

Romberg 求积公式

目 录

1 一般性讨论	1
2 Romberg 求积算法	2
2.1 Richardson 外推加速方法	2
2.2 Romberg 算法	3

1 一般性讨论

Q : 待求量

$Q(h)$: 算法获得的 Q 的近似, h 为参数.

假设:

$$Q - Q(h) = c_1 h^{p_1} + c_2 h^{p_2} + c_3 h^{p_3} + \cdots + c_n h^{p_n} + \cdots \quad (1.1)$$

式中 $0 < p_1 < p_2 < \cdots < p_n < \cdots$

目的: 基于 (1.1) 构造高效算法.

在 (1.1) 中用 $h/2$ 替换 h , 有

$$Q - Q\left(\frac{h}{2}\right) = c_1 \left(\frac{h}{2}\right)^{p_1} + c_2 \left(\frac{h}{2}\right)^{p_2} + c_3 \left(\frac{h}{2}\right)^{p_3} + \cdots + c_n \left(\frac{h}{2}\right)^{p_n} + \cdots \quad (1.2)$$

用以上两式消去 h^{p_1} 项可得

$$Q = \frac{Q(h/2) - 2^{-p_1} Q(h)}{1 - 2^{-p_1}} + c_2^* h^{p_2} + c_3^* h^{p_3} + \cdots + c_n^* h^{p_n} + \cdots$$

记

$$Q_2 = \frac{Q(h/2) - 2^{-p_1} Q(h)}{1 - 2^{-p_1}}, \quad (1.3)$$

则

$$Q - Q_2(h) = \mathcal{O}(h^{p_2}),$$

$$Q - Q_1(h) = \mathcal{O}(h^{p_1}), \quad Q_1(h) = Q(h).$$

反复执行该过程有

$$\begin{cases} Q - Q_{k+1}(h) = \mathcal{O}(h^{p_{k+1}}), \\ Q_{k+1}(h) = \frac{Q_k(h/2) - 2^{-p_k} Q_k(h)}{1 - 2^{-p_k}}, \end{cases} \quad (1.4)$$

称 (1.3) 和 (1.4) 是基于 $Q(h)$ 的 Richardson 外推方法.

- 展开式 (1.1) 至关重要!
- Romberg 求积公式: 复化求积公式 + Richardson 外推方法

2 Romberg 求积算法

2.1 Richardson 外推加速方法

对 $[a, b]$ 进行 n 等分, $h = (b - a)/n$, $x_k = a + kh$,

$$a = x_0 < x_1 < \cdots < x_n = b,$$

记

$$T_1(h) = T(h) = h \left[\frac{f(x_0)}{2} + f(x_1) + \cdots + f(x_{n-1}) + \frac{f(x_n)}{2} \right].$$

注 2.1 需要注意的是, $T(h/2)$ 并不是简单地在上述式中把 h 换成 $h/2$, 而是在二分后的区间划分上做梯形公式.

对梯形公式, 根据 Euler-Maclaurin 公式, 我们有如下定理

定理 2.1 设 $f(x) \in C^\infty[a, b]$, 则

$$\int_a^b f(x)dx = T_1(h) + \sum_{k=1}^{\infty} c_{2k} h^{2k}.$$

上面的结果就是课本 P111 的定理 4.

- 视 $T_1(h)$ 为上一节的 $Q(h)$, 此时 $p_k = 2k$, 则可得

$$T_2(h) = \frac{T_1(h/2) - 2^{-p_1} T_1(h)}{1 - 2^{-p_1}} = \frac{T_1(h/2) - 2^{-2} T_1(h)}{1 - 2^{-2}} = \frac{4T_1(h/2) - T_1(h)}{3},$$

它就是课本 P111 的 $S(h)$.

- 类似地, 由 $T_2(h) = S(h)$ 出发可得

$$T_3(h) = \frac{T_2(h/2) - 2^{-p_2} T_2(h)}{1 - 2^{-p_2}} = \frac{T_2(h/2) - 2^{-4} T_2(h)}{1 - 2^{-4}} = \frac{16T_2(h/2) - T_2(h)}{15},$$

它就是课本中的 $C(h)$.

- 由 $T_3(h) = C(h)$ 出发可得

$$T_4(h) = R(h) = \frac{64T_3(h/2) - T_3(h)}{63}.$$

- 一般地,

$$\begin{aligned} T_{k+1}(h) &= \frac{T_k(h/2) - 2^{-p_k} T_k(h)}{1 - 2^{-p_k}} = \frac{T_k(h/2) - 2^{-2k} T_k(h)}{1 - 2^{-2k}} \\ &= \frac{4^k T_k(h/2) - T_k(h)}{4^k - 1} = \frac{4^k}{4^k - 1} T_k\left(\frac{h}{2}\right) - \frac{1}{4^k - 1} T_k(h) \end{aligned} \quad (2.5)$$

在第一节给出了误差 $Q - Q_{k+1}(h) = \mathcal{O}(h^{p_k+1})$, 对梯形公式则为

$$I - T_{k+1}(h) = \mathcal{O}(h^{p_k+1}) = \mathcal{O}(h^{2k+1}). \quad (2.6)$$

当 h 小于 1 时, 上面的递推公式明显地改善了计算精度. 我们把这个处理方法称为 Richardson 外推加速方法.

注 2.2 为了方便描述, 称 $T_2(h)$ 为步长 h 的梯形公式 $T_1(h) = T(h)$ 加速一次的加速值. 类似地, $T_{k+1}(h)$ 为梯形公式 $T_1(h) = T(h)$ 加速 k 次的加速值.

注 2.3 加速一次对应的步长 h 二等分了一次, 需要计算新的梯形值 $T(\frac{h}{2})$; 加速 k 次对应的步长 h 二等分了 k 次, 需要计算若干个梯形值 $T(\frac{h}{2^j})$, $j = 0, 1, \dots, k$. 正因为如此, 梯形公式的递推化是有必要的, 以避免重复计算.

注 2.4 值得注意的是, 加速公式可以从不同步长出发, 后面要说的 Romberg 算法是从最大步长 $h = b - a$ 出发的. 有些同学会提出疑问: 当区间长度大于 1, 根据误差满足的关系式 (2.6), 误差不是会越来越大吗? 课本上的公式 (4.11), 即 Romberg 求积算法就解决了这个问题.

2.2 Romberg 算法

现在我们来考虑注 2.4 所提出的问题. 想法很简单, 就是从二分后的更小的步长出发进行加速.

为了更清楚地看到过程, 以下分步说明. 为了方便, 记 $h_0 = h = [a, b]$, 二分一次后的步长记为 h_1 , 二分两次的记为 h_2 , 以此类推.

Step 1 当步长为 h_0 时, 从 $T(h_0)$ 出发, 且只加速 0 次, 记此时的梯形值为 T_0^0 .

Step 2 当步长为 h_1 时, 从 $T(h_1)$ 出发, 且只加速 1 次, 记梯形值和 1 次加速值为 T_0^1, T_1^1 (上标对应 h_1 , 下标是序号).

Step 3 当步长为 h_2 时, 从 $T(h_2)$ 出发, 且只加速 2 次, 记梯形值和 2 次加速值为 T_0^2, T_1^2, T_2^2 .

Step 4 依此类推, 当步长为 h_k 时, 从 $T(h_k)$ 出发, 且只加速 k 次, 记梯形值和 k 次加速值为 $T_0^k, T_1^k, \dots, T_k^k$.

注 2.5 显然越往后, 步长越小, 当步长为 h_k 时, 因迭代 k 次, 故误差为 $\mathcal{O}(h_k^{2k+1})$, 这个很小的. 当步长为 h 时, 加速公式为

$$T_m(h) = \frac{4^m}{4^m - 1} T_{m-1}\left(\frac{h}{2}\right) - \frac{1}{4^m - 1} T_{m-1}(h),$$

自然步长为 h_k 时, 加速公式为

$$T_m(h_k) = \frac{4^m}{4^m - 1} T_{m-1}\left(\frac{h_k}{2}\right) - \frac{1}{4^m - 1} T_{m-1}(h_k). \quad (2.7)$$

课本中用 $T_0^{(k)}$ 表示二分 k 次后的梯形值, 显然就是 $T_0^{(k)} = T(h_k)$. 而 $T_0^{(k)}$ 加速 m 次后的值记为 $T_m^{(k)}$, 即 $T(h_k)$ 加速 m 次后的值记为 $T_m^{(k)}$. 注意到, $T(h_k)$ 加速 m 次后的值就是 $T_{m-1}(h_k)$, 因此在课本的记号下, 加速公式为

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

由 (2.7), 我们就可以看到这种记号的根本原因, 只是把括号中的 h_k 的下标搬到 T 的右上角, 而 $\frac{h_k}{2}$ 用 $k+1$ 对应.

注 2.6 Romberg 算法迭代的停止准则是, 当 h_{k-1} 给出的加速终值与 h_k 给出的加速终值变化不大时, 停止迭代. 但课本在构造 T 表时又引入了新的记号, 其目的是保持同一行的上下指标和相同, 可以不按新记号记忆.

References

- [1] A. Demir, N. Omur, Y.T. Ulutas. Parametrized Fibonacci search method with k -Lucas numbers [J]. Applied Mathematics and Computaion, 2008, 198: 355–360.