

3.Cotes公式及其余项

取 $n=4$, 则 $x_k = a + kh$, $k=0,1,\dots,4$, $h = \frac{b-a}{4}$

Cotes系数为

$$C_k^{(n)} = \frac{(-1)^{n-k}}{n \cdot k!(n-k)!} \int_0^n \prod_{\substack{j=0 \\ j \neq k}}^n (t-j) dt$$

$$C_0^{(4)} = \frac{1}{4 \cdot 4!} \int_0^4 (t-1)(t-2)(t-3)(t-4) dt = \frac{7}{90}$$

$$C_1^{(4)} = -\frac{1}{4 \cdot 3!} \int_0^4 t(t-2)(t-3)(t-4) dt = \frac{32}{90}$$

$$C_2^{(4)} = \frac{1}{4 \cdot 2! \cdot 2!} \int_0^4 t(t-1)(t-3)(t-4) dt = \frac{12}{90}$$

$$C_3^{(4)} = -\frac{1}{4 \cdot 3!} \int_0^4 t(t-1)(t-2)(t-4) dt = \frac{32}{90}$$

$$C_4^{(4)} = -\frac{1}{4 \cdot 4!} \int_0^4 t(t-1)(t-2)(t-3) dt = \frac{7}{90}$$

求积公式为

$$\begin{aligned} I_4(f) &= (b-a) \sum_{k=0}^4 C_k^{(4)} f(x_k) \\ &= (b-a) \left[\frac{7}{90} f(x_0) + \frac{32}{90} f(x_1) + \frac{12}{90} f(x_2) + \frac{32}{90} f(x_3) + \frac{7}{90} f(x_4) \right] \\ &= \frac{b-a}{90} [7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4)] \end{aligned}$$

上式称为 **Cotes求积公式**，也称**五点公式**。

记为 $C = I_4(f)$

Cotes公式的余项：

$$R(C) = R(I_4) = \int_a^b R_4(x) dx = -\frac{2(b-a)}{945} \left(\frac{b-a}{4}\right)^6 f^{(6)}(\eta)$$

Cotes公式具有5次代数精度。

Cotes系数表：

n	$C_k^{(n)}$					
1	1/2	1/2				
2	1/6	4/6	1/6			
3	1/8	3/8	3/8	1/8		
4	7/90	16/45	2/15	16/45	7/90	
5	19/288	25/96	25/144	25/144	25/96	19/288
					
8	989/28350	5888/28350	-928/28350	10496/28350	-4540/28350	...

- 注： $n \geq 8$ 时，Cotes系数出现负数，会引起误差增大，计算不稳定。

因此，在实际应用中一般不使用高阶Newton-Cotes公式，而是采用低阶复合求积法(下节)。

7.2.3 偶阶求积公式的代数精度

研究Simpson公式，是二阶Newton-Cotes公式，因此至少具有二次代数精度。

将 $f(x)=x^3$ 代入Simpson公式：

$$I_2(f) = \frac{b-a}{6} \left[a^3 + 4\left(\frac{a+b}{2}\right)^3 + b^3 \right] = \frac{b^4 - a^4}{4}$$

直接对 $f(x)=x^3$ 求积，得

$$I = \int_a^b x^3 dx = \frac{b^4 - a^4}{4}$$

有 $I_2(f) = I$ ，又易证Simpson公式对 $f(x)=x^4$ 不能够准确成立。

故Simpson公式具有3次代数精度。

一般地，可以证明下述论断：

定理： 当 n 为偶数时，Newton-Cotes公式至少具有 $n+1$ 次代数精度.

证明： 只要验证当 n 为偶数时，公式对 $f(x)=x^{n+1}$ 余项为零即可.

由余项公式
$$R[f] = \int_a^b \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega_{n+1}(x) dx$$

又
$$f^{(n+1)}(x) = (n+1)!$$

故
$$R[f] = \int_a^b (x - x_0)(x - x_1) \cdots (x - x_n) dx$$

引进变换 $x=a+th$, 则 $x_j=a+jh$,

$$R[f] = h^{n+2} \int_0^n t(t-1)(t-2)\cdots(t-n)dt$$

若 n 为偶数, 则 $n/2$ 为整数, 再令 $t=u+n/2$, 得

$$R[f] = h^{n+2} \int_{-\frac{n}{2}}^{\frac{n}{2}} (u + \frac{n}{2})(u + \frac{n}{2} - 1)\cdots(u + \frac{n}{2} - n)du$$

此时, 被积函数

$$g(u) = \prod_{j=0}^n (u + \frac{n}{2} - j) = \prod_{j=-n/2}^{n/2} (u - j)$$

是奇函数, 故 $R[f]=0$. 证毕.

例1. 分别使用梯形公式、Simpson公式、Cotes公式

计算定积分：
$$I = \int_1^3 (x^3 - 2x^2 + 7x - 5)dx.$$

解： 梯形公式：

$$T = \frac{b-a}{2}[f(a) + f(b)] = \frac{3-1}{2}[1 + 25] = 26$$

Simpson公式：

$$\begin{aligned} S &= \frac{b-a}{6}[f(a) + 4f(\frac{a+b}{2}) + f(b)] \\ &= \frac{3-1}{6}[1 + 4 \times 9 + 25] = \frac{62}{3} \end{aligned}$$

Cotes公式：

$$\begin{aligned} C &= \frac{3-1}{90} [7f(1) + 32f(1.5) + 12f(2) + 32f(2.5) + 7f(3)] \\ &= \frac{1}{45} [7 + 32 \times \frac{35}{8} + 12 \times 9 + 32 \times \frac{125}{8} + 7 \times 25] = \frac{62}{3} \end{aligned}$$

$$R(S) = -\frac{b-a}{180} \left(\frac{b-a}{2}\right)^4 f^{(4)}(\eta) = 0$$

$$R(C) = 0.$$

7.2.3 复合求积公式

当积分区间 $[a,b]$ 的长度较大，而节点个数 $n+1$ 固定时，直接使用Newton-Cotes公式的余项将会较大。

而如果增加节点个数，即 $n+1$ 增加时，公式的舍入误差又很难得到控制。

为了提高公式的精度，又使算法简单易行，往往使用下述复合方法：

将积分区间 $[a,b]$ 分成若干个子区间，然后在每个小区间上使用低阶Newton-Cotes公式，最后将每个小区间上的积分的近似值相加。

1、复合梯形公式

将 $[a,b]$ n 等分, $h=(b-a)/n$, 在每个子区间 $[x_k, x_{k+1}]$ ($k=0,1,\dots,n-1$)上采用梯形公式, 得

$$\begin{aligned} I &= \int_a^b f(x)dx = \sum_{k=0}^{n-1} \int_{x_k}^{x_{k+1}} f(x)dx \\ &= \frac{h}{2} \sum_{k=0}^{n-1} [f(x_k) + f(x_{k+1})] + R_n(f) \end{aligned}$$

记

$$\begin{aligned} T_n &= \frac{h}{2} \sum_{k=0}^{n-1} [f(x_k) + f(x_{k+1})] \\ &= \frac{h}{2} [f(a) + 2 \sum_{k=1}^{n-1} f(x_k) + f(b)] \end{aligned}$$

——复合梯形公式

复合梯形公式的余项:

设被积函数 $f(x) \in C^2[a, b]$, 由 $R(T) = -\frac{(b-a)^3}{12} f''(\eta)$

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

$$\text{由于} \quad \min_{a \leq x \leq b} \{f''(x)\} \leq \sum_{k=0}^{n-1} \frac{f''(\eta_k)}{n} \leq \max_{a \leq x \leq b} f''(x)$$

$$\text{由介值定理, } \exists \eta \in [a, b], \text{ 使得} \quad \sum_{k=0}^{n-1} \frac{f''(\eta_k)}{n} = f''(\eta)$$

$$\text{即有} \quad I - T_n = -\frac{nh^3}{12} \sum_{k=0}^{n-1} \frac{f''(\eta_k)}{n} = -\frac{nh^3}{12} f''(\eta)$$

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

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

$$\frac{I - T_n}{h^2} = -\frac{1}{12} \sum_{k=0}^{n-1} f''(\eta_k) h = -\frac{1}{12} \sum_{k=0}^{n-1} f''(\eta_k) \Delta x_k$$

$$\approx -\frac{1}{12} \int_a^b f''(x) dx = -\frac{1}{12} [f'(b) - f'(a)] \quad \left(\begin{array}{l} h \rightarrow 0 \\ n \rightarrow \infty \end{array} \right)$$

因此当 n 足够大时,复合梯形公式的余项为

$$I - T_n \approx -\frac{h^2}{12} [f'(b) - f'(a)]$$

2、复合Simpson公式

将 $[a,b]$ n 等分，在每个子区间 $[x_k, x_{k+1}]$ ($k=0,1,\dots,n-1$)上采用Simpson公式，若记 $x_{k+1/2} = x_k + h/2$ ，则可得复合Simpson公式形式为

$$\begin{aligned} S_n &= \frac{h}{6} \sum_{k=0}^{n-1} [f(x_k) + 4f(x_{k+1/2}) + f(x_{k+1})] \\ &= \frac{h}{6} [f(a) + 4 \sum_{k=0}^{n-1} f(x_{k+1/2}) + 2 \sum_{k=1}^{n-1} f(x_k) + f(b)] \end{aligned}$$

复合Simpson公式的余项：

设被积函数 $f(x) \in C^4[a, b]$,

$$R(S) = -\frac{b-a}{180} \left(\frac{b-a}{2}\right)^4 f^{(4)}(\eta)$$

则当 n 足够大时，复合Simpson公式的余项为：

$$I - S_n = \sum_{k=0}^{n-1} \left(-\frac{h^5}{180 \cdot 2^4} f^{(4)}(\eta_k) \right) = -\frac{b-a}{180} \left(\frac{h}{2}\right)^4 f^{(4)}(\eta)$$

$$\approx -\frac{h^4}{180 \cdot 2^4} [f'''(b) - f'''(a)]$$

$$= -\frac{1}{180} \left(\frac{h}{2}\right)^4 [f'''(b) - f'''(a)]$$

3、复合Cotes公式

$$\begin{aligned}\int_a^b f(x)dx &\approx C_n = h \sum_{k=0}^{n-1} \sum_{i=0}^4 C_i^{(4)} f(x_{k+\frac{i}{2}}) \\&= \frac{h}{90} \sum_{k=0}^{n-1} [7f(x_k) + 32f(x_{k+\frac{1}{4}}) + 12f(x_{k+\frac{2}{4}}) + 32f(x_{k+\frac{3}{4}}) + 7f(x_{k+1})] \\&= \frac{b-a}{90n} [7f(a) + \sum_{k=0}^{n-1} [32f(x_{k+\frac{1}{4}}) + 12f(x_{k+\frac{2}{4}}) + 32f(x_{k+\frac{3}{4}})] + 14 \sum_{k=1}^{n-1} f(x_k) + 7f(b)]\end{aligned}$$

复合Cotes公式的余项： 设被积函数 $f(x) \in C^6[a, b]$,

$$\begin{aligned}I - C_n &= \sum_{k=0}^{n-1} \left(-\frac{2h^7}{945 \cdot 4^6} f^{(6)}(\eta_k) \right) = -\frac{2(b-a)}{945} \left(\frac{h}{4} \right)^6 f^{(6)}(\eta) \\&\approx -\frac{2h^6}{945 \cdot 4^6} [f^{(5)}(b) - f^{(5)}(a)] = -\frac{2}{945} \left(\frac{h}{4} \right)^6 [f^{(5)}(b) - f^{(5)}(a)]\end{aligned}$$

比较三种复合公式的余项：

$$I - T_n \approx -\frac{1}{12}h^2[f'(b) - f'(a)] = o(h^2)$$

$$I - S_n \approx -\frac{1}{180}\left(\frac{h}{2}\right)^4[f'''(b) - f'''(a)] = o(h^4)$$

$$I - C_n \approx -\frac{2}{945}\left(\frac{h}{4}\right)^6[f^{(5)}(b) - f^{(5)}(a)] = o(h^6)$$

分别是 h 的2,4,6阶无穷小量,

即 T_n, S_n, C_n 趋于定积分 I 的速度依次更快.

例2. 使用各种复合求积公式计算定积分 $I = \int_0^1 \frac{\sin x}{x} dx$

解: 为简单起见, 依次使用8阶复合梯形公式、4阶复合Simpson公式和2阶复合Cotes公式.

可得各节点的值如下表:

<i>Trapz</i>	<i>Simp.</i>	<i>Cotes</i>	x_i	$f(x_i)$
x_0	x_0	x_0	0	1
x_1	$x_{0+\frac{1}{2}}$	$x_{0+\frac{1}{4}}$	0.125	0.99739787
x_2	x_1	$x_{0+\frac{1}{2}}$	0.25	0.98961584
x_3	$x_{1+\frac{1}{2}}$	$x_{0+\frac{3}{4}}$	0.375	0.97672674
x_4	x_2	x_1	0.5	0.95885108
x_5	$x_{2+\frac{1}{2}}$	$x_{1+\frac{1}{4}}$	0.625	0.93615564
x_6	x_3	$x_{1+\frac{1}{2}}$	0.75	0.90885168
x_7	$x_{3+\frac{1}{2}}$	$x_{1+\frac{3}{4}}$	0.875	0.87719257
x_8	x_4	x_2	1	0.84147098

分别由复合Trapz、Simpson、Cotes公式有

$$T_8 = \frac{1}{16} [f(0) + 2 \sum_{k=1}^7 f(x_k) + f(1)] = 0.94569086$$

精度最低

$$S_4 = \frac{1}{24} [f(0) + 4 \sum_{k=0}^3 f(x_{k+\frac{1}{2}}) + 2 \sum_{k=1}^3 f(x_k) + f(1)] = 0.94608331$$

精度次高

$$C_2 = \frac{1}{180} [7f(0) + \sum_{k=0}^1 [32f(x_{k+\frac{1}{4}}) + 12f(x_{k+\frac{2}{4}}) + 32f(x_{k+\frac{3}{4}})] \\ + 14 \sum_{k=1}^1 f(x_k) + 7f(1)] = 0.94608307$$

精度最高

原积分的精确值为 $I = \int_0^1 \frac{\sin x}{x} dx = 0.946083070367183$

7.3 外推原理和Romberg求积公式

一、外推原理：数值计算中的加速收敛方法。

数值计算中常利用一序列： $F_1, F_2, \dots, F_k, \dots$ 去逼近准确解 F ，然后在理论上给出 $\{F_k\}$ 收敛于 F 的误差估计。

一个有趣的问题：能否在截断误差估计的基础上，通过简易的方法，在 $\{F_k\}$ 基础上产生一个新序列 $\{F_k^*\}$ ，使 $\{F_k^*\}$ 比 $\{F_k\}$ 更快地逼近 F 呢？

——称**加速收敛技巧**

例 设 $f(x)$ 在 $x=0$ 处函数值为 $f(0)$ ，在很多情况下， $f(0)$ 无法求得，只能得到一函数值序列： $f(h)$ ， $f(h/2)$ ，...
($h>0$) 且 h 越小，计算难度越大。

于是问题出现：能否通过序列 $f(h)$ ， $f(h/2)$ ，...构造出一新序列，使其更快收敛于 $f(0)$ 呢？

在某种条件下，这是办得到的。

如利用Taylor展式：

$$f(h) = f(0) + hf'(0) + \frac{1}{2!}h^2 f''(0) + \frac{1}{3!}h^3 f'''(0) + \dots$$

$$f\left(\frac{h}{2}\right) = f(0) + \frac{h}{2} f'(0) + \frac{1}{2!} \left(\frac{h}{2}\right)^2 f''(0) + \frac{1}{3!} \left(\frac{h}{2}\right)^3 f'''(0) + \dots$$

若 $f'(0) \neq 0$ ，则 $f(h)$ ， $f(h/2)$ 逼近 $f(0)$ 的阶数都是 $O(h)$ ，
若令

$$\begin{aligned} f_1(h) &= 2f\left(\frac{h}{2}\right) - f(h) \\ &= f(0) - \frac{1}{4} h^2 f''(0) - \frac{1}{3!} \cdot \frac{3}{4} h^3 f'''(0) + \dots \end{aligned}$$

则当 $f''(0) \neq 0$ 时， $f_1(h)$ 逼近 $f(0)$ 的误差阶为 $O(h^2)$ ，故
序列 $f_1(h)$ ， $f_1(h/2)$ ， \dots ，可更快地收敛到 $f(0)$ 。

若再令

$$\begin{aligned} f_2(h) &= \frac{4}{3} f_1\left(\frac{h}{2}\right) - \frac{1}{3} f_1(h) \\ &= f(0) + \frac{1}{3!} \cdot \frac{1}{8} h^3 f'''(0) + \dots \end{aligned}$$

则当 $f'''(0) \neq 0$ 时, $f_2(h)$ 逼近 $f(0)$ 的误差阶为 $O(h^3)$, 故序列 $f_2(h), f_2(h/2), \dots$, 比 $f_1(h), f_1(h/2), \dots$, 收敛到 $f(0)$ 更快. 再构造 $f_3(h)$, 还可继续加速.

这种利用若干已算出的近似值作适当组合以求得更精确的近似值的加速收敛的方法称**外推算法**.