第三章 数值积分与微分法

章节概述 所有的微积分$\int f(x)dx$核心问题在于求解原函数。数学原理：任意函数均有原函数。
应对两种情况： ① 函数用表格形式给出，不知道$f(x)$，无法直接求导或求积
② 函数的解析表达式结构复杂，不易求导或求积

3.1 Newton-cotes 求积公式

3.1.1 插值型求积公式

基本条件 设$f \in C[a,b]$连续，求$I=\int_a^b f(x)dx$的近似值。

问题简化 利用插值法构造

公式推导 **等分** 将$[a,b]$ n 等分，有**步长**：$h=\frac{b-a}{n}$，则**节点**：$x_k=a+kh\ (k=0,1,2,\dots,n)$

L 插值 有 $f(x)=\sum_{k=0}^{n} f(x_k)l_k(x)+R_n(x)$ 已知系数构造基函数。等号由于误差项存在。

回代 $I=\int_a^b f(x)dx=\sum_{k=0}^{n} f(x_k)\int_a^b l_k(x)dx+\int_a^b R_n(x)dx$ 积分满足交换律

导出 原本$f(x)$用插值法中$f(x_k)_{系数}l_k(x)_{基}$ 令 $A_k=\int_a^b l_k(x)dx=\int_a^b \prod_{\substack{j=0\\j\neq k}}^{n}\frac{x-x_j}{x_k-x_j}dx$ （标准多项式）

区间$[a,b]$是任意的，则该式也是任意的，如果写为 0、1 就好了。我们想要标准化以便于不同数量级数据合并计算。

则称$I_n=\sum_{k=0}^{n} A_k f(x_k)$ 为**插值型求积公式，A_k为求积系数**

余项 取$I\approx I_n$，有$R_n(I_n)=\int_a^b R_n(x)dx=\frac{1}{(n+1)!}\int_a^b f^{(n+1)}(\xi)\prod_{k=0}^{n}(x-x_k)dx \ \Rightarrow I=I_n+R_n(I_n)$

ξ是使用 n 次罗尔定理得到的，与x_k有关系。较难处理。

3.1.2 科茨系数（Cotes 系数）

概述 **为了计算I_n，只需求出A_k**

导出 记$x=a+th$，则$0\le t\le n$，$dx=hdt$，$x-x_j=(t-j)h$，$x_k-x_j=(k-j)h$ 代入上式：

$$A_k=\int_0^n h\prod_{\substack{j=0\\j\neq k}}^{n}\frac{(t-j)h}{(k-j)h}dt=\underbrace{\prod_{\substack{j=0\\j\neq k}}^{n}\frac{h}{k-j}}_{常数}\int_0^n\prod_{\substack{j=0\\j\neq k}}^{n}(t-j)dt=\frac{h(-1)^{n-k}}{\prod_{j=0}^{k-1}(k-j)\prod_{j=k+1}^{n}(j-k)}\int_0^n\prod_{\substack{j=0\\j\neq k}}^{n}(t-j)dt=\frac{h(-1)^{n-k}}{k!(n-k)!}\int_0^n\prod_{\substack{j=0\\j\neq k}}^{n}(t-j)dt$$

（不能为 k，所以分为两部分$\prod_{j=0}^{k-1}(k-j)=k!$，$\prod_{j=k+1}^{n}(j-k)=(n-k)!$，$(-1)^{n-k}$）

$$A_k=(b-a)\frac{(-1)^{n-k}}{n\cdot k!(n-k)!}\int_0^n\prod_{\substack{j=0\\j\neq k}}^{n}(t-j)dt=(b-a)C_k^{(n)}$$ 把区间$[a,b]$去除，做标准化过程

定义 记$C_k^{(n)}$为 cotes 系数，$k=0,1,2,\dots,n$ **该值可预先计算**

注意 ① $C_k^{(n)}$中 n为等分数，k 为节点下标，与区间无关

② $n=1$时，$C_0^{(1)}=-\int_0^1(t-1)dt=\frac{1}{2}$ $C_1^{(1)}=\int_0^1(t)dt=\frac{1}{2}$ 即为**梯形公式**

$n=2$时，$C_0^{(2)}=\frac{1}{4}\int_0^2(t-1)(t-2)dt=\frac{1}{6}$ $C_1^{(2)}=-\frac{1}{2}\int_0^2(t)(t-2)dt=\frac{4}{6}$ $C_2^{(2)}=\frac{1}{4}\int_0^2 t(t-1)dt=\frac{1}{6}$

其余可查 Cotes 系数表

③ $C_i^{(n)} = C_{n-i}^{(n)}$ 每一行中心对称，且求和为 1

④ $n \le 7$时，$C_i^{(n)}$均为正数；$n \ge 8$时，其有正有负 （龙格现象）

3.1.3 Newton-cotes 求积公式

定义 称$I_n = (b-a)\sum\limits_{k=0}^{n} C_k^{(n)} f(x_k)$ 为 Newdon-Cotes 公式。其中，$n = 1,2,4$尤为重要

梯形公式 $T = I_1 = \frac{b-a}{2}[f(a)+f(b)]$

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

Cotes 公式 $C = I_4 = \frac{b-a}{90}[7f(a)+32f(x_1)+12f(x_2)+32f(x_3)+7f(b)]$

3.1.4 代数精度

内容 衡量求积公式的精度。可以理解为评价标准

准确成立 若$I = \int_a^b f(x)dx = \sum\limits_{k=0}^{n} A_k f(x_k) = I_n$ ，则称求积公式对函数 f(x)能**准确成立**

定义 1.1 设有求积公式 $I_n = \sum\limits_{k=0}^{n} A_k f(x_k)$。若 $\forall 0 \le i \le m, \int_a^b P_i(x)dx = \sum_{k=0}^n A_k P_i(x_k)$

但$\int_a^b P_{m+1}(x)dx \ne \sum_{k=0}^n A_k P_{m+1}(x_k)$，则称求积公式具有$m$次代数精度（从 0 开始到达的最大精度）

注意 ① 求积公式具有m次代数精度 $\Leftrightarrow$ $\forall\, 0 \le i \le m$有$\int_a^b x^i dx = \sum_{k=0}^n A_k x^i$,但$\int_a^b x^{m+1}dx \ne \sum_{k=0}^n A_k x^{m+1}$

② 梯形, Simpson, Cotes 公式分别**具有1, 3, 5次（阶）代数精度**.

③ 若f为n次多项式，则$f^{(n+1)}(\xi) = 0$，故$I_n = \sum\limits_{k=0}^{n} A_k f(x_k)$**至少是**$n$次代数精度。

④ **若 n 为偶数**，则 Newdon-Cotes 公式**至少具有 n+1** 次代数精度

⑤ $\sum_{k=0}^n C_k^{(n)} = 1$

3.1.5 几种低阶 Newton-cotes 求积公式的积分余项

梯形公式 $R(T) = \frac{1}{2}\int_a^b f''(\xi_x)\omega_2(x)dx$，$\xi_x \in [a,b]$。由于$\omega_2(x) = (x-a)(x-b)$在$[a,b]$上不变号

所以**积分中值定理**得$R(T) = \frac{1}{2} f''(\eta)\int_a^b (x-a)(x-b)dx = -\frac{1}{90}\left(\frac{b-a}{2}\right)^3 f''(\eta)$，$\eta \in (a,b)$

Simpson 公式 $R(S) = -\frac{b-a}{180}\left(\frac{b-a}{2}\right)^4 f^{(4)}(\eta) = -\frac{1}{90}\left(\frac{b-a}{2}\right)^5 f^{(4)}(\eta)$，$\eta \in (a,b)$

Cotes 公式 $R(C) = -\frac{2(b-a)}{945}\left(\frac{b-a}{2}\right)^6 f^{(6)}(\eta) = \frac{8}{945}\left(\frac{b-a}{2}\right)^7 f^{(6)}(\eta)$, $\eta \in (a,b)$

3.1.6 数值求积公式的数值稳定性

因为相对而言，$C_k^{(n)}$(查表可得)和x_k(n 等分)均能准确计算。所以积分的舍入误差来自$f(x_k)$的计算。

规定 记$\bar{f}(x_k)$为$f(x_k)_{准确值}$的**计算值**。$\varepsilon_k = f(x_k) - \bar{f}(x_k)$ $\bar{I_n} = (b-a)\sum_{k=0}^n C_k^{(n)}\bar{f}(x_k)$

推导 考虑$I_n - \bar{I_n} = (b-a)\sum_{k=0}^n C_k^{(n)}\left(f(x_k)-\bar{f}(x_k)\right) = (b-a)\sum_{k=0}^n C_k^{(n)}\varepsilon_k$

当$C_k^{(n)}$ 全为正数时，$|I_n-\overline{I_n}|\le \max|\varepsilon_k|(b-a)$

推导：$|I_n-\overline{I_n}|=\left|(b-a)\sum_{k=0}^{n}C_k^{(n)}\varepsilon_k\right|=|b-a|\left|\sum_{k=0}^{n}C_k^{(n)}\varepsilon_k\right|\le(b-a)\sum_{k=0}^{n}|C_k^{(n)}\varepsilon_k|\le(b-a)\max|\varepsilon_k|\left|\sum C_k^{(n)}\right|=\max|\varepsilon_k|(b-a)$

当$C_k^{(n)}$存在变号时($n\ge 8$)，$\max|\varepsilon_k|(b-a)\sum_{k=0}^{n}\left|C_k^{(n)}\right|>(b-a)\max|\varepsilon_k|$ 将影响公式的稳定性。

所以高阶会显著影响稳定性。因此**一般实用的是低阶公式**。

3.2 复合求积分

3.2.1 复合求积公式

条件　**考虑$[a,b]$分为小区间，在每个小区间上用低阶公式，再将结果求和。**

设有$x=a+th\ (k=0,1,2,\dots,n)$　　$h=\frac{b-a}{n}$ 在区间$[x_k,x_{k+1}]$使用**梯形公式**

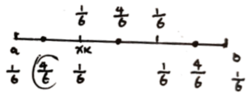
推导　$\int_{x_k}^{x_{k+1}} f(x)dx \approx \frac{h}{2}[f(x_k)+f(x_{k+1})]$

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

称其为**复合梯形公式**

$$T_n=\int_a^b f(x)dx \approx \frac{b-a}{2n}\left[f(a)+2\sum_{k=1}^{n-1} f(x_k)+f(b)\right]$$

同理　**复合型 Simpson 公式**：$S_n=\frac{b-a}{6n}\left[f(a)+2\sum_{k=1}^{n-1}f(x_k)+f(b)+4\sum_{k=0}^{n-1}f(x_{k+\frac{1}{2}})\right]$



复合 Cotes 公式：$C_n=\frac{b-a}{90n}\left[7f(a)+14\sum_{k=1}^{n-1}f(x_k)+7f(b)+32\sum_{k=0}^{n-1}f(x_{k+\frac{1}{4}})+12\sum_{k=0}^{n-1}f(x_{k+\frac{1}{2}})+32\sum_{k=0}^{n-1}f(x_{k+\frac{3}{4}})\right]$

例题　1. 依次使用$n=8$的复合梯形，$n=4$的符合 Simpson，$n=2$的复合 Cotes 计算定积分 $\int_0^1 \frac{\sin x}{x}dx$

首先计算所需各节点的函数值，再利用公式分别计算T_8,S_4,C_2

3.2.2 复合求积公式的余项及收敛的阶

1 复合梯形　**余项**：$I-T_n=-\frac{nh^3}{12}f''(\eta)=-\frac{h^2}{12}(b-a)f''(\eta)\approx-\frac{h^2}{12}[f'(b)-f'(a)]$ (n充分大时)

推导　$I-I_n=\sum_{k=0}^{n-1}\underbrace{\left(-\frac{h^3}{12}f''(\eta_k)\right)}_{\text{每个的误差项}}=-\frac{nh^3}{12}\underbrace{\frac{1}{n}\sum_{k=0}^{n-1}f''(\eta_k)}_{\text{期望}}$ 设在$[a,b]$上连续，

则由连续函数的中值定理知 $\exists\eta\in(a,b)$使得$\frac{1}{n}\sum_{k=0}^{n-1}f''(\eta_k)=f''(\eta)$

即$I-T_n=-\frac{nh^3}{12}f''(\eta)=-\frac{h^2}{12}(b-a)f''(\eta)$ 整体误差只取决于h^2,步长越小精度越高

又有$\lim\limits_{h\to0}\frac{I-T_n}{h^2}=\lim\limits_{h\to0}(-\frac{nh^3}{12}f''(\eta)h)=-\frac{1}{12}\int_a^b f''(x)dx=-\frac{1}{12}[f'(b)-f'(a)]$

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

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

注意：若f在$[a,b]$上具有 2，4，6 阶连续函数，则当$n\to\infty(h\to0)$时，$\boldsymbol{T_n,S_n,C_n\to I}$**,且速度一个比一个快**

收敛阶　公式收敛的速度

定义 2　设I_n为复合求积公式，若$\lim\limits_{h\to0}\frac{I-I_n}{h^p}=C\neq0$，则称求积公式是 p 阶收敛的

注意：T_n,S_n,C_n分别是 2，4，6 阶收敛的。

计算值的精度与$h(n)$有关，还与$f''(f(4)，f(6))$有关，估值困难，通常采用算后估值误差来调整步长.

3.2.3 步长的自动选择

原理 因为$I-T_n=-\frac{b-a}{12}h^2f''(\xi_1)$，$I-T_{2n}=-\frac{b-a}{12}h^2_{为左边一半}f''(\xi_2)$

其中$f''(\xi_1)=\frac{1}{n}\sum_{k=0}^{n-1}f''(\eta_i)$ (n 个 2 阶导数的平均值) $f''(\xi_2)=\frac{1}{2n}\sum_{k=0}^{2n-1}f''(\eta_i)$ (2n 个 2 阶导数的平均值)

大数定律：当n充分大时，$f''(\eta_2)\approx f''(\eta_1)$ 从而 $\frac{I-T_{2n}}{I-T_n}\approx\frac{1}{4}\Rightarrow I-T_{2n}\approx\frac{1}{3}(T_{2n}-T_n)$

左边是T_{2n}的误差，右边是$T_{2n}-T_n$两个已知值。由此，可以在未知I的情况下估计T_{2n}的绝对误差。

即以T_{2n}作为I 的近似值时，其误差约为$\boldsymbol{\Delta=\frac{1}{3}(T_{2n}-T_n)}$ 若规定精度误差限为ε，只需$\Delta\le\varepsilon$

步长折半法 先令$h=b-a$，计算T_1，步长折半后，计算T_2，则$\Delta=\frac{1}{3}(T_2-T_1)$. 若$\Delta\le\varepsilon$，则停止，取$I\approx T_2$

否则步长再次折半，计算T_4，$\Delta=\frac{1}{3}(T_4-T_2)$,……,**直到$\boldsymbol{\Delta\le\varepsilon}$为止**。最后得到的积分值即满足精度要求

同理，由$I-S_n=-\left(\frac{h}{2}\right)^4\frac{(b-a)}{180}f^{(4)}(\xi_2)$可以推出$\boldsymbol{I-S_{2n}\approx\frac{1}{15}(S_{2n}-S_n)}$ 用$\Delta=\frac{1}{15}|S_{2n}-S_n|\le\varepsilon$ 作为复合 Simpson 公式中控制步长的条件。

由$I-C_n=-\frac{2(b-a)}{945}\left(\frac{h}{2}\right)^6f^{(6)}(\xi_3)$可推出$\boldsymbol{I-C_{2n}\approx\frac{1}{63}(C_{2n}-C_n)}$,用$\Delta=\frac{1}{63}|C_{2n}-C_n|\le\varepsilon$作为复合 Cotes 公式中控制步长的条件。

例题 1. 用变步长的复合 Simpson 公式计算定积分$\int_0^1\frac{\sin x}{x}dx$，并给定误差限$\varepsilon=\frac{1}{2}\times10^{-6}$

先取$h=b-a=1,S_1=0.9461459$，再取$h=0.5,S_2=0.94608688,\Delta=\frac{1}{15}|S_1-S_2|=0.36*10^{-4}>\varepsilon$

继续$h=0.25,S_4=0.9460833,\Delta-\frac{1}{15}|S_4-S_2|=2.4*10^{-7}<\varepsilon$ 故$I\approx S_4=0.9460833$

注：在步长折半过程中，出现一批新节点，还有一点旧节点，将重复计算函数值.

3.3 Romberg 算法

3.3.1 复合梯形公式的递推化

目的　找到关系式$T_{2n} = T_n + F$,我们想要找到F

推导　在$[x_k, x_{k+1}]$上利用**梯形公式**：$\frac{h}{2}[f(x_k) + f(x_{k+1})]$，$h = \frac{b-a}{n} = x_{k+1} - x_k$，　$k = 0,1,2,\dots,n-1$

得到$T_n = \frac{h}{2}\sum_{k=0}^{n-1}[f(x_k) + f(x_{k+1})]$ 将$[x_k, x_{k+1}]$折半，设$\boldsymbol{x_{k+\frac{1}{2}}}$**为其中点**：$\left[x_k, x_{k+\frac{1}{2}}\right], \left[x_{k+\frac{1}{2}}, x_{k+1}\right]$

在$[x_k, x_{k+1}]$上积分近似值为$\frac{h_{旧}}{4}\left[f(x_k) + 2f\left(x_{k+\frac{1}{2}}\right) + f(x_{k+1})\right] \Rightarrow T_{2n} = \frac{h}{4}\sum_{k=0}^{n-1}\left[f(x_k) + 2f\left(x_{k+\frac{1}{2}}\right) + f(x_{k+1})\right]$

$= \frac{h}{4}\sum_{k=0}^{n-1}[f(x_k) + f(x_{k+1})] + \frac{h}{2}\sum_{k=0}^{n-1} f\left(x_{k+\frac{1}{2}}\right)$ 即$\boldsymbol{T_{2n} = \frac{1}{2}T_n + \frac{h}{2}\sum_{k=0}^{n-1} f(x_{k+\frac{1}{2}})}$ 新加入的所有点之和乘以当前步长

其中$h = \frac{b-a}{n}$　$x_{k+\frac{1}{2}} = a + \left(k + \frac{1}{2}\right)h = a + (2k+1)\frac{b-a}{2n}$，$k = 0,1,2,\dots,n-1$

所以　$T_1 = \frac{b-a}{2}[f(a) - f(b)]$

$$T_{2n} = \frac{1}{2}T_n + \frac{b-a}{2n}\sum_{k=0}^{n-1} f\left(a + (2k+1)\frac{b-a}{2n}\right),\ n = 2^0, 2^1, 2^2, \dots\dots$$

注意　为了编程方便，通常取$n = 2^{i-1}$记$T_{2n} = T_{2^i} \triangleq T_0(i)$，从而有

$$\begin{cases} T_0 = \frac{b-a}{2}[f(a) + f(b)] \\ T_0(i) = \frac{1}{2}T_0(i-1) + \frac{b-a}{2^i}\sum_{k=0}^{2^{i-1}-1} f\left(a + (2k+1)\frac{b-a}{2^i}\right) \end{cases} \quad i = 1,2,\dots$$

例题　1. 用递推的梯形公式计算$\int_0^1 \frac{\sin x}{x}dx$，误差限$\varepsilon = 10^{-3}$

误差估计$\Delta = \frac{1}{3}|T_0(k) - T_0(k-1)| \le \varepsilon$

取$I \approx T_0(3) = 0.94569086$，其与精确值$I = 0.946083$的误差绝对值为$3.924 * 10^{-4} < 10^{-3}$

k	x_j-新节点	$f(x_j)$	$T_0(k)$	Δ
0	0	1.00000000	0.92073549	
	1	0.84147098		
1	0.5	0.95885108	0.93979328	6.352×10^{-3}
2	0.25	0.98961584	0.94451352	1.573×10^{-3}
	0.75	0.90885168		
3	0.125	0.99739788	0.94569086	3.924×10^{-4}
	0.375	0.97672674		
	0.625	0.93615564		
	0.875	0.94569087		

3.3.2 外推加速公式

梯形、Simpson、Cotes 公式之间的关系　$\boldsymbol{S_n = \frac{4}{3}T_{2n} - \frac{1}{3}T_n}$　　$\boldsymbol{C_n = \frac{16}{15}S_{2n} - \frac{1}{15}S_n}$

推导　由$I - T_{2n} \approx \frac{1}{3}(T_{2n} - T_n)$可知$I \approx \frac{4}{3}T_{2n} - \frac{1}{3}T_n = \frac{4T_{2n} - T_n}{3}$

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

$= \frac{h}{2}\left[f(a) + 2\sum_{k=1}^{n-1} f(x_k) + f(b) + 4\sum_{k=0}^{n-1} f\left(x_{k+\frac{1}{2}}\right)\right] = 3S_n \Rightarrow S_n = \frac{4}{3}T_{2n} - \frac{1}{3}T_n$

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

外推加速公式　称上两式为外推加速公式，将精度较低的公式加工成精度较高的公式

Romberg 算法　进一步推广：$\boldsymbol{R_n = \frac{64}{63}C_{2n} - \frac{1}{63}C_n}$ **称$\{R_n\}$为龙贝格序列**

注意　① $\{T_n\}$、$\{S_n\}$、$\{C_n\}$和$\{R_n\}$的收敛阶依次为 2, 4, 6, 8

② 设$2n = 2^i$,记$T_1(k-1) = S_{2^{k-1}} = S_n$ 则有 $T_1(k-1) = \frac{4}{3}T_0(k) - \frac{1}{3}T_0(k-1)$

记$T_2(k-1)=C_{2^{k-1}}=C_n$ 则有 $T_2(k-1)=\frac{16}{15}T_1(k)-\frac{1}{15}T_1(k-1)$

记$T_3(k-1)=R_{2^{k-1}}=R_n$ 则有 $T_3(k-1)=\frac{64}{63}T_2(k)-\frac{1}{63}T_2(k-1)$

统一表为$T_m(k-1)=\frac{4^m T_{m-1}(k)-T_{m-1}(k-1)}{4^m-1}, m=1,2,3\ldots$

③ 利用外推法，将粗糙近似值逐步加工成精细近似值的方法称为 Romberg 算法

④ 理论上可继续加速下去，但当m比较大时,$4^m-1\approx 4^m$，此时有$T_m(k-1)\approx T_m(k)$,已无必要

⑤ **变步长的两种方法**:

$\Delta=\frac{1}{4^4-1}|T_3(k)-T_3(k-1)|\le|T_3(k)-T_3(k-1)|<\varepsilon$ $\quad\quad\Delta=|T_m(0)-T_{m-1}(0)|<\varepsilon$

⑥ **计算过程使用 T 数表**:

k	$T_0(k)$	$T_1(k\text{-}1)$	$T_2(k\text{-}2)$	$T_3(k\text{-}3)$
0	$T_0(0)$			
1	$T_0(1)$	$T_1(0)$		
2	$T_0(2)$	$T_1(1)$	$T_2(0)$	
3	$T_0(3)$	$T_1(2)$	$T_2(1)$	$T_3(0)$

由最简单的梯形公式可计算到最终值

例题 1. 使用 Romberg 算法计算$\int_0^1\frac{\sin x}{x}dx$，误差限$\varepsilon=10^{-7}$

k	$T_0(k)$	$T_1(k\text{-}1)$	$T_2(k\text{-}2)$	$T_3(k\text{-}3)$	
0	0.92073549				
1	0.93979328	0.94614588			
2	0.94451352	0.94608693	0.946083		
3	0.94569086	0.94608331	0.94608306	0.94608307	6. 2081E-08

3.4 数值微分

3.4.1 插值型求导公式

根据函数在给定节点的函数值来推算其**导数的近似值**. 方法：利用插值，样条先得到近似函数.

条件 设$f(x_i)=f_i, i=0,1,2,\ldots,n$ $P_n(x)$为$f(x)$的插值多项式

则$f(x)=P_n(x)+R_n(x)\Rightarrow f'(x)=P_n'(x)+R_n'(x)$

定义 称$f'(x)\approx P_n'(x)$ 为**插值型求导公式**

推导 设$P_n(x)=L_n(x)$ 则$R_n(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x)$ $R_n'(x)=\frac{1}{(n+1)!}\left[f^{(n+1)}(\xi_x)'\omega_{n+1}(x)+f^{(n+1)}(\xi_x)\,\omega_{n+1}'(x)\right]$

由于$f^{(n+1)}(\xi_x)'$难以确定，故只考虑$x=x_k, k=0,1,2,\ldots,n$的情况 只考虑间断点已知情况，函数值无误差

设$f'(x)=L_n'(x)+R_n'(x)$ 只考虑$x=x_k, k=0,1,2,\ldots,n$的情形 导数值仍存在误差

已知有$\prod_{j=0}^n\left((x-x_j)\right)'_{x=x_k}=\prod_{\substack{j=0\\j\ne k}}^n(x_k-x_j)$ 则:

$$f'(x_k)=L_n'(x_k)+\frac{f^{(n+1)}(\xi_x)}{(n+1)!}\left(\prod_{j=0}^n(x-x_j)\right)'_{x=x_k}$$

取$f'(x)\approx L_n'(x)$，误差为$E_n(x_k)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{\substack{j=0\\j\ne k}}^n(x_k-x_j)$

注：由于高阶插值不稳定，实际采用 n=1, 2, 4.分别称为二点，三点，五点公式.

两点公式（$n=1$） $L_1'(x)=\left(\frac{x-x_1}{x_0-x_1}\right)'f_0+\left(\frac{x-x_0}{x_1-x_0}\right)'f_1=\frac{f_1-f_0}{x_1-x_0}$ $\quad E_1(x_k)=\frac{f''(\xi)}{2}\prod_{\substack{j=0\\j\ne k}}^{1}(x_k-x_j), k=0,1$

$$\xRightarrow{h=x_1-x_0}\begin{cases}f'(x_0)=\frac{1}{h}(f_1-f_0)-\frac{h}{2}f''(\xi)\\ f'(x_1)=\frac{1}{h}(f_1-f_0)+\frac{h}{2}f''(\xi)\end{cases}$$

称为**带余项的两点(数值求导)公式**

计算用式 $f'(x_0)\approx\frac{1}{h}(f_1-f_0)\approx f'(x_1)$

等距节点下的三点公式与五点公式

① $\boldsymbol{n=2\quad x_k=x_0+kh,\ k=0,1,2}$

$$L_2'(x)=f_0l_0'(x)+f_1l_1'(x)+f_2l_2'(x)\qquad l_0'(x)=\left(\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}\right)'=\frac{x-x_2+x-x_1}{(x_0-x_1)(x_0-x_2)}=\frac{2x-x_1-x_2}{2h^2}$$

$$l_1'(x)=\left(\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}\right)'=\frac{x-x_2+x-x_0}{(x_1-x_0)(x_1-x_2)}=\frac{2x-x_0-x_2}{-h^2}\qquad l_2'(x)=\left(\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}\right)'=\frac{x-x_0+x-x_1}{(x_2-x_0)(x_2-x_1)}=\frac{2x-x_0-x_1}{2h^2}$$

$$\Longrightarrow\begin{cases}f'(x_0)=\frac{1}{2h}(-3f_0+4f_1-f_2)+\frac{h^3}{3}f'''(\xi)\\ f'(x_1)=\frac{1}{2h}(-f_0+f_2)-\frac{h^6}{6}f'''(\xi)\quad \text{中点公式}\\ f'(x_2)=\frac{1}{2h}(f_0-4f_1+3f_2)+\frac{h^2}{3}f'''(\xi)\end{cases}$$

用当前点的左右两值计算当前导数

称为**带余项的三点(数值求导)公式**，其中**中点公式**较其余二式简单且精度高

有x_0,x_1，假设步长为 1，则$f(x_0)-f(x_1)\approx f(x_1)-f(x_0)\approx f'(x_{0.5})$ 实际情况下，得到的是中点，有半个单位误差。如果仍然使用普通方法，则二阶导有一个单位的误差。因此，若第一次向右取点，则第二次要向左取点。这样可以抵消误差。如果算法要迭代求解，尽量不要直接使用库中的标准化函数。

② $\boldsymbol{n=4}$

$$\begin{cases}f'(x_0)=\frac{1}{12h}(-25f_0+48f_1-36f_2+16f_3-3f_4)+\frac{h^4}{5}f^{(5)}(\xi)\\ f'(x_1)=\frac{1}{12h}(-3f_0-10f_1+18f_2-6f_3+f_4)-\frac{h^4}{20}f^{(5)}(\xi)\\ f'(x_2)=\frac{1}{12h}(f_0-8f_1+8f_2-f_4)-\frac{h^4}{30}f^{(5)}(\xi)\\ f'(x_3)=\frac{1}{12h}(-f_0+6f_1-18f_2+10f_3+3f_4)-\frac{h^4}{20}f^{(5)}(\xi)\\ f'(x_4)=\frac{1}{12h}(3f_0-16f_1+36f_2-48f_3+25f_4)+\frac{h^4}{5}f^{(5)}(\xi)\end{cases}$$

称为**带余项的五点(数值求导)公式**

注意 ① 对于给定数值表,一般在所求导数节点的**两侧各选两个节点（居中原则）**，若一侧不足，则从另一侧补充.

② 为方便上机，将五点公式写为一般形式，$f'(x_k)=\frac{1}{12h}\sum_{j=i}^{i+4}A_j^{(k)}f_j\quad(k=i,i+1,i+2,i+3,i+4)$

$$\text{由}k\text{求}i:\ i=\begin{cases}0 & k<2\\ k-2 & 2\le k\le n-2\\ n-4 & k>n-2\end{cases}$$

步长选择 m 点公式的一般形式：$f'(x_k)=\frac{1}{C^{(m)}h}\sum_{j=i}^{i+m-1}A_{jm}^{(k)}f(x_j)+\frac{h^{m-1}}{b_k^{(m)}}f^{(m)}(\xi)$

当$h\to 0$时，截断误差为$\frac{h^{m-1}}{b_k^{(m)}}f^{(m)}(\xi)\to 0$ 但近似值的**舍入误差可能增大**，所以 h 的选择要兼顾两者

例题 1. 已知$f(x)=\ln x$利用两点公式计算$f'(1.8)$的近似值. 依次取$h=10-n(n=0,1,2,\dots,6)$.

由两点公式$f'(1.8)\approx\frac{1}{h}(f(1.8+h)-f(1.8))$ 截断误差：$|E(1.8)|\le\frac{h}{2}\max_{1.8\le x\le 1.8+h}|f''(x)|=\frac{h}{2\times 1.8^2}=\frac{h}{6.48}$

h	f(1.8)	E(1.8)	相邻两次计算的差
1	0.4418328	1.54321×10^{-1}	9.884×10^{-2}
0.1	0.5406722	1.54321×10^{-2}	1.335×10^{-2}
0.01	0.5540180	1.54321×10^{-3}	1.382×10^{-3}
0.001	0.5554000	1.54321×10^{-4}	1.491×10^{-4}
0.0001	0.5555491	1.54321×10^{-5}	9.486×10^{-5}
0.00001	0.555644	1.54321×10^{-6}	8.11×10^{-4}
0.000001	0.556455	1.54321×10^{-7}	

2. 利用三点公式求各节点处的数值导数

i	0	1	2	3	4	5
x_i	0.1	0.2	0.3	0.4	0.5	0.6
$f(x_i)$	1.2051709	1.4214028	1.6498588	1.8918247	2.1487213	2.4221188

h=0.1，两端点处的导数值 f '(x0), f '(x5)用三点公式，其余各节点用中点公式.

$f'(x_0) \approx \frac{1}{2h}(-3f_0 + 4f_1 - f_2)$ $\quad f'(x_5) \approx \frac{1}{2h}(f_3 - 4f_4 + 3f_5)$ $\quad f'(x_i) \approx \frac{1}{2h}(-f_{i-1} + f_{i+1})\ (i = 1,2,3,4)$

i	0	1	2	3	4	5
x_i	0.1	0.2	0.3	0.4	0.5	0.6
$f(x_i)$	1.2051709	1.4214028	1.6498588	1.8918247	2.1487213	2.4221188
$f'(x_i)$	2.1011985	2.2234395	2.3521095	2.4943125	2.6514705	2.8164795

3.4.2 样条求导公式（未教授）

引入 插值求导公式只能求节点处的导数，若求其他点的导数，就无能为力了. 下面考虑样条函数:

设定 $S^{(i)}(x) \approx f^{(i)}(x)\ (i = 0,1,2)\ \ \forall x \in [a,b]$ 设$f \in [a,b]$，为简化公式，取等距节点:

$x_j = a + jh \ \ (j = 0,1,2,\dots,n) \ \ h = \frac{b-a}{n}$

1 求节点导数近似值m_j

由$\begin{bmatrix} \lambda_1 & 2 & \mu_1 & & & & \\ & \lambda_2 & 2 & \mu_2 & & & \\ & & \dots & \dots & \dots & & \\ & & & & \lambda_{n-1} & 2 & \mu_{n-1} \end{bmatrix}\begin{bmatrix} m_0 \\ m_1 \\ \dots \\ m_n \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ \dots \\ g_{n-1} \end{bmatrix}$得 $\lambda_j = \mu_j = \frac{1}{2}\ \ \widetilde{g_j} = \frac{3}{2h}(f_{j+1} - f_j)\ (j = 1,2,\dots,n-1)$

进一步可得$\frac{1}{2}m_{j-1} + 2m_j + \frac{1}{2}m_{j+1} = \widetilde{g_j} = \frac{3}{2h}(f_{j+1} - f_j) \Rightarrow\ m_{j-1} + 4m_j + m_{j+1} = \frac{3}{h}(f_{j+1} - f_j)$

所以有 $\begin{bmatrix} 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & \dots & \dots & \dots & & \\ & & & & 1 & 4 & 1 \end{bmatrix}\begin{bmatrix} m_0 \\ m_1 \\ \dots \\ m_n \end{bmatrix} = \begin{bmatrix} 2\widetilde{g_1} \\ 2\widetilde{g_2} \\ \dots \\ 2\widetilde{g_{n-1}} \end{bmatrix}$ 设有边界条件: $m_0 = f_0'\ \ m_n = f_n'$ 则

$m_0 + 4m_1 + m_2 = \frac{3}{h}(f_2 - f_0) \Rightarrow 4m_1 + m_2 = \frac{3}{h}(f_2 - f_0) - f_0'$

$m_{n-2} + 4m_{n-1} + m_n = \frac{3}{h}(f_n - f_{n-2}) \Rightarrow 4m_{n-2} + m_{n-1} = \frac{3}{h}(f_n - f_{n-2}) - f_n'$

$\Rightarrow \begin{bmatrix} 4 & 1 & & & & \\ 1 & 4 & 1 & & & \\ & \dots & \dots & \dots & & \\ & & & 1 & 4 & 1 \\ & & & & 1 & 4 \end{bmatrix}\begin{bmatrix} m_1 \\ m_2 \\ \dots \\ m_{n-2} \\ m_{n-1} \end{bmatrix} = \begin{bmatrix} 2\widetilde{g_1} - f_0' \\ 2\widetilde{g_2} \\ \dots \\ 2\widetilde{g_{n-2}} \\ 2\tilde{g}_{n-1} - f_n' \end{bmatrix}$ 由此可解得: $f'(x_j) \approx m_j\ \ ,j = 1,2,\dots,n-1$

2. 样条求导

解出$f'(x_j) \approx m_j\ \ ,j = 1,2,\dots,n-1$

由$S_k(x) = \frac{h_k+2(x-x_k)}{h_k^3}(x - x_{k+1})^2 y_k + \frac{h_k-2(x-x_{k+1})}{h_k^3}(x - x_k)^2 y_{k+1} + \frac{(x-x_k)(x-x_{k+1})^2}{h_k^2}\boldsymbol{m_k} + \frac{(x-x_k)^2\,(x-x_{k+1})}{h_k^2}\boldsymbol{m_{k+1}}$

得$S(x) = \frac{h+2(x-x_j)}{h^3}(x - x_{j+1})^2 f_j + \frac{h+2(x-x_{j+1})}{h^3}(x - x_j)^2 f_{j+1} + \frac{(x-x_j)(x-x_{j+1})^2}{h^2}\boldsymbol{m_j} + \frac{(x-x_j)^2\,(x-x_{j+1})}{h^2}\boldsymbol{m_{j+1}}$

求导$S'(x) = \frac{2(x-x_{j+1})(3x-2x_j-x_{j+1}+h)f_j - 2(x-x_{j+1})(3x-2x_j-2x_{j+1}-h)f_{j+1}}{h^3} + \frac{(x-x_{j+1})(3x-2x_j-x_{j+1})m_j - 2(x-x_j)(3x-x_j-2x_{j+1})m_{j+1}}{h^2}$

$x \in [x_j, x_{j+1}]$ 代入m_j可求出$f'(x) \approx S'(x)$

又$S''(x) = \frac{6(x_j+x_{j+1}-2x)(f_{j+1}-f_j)}{h^3} + \frac{(6x-2x_j-4x_{j+1})m_j + (6x-4x_j-2x_{j+1})m_{j+1}}{h^2}$ $\quad x \in [x_j, x_{j+1}]$ 代入m_j可求出$f''(x) \approx S''(x)$

注意: 可以先用较简单的三点或五点公式求出个节点处的一阶导数值，然后再利用上面两式求出非节点处的一阶及二阶导数值. 这样可以回避求三转角方程组.

例题 1. 利用例 2 的函数表以及求得节点处的一阶导数值。用三次样条微分公式求和的近似值. $(f(x)=e^x+x)$.

① $x = 0.25 \in [0.2, 0.3], h = 0.1$

即$x_j = 0.2, x_{j+1} = 0.3, f_j = 1.4214028,\ f_{j+1} = 1.64985888, m_j = 2.2234395, m_{j+1} = 2.3521095.$

② 代入公式得$f'(0.25) \approx S'(0.25) = 2.28295275$

代入公式得$f''(0.25) \approx S''(0.25) = 1.2867$