《地球物理计算方法》

第2章　数值积分

数值积分

1、机械求积

2、Newton-cotes积分公式

3、复化求积方法

4、Romberg求积公式

5、Gauss积分公式

6、数值微分

问题的提出

函数积分：

$$I=\int_a^b f(x)\mathrm{d}x.$$

若能求出被积函数$f(x)$的一个原函数$F(x)$，则定积分I能根据牛顿-莱布尼茨公式求出，即

$$\boxed{\int_a^b f(x)dx}=F(b)-F(a)$$

$$F'(x)=f(x)$$

函数积分面临的困难：

(1) 被积函数$f(x)$并不一定能够找到用初等函数的有限形式表示的原函数F(x)，例如：

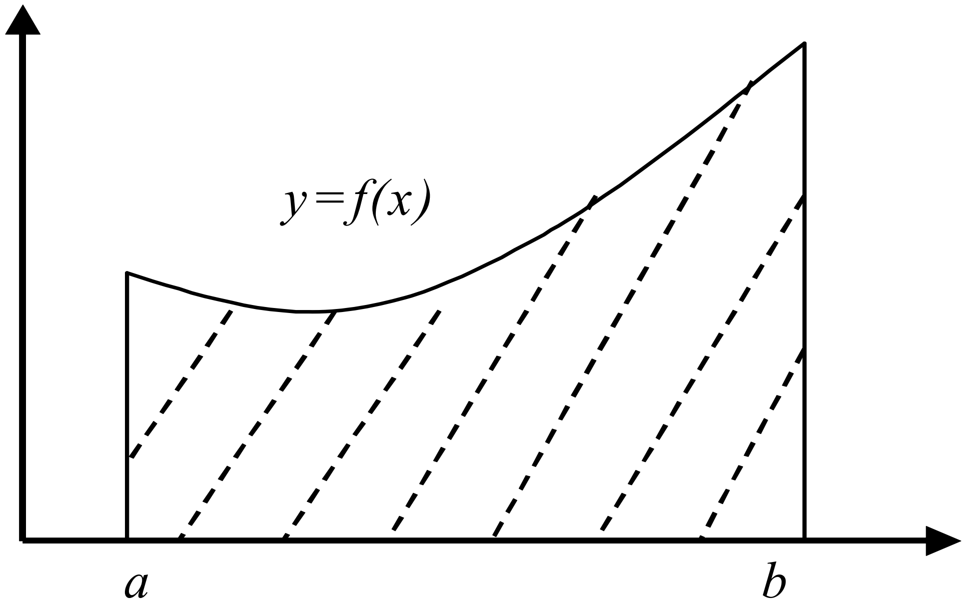
$$\int_0^1 \frac{\sin x}{x}dx \quad 和 \quad \int_0^1 e^{-x^2}dx$$

(2) 还有被积函数$f(x)$的原函数能用初等函数表示，但表达式太复杂，例如函数 $f(x)=x^2\sqrt{2x^2+3}$ ，其原函数F(x)表达式为：

$$F(x)=\frac{1}{4}x^3\sqrt{2x^2+3}+\frac{3}{16}x\sqrt{2x^2+3}-\frac{9}{16\sqrt{2}}\ln(\sqrt{2}x+\sqrt{2x^2+3})+C$$

函数积分面临的困难：

(3) 被积函数$f(x)$没有具体的解析表达式，其函数关系由表格或图形表示.



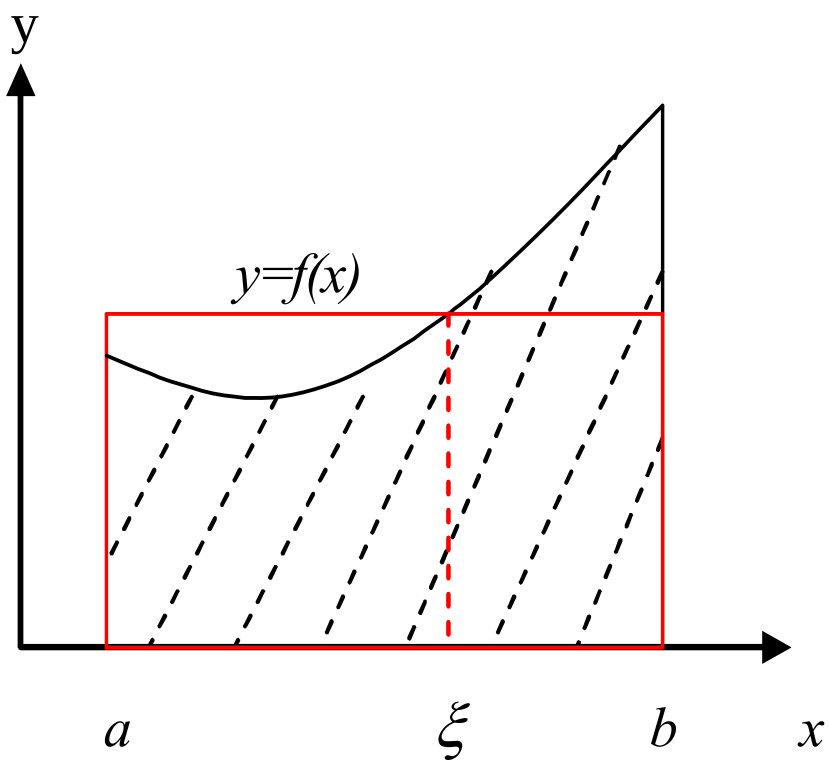
1、机械求积的概念

根据积分中值定理，函数积分：

平均高度

$$\int_a^b f(x)dx=(b-a)f(\xi),\quad \xi\in(a,b)$$

$f(\xi)$为区间$[a,b]$上的平均高度。只要对平均高度$f(\xi)$提供一种数值计算方法，便可以获得一种数值积分方法。



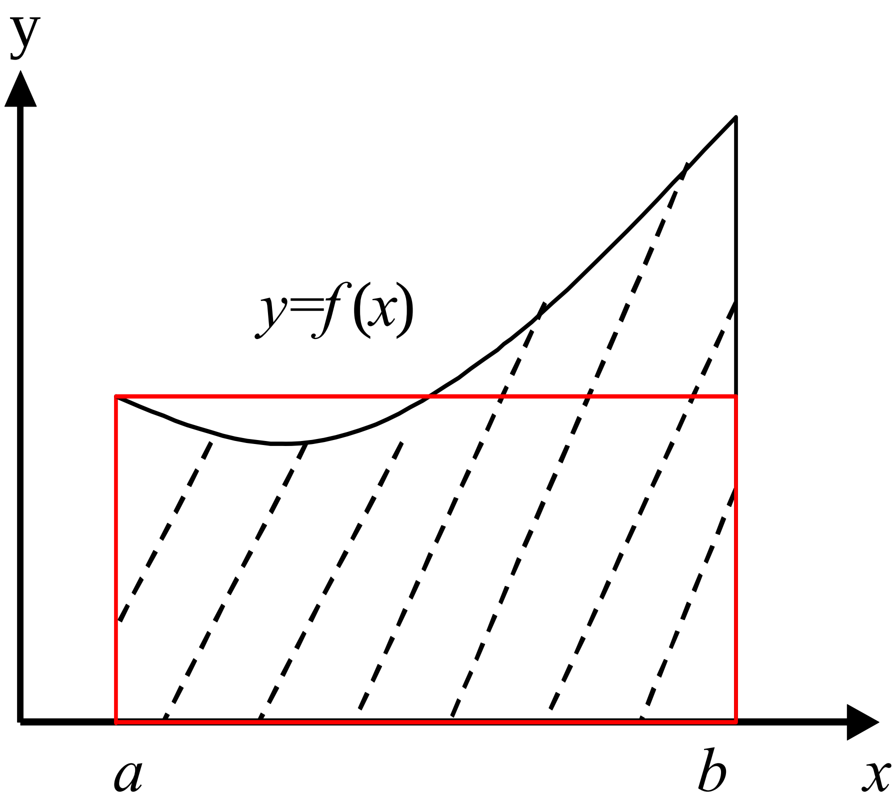
1、机械求积的概念

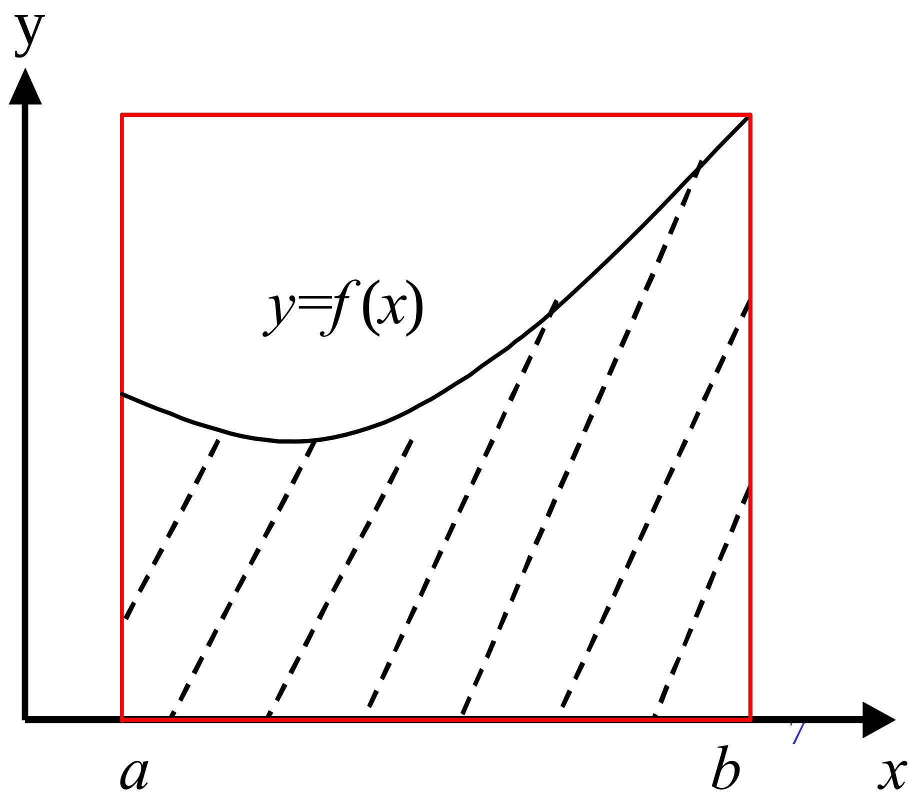
左矩形公式 $f(\xi) \approx f(a)$

$$\int_a^b f(x)dx \approx (b-a)f(a)$$

右矩形公式 $f(\xi) \approx f(b)$

$$\int_a^b f(x)dx \approx (b-a)f(b)$$



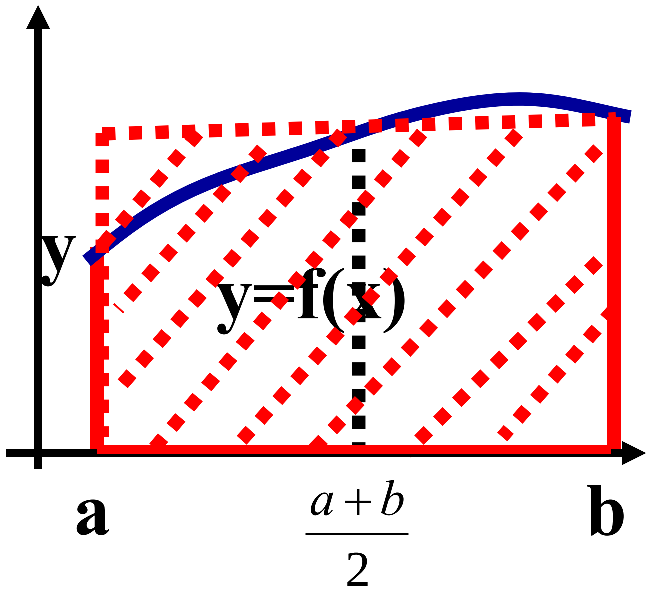


1、机械求积的概念

中矩形公式

$$f(\xi) \approx f(\frac{a+b}{2})$$

$$\int_a^b f(x)dx \approx (b-a) f(\frac{a+b}{2})$$

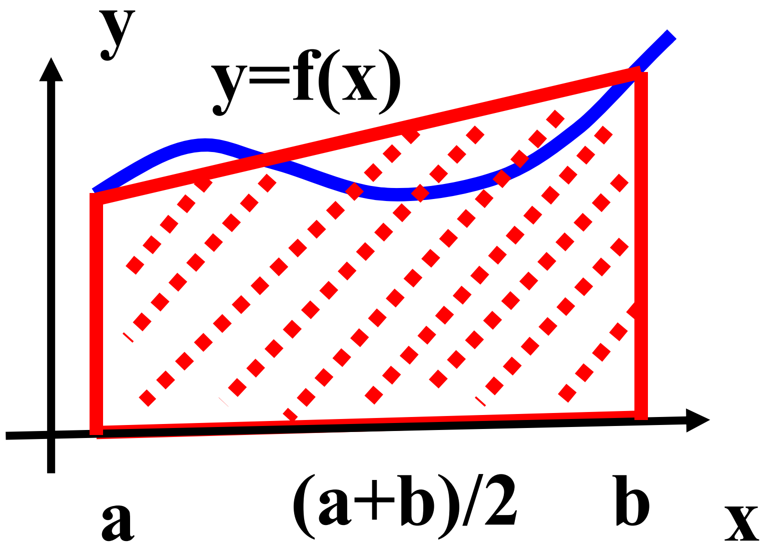


1、机械求积的概念

梯形公式

$$f(\xi) \approx \frac{f(a)+f(b)}{2}$$

$$\boxed{\int_a^b f(x)dx \approx \frac{1}{2}(b-a)\left[f(a)+f(b)\right]}$$

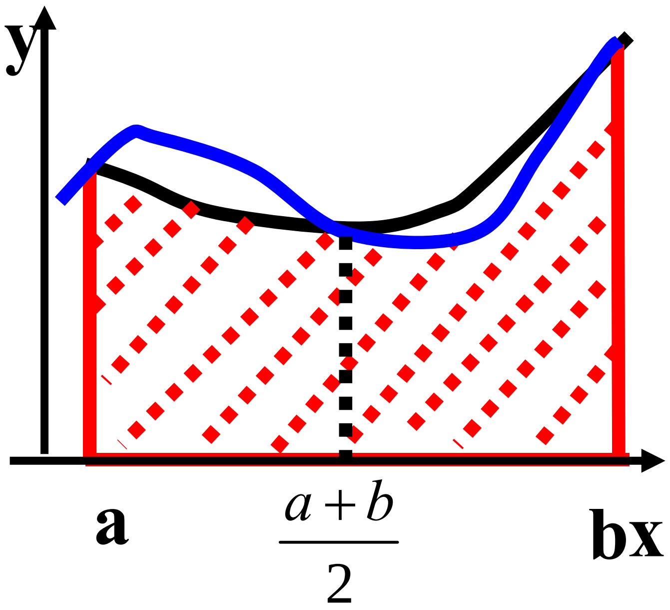


1、机械求积的概念

辛甫生（Simpson）公式

$$f(\xi)\approx\frac{1}{6}[f(a)+4f(\frac{a+b}{2})+f(b)]$$

$$\int_a^b f(x)dx\approx\frac{1}{6}(b-a)\left[f(a)+4f(\frac{a+b}{2})+f(b)\right]$$



1、机械求积的概念

机械求积公式：

特点：（1）求积系数A_k与节点x_k的选取有关，不依赖于被积函数$f(x)$的具体形式；(2)直接利用被积函数计算积分值，避免求原函数的困难。

一般地

$$\int_a^b f(x)dx \approx \sum_{k=0}^{n} A_k f(x_k)$$

求积节点

求积系数，即x_k的权

认识：

◆求积分用某些节点函数值的加权平均来计算$f(\xi)$ ；

◆求积系数A_k（权系数）,与函数没有关系，只与求积节点有关。

问题：

1．求积系数及节点如何确定？

2．此公式与插值多项式有何联系？

3．公式的计算精度如何判断？如何提高计算精度？

例 设积分区间[a, b]为[0, 2]，分别用梯形和Simpson公式

$$\int_0^2 f(x)dx \approx \frac{2-0}{2}[f(0)+f(2)]$$

$$\int_0^2 f(x)dx \approx \frac{2-0}{6}[f(0)+4f(1)+f(2)]$$

计算 $f(x)=1, x, x^2, x^3, x^4, e^x$ 时积分结果并与准确值进行比较。

解:梯形公式和Simpson公式的计算结果与准确值比较如下表所示

$f(x)$	1	x	x^2	x^3	x^4	e^x
准确值	2	2	2.67	4	6.40	6.389
梯形公式	2	2	4	8	16	8.389
辛甫生公式	2	2	2.67	4	6.67	6.421

2、代数精度的概念

定义：设求积公式 $\int_a^b f(x)dx \approx \sum_{k=0}^{n} A_k f(x_k)$

对于一切次数小于等于*m*的多项式是准确的，而对于次数为*m*+1的多项式是不准确的，则称该求积公式具有*m*阶代数精度（简称精度）

等价定义：设求积公式 $\int_a^b f(x)dx \approx \sum_{k=0}^{n} A_k f(x_k)$

对于 $1, x, x^2, \cdots, x^m$ **是准确的，而对于** x^{m+1} **是不准确的，则称该求积公式具有*m*阶代数精度。**

注：利用代数精度作为条件构建求积公式。

例　试构造求积公式 $\int_a^b f(x)dx \approx A_0 f(x_0) + A_1 f(x_1)$

使其具有1阶代数精度.

解：对于$f(x)$=1，x是否精确相等；

当$f(x)=1$时，左 $=\int_a^b 1dx = b-a$，右 $=(A_0+A_1)$

当$f(x)=x$时，左 $=\int_a^b xdx = \frac{1}{2}(b^2-a^2)$，右$=[A_0a+A_1b]$;

得到：　$A_0 = A_1 = \frac{b-a}{2}$;

例 若对于给定的一组求积节点 $x_k(k=0,1,2,\cdots,n)$ 相应的求积公式

$$\int_a^b f(x)dx \approx \sum_{k=0}^{n} A_k f(x_k)$$

至少具有n次代数精度，试确定其求积系数。

解 由已知对于 $f(x)=1, x, x^2, \cdots, x^n$，求积公式

均成立等式，得：$\int_a^b f(x)dx \approx \sum_{k=0}^{n} A_k f(x_k)$

解：对于$f(x)=1$，x，…，x^n均精确相等；令左=右

$$\begin{cases} A_0 + A_1 + \cdots + A_n = b - a \\ x_0 A_0 + x_1 A_1 + \cdots + x_n A_n = \dfrac{b^2 - a^2}{2} \\ \qquad\qquad \cdots \\ x_0^n A_0 + x_1^n A_1 + \cdots + x_n^n A_n = \dfrac{b^{n+1} - a^{n+1}}{(n+1)!} \end{cases}$$

其系数矩阵

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_0 & x_1 & \cdots & x_n \\ x_0^2 & x_1^2 & \cdots & x_n^2 \\ \vdots & \vdots & & \vdots \\ x_0^n & x_1^n & \cdots & x_n^n \end{bmatrix}$$

当$x_k\ (k=0,1,\ldots,n)$互异时非奇异，故A_k有唯一解.

在求积节点给定的情况下，求积公式的构造本质上是个解线性方程组的代数问题。

3、插值型求积公式

过函数$f(x)$的$n+1$节点x_0，x_1，……，x_n的函数值，作n次插值多项式函数 $p_n(x)$：

$$f(x) \approx p_n(x) = \sum_{k=0}^{n} f(x_k) l_k(x)$$

式中

$$l_k(x) = \prod_{\substack{j=0 \\ j \neq k}}^{n} \frac{x - x_j}{x_k - x_j}$$

由于多项式求积分容易，令

$$\int_a^b f(x)dx \approx \int_a^b p_n(x)dx$$

插值表示：

$$\sum_{k=0}^{n} A_k f(x_k) = \int_a^b \sum_{k=0}^{n} f(x_k) l_k(x) dx$$

$$\sum_{k=0}^{n} A_k f(x_k) = \sum_{k=0}^{n} f(x_k) \int_a^b l_k(x) dx$$

这样求解插值系数：$A_k = \int_a^b l_k(x) dx$

$$l_k(x) = \prod_{\substack{j=0 \\ j \neq k}}^{n} \frac{x - x_j}{x_k - x_j}, k = 0 \sim n$$

——插值型求积方法

插值型求积代数精度

定理　n+1个节点的求积公式

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} A_i f(x_i)$$

至少具有n次代数精度的充要条件是此公式为插值型求积公式.

数值积分

1、机械求积

2、Newton-cotes积分公式

3、复化求积方法

4、Romberg求积公式

5、Gauss积分公式

6、数值微分

Newton-cotes积分公式

把区间[a，b]分为**n等份**，步长为h（h=(b－a)/n）,则n+1个点（**取等分点**）分别为：$x_k=a+kh$, 由这n＋1个点构造的插值型求积公式为：

$$I_n \approx (b-a)\sum_{k=0}^{n} C_k f(x_k)$$

若此求积公式至少具有n阶的代数精度，则称此求积公式为**n阶的** **Newton-Cotes公式.**

1、公式的导出

积分公式的构造：

(1) 利用精度概念求积分系数

对于

$$\int_a^b f(x)dx \approx (b-a)\sum_{k=0}^{n} C_k f(x_k)$$

则此求积公式对于 $1, x, x^2, \ldots\ldots$ 应成立等式条件满足。

1、公式的导出

（2）利用插值公式求积分系数

$$C_k = \frac{1}{b-a}\int_a^b l_k(x)dx\,,\ k = 0,1,\cdots,n$$

$$C_k = \frac{1}{b-a}\int_a^b \prod_{\substack{j=0\\ j\neq k}}^{n} \frac{x-x_j}{x_k-x_j}dx \xrightarrow[\quad]{\substack{x=a+th\\ x-x_j=(t-j)h\\ x_k-x_j=(k-j)h}} \frac{1}{nh}\int_0^n h\prod_{\substack{j=0\\ j\neq k}}^{n}\frac{t-j}{k-j}dt$$

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

一般多项式积分

1、公式的导出

◆既然柯特斯系数跟被积函数、积分区间无关，只与代替被积函数的多项式次数n有关，就可以在求积分前预先算出来。

◆只要计算出柯特斯系数，就可以按下式计算积分（如n=4）

$$\int_a^b f(x)dx \approx \int_a^b p_4(x)dx = (b-a)\sum_{i=0}^{4} c_i^{(4)} f(x_i)$$

$$= (b-a)\left[c_0^{(4)} f(x_0) + c_1^{(4)} f(x_1) + c_2^{(4)} f(x_2) + c_3^{(4)} f(x_3) + c_4^{(4)} f(x_4)\right]$$

不同n的柯特斯系数表及代数精度

n	注:n为小区间个数 C_k	m
1	$\frac{1}{2}\{1,1\}$	1
2	$\frac{1}{6}\{1,4,1\}$	3
3	$\frac{1}{8}\{1,3,3,1\}$	3
4	$\frac{1}{90}\{7,32,12,32,7\}$	5
5	$\frac{1}{288}\{19,75,50,50,75,19\}$	5
6	$\frac{1}{840}\{41,216,27,272,27,216,41\}$	7
7	$\frac{1}{17280}\{751,3577,1323,2989,2989,1323,3577,751\}$	7
8	$\frac{1}{28350}\{989,5888,-928,10496,-4540,10496,-928,5888,989\}$	9

基本性质

1、 $\sum_{k=0}^{n} C_k = 1$

2、$C_{n-k} = C_k, k = 0 \sim n$, (P61公式10) 对称性；

3、当 $n \le 7$ 时，其 Newton—Cotes 系数为正；当 $n \ge 8$ 时，其 Newton—Cotes 系数有正、有负.

根据柯特斯系数表，当$n \ge 8$时, C_k出现负值, $N - C$公式不稳定.

牛顿-柯特斯公式

1、梯形公式

当n＝1时（即2个点，1等份），有梯形公式（1次代数精度）：

$$I_1 = \frac{b-a}{2}\left[f(a)+f(b)\right]$$

2、辛普生公式

当n＝2时（即3个点，2等份），有辛普生公式（3次代数精度）：

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

3. 柯特斯公式

当n＝4时（即5个点，4等份），有柯特斯公式（5次代数精度）

$$I_4 = \frac{b-a}{90}\left[7f\left(x_0\right)+32f\left(x_1\right)+12f\left(x_2\right)+32f\left(x_3\right)+7f\left(x_4\right)\right]$$

$$h=\frac{b-a}{4}, x_0=a, x_i=a+ih, i=0,1,2,3,4$$

其代数精度至少是**4;**

例:分别用梯形公式、辛甫(pu)生公式和柯特斯公式计算定积分 $\int_{0.5}^{1}\sqrt{x}\,\mathrm{d}x$ 的近似值(计算结果取5位有效数字)

(1) **用梯形公式计算**

$$\int_{0.5}^{1}\sqrt{x}\mathrm{d}x \approx \frac{1-0.5}{2}[f(0.5)+f(1)] = 0.25\times[0.70711+1] = 0.4267767$$

（2） **用辛甫生公式**

$$\int_{0.5}^{1}\sqrt{x}\,\mathrm{d}x \approx \frac{1-0.5}{6}[\sqrt{0.5}+4\times\sqrt{(0.5+1)/2}+\sqrt{1}]$$

$$=\frac{1}{12}\times[0.70711+4\times 0.86603+1]=0.43093403$$

（3）　用柯特斯公式计算，系数为

$$\frac{7}{90},\frac{32}{90},\frac{12}{90},\frac{32}{90},\frac{7}{90}$$

$$\int_{0.5}^{1}\sqrt{x}\mathrm{d}x \approx \frac{1-0.5}{90}[7\times\sqrt{0.5}+32\times\sqrt{0.625}+12\times\sqrt{0.75}+32\times\sqrt{0.875}+7\times\sqrt{1}]$$

$$=\frac{1}{180}\times[4.94975+25.29822+10.39223+29.93326+7]=0.43096407$$

积分的准确值为

$$\int_{0.5}^{1}\sqrt{x}\,\mathrm{d}x=\frac{2}{3}x^{\frac{3}{2}}\Big|_{0.5}^{1}=0.43096441$$

梯形	辛甫生	柯特斯
0.4267767	0.43093403	0.43096407

可见，三个求积公式的精度逐渐提高.

2、低阶求积公式的代数精度

定理　$n+1$个等距节点$x_k=a+kh\,,k=0\sim n$的牛顿-柯特斯公式

$$\int_a^b f(x)dx \approx (b-a)\sum_{k=0}^{n} C_k f(x_k)$$

至少具有n阶的精度；当n为偶数时，其精度可以达到$n+1$阶.

所以，n为偶数时的积分公式更加常用。

3、低阶求积公式的余项

牛顿-柯特斯公式是一个插值型数值求积公式，当用插值多项式$P_n(x)$代替$f(x)$积分时(代数精度为n)，其截断误差R(f)为：

$$R(f)=\int_a^b f(x)dx-\int_a^b p_n(x)dx=\int_a^b (f(x)-p_n(x))dx$$

$$R_n(x)=f(x)-p_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x)$$

其中 $\omega_{n+1}(x)=(x-x_0)(x-x_1)\cdots(x-x_n)=\prod_{i=0}^{n}(x-x_i)$

便可得出它的截断误差：

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

1、梯形公式

如果取n=1；用一次多项式代替被积函数(代数精度为1），

根据牛顿-柯特斯求积公式的误差理论式，梯形求积公式的误差估计为：

$$R_1(f)=\frac{1}{(2)!}\int_a^b f^{(2)}(\xi)\omega_2(x)dx=\frac{f^{(2)}(\xi)}{(2)!}\int_a^b (x-a)(x-b)dx=\frac{-(b-a)^3}{12}f''(\xi)$$

2、辛甫生公式

由于**辛甫生公式**n=2；该求积公式是用二次多项式逼近被积函数推得的，原则上它的代数精度为2。但因多项式次数是偶数，据前面的定理知道，它的代数精度为3。

根据牛顿-柯特斯求积公式的误差理论式，**辛甫生公式**求积公式的误差估计为：

$$R_2(x)=\frac{1}{4!}[-\frac{(b-a)^5}{120}f^{(4)}(\xi)]=-\frac{(b-a)^5}{2880}f^{(4)}(\xi),\xi\in[a,b]$$

1、机械求积

2、Newton-cotes积分公式

3、复化求积方法

4、Romberg求积公式

5、Gauss积分公式

6、数值微分

问题的提出

Newton-Cotes公式是取等距节点作为求积节点，随着求积节点的增多, 求积公式的代数精度会提高。

例：分别用梯形公式、辛甫生公式和柯特斯公式计算定积分 $\int_{0.5}^{1}\sqrt{x}\mathrm{d}x$ 的近似值（计算结果取5位有效数字）

（1） 用梯形公式计算

$$\int_{0.5}^{1}\sqrt{x}\mathrm{d}x \approx \frac{1-0.5}{2}[f(0.5)+f(1)] = 0.25\times[0.70711+1] = 0.4267767$$

（2） 用辛甫生公式

$$\int_{0.5}^{1}\sqrt{x}\mathrm{d}x \approx \frac{1-0.5}{6}[\sqrt{0.5}+4\times\sqrt{(0.5+1)/2}+\sqrt{1}]$$

$$=\frac{1}{12}\times[0.70711+4\times0.86603+1] = 0.43093403$$

（3）　用柯特斯公式计算，系数为

$$\frac{7}{90},\frac{32}{90},\frac{12}{90},\frac{32}{90},\frac{7}{90}$$

$$\int_{0.5}^{1}\sqrt{x}\mathrm{d}x \approx \frac{1-0.5}{90}[7\times\sqrt{0.5}+32\times\sqrt{0.625}+12\times\sqrt{0.75}+32\times\sqrt{0.875}+7\times\sqrt{1}]$$

$$=\frac{1}{180}\times[4.94975+25.29822+10.39223+29.93326+7]=0.43096407$$

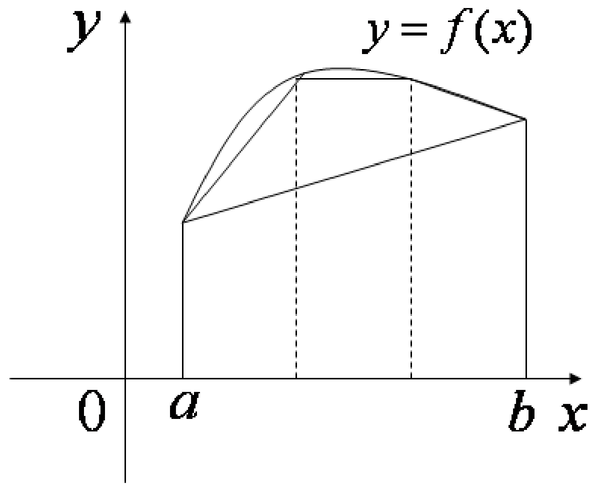
积分的准确值为

$$\int_{0.5}^{1}\sqrt{x}\mathrm{d}x=\frac{2}{3}x^{\frac{3}{2}}\Big|_{0.5}^{1}=0.43096441$$

梯形	辛甫生	柯特斯
0.4267767	0.43093403	0.43096407

可见，三个求积公式的精度逐渐提高.

问题：高次求积公式并不一定能取得好的效果，如：n≥8



办法：分段求积分→复化求积方法

将积分区间$[a,b]$划分为n等分，步长$h=\dfrac{b-a}{n}$，分点

$x_k=a+kh(k=0,1,\ldots,n)$，

由定积分性质知 $I=\int_a^b f(x)dx=\sum_{k=0}^{n-1}\int_{x_k}^{x_{k+1}} f(x)dx$

把积分区间$[a, b]$平均分成若干小区间$[x_k, x_{k+1}]$

第一步，在每个小区间上采用次数不高的Newton-Cotes求积公式，如梯形公式或辛甫生公式;

第二步，将每个区间上的近似积分值求和，用所得的值近似代替原积分值。

如此得到的求积公式称为复化求积公式。

复化求积公式可以克服高次Newton-Cotes公式计算不稳定的问题，运算简单且易于在计算机上实现。

1、复化求积公式

1. 复化梯形公式 T_n

$$I \approx \sum_{i=0}^{n-1} \frac{h}{2}\left[f(x_i) + f(x_{i+1})\right]$$

$$= \frac{h}{2}\left[f(x_0) + 2(f(x_1) + f(x_2) + \cdots + f(x_{n-1})) + f(x_n)\right]$$

$$= \frac{h}{2}\left[f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b)\right] = T_n$$

1、复化求积公式

2. 复化辛甫生公式 S_n

$$I=\sum_{k=0}^{n-1}\int_{x_i}^{x_{i+1}} f(x)dx \approx \sum_{i=0}^{n-1}\frac{h}{6}\left[f(x_i)+4f(x_{i+\frac{1}{2}})+f(x_{i+1})\right]$$

$$=\frac{h}{6}\left[f(a)+4\sum_{i=0}^{n-1}f(x_{i+\frac{1}{2}})+2\sum_{i=1}^{n-1}f(x_i)+f(b)\right]$$

$$S_n=\frac{h}{6}\left[f(a)+4\sum_{i=0}^{n-1}f(x_{i+\frac{1}{2}})+2\sum_{i=1}^{n-1}f(x_i)+f(b)\right]$$

其中 $x_{i+\frac{1}{2}}=x_i+\frac{1}{2}h$

1、复化求积公式

3.复化柯特斯公式

$$C_n = \frac{h}{90}\left[7f(a) + 32\sum_{i=0}^{n-1} f(x_{i+\frac{1}{4}}) + 12\sum_{i=0}^{n-1} f(x_{i+\frac{1}{2}})\right.$$

$$\left. +32\sum_{i=0}^{n-1} f(x_{i+\frac{3}{4}}) + 14\sum_{i=1}^{n-1} f(x_i) + 7f(b)\right]$$

其中， $x_{i+\frac{1}{4}}, x_{i+\frac{1}{2}} x_{i+\frac{3}{4}}$ 为子区间内的4等分点。

复化求积公式的余项

1. 复化梯形公式的余项

复化梯形公式T_n的余项

$$R = \mathrm{I} - T_{\mathrm{n}} = \sum_{\mathrm{i}=0}^{n\text{-}1}[-\frac{1}{12}h^3 f''(h_i)],\ h_i \in [x_i, x_{i+1}],$$

若$f''(x)$在$[a,b]$上连续，则

$$R = -\frac{n}{12}h^3 f''(\xi) = -\frac{b-a}{12}h^2 f''(\xi),\quad \xi \in [a,b].$$

$\therefore n \to \infty, h^2 \to 0, R \to 0$ 是收敛的.

复化求积公式的余项

2. 复化simpson公式的余项

余项 $R = I - S_n = -\dfrac{h}{180}\left(\dfrac{h}{2}\right)^4 \sum_{k=0}^{n-1} f^{(4)}(\eta_k),\ \eta_k \in (x_k, x_{k+1}),$

当$f \in C^4[a,b]$时,

$$R = I - S_n = -\frac{b-a}{180}\left(\frac{h}{2}\right)^4 f^{(4)}(\eta) = -\frac{b-a}{2880} h^4 f^{(4)}(\eta), \eta \in (a,b).$$

$$\therefore n \to \infty, h^4 \to 0, R \to 0$$

具有相应的收敛性和稳定性.

复化求积公式的截断误差

复化梯形公式的截断误差：

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

复化辛普生公式的截断误差：

$$R_S(f)=\sum_{k=1}^{m}-\frac{(2h)^5}{2880}mf^{(4)}(\xi)=-\frac{b-a}{180}h^4f^{(4)}(\xi)\quad \xi\in(a,b)$$

步长h越小，截断误差越小。

实例分析

例：

依次用n=8的复化梯形公式、n=4的复化Simpson公式计算定积分

$$I=\int_0^1 \frac{\sin x}{x}\mathrm{d}x$$

解:首先计算出所需各节点的函数值, n=8时， $h=\frac{1}{8}=0.125$

实例分析

1. 复化梯形公式得

$$T_8=\frac{1}{16}\Big[f(0)+2f(0.125)+2f(0.25)+2f(0.375)+2f(0.5)$$

$$+2f(0.625)+2f(0.75)+2f(0.875)+f(1)\Big]$$

$$=0.9456909$$

2. 复化Simpson公式得

$$S_4=\frac{1}{24}\Big[f(0)+f(1)+2(f(0.25)+f(0.5)+f(0.75))$$

$$+4(f(0.125)+f(0.375)+f(0.625)+f(0.875))\Big]$$

$$=0.9460832$$

实例分析

各种方法精度的比较：

比较： 准确值 I=0. 9460831

$$T_1 = 0.9270354 \qquad \boxed{T_8 = 0.9456909}$$

$$S_1 = 0.9461359 \qquad \boxed{S_4 = 0.9460832}$$

1、机械求积

2、Newton-cotes积分公式

3、复化求积方法

4、Romberg求积公式

5、Gauss积分公式

6、数值微分

1、梯形法的递推（变步长法）

复化积分可以提高精度，h->0时，I(h)=I。

复化积分存在的问题：

1、在使用求积公式之前必须先给出步长h;

2、h->0,$n \to \infty$,计算量太大。

解决问题的方法：

变步长计算，逐次二分，直到求得的积分值满足精度为止。同时继承前面已算出的积分值。

1、梯形法的递推（变步长法）

建立T_{2n}与T_n之间的递推关系：

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

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

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

得到递推关系：

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

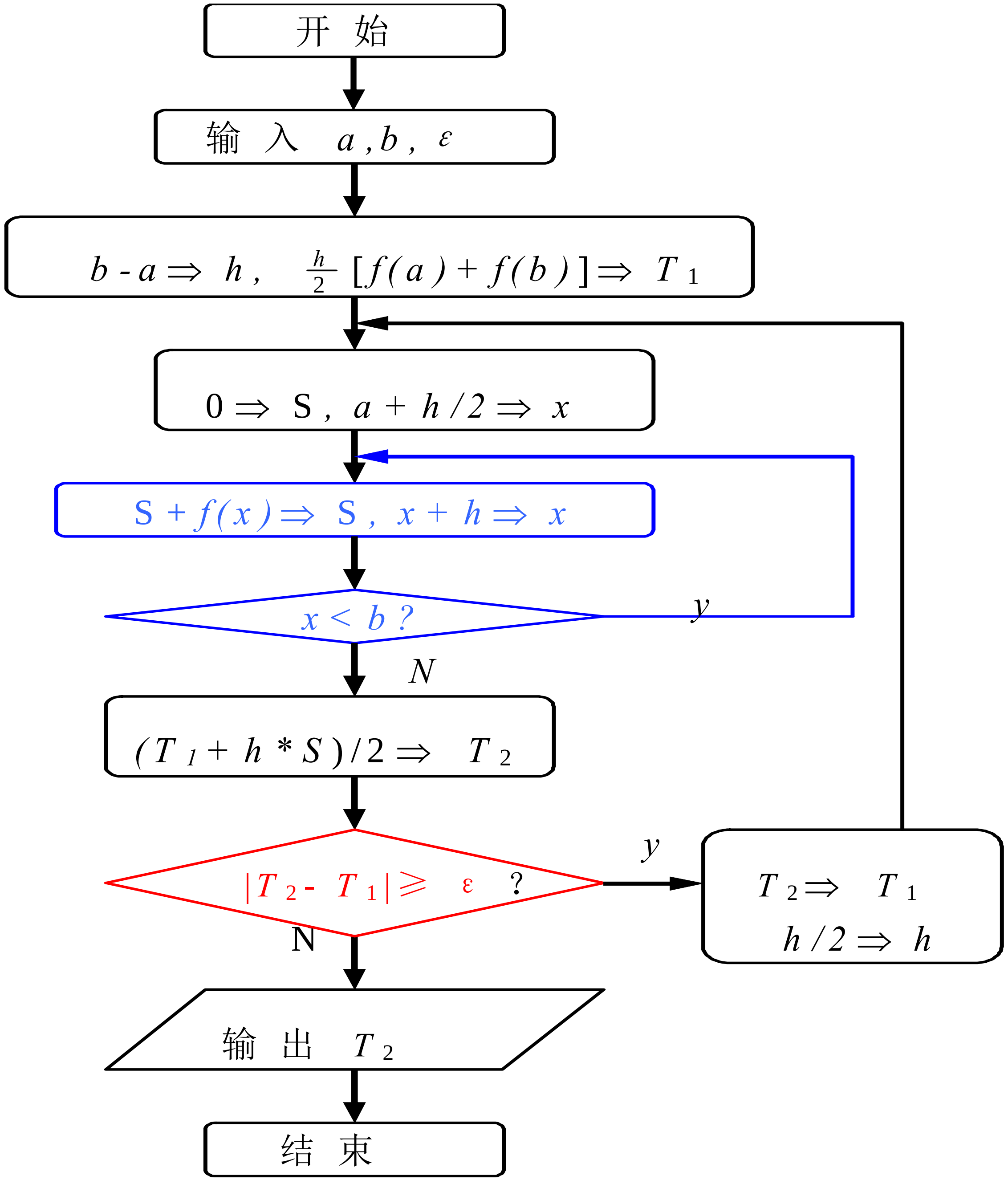
$$x_{k+\frac{1}{2}} = a + (k+\frac{1}{2})h$$

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

注：T_{2n}的全部节点中有n+1个是二分前的原有节点，可作为已知值使用，而它的后一项只涉及二分时新增加的节点$x_{k+1/2}$，该递推公式避免了旧节点的重复计算，节约了计算量。

计算流程图

变步长梯形法的流程图



实例分析

例 用变步长梯形求积法计算定积分 $I=\int_0^1 \frac{\sin x}{x}\mathrm{d}x$ ，$\varepsilon=10^{-2}$．

解：由

$$f(x)=\frac{\sin x}{x}, f(0)=1, f(1)=0.8410709$$

得

$$T_1=\frac{1}{2}[f(0)+f(1)]=0.9207355$$

二分一次

由于 $f(\frac{1}{2})=0.958851$，**故有**

$$T_2=\frac{1}{2}T_1+\frac{1}{2}f(\tfrac{1}{2})=0.9397933$$

$$|T_2-T_1|=0.019>\varepsilon$$

实例分析

二分二次

新分点上的函数值

$$f(\tfrac{1}{4}) = 0.9896158 \quad , f(\tfrac{3}{4}) = 0.9088516$$

$$T_4 = \frac{1}{2}T_2 + \frac{1}{4}\left[f(\tfrac{1}{4}) + f(\tfrac{3}{4})\right] = 0.9445135$$

$$\because \left|T_4 - T_2\right| < 0.005 < \varepsilon \quad , \qquad \therefore I \approx T_4 = 0.945$$

2、龙贝格公式

1、梯形法的加速

➢梯形法的算法简单，但精度低，收敛慢。

办法：可以采用进行外推算法，加工梯形值提高精度。

根据梯形法的误差公式，积分值T_n的误差与h^2成正比，因此步长减半后误差将减至1/4，即有

$$\frac{I - T_{2n}}{I - T_n} \approx \frac{1}{4}$$

移项整理，知

$$I - T_{2n} \approx \frac{1}{3}(T_{2n} - T_n)$$

这是T_{2n}的事后估计法，可以用这个误差来补偿数值结果；

$$\overline{T} = T_{2n} + \frac{1}{3}(T_{2n} - T_n) = \frac{4}{3}T_{2n} - \frac{1}{3}T_n$$

这个值正好是辛普生公式的结果。也可写为

$$S_n = (1+\omega)T_{2n} - \omega T_n \qquad \omega = \frac{1}{3}$$

2、 辛普生法的再加速

根据辛甫生法的误差公式，积分值T_n的误差与h^4成正比，因此步长减半后误差将减至1/16，即有

$$\frac{I-S_{2n}}{I-S_n}\approx\frac{1}{16}$$

移项整理，知 $I\approx\frac{16}{15}S_{2n}-\frac{1}{15}S_n$

这个值正好是柯特斯公式的结果。也可写为 $C_n\approx\frac{16}{15}S_{2n}-\frac{1}{15}S_n$

$$C_n=(1+\omega)S_{2n}-\omega S_n \quad \omega=\frac{1}{15}$$

3、Romberg公式

重复同样的手续，依据柯特斯法的误差公式可进一步导出下列龙贝格公式：

$$R_n = \frac{64}{63} C_{2n} - \frac{1}{63} C_n$$

这个公式也可写为，

$$R_n = (1+\omega)C_{2n} - \omega C_n \quad \omega = \frac{1}{63}$$

注：龙贝格求积公式也称为逐次分半加速法。它是在梯形公式、辛普森公式和柯特斯公式之间的关系的基础上，构造出一种加速计算积分的方法。 作为一种外推算法，它在不增加计算量的前提下提高了误差的精度。

一般递推公式:

设k次二分得到的m次加速值为$T_{\mathrm{m}}^{(k)}$，Romberg算法的一般公式为,

$$T_{\mathrm{m}}^{(k)}=\frac{4^m}{4^m-1}T_{m-1}^{(k+1)}-\frac{1}{4^m-1}T_{m-1}^{(k)}.$$

其中，k表示每个区间二分的次数，等分的区间总个数为$n=2^k,k=0,1,2,\ldots$;
m表示第几次加速，$m=1,2,\ldots$。

注：Romberg算法的代数精度为2m次，收敛阶为2(m+1)阶。

4、 Romberg算法的实现

计算流程见P70, 实现步骤:

① 用梯形公式计算积分近似值 $T_1=\frac{b-a}{2}[f(a)+f(b)]$

T_1

② 按变步长梯形公式计算积分近似值

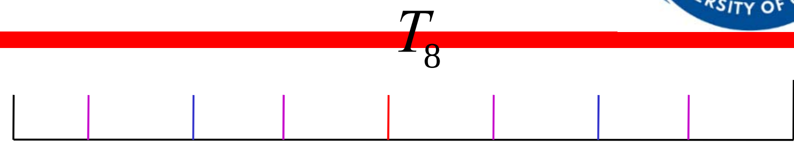
将区间逐次分半, 令区间长度 $h=\frac{b-a}{2^k}\quad(k=0,1,2,\cdots)$

T_2

计算

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

T_4



③ 按加速公式求加速值

梯形加速公式：
$$S_n = T_{2n} + \frac{T_{2n} - T_n}{3}$$

辛甫生加速公式：
$$C_n = S_{2n} + \frac{S_{2n} - S_n}{15}$$

龙贝格求积公式：
$$R_n = C_{2n} + \frac{C_{2n} - C_n}{63}$$

④ 精度控制

$$|\Delta| < \varepsilon$$ （其中ε为允许的误差）

Δ：表示二分前后的差。

龙贝格加速过程

	$\frac{\Delta}{3}$	$\frac{\Delta}{15}$	$\frac{\Delta}{63}$
T_1			
T_2	S_1		
T_4	S_2	C_1	
T_8	S_4	C_2	R_1
T_{16}	S_8	C_4	R_2

Δ：表示二分前后的差。

Romberg 算法:

① $T_1=T_0^{(0)}$ $\quad < \varepsilon ?$

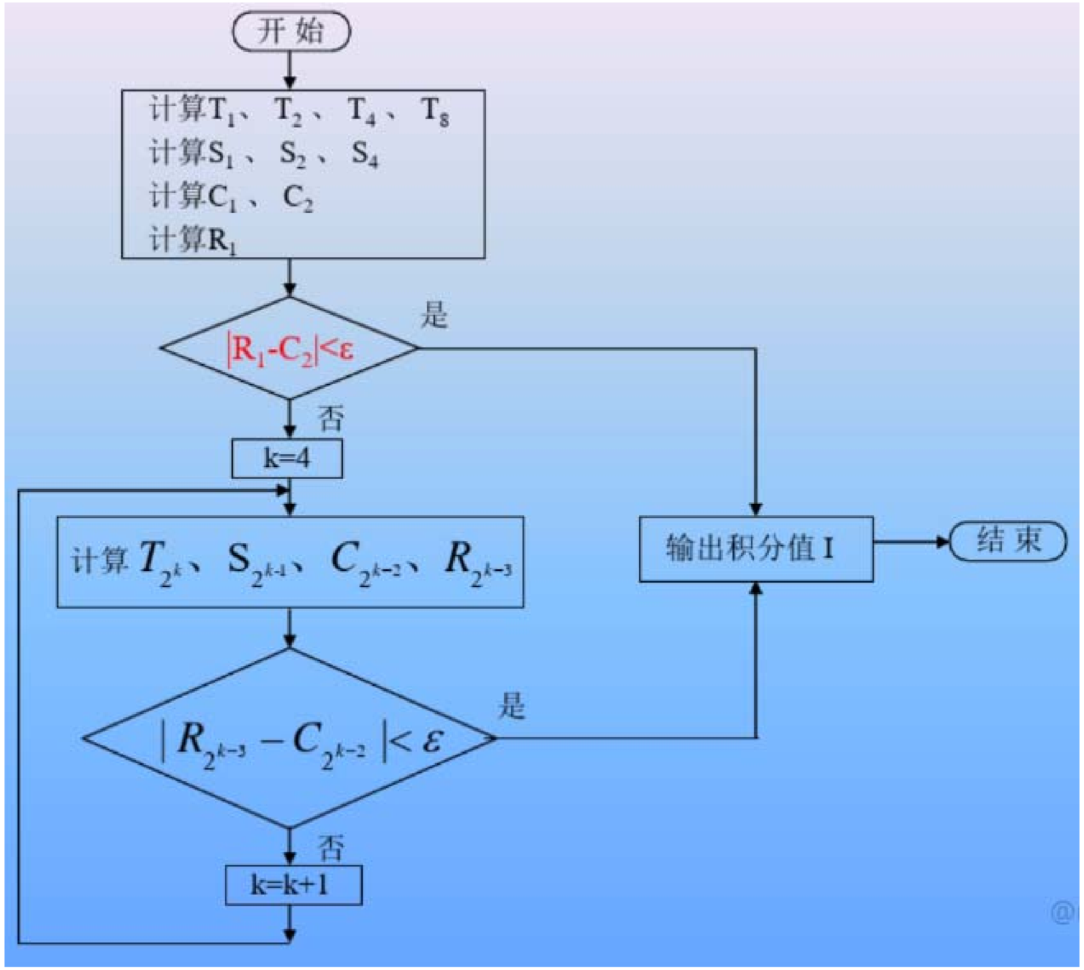
② $T_2=T_0^{(1)}$ ③ $S_1=T_1^{(0)}$ $\quad < \varepsilon ?$

④ $T_4=T_0^{(2)}$ ⑤ $S_2=T_1^{(1)}$ ⑥ $C_1=T_2^{(0)}$ $\quad < \varepsilon ?$

⑦ $T_8=T_0^{(3)}$ ⑧ $S_4=T_1^{(2)}$ ⑨ $C_2=T_2^{(1)}$ ⑩ $R_1=T_3^{(0)}$

••• ••• ••• ••• ••• •••

龙贝格算法流程图



开 始
计算T1、T2、T4、T8
计算S1、S2、S4
计算C1、C2
计算R1
|R1-C2|<ε
是
否
k=4
计算 T_{2^k}、$S_{2^{k-1}}$、$C_{2^{k-2}}$、$R_{2^{k-3}}$
$|R_{2^{k-3}} - C_{2^{k-2}}| < \varepsilon$
是
否
k=k+1
输出积分值 I
结 束

例　用Romberg算法计算定积分 $I=\int_0^1 \frac{\sin(x)}{x}\mathrm{d}x$

解:构造龙贝格算法表

k	T_{2^k}	S_{2^k}	C_{2^k}	R_{2^k}
0	0.9207355			
1	0.9397933	0.9461459		
2	0.9445135	0.9460869	0.9400830	
3	0.9456909	0.9460833	0.9460831	0.9460831