,

得到复化*Simpson*公式。

$$S_n(f)=\sum_{k=0}^{n-1}\frac{h}{6}[f(x_k)+4f(x_{k+\frac{1}{2}})+f(x_{k+1})],\text{或}$$

$$S_n(f)=\frac{h}{6}[f(x_0)+2\sum_{k=0}^{n\text{-}1}f(x_k)+f(x_n)+4\sum_{k=0}^{n\text{-}1}f(x_{k+\frac{1}{2}})]$$

复化*Simpson*公式的截断误差:

$$I(f)-S_n(f)=\frac{b-a}{180}f^{(4)}(\eta)(\frac{h}{2})^4\approx\frac{1}{180}[f^{(3)}(a)-f^{(3)}(b)](\frac{h}{2})^4;$$

6.3.3 复化柯特斯公式

记$x_{k+\frac{1}{4}} = x_k + \frac{1}{4}h, x_{k+\frac{1}{2}} = x_k + \frac{1}{2}h, x_{k+\frac{3}{4}} = x_k + \frac{3}{4}h,$在每个区间

应用$Cotes$公式，得到复化$Cotes$公式。

$$C_n(f) = \sum_{k=0}^{n-1} \frac{h}{90}[7f(x_k) + 32f(x_{k+\frac{1}{4}}) + 12f(x_{k+\frac{1}{2}}) + 32f(x_{k+\frac{3}{4}})$$
$$+7f(x_{k+1})]$$

复化$Cotes$公式的截断误差：

$$I(f) - C_n(f) = \frac{2(b-a)}{945} f^{(6)}(\eta)(\frac{h}{2})^6 \approx \frac{2}{945}[f^{(5)}(a) - f^{(5)}(b)](\frac{h}{2})^6$$

例1.取9个等距节点(包括区间端点)，用复化梯形公式和复化*Simpson*公式求积分$\int_0^1 \frac{4}{1+x^2}dx$的近似值。

x_k	$f(x_k)$	梯形组合系数	Simpson组合系数
0	4.000000	1	1
1/8	3.938461	2	4
1/4	3.764705	2	2
3/8	3.506849	2	4
1/2	3.200000	2	2
5/8	2.876404	2	4
3/4	2.560000	2	2
7/8	2.265486	2	4
1	2.000000	1	1

对梯形公式$\sum = 50.2238218$,

从而$T_8(f) = 3.138989$;

对Simpson公式$\sum = 75.398224$,

从而$S_4(f) = 3.141593$.

与积分的精确值$\pi = 3.141592653\cdots$比较,

T_8只有三位有效数字，而S_4有七位有效数字.

6.3.4 复化求积公式的阶：

1.定义：设有一个复化求积公式$I_n(f)$,如果

$$\lim_{h\to 0}\frac{I(f)-I_n(f)}{h^p}=c,$$ 其中 ,c为与$h(h=\dfrac{b-a}{n})$无关的非零常数,

则称该求积公式是p阶收敛的。

复化梯形公式的截断误差为：$I(f)-T_n(f)=-\dfrac{b-a}{12}f''(\eta)h^2$

$$I(f)-T_n(f)=-\frac{h^3}{12}\sum_{k=0}^{n-1}f''(\eta_k);$$

$$\frac{I(f)-T_n(f)}{h^3}=-\frac{1}{12}\underline{\sum_{k=0}^{n-1}f''(\eta_k)}=c??$$

$$\lim_{h\to 0}\frac{I(f)-T_n(f)}{h^2}=\lim_{h\to 0}-\frac{h}{12}\underline{\sum_{k=0}^{n-1}f''(\eta_k)}=\frac{1}{12}[f'(a)-f'(b)]$$

$$\lim_{h\to 0}\frac{I(f)-T_n(f)}{h^2}=\lim_{h\to 0}-\frac{b-a}{12}f''(\eta)=???$$

$$I(f)-T_n(f)=-\frac{h^3}{12}\sum_{k=0}^{n-1}f''(\eta_k);$$

$$\frac{I(f)-T_n(f)}{h^3}=-\frac{1}{12}\underline{\sum_{k=0}^{n-1}f''(\eta_k)}=c??$$

$$\lim_{h\to 0}\frac{I(f)-T_n(f)}{h^2}=\lim_{h\to 0}-\frac{h}{12}\underline{\sum_{k=0}^{n-1}f''(\eta_k)}=\frac{1}{12}[f'(a)-f'(b)]$$

同理，$$\lim_{h\to 0}\frac{I(f)-S_n(f)}{(\frac{h}{2})^4}=\frac{1}{180}(f^{(3)}(a)-f^{(3)}(b))$$

$$\lim_{h\to 0}\frac{I(f)-C_n(f)}{(\frac{h}{4})^6}=\frac{2}{945}(f^{(5)}(a)-f^{(5)}(b))$$

说明：可以看出，复化梯形公式、复化*Simpson*公式、和复化*Cotes*公式分别是二阶、四阶和六阶收敛的。

如何使用插值型求积公式达到近似值的精度要求呢？

2.步长的自动选择

加密节点 $\rightarrow$ 提高精度,使用公式前如何给出合适步长?

一般采用区间逐次二分，反复利用求积公式计算，

直至所求得的前后二次积分的差满足精度为止。

终止条件的合理性说明 ：

设有一个p阶收敛的复化求积公式$I_n(f)$,则当h充分小时

有$I(f)-I_n(f)\approx ch^p$及$I(f)-I_{2n}(f)\approx c\left(\frac{h}{2}\right)^p$

有以上两式得$I(f)-I_{2n}(f)\approx \frac{1}{2^p}[I(f)-I_n(f)]$

即每二分一次,截断误差缩小2^p倍.

两边同时乘以 2^p,移项得

$$I(f)-I_{2n}(f)\approx\frac{1}{2^p-1}[I_{2n}(f)-I_n(f)]. \qquad (6.5)$$

对于给定的精度ε,当$|I_{2n}(f)-I_n(f)|\le\varepsilon$时,我们有

$$|I(f)-I_{2n}(f)|\approx\left|\frac{1}{2^p-1}[I_{2n}(f)-I_n(f)]\right|\le\frac{1}{2^p-1}\varepsilon\le\varepsilon,$$

因而可取$I_{2n}(f)$作为$I(f)$的近似值.

观察（6.5）式，能发现什么？

由(6.5)式得到的复化求积公式误差表达式及其启示：

1.梯形公式：$I(f)-T_{2n}(f)\approx\frac{1}{3}[T_{2n}(f)-T_n(f)]$;

移项得：$I(f)\approx T_{2n}(f)+\frac{1}{3}[T_{2n}(f)-T_n(f)]=\frac{4}{3}T_{2n}(f)-\frac{1}{3}T_n(f)$;

令$\tilde{T}(f)=T_{2n}(f)+\frac{1}{3}[T_{2n}(f)-T_n(f)]=\frac{4}{3}T_{2n}(f)-\frac{1}{3}T_n(f)$

由于$\frac{4}{3}T_{2n}(f)-\frac{1}{3}T_n(f)$

$$=\frac{4}{3}\sum_{k=0}^{n-1}\{\frac{h}{4}[f(x_k)+f(x_{k+\frac{1}{2}})]+[\frac{h}{4}[f(x_{k+\frac{1}{2}})+f(x_{k+1})]\}$$

$$-\frac{1}{3}\sum_{k=0}^{n-1}\frac{h}{2}[f(x_k)+f(x_{k+1})]$$

$$=\sum_{k=0}^{n-1}\{\frac{h}{3}[f(x_k)+2f(x_{k+\frac{1}{2}})+f(x_{k+1})]-\frac{h}{6}[f(x_k)+f(x_{k+1})]\}$$

$$=\sum_{k=0}^{n-1}\frac{h}{6}[f(x_k)+4f(x_{k+\frac{1}{2}})+f(x_{k+1})=S_n(f);$$

即$S_n(f)=\frac{4}{3}T_{2n}(f)-\frac{1}{3}T_n(f)$;

发挥了复化梯形公式算法简单的优点,
同时形成一个新的算法.
对于*Simpson*公式和*Cotes*公式是否也可以产生精度
更高的新算法?

2.复化$Simpson$公式：

$I(f)-S_{2n}(f)\approx\frac{1}{15}[S_{2n}(f)-S_n(f)]$;移项，得

$$I(f)\approx S_{2n}(f)+\frac{1}{15}[S_{2n}(f)-S_n(f)]=\frac{16}{15}S_{2n}(f)-\frac{1}{15}S_n(f)$$

而$C_n(f)=\frac{16}{15}S_{2n}(f)-\frac{1}{15}S_n(f).$

3.对复化$Cotes$公式重复上面的过程可得：

$$I(f)\approx C_{2n}(f)+\frac{1}{63}[C_{2n}(f)-C_n(f)]=\frac{64}{63}C_{2n}(f)-\frac{1}{63}C_n(f);$$

记$R_n(f)=\dfrac{64}{63}C_{2n}(f)-\dfrac{1}{63}C_n(f)$,称为$Romberg$公式,

具有7次代数精度，截断误差为$O(h^8)$.

6.4 Romberg求积算法

区间等分数2^k	T_{2^k}	$S_{2^{k-1}}$	$C_{2^{k-2}}$	$R_{2^{k-3}}$
1	T_1			
2	T_2	S_1		
3	T_4	S_2	C_1	
4	T_8	S_4	C_2	R_1
5	T_{16}	S_8	C_4	R_2
6	T_{32}	S_{16}	C_8	R_4
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

1.算法终止条件：$\frac{1}{63}|C_{2^k}-C_{2^{k-1}}|\le\varepsilon$或$\frac{1}{255}|R_{2^{k-1}}-R_{2^{k-2}}|\le\varepsilon.$

2.算法中计算量的节约：

复化梯形公式二分前记为$T_n(f)$,二分后记为$T_{2n}(f)$,

记$h=\frac{b-a}{n}$为$\boxed{\text{二分前}}$的步长，则

$$T_{2n}(f)=\sum_{k=0}^{n-1}\frac{h}{4}[f(x_k)+2f(x_{k+\frac{1}{2}})+f(x_{k+1})]$$

$$=\frac{1}{2}\sum_{k=0}^{n-1}\frac{h}{2}[f(x_k)+f(x_{k+1})]+\frac{h}{2}\sum_{k=0}^{n-1}f(x_{k+\frac{1}{2}})$$

$$=\frac{1}{2}T_n(f)+\frac{h}{2}\sum_{k=0}^{n-1}f(x_{k+\frac{1}{2}})$$

例2.用Romberg公式求积分$I=\int_0^2 e^{-x^2}dx$.

解：设$f(x)=e^{-x^2}$,则有

$$T_1=\frac{2}{2}[f(0)+f(2)]=1.0183156380$$

$$T_2=\frac{1}{2}T_1+\frac{2}{2}f(1)=0.8770372602$$

$$S_1=\frac{1}{3}(4T_2-T_1)=0.8299444676$$

$$T_4 = ?; S_2 = ?$$

$$C_1 = \frac{1}{15}(16S_2 - S_1) = 0.8852702890$$

$$T_8 = ?; S_4 = ?; C_2 = ?;$$

$$R_1 = \frac{1}{63}(64C_2 - C_1) = 0.8820317809$$

$$T_{16} = ?; \quad S_8 = ?; \quad C_4 = ?;$$

终止条件？？

$$R_2 = \frac{1}{63}(64C_4 - C_2) = 0.8820813731$$

因为$\frac{1}{63}|C_4 - C_2| = 0.158 \times 10^{-7}$，所以所求

I的近似值为0.8820814.

用Matlab实现龙贝格(Romberg)求积

```
function t=naromberg(fname,a,b,e) %fname为被积
函数；a,b分别为上下限，e为精度
if nargin<4,e=1e-4;end
  i=1;j=1;h=b-a;
T(i,1)=h/2*(feval(fname,a)+feval(fname,b));
T(i+1,1)=T(i,1)/2+sum(feval(fname,a+h/2:h:b-
h/2+0.001*h))*h/2;
T(i+1,j+1)=4^j*T(i+1,j)/(4^j-1)-T(i,j)/(4^j-1);
  while abs(T(i+1,i+1)-T(i,i))>e,
        i=i+1;h=h/2;
```

```
T(i+1,1)=T(i,1)/2+sum(feval(fname,a+h/2:h:b-h/2+0.001*h))*h/2;
    for j=1:i
        T(i+1,j+1)=4^j*T(i+1,j)/(4^j-1)-T(i,j)/(4^j-1);
    end
end
T
t=T(i+1,j+1);
```

E:\Matlab6.5\work\naromberg.m

File Edit View Text Debug Breakpoints Web Window Help

```
function t=naromberg(fname,a,b,e) %fname为被积函数；a,b分别为上下限，e为精度
if nargin<4,e=1e-4;end
i=1;j=1;h=b-a;
T(i,1)=h/2*(feval(fname,a)+feval(fname,b));
T(i+1,1)=T(i,1)/2+sum(feval(fname,a+h/2:h:b-h/2+0.001*h))*h/2;
T(i+1,j+1)=4^j*T(i+1,j)/(4^j-1)-T(i,j)/(4^j-1);
while abs(T(i+1,i+1)-T(i,i))>e,
    i=i+1;h=h/2;
    T(i+1,1)=T(i,1)/2+sum(feval(fname,a+h/2:h:b-h/2+0.001*h))*h/2;
    for j=1:i
        T(i+1,j+1)=4^j*T(i+1,j)/(4^j-1)-T(i,j)/(4^j-1);
    end
end
T
t=T(i+1,j+1);
```

用Matlab求解结果如右图：

Command Window

```
>> format long;naromberg(inline('exp(-x.^2)'),0,2,0.5e-7),format short;

T =

  Columns 1 through 5

   1.01831563888873                  0                  0                  0                  0
   0.87703726061581   0.82994446785817                  0                  0                  0
   0.88061863412454   0.88181242529412   0.88527028912318                  0                  0
   0.88170379133213   0.88206551040133   0.88208238274181   0.88203178105322                  0
   0.88198624526578   0.88208039657699   0.88208138898870   0.88208137321484   0.88208156769391
   0.88205755780121   0.88208132864636   0.88208139078431   0.88208139081282   0.88208139088183
   0.88207542961079   0.88208138688065   0.88208139076294   0.88208139076260   0.88208139076241

  Columns 6 through 7

                  0                  0
                  0                  0
                  0                  0
                  0                  0
                  0                  0
   0.88208139070899                  0
   0.88208139076229   0.88208139076230


ans =

   0.88208139076230
```

基本要求:

1.掌握复化梯形公式和复化Simpson公式;

2.掌握复化梯形公式和复化Simpson公式的截断误差;

3.熟悉Romberg算法.

作业：见课堂在线

6.5 Gauss求积公式

6.5.1 问题

已知$n+1$个节点的插值型求积公 式

$$\begin{cases} \int_a^b f(x)dx \approx \sum_{k=0}^{n} A_k f(x_k) \\ A_k = \int_a^b l_k(x)dx = \int_a^b \prod_{\substack{k=0 \\ i\neq k}}^{n} \frac{x-x_i}{x_k-x_i}dx \end{cases} \quad (6.6)$$

的代数精度为n或$n+1$(当$n+1$位奇数时),

问:$[a,b]$内节点数$n+1$一定时，如何选取节点 x_k的位置，以使得插值型求积公式 代数精度达最高?

6.5.2 *Gauss*型求积公式问题

1.定义：如果插值型求积公式（6.6）的代数精度达到 $2n+1$, 则称该公式为*Gauss*型求积公式。此时称 x_k 为 *Gauss*点，A_k 为*Gauss*系数。精度还能更高吗？

2.取 $f(x)=\omega_{n+1}{}^2(x)=(x-x_0)^2(x-x_1)^2\cdots(x-x_n)^2$

$2n+2$次，带入（6.6），得左边 $=\int_a^b \omega_{n+1}{}^2(x)dx>0$

右边 $=\sum_{k=0}^{n} A_k\omega_{n+1}{}^2(x_k)=0$, 故左边 $\neq$ 右边,

即求积公式(6.6)的代数精度达不到 $2n+2$。

同时得，*Gauss*求积公式的代数精度最高。

3.如何求$Gauss$点

定理：对于插值型求积公式（6.6），其节点x_k（$k=0,1,\cdots,n$）是高斯点的充要条件 是$(n+1)$次多项式

$$W_{n+1}(x)=(x-x_0)(x-x_1)\cdots(x-x_n)$$

与任意次数不超过 n的多项式$p(x)$均正交，即

$$\int_a^b p(x)W_{n+1}(x)dx=0.$$

即区间$[a,b]$上的高斯点就是区间上的$(n+1)$次正交多项式$\mathrm{W}_{n+1}(x)$的零点。证明略。

区间$[-1,1]$内的n次正交多项式为勒让德多项式

$$P_n(x)=\frac{n!}{(2n)!}\frac{d^n(x^2-1)^n}{dx^n}$$

由此，可以得到区间$[-1,1]$内的高斯点和高斯求积系数。想更多了解正交多项式及其类型的可参考课堂在线共享资料西安交大教材。

区间$[-1,1]$内的高斯点和高斯求积 系数见课本表6-5-1.

例3：对$\int_{-1}^{1} f(x)dx$写出3点高斯公式。

解：$\int_{-1}^{1} f(x)dx \approx 0.5555555 f(-0.7745967) + 0.8888889 f(0)$

$+ 0.5555555 f(0.7745967)$

例4.建立计算积分$\int_{2}^{10} f(x)dx$的高斯求积公式，

使其具有3次代数精度.

对于积分$\int_a^b f(x)dx$，作变换

$$x=\frac{a+b}{2}+\frac{b-a}{2}t,$$则有

$$\int_a^b f(x)dx=\int_{-1}^{1}\frac{b-a}{2}f(\frac{a+b}{2}+\frac{b-a}{2}t)dt;$$

于是，若设$\int_{-1}^{1}g(t)dt\approx\sum_{k=0}^{n}\tilde{A}_k g(t_k)$是区间$[-1,1]$上的$n+1$个求积

节点的高斯公式，则$\int_a^b f(x)dx\approx\sum_{k=0}^{n}(\frac{b-a}{2}\tilde{A}_k)f(\frac{a+b}{2}+\frac{b-a}{2}t_k)$，

此式为$n+1$点的高斯公式，其求积节点和求积系数为：

$$x_k=\frac{a+b}{2}+\frac{b-a}{2}t_k,\ A_k=\frac{b-a}{2}\tilde{A}_k.$$

4.高斯求积公式的优缺点：

优点:(1)代数精度最高

（2）*Gauss*系数≥ 0，可证明*Gauss*求积公式计算过程稳定。

缺点：增加节点时，计 算量大，不能利用已算 出的函数值。实际使用时也可分段 使用节点少的*Gauss*求积公式。

基本要求:

1.了解Gauss求积公式的思想;

2.会使用Gauss求积公式的节点和系数表

6.6 重积分的计算

思想沿用已有求积公式。

以矩形域$D=\{(x,y)\,|\,a\le x\le b, c\le y\le d\}$上的重积分为例。

设$\mathrm{I}(f)=\iint\limits_D f(x,y)dxdy$

根据积分中值定理，$\mathrm{I}(f)=f(\xi,\eta)(b-a)(d-c)$

其中,$(\xi,\eta)\in D$, $f(\xi,\eta)$是否可以给出？？？

一重积分有哪些形式？？？

是否可以同理给出重积分公式?

首先将重积分华为累次积分

$$I(f)=\int_a^b[\int_c^d f(x,y)dy]dx,\text{记}g(x)=\int_c^d f(x,y)dy$$

$$\text{则}I(f)=\int_a^b g(x)dx$$

其次，利用梯形公式逐步进行推导

最后，得

$$I(f)=\frac{[f(a,c)+f(a,d)+f(b,c)+f(b,d)]}{4}(b-a)(d-c)$$

称之为求二重积分的梯形公式。

同理，为了提高求积公式精度，也可利用复化梯形公式。

$$T_{m,n}(f)=\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\frac{hk}{4}[f(x_i,y_i)+f(x_i,y_{i+1})+f(x_{i+1},y_i)+f(x_{i+1},y_{i+1})]$$

$$=hk\sum_{i=0}^{m}\sum_{j=0}^{n}\omega_{ij}f(x_i,y_i),$$

其中，$\omega_{ij}=\begin{cases}1, & 1\le i\le m-1,\quad 1\le j\le n-1\\ \dfrac{1}{4}, & (i,j)=(0,0),(m,0),(0,n),(m,n);\\ \dfrac{1}{2}, & \text{其他}\end{cases}$

称之为求二重积分的复化梯形公式的外推公式。

6.7 数值微分

6.7.1 问题的提出

根据函数在一些离散点上的函数值推算某点处导数的近似值的方法为数值微分。

简单方法：

1）向前差商：$f'(x_0) \approx \dfrac{f(x_0+h)-f(x_0)}{h}$

2)向后差商：$f'(x_0) \approx \dfrac{f(x_0)-f(x_0-h)}{h}$

3)中心差商：$f'(x_0) \approx \dfrac{f(x_0+h)-f(x_0-h)}{2h}$

选哪一种呢？ $D(h)=\dfrac{f(x_0+h)-f(x_0-h)}{2h}$

对应的截断误差：$f'(x_0)-D(h)=-\dfrac{1}{6}h^2 f'''(x_0+\theta h)$

其中$\theta\in$（-1,1）

可见：h越小，越精确。

但是h越小，$f(x_0+h)$和$f(x_0-h)$越接近，

造成？？？

因此，步长不宜太小。那如何解决呢？

解决方法：二分步长，误差事后估计的方式。

6.7.2插值型求导公式及截断误差

应用插值多项式，生成离散数据的多项式函数，再求导。这样建立的数值微分公式$f'(x) \approx p_n'(x)$，称为插值型求导公式。

对应的截断误差：$f'(x_k) - p_n'(x_k) = \dfrac{f^{(n+1)}(\xi)}{(n+1)!} W'_{n+1}(x_k)$