第九章 数值积分和数值微分

殷东生
yindongsheng@tsinghua.edu.cn

清华大学数学科学系

2023年秋季学期

在科学与工程问题中经常遇到定积分的计算：

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

如：

- 常微分方程的求解（电子电路、电力系统的暂态计算）；
- 机械、建筑中各种体积分、面积分的计算；
- 金融中期权、股票等收益、风险的计算；
- 工程中有限元方法、边界元方法需要计算积分；
- 各种积分变换：Fourier变换、Laplace 变换、卷积等；
- 偏微分方程的求解：如Poisson方程的解可以通过Green函数的积分得到。

定积分的计算

数学上，设$f(x)$在$[a,b]$上可积，$F(x)$是$f(x)$的一个原函数，则对$\forall x \in [a,b]$有 Newton-Leibniz公式

$$\int_a^x f(t)dt = F(x) - F(a).$$

求出原函数后定积分的计算问题解决！

但是对于实际问题Newton-Leibniz公式是远远不够的。

- 首先，在整个可积函数类中，能够用初等函数表示的定积分只是很小的一部分。对于绝大部分 理论上存在定积分的函数，如

$$e^{x^2}, \qquad \frac{\sin x}{x}(x \neq 0)$$

等并不能用Newton-Leibniz公式求得 定积分的值。

- 其次，即使被积函数很简单，但其不定积分可能十分复杂。如被积函数 $f(x) = \frac{1}{x^6+1}$,它的 不定积分为

$$\frac{1}{3}\arctan x + \frac{1}{6}\arctan\left(\frac{x^2-1}{x}\right) + \frac{1}{4\sqrt{3}}\ln\frac{x^2+x\sqrt{3}+1}{x^2-x\sqrt{3}+1} + C,$$

  表达式很复杂，不能求得精确解，实际计算会有很大的困难。

- 在实际问题中，许多函数只是通过测量、试验等方法给出若干个离散点上的函数值，Newton-Leibnize 公式难于利用。
- 某些实际应用中如建筑物体积的计算，由于建筑物的造型不规则，无法直接求解积分，需要测量后数值积分。

实际应用中必须考虑定积分的近似方法，而数值积分是最重要的一种近似方法。

中点公式

设$f(x)\in C[a,b]$，如果已知$x=\frac{a+b}{2}$处的函数值$f(\frac{a+b}{2})$，则可以得到一个最简单的求积 公式

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

注记

若$f(x)>0$,此公式的几何意义是用长为$b-a$，宽为$f(\frac{a+b}{2})$的矩形面积来近似曲边梯形的面积，因此被称为中点公式(*midpoint formula*)。

若$f(x)\in C^2[a,b]$,则中点公式的截断误差为

$$\int_a^b f(x)dx - f\left(\frac{a+b}{2}\right)(b-a) = \frac{(b-a)^3}{24}f''(\xi), \xi\in(a,b).$$

图: 中点公式

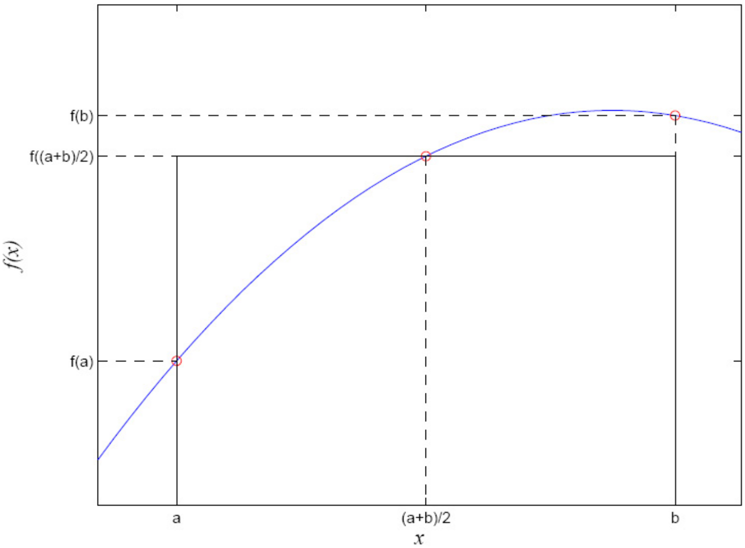
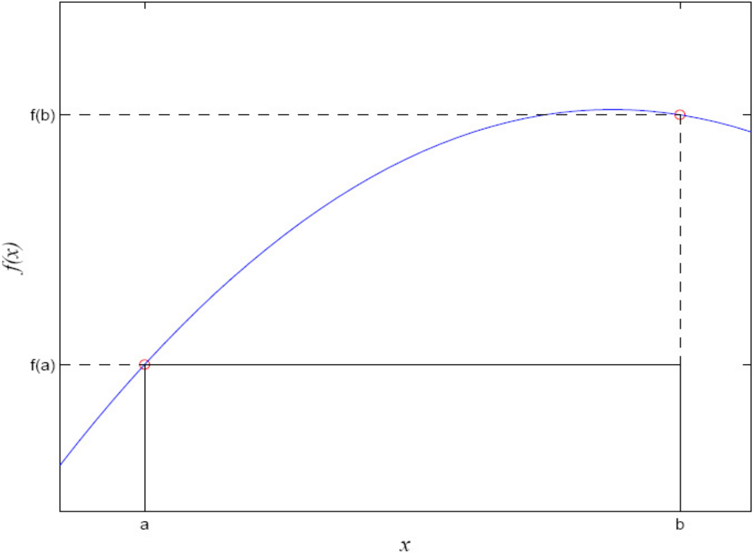


图: 矩形公式



梯形公式

如果已知$x=a,b$两点的函数值$f(a)$,$f(b)$，则有$f(x)$的一次插值多项式

$$L_1(x)=\frac{x-b}{a-b}f(a)+\frac{x-a}{b-a}f(b),$$

从而近似的有

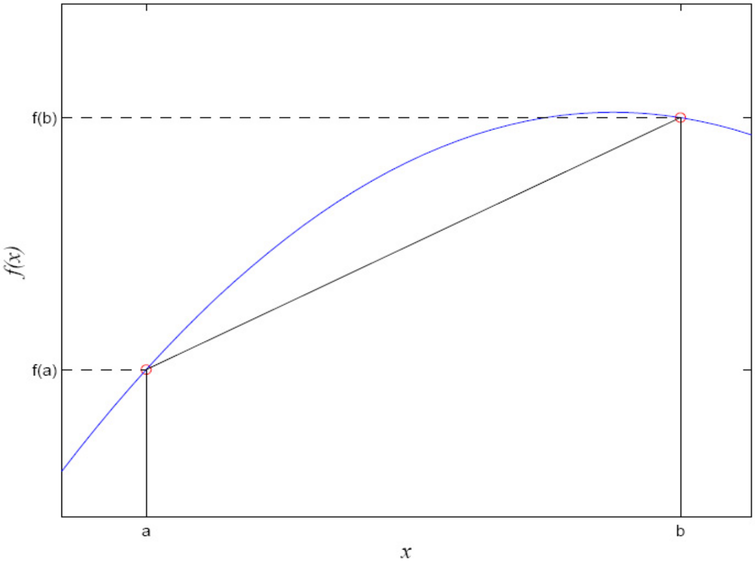
$$I(f)=\int_a^b f(x)dx\approx I_1(f)=\int_a^b L_1(x)dx=\frac{f(a)+f(b)}{2}(b-a).$$

上式定义的积分公式称为梯形公式(trapezoidal formula)。

注记

公式的几何意义就是用梯形面积近似曲线$f(x)$上的面积$\int_a^b f(x)dx$.

图: 梯形公式



为了讨论梯形公式的误差 $E_1(f) = I(f) - I_1(f)$ 需要微积分中的积分中值定理。

定理 (积分第一中值定理)

设$f(x), g(x) \in C[a,b]$，且$g(x)$在$[a,b]$上不变号，则存在$\xi \in [a,b]$使得

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

定理

设$f \in C^2[a,b]$，则梯形求积公式的截断误差为

$$E_1(f) = \int_a^b f(x)dx - \frac{b-a}{2}[f(a)+f(b)] = -\frac{(b-a)^3}{12}f''(\xi), \xi \in [a,b].$$

推论

设$f \in C^2[a,b]$,则有

$$|E_1(f)| = |I(f) - I_1(f)| \le \frac{1}{12}\|f''\|_\infty (b-a)^3.$$

例

用梯形公式计算

$$I = \int_0^1 \frac{1}{1+x} dx$$

解： 由梯形公式有

$$I_1(f) = \frac{1}{2}\left(1 + \frac{1}{2}\right) = \frac{3}{4} = 0.75$$

I的准确值为$\ln 2 \approx 0.693147$，计算的误差 $I - I(f) \approx -0.0569$。

$$f(x) = \frac{1}{1+x}, f''(x) = \frac{2}{(1+x)^3}, \|f''\|_\infty = 2,$$

用推论得估计

$$|I(f) - I_1(f)| \leq \frac{1}{12} \times 2 \times 1 = \frac{1}{6} \approx 0.16667.$$

Simpson公式

为了使计算定积分$I(f)$更准确，用$f(x)$在$[a,b]$上的二次插值函数$L_2(x)$来代替$L_1(x)$。插值节点取为$x_0=a, x_1=\frac{a+b}{2}, x_2=b$，则

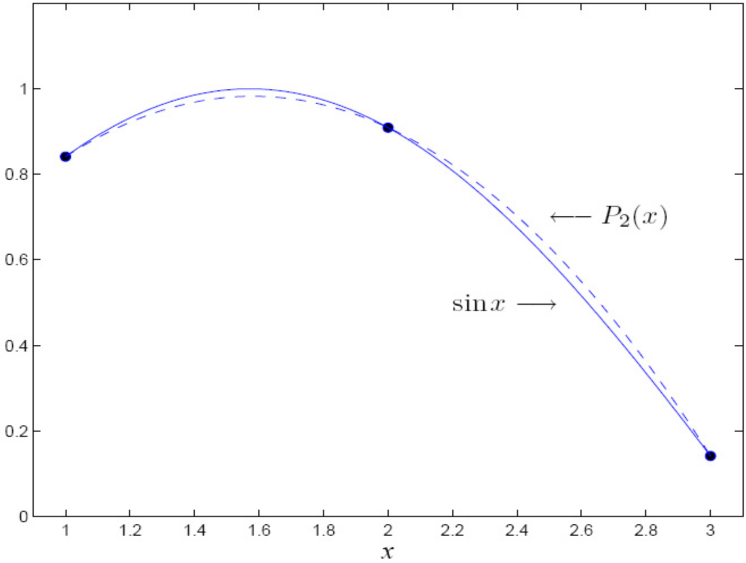
$$\begin{aligned} L_2(x) &= \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}f(x_0)+\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}f(x_1) \\ &\quad +\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}f(x_2). \\ I(f) &\approx \int_a^b L_2(x)dx=\frac{b-a}{6}\left[f(a)+4f(\frac{a+b}{2})+f(b)\right]. \end{aligned}$$

令

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

称$I_2(f)$为Simpson求积公式或抛物型公式。

图: Simpson公式



定理

如果$f(x) \in C^4[a,b]$，则$Simpson$积分公式的误差为

$$E_2(f) = \int_a^b f(x)dx - I_2(f) = -\frac{(b-a)^5}{2880} f^{(4)}(\xi), \xi \in (a,b).$$

则

$$|I(f) - I_2(f)| \leq \frac{(b-a)^5}{2880} \|f^{(4)}\|_\infty.$$

例

用*Simpson*公式计算

$$I(f) = \int_0^1 \frac{1}{1+x}dx$$

解：用Simpson公式有

$$\begin{aligned} I_2(f) &= \frac{1}{6}\left[1 + \frac{4}{1+\frac{1}{2}} + \frac{1}{2}\right] \approx 0.694\,444 \\ I(f) - I_2(f) &\approx -0.001\,297 \end{aligned}$$

在$[a,b]$上给定节点$a \leq x_0 < x_1 < \cdots < x_n \leq b$以及相应的函数值$\{f(x_i)\}_{i=0}^n$，则 可以得到$n$次Lagrange插值多项式

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

设其余项为$R_n(x)$，那么

$$f(x) = L_n(x) + R_n(x), x \in [a,b].$$

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

其中

$$\begin{aligned} A_k &= \int_a^b l_k(x)dx, k = 0, 1, \cdots, n, \\ E_n(f) &= \int_a^b R_n(x)dx \end{aligned}$$

插值型求积公式

从上面的推导中有

- $x_k, k = 0, 1, \cdots, n$称为求积节点；
- $A_k, k = 0, 1, \cdots, n$为求积系数；
- $E_n(f)$为求积公式的误差。
- $A_k, x_k, k = 0, 1, \cdots, n$和$f$无关。
- 一般称

$$\int_a^b f(x)dx \approx \sum_{k=0}^{n} A_k f(x_k), \tag{0.1}$$

为插值型求积公式。

- 中点公式、梯形公式和Simpson公式都是插值型求积公式。

由Lagrange插值多项式的余项可得

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

若

$$f(x) \in \mathcal{P}_n = \operatorname{span}\{1, x, \cdots, x^n\},$$

则

$$E_n(f) = 0,$$

亦即

$$\int_a^b f(x)dx = \sum_{k=0}^{n} A_k f(x_k), \quad \forall f(x) \in \mathcal{P}_n$$

这一现象反映了数值求积公式的精度，数学上用代数精度这一概念来描述。

代数精度

定义 (代数精度)

如果求积公式对所有$p \in \mathcal{P}_m$准确成立，即$E_n(p) = 0$，而对某个$q \in \mathcal{P}_{m+1}$，有$E_n(q) \neq 0$,那么称求积公式具有m次代数精度

由求积公式有

$$\begin{aligned} & E_n(\alpha f + \beta g) = \alpha E_n(f) + \beta E_n(g) \\ \Longrightarrow \quad & E_n(p) = 0, \forall p \in \mathcal{P}_m, E_n(q) \neq 0, \exists q \in \mathcal{P}_{m+1} \\ \Longleftrightarrow \quad & E_n(x^k) = 0, k = 0, 1, \cdots, m, \text{ 且 } E_n(x^{m+1}) \neq 0. \end{aligned}$$

定义

若$E_n(x^k) = 0, k = 0, 1, \cdots, m$，则称求积公式至少具有$m$次代数精度，若还有 $E_n(x^{m+1}) \neq 0$，则称求积公式具有m次代数精度。

根据代数精度的定义和余项表达式有

定理

对于 $n+1$ 个节点的插值型求积公式*(0.1)*，代数精度至少为n。

有了代数精度的概念，即使积分区间$[a,b]$不是很小，也可以讨论求积公式的精度。

1. 中点公式和梯形公式的代数精度都是一阶。
2. Simpson公式的代数精度为3阶。
3. 同样是1阶代数精度，中点公式只需要一个节点，而梯形公式要两个。

将区间$[a,b]n$等分，令

$$h=\frac{b-a}{n}, x_k=a+kh, k=0,1,\cdots,n$$

则

$$\begin{aligned}
L_n(x) &= \sum_{k=0}^{n} f(x_k)l_k(x), \\
l_k(x) &= \prod_{j=0,j\neq k}^{n} \frac{x-x_j}{x_k-x_j}. \\
\int_a^b f(x)dx &\approx \int_a^b \sum_{k=0}^{n} f(x_k)l_k(x)dx \\
&= \sum_{k=0}^{n} f(x_k)\int_a^b l_k(x)dx \\
&= (b-a)\sum_{k=0}^{n}\frac{1}{b-a}\int_a^b l_k(x)dx\cdot f(x_k).
\end{aligned}$$

$$令\ C_k^{(n)} = \frac{1}{b-a}\int_a^b l_k(x)dx$$

则有

$$\int_a^b f(x)dx \approx (b-a)\sum_{k=0}^{n} C_k^{(n)} f(x_k). \tag{0.2}$$

公式(0.2)称为Newton-Cotes求积公式。$C_k^{(n)}$称为Cotes求积系数。
由于节点是等距的，做变换

$$x = a + th, \qquad x_k = a + kh$$

所以

$$C_k^{(n)} = \frac{1}{b-a}\int_a^b \prod_{j=0,j\neq k}^{n} \frac{x-x_j}{x_k-x_j} = \frac{1}{n}\int_a^b \prod_{j=0,j\neq k}^{n} \frac{t-j}{k-j}dt.$$

进一步化简有

$$C_k^{(n)} = \frac{(-1)^{n-k}}{k!(n-k)!}\frac{1}{n}\int_0^1 \prod_{j=0, j\neq k}^{n} (t-j)dt. \tag{0.3}$$

注记

因此Cotes系数$C_k^{(n)}, k = 0, 1, \cdots, n$不仅与被积函数无关，而且与求积区间也没有关系。

由上式知道

$$C_k^{(n)} = C_{n-k}^{(n)}, k = 0, 1, \cdots, [\frac{n}{2}].$$

利用(0.3)可以求解Cotes系数。令$f(x) = 1$，则

$$\sum_{k=0}^{n} C_k^{(n)} = 1.$$

常用 Newton-Cotes 公式

1. $n=1, n=2$分别为梯形公式和Simpson公式。
2. 当$n=3$时，Newton-Cotes公式为Simpson3/8公式，令$h=\frac{b-a}{3}$,则
$$\int_a^b f(x)dx \approx \frac{3}{8}h\left[f(a)+3f(a+h)+3f(b-h)+f(b)\right].$$
3. 当$n=4$时，Newton-Cotes公式为Boole公式或称为Cotes公式。
令$h=\frac{b-a}{4}$，那么有
$$\begin{aligned}\int_a^b f(x)dx \quad\approx\quad & \frac{2h}{45}\left[7f(a)+32f(a+h)+12f\left(\frac{a+b}{2}\right)\right.\\ & \left.+32f(b-h)+7f(b)\right].\end{aligned}$$

实际计算的时候，数据有误差

$$f(x_k)+\epsilon_k=\tilde{f}(x_k), k=0,1,\cdots,n.$$

记$\epsilon=\max_{0\le k\le n}|f(x_k)-\tilde{f}(x_k)|$，则有

$$\begin{aligned}
\left|\sum_{k=0}^{n} f(x_k)C_k^{(n)}-\sum_{k=0}^{n}\tilde{f}(x_k)C_k^{(n)}\right| &= \left|\sum_{k=0}^{n} C_k^{n}(\tilde{f}(x_k)-f(x_k))\right| \\
\le \quad \sum_{k=0}^{n}|C_k^{(n)}|\cdot|f(x_k)-\tilde{f}(x_k)| &\le \max_{0\le k\le n}|f(x_k)-\tilde{f}(x_k)|\cdot\sum_{k=0}^{n}\left|C_k^{(n)}\right| \\
\le \quad \epsilon\sum_{k=0}^{n}|C_k^{(n)}|
\end{aligned}$$

如果$C_k^{(n)}>0, k=0,1,\cdots,n$，则有

$$|\sum_{k=0}^{n} f(x_k)C_k^{(n)}-\sum_{k=0}^{n}\tilde{f}(x_k)C_k^{(n)}|\le\epsilon\sum_{k=0}^{n}|C_k^{(n)}|=\epsilon$$

所以求积公式是稳定的。

当$n\geq 8$是,$C_k^{(n)}$出现负值，亦即

$$\sum_{k=0}^{n}|C_k^{(n)}|\geq|\sum_{k=0}^{n}C_k^{(n)}|=1.$$

特别的假定$C_k^{(n)}(\tilde{f}(x_k)-f(x_k))>0$，且$|f(x_k)-\tilde{f}(x_k)|=\alpha$，则

$$\begin{aligned}\left|\sum_{k=0}^{n}\tilde{f}(x_k)C_k^{(n)}-\sum_{k=0}^{n}f(x_k)C_k^{(n)}\right| &= \sum_{k=0}^{n}C_k^{(n)}[\tilde{f}(x_k)-f(x_k)]\\ &= \sum_{k=0}^{n}|C_k^{(n)}|\cdot|\tilde{f}(x_k)-f(x_k)|>\alpha\end{aligned}$$

注记

当$n\geq 8$时，$C_k^{(n)}$有正有负，必然导致其中有的系数绝对值大于*1*，由于舍入误差的影响，误差 会被放大而导致计算不准确，这种不稳定性使得在$n\geq 8$时*Newton-Cotes*公式基本不被采用。

考虑定积分

$$I(f) = \int_1^{10} \frac{1}{x} dx \approx 2.302585.$$

如果用梯形公式求其近似值，

$$I_1(f) = \frac{9}{2}(\frac{1}{10} + 1) = 4.9$$

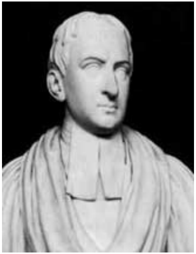
用Simpson求积公式，

$$I_2(f) = \frac{9}{6}(1 + \frac{4}{5.5} + \frac{1}{10}) = 2.740909.$$

用$I_1(f)$和$I_2(f)$计算上面的定积分都不准确，要得到$I(f)$的很精确的值，必须采用高阶的Newton-Cotes求积公式，而高阶的Newton-Cotes公式($n \geq 8$)是不稳定的。

数学家简介 (Cotes)

Roger Cotes（*1682-1716*），英国数学家，*Newton*的学生。一生只发表一篇文章，但对指数函数和积分计算有重要贡献。引入*Euler*公式的第一人，在插值和表格构造法有重要贡献，预见了最小二乘法。*26*岁成为剑桥三一学院的教授。对于他的英年早逝，*Newton*说：“*If he had lived, we might have known something.*”



中国作诗最多的人：“十全老人”爱新觉罗.弘历，作诗42250多首，全唐朝2700位诗人300年流传下来的有48900首。

唐代张若虚，一生仅留下两首诗，其中一首《春江花月夜》“孤篇横绝，竟为大家”!闻一多在《宫体诗的自赎》中评价：“诗中的诗，顶峰上的顶峰”！

南宋的陆游的诗词共流传下来9400多首，

诗：“江声不尽英雄恨，天意无私草木秋”

词：“小楼一夜听春雨，深巷明朝卖杏花”

提高精度的另外一个方法是把$[a,b]$分成若干子区间，然后在子区间上采用低阶求积公式，这种方法称为复合求积方法。
将$[a,b]$分成n等分$x_k=a+kh, h=\frac{b-a}{n}, k=0,1,\cdots,n$。在每个子区间$[x_{k-1},x_k], k=0,1,\cdots,n$上用梯形公式，则有

$$
\begin{aligned}
\int_a^b f(x)dx &= \sum_{k=1}^{n}\int_{x_{k-1}}^{x_k} f(x)dx = \frac{h}{2}\sum_{k=1}^{n}[f(x_{k-1})+f(x_k)] + E_n(f). \\
\text{令 } T_n(f) &= \frac{h}{2}\sum_{k=1}^{n}[f(x_{k-1})+f(x_k)] \\
&= \frac{h}{2}\left[f(a)+2\sum_{k=1}^{n-1}f(x_k)+f(b)\right]
\end{aligned}
$$

称为复合梯形公式。

复合梯形公式的误差

由梯形公式的误差有

$$\int_{x_{k-1}}^{x_k} f(x)dx = \frac{h}{2}[f(x_{k-1}) + f(x_k)] - \frac{h^3}{12}f''(\eta_k), \eta_k \in [x_{k-1}, x_k].$$

$$\Longrightarrow \quad \int_a^b f(x)dx = T_n(x) - \frac{h^3}{12}\sum_{k=1}^{n} f''(\eta_k).$$

$$\Longrightarrow \quad E_n(f) = -\frac{b-a}{12}h^2\frac{1}{n}\sum_{k=1}^{n} f''(\eta_k).$$

因为

$$\min_{1\le k\le n} f''(\eta_k) \le \frac{1}{n}\sum_{k=1}^{n} f''(\eta_k) \le \max_{1\le k\le n} f''(\eta_k).$$

所以

$$f \in C^2[a,b] \qquad \Longrightarrow f''(\eta) = \frac{1}{n}\sum_{k=1}^{n} f''(\eta_k)$$

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

复合梯形公式的误差是$O(h^2)$，且

$$\lim_{h\to 0} E_n(f) = 0,$$

亦即

$$\lim_{n\to\infty} T_n(f) = \int_a^b f(x)dx.$$

设$f \in C[a,b]$，把$T_n(f)$改写为

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

实际上，复合梯形公式收敛只需$f(x) \in C[a,b]$，此时

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

因为$f(x) \in C[a,b]$，所以$f(x)$在$[a,b]$上可积，则

$$\begin{aligned} \lim_{n\to+\infty} \frac{b-a}{n}\sum_{k=0}^{n-1} f(x_k) &= \lim_{n\to+\infty} \frac{b-a}{n}\sum_{k=1}^{n} f(x_k) \\ &= \int_a^b f(x)dx, \end{aligned}$$

$$\Longrightarrow \lim_{n\to\infty} T_n(f) = \int_a^b f(x)dx.$$

但此时没有收敛阶。如果求积公式的精度还不够，则可以把子区间$[x_{k-1}, x_k]$对分。在$2n$个子区间上 用复合梯形公式计算$T_{2n}(f)$。

$T_n(f)$的分点也是$T_{2n}(f)$的分点，实际计算$T_{2n}(f)$的时候，只要把新分点

$$x_{k-\frac{1}{2}} = x_k - \frac{h}{2}, h = \frac{1}{n}(b-a), k = 1, 2, \cdots, n$$

上的函数值$f(x_{k-\frac{1}{2}})$加到$T_n(f)$中，即

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

若令

$$H_n(f) = h\sum_{k=1}^{n} f(x_{k-\frac{1}{2}}),$$

那么有

$$T_{2n}(f) = \frac{1}{2}[T_n(f) + H_n(f)].$$

Matlab求积函数trapz

Q = trapz(Y) returns the approximate integral of Y via the trapezoidal method with unit spacing. The size of Y determines the dimension to integrate along:

- If Y is a vector, then trapz(Y) is the approximate integral of Y.
- If Y is a matrix, then trapz(Y) integrates over each column and returns a row vector of integration values.
- If Y is a multidimensional array, then trapz(Y) integrates over the first dimension whose size does not equal 1. The size of this dimension becomes 1, and the sizes of other dimensions remain unchanged.

1. Q = trapz(X,Y) integrates Y with spacing increment X. By default, trapz operates on the first dimension of Y whose size does not equal 1. length(X) must be equal to the size of this dimension . If X is a scalar, then trapz(X,Y) is equivalent to X*trapz(Y). example
2. Q = trapz(——,dim) integrates along the dimension dim using any of the previous syntaxes. You must specify Y, and optionally can specify X. The length of X, if specified, must be the same as size(Y,dim). For example, if Y is a matrix, then trapz(X,Y,2) integrates each row of Y.

trapz

例

X = 0:pi/100:pi; %*Create a domain vector, X.*
Y = sin(X); %*Calculate the sine of X and store the result in Y.*
Q = trapz(X,Y) % *Integrate the function values contained in Y using trapz.*
Q =1.9998

例

x = -3:.1:3;
y = -5:.1:5;
[X,Y] = meshgrid(x,y); %*Created a grid of domain values.*
F = X.^2 + Y.^2; % *Calculate the function* $f(x,y) = x^2 + y^2$ *over the grid.*
I = trapz(y,trapz(x,F,2)) % *Use trapz to approximate the double integral*
I = 680.2000

把区间$[a,b]$分为n等分，在每个子区间$[x_{k-1},x_k]$上用Simpson公式，即

$$\int_a^b f(x)dx=\sum_{k=1}^{n}\int_{x_{k-1}}^{x_k} f(x)dx \approx \frac{h}{6}\sum_{k=1}^{n}[f(x_{k-1})+4f(x_{k-\frac{1}{2}})+f(x_k)]$$

这里$x_{k-\frac{1}{2}}=\frac{1}{2}(x_{k-1}+x_k)$. 则

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

这个公式称为复合Simpson求积公式。公式可以简写为

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

如果$f \in C^4[a,b]$，则存在$\eta_k \in [x_{k-1}, x_k]$使得

$$\int_a^b f(x)dx = \sum_{k=1}^{n} \left\{ \frac{h}{6}[f(x_{k-1}) + 4f(x_{k-\frac{1}{2}}) + f(x_k)] - \frac{h^5}{2880} f^{(4)}(\eta_k) \right\}.$$

求和以后有

$$E_n(f) = -\frac{1}{2880}(b-a)h^4 \left(\frac{1}{n} \sum_{k=1}^{n} f^{(4)}(\eta_k) \right).$$

类似与复合梯形公式误差的推导有

$$\int_a^b f(x)dx = S_n(f) - \frac{1}{2880}(b-a)h^4 f^{(4)}(\eta), \ \eta \in [a,b].$$

复合Simpson公式的误差为$O(h^4)$。当$f \in C[a,b]$时，复合梯形求积公式是收敛的。实际上

$$S_n(f) = \frac{1}{3}T_n(f) + \frac{2}{3}H_n(f).$$

例

若用复合梯形公式求$I=\int_0^1 e^{-x}dx$的近似值，问将要积分区间$[0,1]$分成多少等分才能使误差 不超过$\frac{1}{2}\times 10^{-4}$？若用复合$Simpson$求积公式呢？

解： 由复合梯形公式的余项有

$$E_n(f)=-\frac{b-a}{12}h^2f''(\eta)=-\frac{1}{12}\left(\frac{1}{n}\right)^2 f''(\eta),\eta\in[a,b].$$

而$f(x)=e^{-x},f''(x)=e^{-x},\max_{x\in[0,1]}e^{-x}=1$，则

$$|E_n(f)|\leq\frac{1}{12n^2}\leq\frac{1}{2}\times 10^{-4}.$$

当$n\geq 40.8$时上式满足，取$n=41$。而复合Simpson求积公式有

$$|E_n(f)|\leq\frac{1}{2880}\frac{1}{n^4}\leq\frac{1}{2}\times 10^{-4}.$$

当$n\geq 2$时满足要求，取$n=2$即可。

Matlab数值积分函数quad

q = quad(fun,a,b) tries to approximate the integral of function fun from a to b to within an error of 1e-6 using recursive adaptive Simpson quadrature. fun is a function handle. Limits a and b must be finite. The function y = fun(x) should accept a vector argument x and return a vector result y, the integrand evaluated at each element of x.

1. q = quad(fun,a,b,tol) uses an absolute error tolerance tol instead of the default which is 1.0e-6. Larger values of tol result in fewer function evaluations and faster computation, but less accurate results.
2. q = quad(fun,a,b,tol,trace) with non-zero trace shows the values of [fcnt a b-a Q] during the recursion.
3. [q,fcnt] = quad(...) returns the number of function evaluations.

The function quadl may be more efficient with high accuracies and smooth integrands.

Matlab数值积分函数dblquad

q = dblquad(fun,xmin,xmax,ymin,ymax) calls the quad function to evaluate the double integral fun(x,y) over the rectangle xmin <= x <= xmax, ymin <= y <= ymax. The input argument, fun, is a function handle that accepts a vector x, a scalar y, and returns a vector of integrand values.

1. q = dblquad(fun,xmin,xmax,ymin,ymax,tol) uses a tolerance tol instead of the default, which is 1.0e-6.
2. q = dblquad(fun,xmin,xmax,ymin,ymax,tol,method) uses the quadrature function specified as method, instead of the default quad. Valid values for method are @quadl or the function handle of a user-defined quadrature method that has the same calling sequence as quad and quadl.

类似的函数还有：quad2d,quadgk等，见Matlab帮助。

dblquad

%Pass function handle @integrnd to dblquad:

```
Q = dblquad(@integrnd,pi,2*pi,0,pi);
```

%where the function integrnd.m is:

```
function z = integrnd(x, y)
z = y*sin(x)+x*cos(y);
```

%Pass anonymous function handle F to dblquad:

```
F = @(x,y)y*sin(x)+x*cos(y);
Q = dblquad(F,pi,2*pi,0,pi);
```

%Nonsquare regions can be handled by setting the integrand to zero outside of the region. For example, the volume of a hemisphere is:

```
dblquad(@(x,y)sqrt(max(1-(x.^2+y.^2),0)), -1, 1, -1, 1)
```

or

```
dblquad(@(x,y)sqrt(1-(x.^2+y.^2)).*(x.^2+y.^2<=1), -1, 1, -1, 1)
```

用$Q_1(h)$去近似Q，如果

$$Q - Q_1(h) = c_1 h^{p_1} + c_2 h^{p_2} + c_3 h^{p_3} + \cdots + c_k h^{p_k} + \cdots , \tag{0.4}$$

其中c_k, p_k为常数，且$0 < p_1 < p_2 < \cdots$。截断误差为$O(h^{p_1})$。
取$h = h/2$，即把步长减小一倍，则

$$\begin{aligned} Q - Q_1\left(\frac{h}{2}\right) &= c_1\left(\frac{h}{2}\right)^{p_1} + c_2\left(\frac{h}{2}\right)^{p_2} + c_3\left(\frac{h}{2}\right)^{p_3} + \cdots \\ &= 2^{-p_1} c_1 h^{p_1} + 2^{-p_2} c_2 h^{p_2} + 2^{-p_3} c_3 h^{p_3} + \cdots , \end{aligned} \tag{0.5}$$

把式(0.5)乘于2^{p_1}后减去(0.4)有

$$Q - \frac{2^{p_1} Q_1(h/2) - Q_1(h)}{2^{p_1} - 1} = c_2^* h^{p_2} + c_3^* h^{p_3} + \cdots , \tag{0.6}$$

而

$$c_k^* = \frac{c_k(2^{p_1-p_k}-1)}{2^{p_1}-1}$$

记

$$Q_2(h) = \frac{2^{p_1}Q_1(h/2)-Q_1(h)}{2^{p_1}-1},$$

则

$$Q - Q_2(h) = c_2^* h^{p_2} + c_3^* h^{p_3} + \cdots,$$

亦即Q_2的截断误差为$O(h^{p_2})$，从$Q_2(h)$出发用类似的方法可构造$Q_3(h)$，使得它的截断误差为$O(h^{p_3})$。

这种从低阶精度格式的截断误差的渐进展开式出发，作简单线性组合来得到高阶精度格式的方法 称为Richardson外推收敛加速技术。它在数值积分、数值微分和微分方程数值解等领域 有广泛的应用。

定理 (Richardson外推)

假定$Q(h)$，当$h \to 0$时有$Q(h) \to Q^*$(Q^*与h无关)并有

$$Q^* - Q(h) = \sum_{k=1}^{\infty} \alpha_k h^{p_k}, \quad 0 < p_1 < p_2 < \cdots$$

其中p_k, α_k为与h无关的常数，$\alpha_k \neq 0$。则有

$$\begin{aligned} Q_1(h) &= Q(h) \\ Q_{m+1}(h) &= \frac{Q_m(qh) - q^{p_m} Q_m(h)}{1 - q^{p_m}}, \quad m = 1, 2, \cdots, \ 0 < q < 1. \end{aligned}$$

定义的序列$\{Q_m(h)\}$有

$$Q^* - Q_{m+1}(h) = \sum_{k=1}^{\infty} \alpha_{m+k}^{(m+1)} h^{p_{m+k}}$$

其中$\alpha_{m+k}^{(m+1)}$为与h无关的非零常数。

定义 (Bernoulli多项式)

n次Bernoulli多项式$B_n(x)$是指满足

$$\begin{cases} B_0(x) = 1 \\ B_n'(x) = nB_{n-1}(x) \text{且} \int_0^1 B_n(x)dx = 0, n \in \mathbb{N}. \end{cases}$$

的多项式序列，由此可得Bernoulli多项式的表达式

$$B_n(x) = \sum_{k=0}^{n} \binom{n}{k} b_k x^{n-k}$$

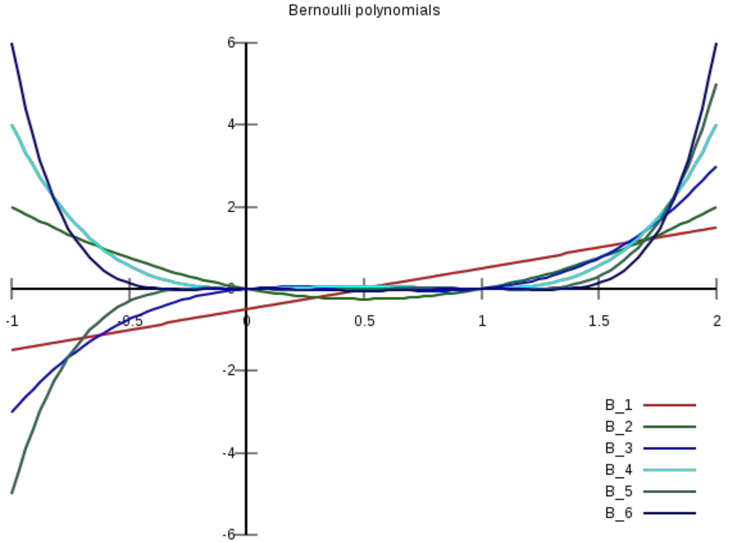
其中$b_n = B_n(0)(n = 0, 1, \cdots)$称为Bernoulli数

显然$\int_0^1 B_n(x) = 0$等价于$B_n(0) = B_n(1), n = 2, 3, \cdots$.

由定义可以得到Bernoulli多项式

$$
\begin{aligned}
B_0(x) &= 1, \\
B_1(x) &= x - \frac{1}{2}, \\
B_2(x) &= x^2 - x + \frac{1}{6}, \\
B_3(x) &= x^3 - \frac{3}{2}x^2 + \frac{1}{2}x, \\
B_4(x) &= x^4 - 2x^3 + x^2 - \frac{1}{30}. \\
B_5(x) &= x^5 - \frac{5}{2}x^4 + \frac{5}{3}x^3 - \frac{1}{6}x, \\
B_6(x) &= x^6 - 3x^5 + \frac{5}{2}x^4 - \frac{1}{2}x^2 + \frac{1}{42}.
\end{aligned}
$$

图: Bernoulli多项式

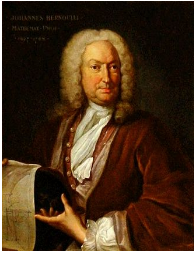


数学家简介 (Jacob Bernoulli)

Jacob Bernoulli the elder(1654-1705)，瑞士数学家，*Bernoulli*家族的第二代。是最早认识到*Newton*和*Leibniz*微积分的重要性的数学家之一。在无穷级数、力学、变分法和概率论等有重要贡献，大数定律的发现者，*Bernoulli*分布。奠定了常微分方程的理论基础。

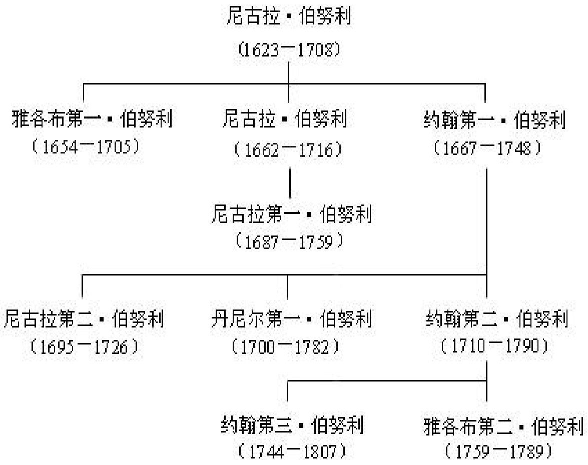
数学家简介 (Johann Bernoulli)

Johann Bernoulli(1667-1748)，瑞士数学家，*Jacob Bernoulli*的弟弟和学生，*Daniel Bernoulli*的父亲。在微分方程（*Bernoulli*方程）和概率论的奠基人之一。发现了洛必达法则和换元积分。是*Euler*和*L'Hôpita*的导师。



数学家简介 (Daniel Bernoulli)

Daniel Bernoulli(1700-1782)，瑞士数学家，*Jahann Bernoulli*的儿子和学生，*Bernoulli*家族中最杰出的人物。*Euler*在圣彼得堡时是他的助手，在巴塞尔担任过解剖学、植物学、生理学、物理学和哲学教授。用数学的方法去研究力学，尤其是流体力学，*1738*年出版《流体力学》，发现*Bernoulli*方程，号称数学物理方法的奠基人。曾因天文学、地球引力、潮汐、磁学、洋流、船体稳定和振动理论获得巴黎科学院*10*次以上的奖励。在概率和数理统计领域也有先驱工作。伯努利原理是能量守恒定律的一个特别的范例，这个原理描述了力学中潜在的数学，促成*20*世纪现在的两个重要的技术的应用：化油器和机翼。


尼古拉·伯努利
(1623—1708)
雅各布第一·伯努利
(1654—1705)
尼古拉·伯努利
(1662—1716)
约翰第一·伯努利
(1667—1748)
尼古拉第一·伯努利
(1687—1759)
尼古拉第二·伯努利
(1695—1726)
丹尼尔第一·伯努利
(1700—1782)
约翰第二·伯努利
(1710—1790)
约翰第三·伯努利
(1744—1807)
雅各布第二·伯努利
(1759—1789)

对复合梯形求积公式有下面的定理

定理 (Euler-Maclaurin公式)

设$f \in C^m[a,b](m=3,4,\cdots)$，对复合梯形求积公式$T_n(f)$有

$$\begin{aligned}\int_a^b f(x)dx &= T_n(f) - \sum_{j=1}^{[\frac{m}{2}]} \frac{b_{2j}h^{2j}}{(2j)!}[f^{(2j-1)}(b) - f^{(2j-1)}(a)] \\ &\quad +(-1)^m h^m \int_a^b \tilde{B}_m\left(\frac{x-a}{h}\right) f^{(m)}(x)dx\end{aligned}$$

此处$[\frac{m}{2}]$是指小于或等于$\frac{m}{2}$的最大整数，b_{2j}是$Bernoulli$数。

注记

如果$f(x)$是周期函数，则h^{2j}的系数$f^{(2j-1)}(b)-f^{(2j-1)}(a)=0$，最后仅剩下$h^m$。这表明梯形公式对周期函数的积分具有谱精度，亦即$f$有多光滑，精度就有多高。

注记

把*Richardson*外推加速收敛技术应用与复合梯形求积公式的误差表达式–*Euler-Maclaurin*展开

$$\int_a^b f(x)dx - T_n(f) = -\sum_{j=1}^{[\frac{m}{2}]} \frac{b_{2j}h^{2j}}{(2j)!}[f^{(2j-1)}(b) - f^{(2j-1)}(a)]$$

就可以得到*Romberg*求积方法。

由Euler-Maclaurin公式可知复合梯形求积公式可表为

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

利用加速外推技术可得

$$\int_a^b f(x)dx = T_2(h) + \sum_{k=2}^{\infty} c_{2k}^{(2)} h^{2k}, \tag{0.7}$$

这里

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

利用公式(0.7)进行Richardson外推加速技术有

$$\int_a^b f(x)dx = T_3(h) + \sum_{k=3}^{\infty} c_{2k}^{(3)} h^{2k}, \tag{0.8}$$

$$T_3(h) = \frac{4^2 T_2(h/2) - T_2(h)}{4^2 - 1}.$$

一般的可以递归定义求积序列

$$T_{k+1}(h) = \frac{4^k T_k(h/2) - T_k(h)}{4^k - 1}, k = 1, 2, \cdots, \tag{0.9}$$

易证

$$\int_a^b f(x)dx - T_{k+1}(h) = O(h^{2(k+1)}), k = 1, 2, \cdots.$$

上面的对复合梯形公式用Richardson 外推法提高精度的方法就是Romberg求积方法。

注记

$T_2(h)$是复合*Simpson*公式，$T_3(h)$是复合*Boole*公式。

Newton-Cotes求积公式的求积节点是等距分布的。当n是偶数时，Newton-Cotes公式的代数精度为$n+1$, Simpson公式的代数精度为3, n为奇数时，代数精度为n,如梯形公式的代数精度为1。$n+1$个等距 节点的求积公式的代数精度至多为$n+1$。

例

对求积公式

$$\int_{-1}^{1} f(x)dx \approx A_0 f(x_0) + A_1 f(x_1),$$

确定节点x_0, x_1以及求积系数A_0, A_1使之代数精度尽可能高？

注记

从这个例子可以看出，适当选取求解节点和求积系数，可把两个节点的梯形公式的1次代数精度提高到3次代数精度。

Q：是否可以通过改变求积节点使得两节点的求积公式的代数精度更高？

A：代数精度最高就是3，不能再提高。

取

$$f(x) = (x - x_0)^2(x - x_1)^2, \int_{-1}^{1} f(x)dx > 0.$$

而$f(x_0) = f(x_1) = 0$，求积公式

$$A_0 f(x_0) + A_1 f(x_1) = 0,$$

所以上面的求积公式的代数精度最多为3。

考虑带权积分

$$\int_a^b \rho(x)f(x)dx,$$

如何选取$n+1$个节点和$n+1$个求积系数

$$a \leq x_0 < x_1 < \cdots < x_n = b, A_0, A_1, \cdots, A_n$$

使得

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

至少有$2n+1$次代数精度？

$$I_n(x^j) = I(x^j), j = 0, 1, \cdots, 2n+1$$

得关于求积节点和系数的$2n+2$个非线性方程组。

求解非常困难，计算量大！

定义 (Gauss型求积公式)

若对已知$n+1$个互异节点$a \le x_0 < x_1 < \cdots < x_n \le b$及常数$A_0, A_1, \cdots, A_n$使得 插值型公式

$$I(f) = \int_a^b f(x)\rho(x)dx \approx \sum_{k=1}^{n} A_k f(x_k) = I_n(f), \tag{0.10}$$

的代数精度为$2n+1$，则称其为*Gauss型积分公式*，相应的节点为*Gauss点*。

如何确定Gauss点以及如何求系数$A_0, A_1, \cdots, A_n$?
令

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

插值型求积公式的截断误差

$$\int_a^b \rho(x)f(x)dx = \sum_{k=0}^{n} A_k f(x_k) + \frac{1}{(n+1)!}\int_a^b \rho(x)f^{(n+1)}(\xi(x))\omega_{n+1}(x)dx$$

当$f(x)\in\mathcal{P}_{2n+1}$时$f^{(n+1)}(x)$为$n$次多项式。如果

$$\int_a^b \rho(x)p(x)\omega_{n+1}(x)dx=0,\ \forall p\in\mathcal{P}_n$$

即$\omega_{n+1}(x)$与$\forall p(x)\in\mathcal{P}_n$正交。

$\omega_{n+1}(x)$为$[a,b]$上以权$\rho(x)$的$n+1$次正交多项式即可。

设$\phi_{n+1}(x)$是$[a,b]$上权为$\rho(x)$的$n+1$次正交多项式

$$\phi_{n+1}(x)=a_{n+1}x^{n+1}+a_nx^n+\cdots+a_1x+a_0.$$

则$\phi_{n+1}(x)$在(a,b)上有$n+1$个不同零点$t_0,t_1,\cdots,t_n$,

$$\phi_{n+1}(x)=a_{n+1}(x-t_0)(x-t_1)\cdots(x-t_n).$$

若$x_0,x_1,\cdots,x_n$取为$t_0,t_1,\cdots,t_n$时就有

$$\int_a^b \rho(x)p(x)\omega_{n+1}(x)dx=\frac{1}{a_{n+1}}\int_a^b \rho(x)p(x)\phi_{n+1}(x)dx=0.$$

$n+1$个节点的求积公式的代数精度可以大于$2n+1$?

$$f(x)=\omega_{n+1}^2(x)=\prod_{i=0}^{n}(x-x_i)^2 \Longrightarrow \int_a^b \rho(x)f(x)dx>0.$$

但是$\sum_{k=0}^{n}A_kf(x_k)=0$。所以对$n+1$个节点的求积公式Gauss公式的代数精度最高。

定理

插值型求积公式(0.10)具有$2n+1$次代数精度$\Longleftrightarrow$ 求积公式的节点$x_0,x_1,\cdots,x_n$为$[a,b]$上权函数为$\rho(x)$的$n+1$次正交多项式的零点。

命题

在$[a,b]$上带权$\rho(x)$的正交多项式$\phi_{n+1}(x)$的根$a \le x_0 < x_1 < \cdots < x_n \le b$就是$Gauss$型求积公式的$Gauss$点。

例

确定求积公式

$$\int_0^1 \sqrt{x}f(x)dx \approx A_0 f(x_0) + A_1 f(x_1)$$

的系数A_0, A_1以及节点x_0, x_1使该求积公式具有最高代数精度。

例

求$Gauss$求积公式

$$\int_{-1}^{1} f(x)dx \approx A_0 f(x_0) + A_1 f(x_1),$$

的节点x_0, x_1及求积系数A_0, A_1。

定理 (Gauss型求积公式的余项)

设$f \in C^{2n+2}[a,b]$,则Gauss型求积公式的余项表达式为

$$E_n(f) = \frac{1}{(2n+2)!} f^{(2n+2)}(\eta) \int_a^b \rho(x)[\omega_{n+1}(x)]^2 dx, \eta \in [a,b]. \qquad (0.11)$$

证明： 由Hermite插值公式有$2n+1$次多项式$H_{2n+1}(x)$满足

$$\begin{aligned}
H_{2n+1}(x_i) &= f(x_i), H'_{2n+1}(x_i) = f'(x_i), i = 0, 1, \cdots, n. \\
f(x) &= H_{2n+1}(x) + \frac{\omega_{n+1}^2(x)}{(2n+2)!} f^{(2n+2)}(\eta).\eta \in [a,b] \\
\int_a^b \rho(x) H_{2n+1}(x) dx &= \sum_{k=0}^{n} A_k H_{2n+1}(x_k) = \sum_{k=0}^{n} A_k f(x_k). \\
E_n(f) &= \frac{1}{(2n+2)!} f^{(2n+2)}(\eta) \int_a^b \rho(x) \omega_{n+1}^2(x) dx
\end{aligned}$$

在Newton-Cotes公式中，$n \geq 8$时由于求积系数出现负值导致算法不稳定，对Gauss型求积公式的系数有

定理

*Gauss*型求积公式的求积系数$A_k, k = 0, 1, \cdots, n$皆为正。

考虑Gauss求积公式的数值稳定性，记Gauss求积公式为

$$I_n(f) = \sum_{k=0}^{n} A_k f(x_k).$$

设$\tilde{f}(x_k)$是$f(x_k), k = 0, 1, \cdots, n$的近似值。

定义 (数值稳定性)

对$\forall\epsilon > 0$，若$\exists\delta > 0$，只要$|\tilde{f} - f| \leq \delta$，就有

$$|I_n(f) - I_n(\tilde{f})| \leq \epsilon.$$

则称求积公式$I_n(f)$是稳定的。

定理 (Gauss求积公式的数值稳定性)

*Gauss*求积公式是数值稳定的。

定义 (收敛性)

若求积公式满足

$$\lim_{n\to\infty,h\to 0}\sum_{k=0}^{n}A_k f(x_k)=\int_a^b f(x)\rho(x)dx$$

其中$h=\max_{1\le i\le n}(x_i-x_{i-1})$，则称求积公式是收敛的。

定理 (Gauss求积公式的收敛性)

设$f(x)\in C[a,b]$，则对$Gauss$型求积公式有

$$\lim_{n\to\infty} I_n(f)=\int_a^b f(x)\rho(x)dx.$$

设在区间$[a,b]=[-1,1]$上的以$\rho(x)=1$为权的正交多项式为Legendre多项式$P_n(x)$:

$$P_0(x)=1,\ P_n(x)=\frac{1}{2^n n!}\frac{\mathrm{d}^n}{\mathrm{d}x^n}[(x^2-1)^2], n=1,2,\cdots.$$

此时的Guass型求积公式称为Gauss-Legendre求积公式:

$$\int_{-1}^{1} f(x)dx=\sum_{k=0}^{n} A_k f(x_k)+E_n(f). \tag{0.12}$$

其中Gauss节点$x_0,x_1,\cdots,x_n$是$P_{n+1}(x)$的零点, 系数

$$A_k=\frac{2}{(1-x_k^2)[P'_{n+1}(x_k)]^2}, k=0,1,\cdots,n.$$

余项

$$E_n(f)=\frac{2^{2n+3}[(n+1)!]^4}{(2n+3)[(2n+2)!]^3}f^{(2n+2)}(\eta), \eta\in[-1,1].$$

1. $n=0$时仅有一个节点$P_1(x)=x$的零点0，则$x=0$，代数精度为1，令$f(x)=1$得 $A_0=2$，求积公式为

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

2. $n=1$时有

$$\int_{-1}^{1} f(x)dx \approx f\left(-\frac{\sqrt{3}}{3}\right)+f\left(\frac{\sqrt{3}}{3}\right).$$

3. $n=2$时有3个节点为$P_3(x)=\frac{1}{2}(5x^3-3x)$的零点

$$x_0=-\sqrt{\frac{3}{5}}, x_1=0, x_2=\sqrt{\frac{3}{5}},$$

令$f(x)=1,x,x^2$得$A_0=\frac{5}{9}, A_1=\frac{8}{9}, A_2=\frac{5}{9}$，则

$$\int_{-1}^{1} f(x)dx \approx \frac{1}{9}\left[5f\left(-\sqrt{\frac{3}{5}}\right)+8f(0)+5f\left(\sqrt{\frac{3}{5}}\right)\right].$$

例

用两个节点*Gauss-Legendre*求积公式计算$\int_{-1}^{1}\frac{1}{x+2}dx$的近似值。

解： Gauss型求积公式为

$$\begin{aligned}\int_{-1}^{1}\frac{1}{x+2}dx &\approx \frac{1}{2-\frac{\sqrt{3}}{3}}+\frac{1}{2+\frac{\sqrt{3}}{3}}\\ &= 1.0909091.\end{aligned}$$

积分的准确值为

$$\ln 3 = 1.0986123.$$

梯形公式计算

$$T(f) = 1.3333333.$$

Simpson公式计算

$$S(f) = 1.1111111.$$

对于任意区间$[a,b]$上的定积分

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

要用变量替换的方法把$[a,b]$变换到$[-1,1]$上。令

$$x=\frac{a+b}{2}+\frac{b-a}{2}t,$$

则

$$\begin{aligned}\int_a^b f(x)dx &= \frac{b-a}{2}\int_{-1}^1 f(\frac{a+b}{2}+\frac{b-a}{2}t)dt \\ &\approx \frac{b-a}{2}\sum_{k=0}^n A_k f(\frac{a+b}{2}+\frac{b-a}{2}x_k).\end{aligned}$$

在$[-1,1]$上的权函数$\rho(x)=\frac{1}{\sqrt{1-x^2}}$的正交多项式为Chebyshev多项式

$$T_n(x)=\cos(n\arccos x), n=0,1,\cdots$$

此时的Gauss型求积公式

$$\int_{-1}^{1}\frac{f(x)}{\sqrt{1-x^2}}dx\approx\sum_{k=0}^{n}A_kf(x_k)+E_n(f) \tag{0.13}$$

称为Gauss-Chebyshev求积公式，$x_0,x_1,\cdots,x_n$为$T_{n+1}(x)$的零点。

*Gauss*节点 $\quad x_k=\cos\left(\frac{2k+1}{2(n+1)}\pi\right), k=0,1,\cdots,n,$

求积系数 $\quad A_k=\frac{\pi}{n+1}, k=0,1,\cdots,n.$

误差余项 $\quad E_n(f)=\frac{\pi}{2^{2n+1}(2n+2)!}f^{(2n+2)}(\xi), \xi\in[-1,1].$

Gauss-Chebyshev求积公式可用来求含因子$\frac{1}{\sqrt{1-x^2}}$的奇异积分。

例

用$n=2$的*Gauss-Chebyshev*求积公式计算$\int_{-1}^{1}\frac{x^4}{\sqrt{1-x^2}}dx$。

解:

$$
\begin{aligned}
x_0 &= \cos\frac{\pi}{6}=\frac{\sqrt{3}}{2}, x_1=\cos\frac{\pi}{2}=0, x_2=\cos\frac{5\pi}{6}=-\frac{\sqrt{3}}{2}. \\
A_0 &= A_1=A_2=\frac{\pi}{3}.
\end{aligned}
$$

则

$$
\int_{-1}^{1}\frac{x^4}{\sqrt{1-x^2}}dx\approx\frac{\pi}{3}\Big[(\frac{\sqrt{3}}{2})^4+(-\frac{\sqrt{3}}{2})^4\Big]=\frac{3}{8}\pi.
$$

求积公式的代数精度为$2n+1=5$，因而上式精确成立，亦即

$$
\int_{-1}^{1}\frac{x^4}{\sqrt{1-x^2}}=\frac{3}{8}\pi.
$$

Gauss-Laguerre求积公式

无穷区间$(0,\infty)$上以$\rho(x)=e^{-x}$为权的正交多项式为Laguerre多项式，相应的求积公式为Gauss-Laguerre求积公式

$$\int_0^{\infty} e^{-x} f(x)dx = \sum_{k=0}^{n} A_k f(x_k) + E_n(f),$$

其中$x_0, x_1, \cdots, x_n$是$n+1$阶Laguerre多项式$Q_{n+1}(x)$的根，求积系数

$$A_k = \frac{[(n+1)!]^2 {x_k}^2}{[Q_{n+1}(x_k)]},$$

误差余项

$$E_n(f) = \frac{[(n+1)!]^2}{(2n+1)!} f^{(2n+2)}(\eta),\ \eta \in (0,\infty).$$

Gauss-Hermite求积公式

区间$(-\infty,\infty)$上权函数为$\rho(x)=e^{-x^2}$的正交多项式为$Hermite$多项式，相应的求积 公式为Gauss-Hermite求积公式

$$\int_{-\infty}^{\infty} f(x)e^{-x^2}dx=\sum_{k=0}^{n} A_k f(x_k)+E_n(f).$$

其中$x_0,x_1,\cdots,x_n$是$n+1$次Hermite多项式的零点，求积系数

$$A_k=2^{n+2}(n+1)!\frac{\sqrt{\pi}}{[H_{n+2}(x_k)]^2}$$

误差余项为

$$E_n(f)=\frac{(n+1)!\sqrt{\pi}}{2^{n+1}(2n+2)!}f^{(2n+2)}(\eta),\eta\in(-\infty,\infty).$$

Matlab数值积分函数integral

1. q = integral(fun,xmin,xmax) numerically integrates function fun from xmin to xmax using global adaptive quadrature and default error tolerances.
2. q = integral(fun,xmin,xmax,Name,Value) specifies additional options with one or more Name,Value pair arguments.
3. q = integral2(fun,xmin,xmax,ymin,ymax) approximates the integral of the function z = fun(x,y) over the planar region xmin $\leqslant$ x $\leqslant$ xmax and ymin(x) $\leqslant$ y $\leqslant$ ymax(x). example
4. q = integral2(fun,xmin,xmax,ymin,ymax,Name,Value) specifies additional options with one or more Name,Value pair arguments.
5. q = integral3(fun,xmin,xmax,ymin,ymax,zmin,zmax) approximates the integral of the function z = fun(x,y,z) over the region xmin $\leqslant$ x $\leqslant$ xmax, ymin(x) $\leqslant$ y $\leqslant$ ymax(x) and zmin(x,y) $\leqslant$ z $\leqslant$ zmax(x,y). example
6. q = integral3(fun,xmin,xmax,ymin,ymax,zmin,zmax,Name,Value) specifies additional options with one or more Name,Value pair arguments.

integral 的例子

%*Create the function* $f(x) = \ln(x)$.

```
fun = @(x)log(x);
```

%*Evaluate the integral from x=0 to x=1 with the default error tolerances.*

```
format long
q1 = integral(fun,0,1)
q1 =
   -1.000000010959678
```

%*Evaluate the integral again, specifying 12 decimal places of accuracy.*

```
q2 = integral(fun,0,1,'RelTol',0,'AbsTol',1e-12)
q2 =
  -1.000000000000010
```

integral2的例子

%Create the anonymous function.

```
fun = @(x,y) 1./( sqrt(x + y) .* (1 + x + y).^2 );
```

%Integrate over the triangular region bounded by $0 \le x \le 1$ and $0 \le y \le 1 - x$.

```
ymax = @(x) 1 - x;
q = integral2(fun,0,1,0,ymax)
q =
   0.2854
```

%Evaluate Double Integral in Polar Coordinates

```
fun = @(x,y) 1./( sqrt(x + y) .* (1 + x + y).^2 );
polarfun = @(theta,r) fun(r.*cos(theta),r.*sin(theta)).*r;
```

% Define a function for the upper limit of r.

```
rmax = @(theta) 1./(sin(theta) + cos(theta));
```

%Integrate over the region bounded by $0 \le \theta \le \pi/2$ and $0 \le r \le r_{max}$.

```
q = integral2(polarfun,0,pi/2,0,rmax)
q =
  0.2854
```

integral3的例子

%Define the anonymous parameterized function $f(x,y,z) = 10/(x^2+y^2+z^2+a)$.

```
a = 2;
f = @(x,y,z) 10./(x.^2 + y.^2 + z.^2 + a);
```

%Evaluate the triple integral over the region $-\infty \le x \le 0$, $-100 \le y \le 0$, and $-100 \le z \le 0$.

```
format long
q1 = integral3(f,-Inf,0,-100,0,-100,0)
q1 =
   2.734244598320928e+03
```

%Evaluate the integral again and specify accuracy to approximately 9 significant digits.

```
q2 = integral3(f,-Inf,0,-100,0,-100,0,'AbsTol', 0,'RelTol',1e-9)
q2 =
   2.734244599944285e+03
```

反常积分的计算

反常积分通常是指被积函数在有限区间$[a,b]$上无界的积分。假定要计算积分$\int_a^b f(x)dx$，其中被积函数$f(x)$在某点$c \in [a,b]$的邻域内可以无界。通常有区间截断法、变量替换法和用Gauss型积分等方法。

区间的截断

设$I=\int_a^b f(x)dx$, a点为奇点，则可将I分为

$$I=\int_a^{a+\delta} f(x)dx+\int_{a+\delta}^b f(x)dx,$$

如果$\exists\delta>0$使得

$$\left|\int_a^{a+\delta} f(x)dx\right| \le \epsilon,$$

则第一个积分可忽略不计，近似地有正常积分

$$I \approx \int_{a+\delta}^b f(x)dx.$$

例

设$g \in C[0,1]$,且$|g(x)| \le 1, x \in [0,1]$，试估计

$$\int_0^r \frac{g(x)}{x^{\frac{1}{2}} + x^{\frac{1}{3}}} dx.$$

解： 在$[0,1]$上有$x^{\frac{1}{2}} \le x^{\frac{1}{3}}$，则

$$\left| \frac{g(x)}{x^{\frac{1}{2}} + x^{\frac{1}{3}}} \right| \le \frac{1}{2x^{\frac{1}{2}}}$$

所以

$$\left| \int_0^r \frac{g(x)}{x^{\frac{1}{2}} + x^{\frac{1}{3}}} dx \right| \le \frac{1}{2} \int_0^r \frac{1}{x^{\frac{1}{2}}} dx = r^{\frac{1}{2}}.$$

如果要求精度达到10^{-3}，取$r \le 10^{-6}$。计算

$$I \approx \int_r^1 \frac{g(x)}{x^{1/2} + x^{1/3}} dx.$$

变量替换

可用变量替换法来消除奇点，如 $f \in C[0,1]$，变量 $t^n = x$ 可将积分

$$\int_0^1 x^{-\frac{1}{n}} f(x)dx, \quad n \geq 2$$

变成正常积分

$$n\int_0^1 f(t^n)t^{n-2}dt.$$

常用的变量替换法有

$$\begin{aligned}
\int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}}dx &= \int_0^{\pi} f(\cos t)dt, \\
\int_{-1}^{1} \frac{f(x)}{\sqrt{x(1-x)}}dx &= 2\int_0^{\pi} f(\sin^2 t)dt.
\end{aligned}$$

用Gauss型积分来计算反常积分

可以利用Gauss-Chebyshev公式来计算奇异积分

$$I(f)=\int_{-1}^{1}\frac{f(x)}{\sqrt{1-x^2}}dx,\ \ f(x)\in C[-1,1].$$

的近似值。如

$$I=\int_{-1}^{1}\frac{1}{\sqrt{1-x^4}}dx=\int_{-1}^{1}\frac{1}{\sqrt{1+x^2}}\frac{1}{\sqrt{1-x^2}}dx.$$

对$f(x)=\frac{1}{\sqrt{1+x^2}}$用Gauss-Chebyshev公式

$$I\approx\frac{\pi}{n}\sum_{k=1}^{n}f\left(\cos\frac{(2k-1)}{2n}\pi\right).$$

即可计算出近似值。

还有一些积分可利用已知的Gauss型求积公式得到。如在 $[0,1]$ 上权函数 $\rho(x)=\frac{1}{\sqrt{x}}$，它的正交多项式族为

$$q_k(x)=P_{2k}(\sqrt{x}), k=0,1,\cdots,n.$$

这里 $P_{2k}(x)$ 是Legendre多项式。若 $\tilde{x}_k$ 与 $\tilde{A}_k$ 为 $2n+1$个节点的Gauss-Legendre求积公式的节点和系数，则有

$$I(f)=\int_0^1 \frac{1}{\sqrt{x}}f(x)dx=\sum_{k=0}^{n} A_k f(x_k)+E_n(f)$$

其中$x_k=\overline{x}_k^2, A_k=2\overline{A}_k$。余项为

$$E_n(f)=\frac{2^{4n+5}[(2n+2)!]^3}{(4n+5)[(4n+4)!]^2}f^{(2n+2)}(\eta).$$

对积分

$$I = \int_0^1 \left(\frac{x}{1-x}\right)^{1/2} f(x)dx.$$

奇点为$x=1$。在$[0,1]$上以$\rho(x) = \left(\frac{x}{1-x}\right)^{1/2}$的正交多项式族为

$$q_k(x) = \frac{1}{\sqrt{x}} T_{2k+1}(\sqrt{x}), k = 0, 1, \cdots,$$

其中$T_{2k+1}(x)$为Chebyshev多项式，相应的求积公式为

$$\begin{aligned} I &= \int_0^1 \left(\frac{x}{1-x}\right)^{1/2} f(x)dx = \sum_{k=0}^{n} A_k f(x_k) + E_n(f), \\ x_k &= \cos^2 \frac{(2k+1)}{2(2n+3)}\pi, A_k = \frac{2\pi}{2n+3} x_k, k = 0, 1, \cdots, n, \\ E_n(f) &= \frac{\pi}{2^{4n+5}(2n+2)!} f^{(2n+2)}(\eta). \end{aligned}$$

Kantorovitch方法

亦即通过被积函数的等价来“消除”奇点的方法。如

$$\begin{aligned}\int_0^1 \frac{\cos x}{x^{1/2}}dx &= \int_0^1 \frac{1}{x^{1/2}} + \int_0^1 \frac{\cos x - 1}{x^{1/2}}dx \\ &= 2 + \int_0^1 \frac{\cos x - 1}{x^{1/2}}dx\end{aligned}$$

在$x=0$附近有

$$\cos x - 1 \approx \frac{x^2}{2}.$$

则上式最右边积分的被积函数属于$C[0,1]$，为准确计算可以再消一次

$$\int_0^1 \frac{\cos x}{x^{1/2}}dx = \int_0^1 \frac{1-\frac{1}{2}x^2}{x^{1/2}}dx + \int_0^1 \frac{\cos x - 1 + \frac{1}{2}x^2}{x^{1/2}}dx.$$

对于这种更简单的被积函数，可以用$\cos x$的展开。

对一般的奇异积分
$$\int_a^b f(x)dx,$$
选取 $g(x)$ 使得它与 $f(x)$ 有相同的零点，但在 $[a,b]$ 上是初等可积的，且 $f-g$ 有必要的光滑性，则

$$\int_a^b f(x)dx = \int_a^b g(x)dx + \int_a^b [f(x)-g(x)]dx.$$

设f具有如下形式，则

$$f(x) = (x-c)^{\alpha}\varphi(x),\ c \in [a,b], -1 < \alpha < 0.$$

φ在$[a,b]$上足够光滑，φ在$x=c$处有Taylor展开，则

$$f(x) = f_1(x) + f_2(x),$$

其中

$$
\begin{aligned}
f_1(x) &= \Big[\varphi(c)(x-c)^\alpha + \frac{1}{1!}\varphi'(c)(x-c)^{\alpha+1} + \frac{1}{2!}\varphi''(c)(x-c)^{\alpha+2} \\
&\quad + \cdots + \frac{1}{k!}\varphi^{(k)}(c)(x-c)^{\alpha+k}\Big] \\
f_2(x) &= (x-c)^\alpha\Big[\varphi(x) - \varphi(c) - \frac{1}{1!}\varphi'(c)(x-c) - \frac{1}{2!}\varphi''(c)(x-c)^2 \\
&\quad - \cdots - \frac{1}{k!}\varphi^{(k)}(x-c)^k\Big]
\end{aligned}
$$

$f_1(X)$ 直接可积，$f_2(x)$ 是连续函数且直到它的$k-1$阶导数亦连续，所以可以用标准的求积公式进行计算。

例

$$\text{计算积分} I = \int_0^{0.5} \frac{1}{\sqrt{x(1-x)}} dx$$

解:被积函数在$x=0$处有奇性，改写为

$$f(x) = x^{-\frac{1}{2}}(1-x)^{-\frac{1}{2}}.$$

则$\alpha = -\frac{1}{2}, c = 0, \varphi(x) = (1-x)^{-\frac{1}{2}}$，Taylor展开有

$$\begin{aligned}
\varphi(x) &= 1 + \frac{1}{2}x + \frac{3}{8}x^2 + \frac{5}{16}x^3 + \frac{35}{128}x^4 + E_n(f), \\
f(x) &= \left[x^{-\frac{1}{2}} + \frac{1}{2}x^{\frac{1}{2}} + \frac{3}{8}x^{\frac{3}{2}} + \frac{5}{16}x^{\frac{5}{2}} + \frac{35}{128}x^{\frac{7}{2}}\right] + \frac{\psi(x)}{\sqrt{x}}, \\
\psi(x) &= \frac{1}{\sqrt{1-x}} - \left(1 + \frac{1}{2}x + \frac{3}{8}x^2 + \frac{5}{16}x^3 + \frac{35}{128}x^4\right).
\end{aligned}$$

因为

$$\psi(0) = 0,$$

所以

$$I = \int_0^{0.5} \left(x^{-\frac{1}{2}} + \frac{1}{2}x^{\frac{1}{2}} + \frac{3}{8}x^{\frac{3}{2}} + \frac{5}{16}x^{\frac{5}{2}} + \frac{35}{128}x^{\frac{7}{2}} \right) = 1.5691585 + I_1$$

其中

$$I_1 = \int_0^{0.5} \frac{\psi(x)}{\sqrt{x}} dx$$

用$n = 10$的复合Simpson求积公式可得$I_1 = 0.0016385$。所以

$$I = \int_0^{0.5} \frac{1}{\sqrt{x(1-x)}} dx = 1.5707970.$$

积分的精确值为$\frac{\pi}{2} = 1.5707693$，方法有很高的精度。

无穷区间积分-区间截断

将无穷区间截去“尾巴”转化为有限区间，若选$R > M$，使得

$$\left|\int_R^{\infty} f(x)dx\right| \le \epsilon,$$

则

$$\begin{aligned} I &= \int_0^{\infty} f(x)dx \\ &= \int_0^{R} f(x)dx + \int_R^{\infty} f(x)dx. \end{aligned}$$

只需要计算第一个积分

$$\int_0^{R} f(x)dx$$

即可。

如计算

$$I=\int_0^{\infty} e^{-x^2}dx$$

由于

$$\int_R^{\infty} e^{-x^2}dx \le \int_R^{\infty} e^{-Rx}dx = \frac{1}{R}e^{-R^2}.$$

取$R=4$，则

$$\frac{1}{4}e^{-R^2}\approx 10^{-8},$$

则

$$I=\int_0^{4} e^{-x^2}dx$$

就满足要求。

变量替换把无穷区间变为有限区间

1. 用变量替换$t=\frac{x}{1+x}$或$t=e^{-x}$可以将区间$[0,\infty)$变换到区间$[0,1]$。
2. 类似用$t=\frac{e^x-1}{e^x+1}$将区间$(-\infty,\infty)$变换到$(-1,1)$。如果变换后的被积函数有界，则可 用正常积分的方法计算，或者会导致反常积分。

例如用替换$t=e^{-x}$可得

$$\int_0^{\infty} f(x)dx=\int_0^1 \frac{f(-\ln t)}{t}dt$$

如果

$$g(t)=\frac{1}{t}f(-\ln t)$$

在$[0,1]$上有界, 就可以用计算正常积分的方法来计算。

$$f(x)=e^{-x}\Rightarrow \int_0^{\infty} e^{-x}dx=\int_0^1 dt=1.$$

数值微分问题的提法

设$f(x)$在$[a,b]$上定义，$a \leq x_0 < x_1 < \cdots < x_n \leq b$ 是$[a,b]$中的$n+1$个节点。数值微分问题的典型提法是：

定义 (数值微分)

已知函数在这些节点上的值$f(x_i), i = 0, 1, \cdots, n$，利用这些函数值求$f$的各阶导数值$f', f'', \cdots$等在$x_i, i = 0, 1 \cdots, n$处的近似值称为数值微分。

常用的构造数值微分的方法主要有下面几种：

1. Taylor展开法；
2. 隐式微分格式；
3. 插值型求导公式；
4. 利用数值积分来构造微分；
5. Richardson外推法。

设节点是等距的，亦即$x_{i+1}=x_i+h, h=\frac{b-a}{n}$，若$f(x)$有$k$阶微商，对$i=1,2,\cdots,n-1$进行Taylor展开有：

$$f(x_{i+1}) = f(x_i)+hf'(x_i)+\cdots+\frac{h^k}{k!}f^{(k)}(\xi_1), \tag{0.14}$$

$$f(x_{i-1}) = f(x_i)-hf'(x_i)+\cdots+\frac{(-h)^k}{k!}f^{(k)}(\xi_2), \tag{0.15}$$

其中$\xi_1\in(x_i,x_{i+1}),\xi_2\in(x_{i-1},x_i)$, 取$k=2,3,4$有

$$\begin{aligned}
f'(x_i) &= \frac{f(x_{i+1})-f(x_i)}{h}-\frac{1}{2}hf''(\xi_1),\\
f'(x_i) &= \frac{f(x_i)-f(x_{i-1})}{h}+\frac{1}{2}hf''(\xi_2),\\
f'(x_i) &= \frac{f(x_{i+1})-f(x_{i-1})}{2h}-\frac{1}{6}h^2f'''(\xi_3),\\
f''(x_i) &= \frac{f(x_{i+1})-2f(x_i)+f(x_{i-1})}{h^2}-\frac{1}{12}h^2f^{(4)}(\xi_4),
\end{aligned}$$

1. 一阶微商的向前差商近似公式

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_i)}{h}; \quad O(h)$$

2. 一阶微商的向后差商近似公式

$$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{h}; \quad O(h)$$

3. 一阶微商的中心差商近似公式

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_{i-1})}{2h}; \quad O(h^2)$$

4. 二阶微商的中心差商近似公式

$$f''(x_i) \approx \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2}. \quad O(h^2)$$

形式上看，步长h越小，数值微分公式的截断误差越小，但是由于舍入误差的影响，h不能取得太小。
若$f(x_{i-1})$,$f(x_{i+1})$分别有舍入误差ϵ_1, ϵ_2，令$\epsilon = \max\{\epsilon_1, \epsilon_2\}$，亦即

$$\tilde{f}(x_{i-1}) = f(x_{i-1}) + \epsilon_1, \tilde{f}(x_{i+1}) = f(x_{i+1}) + \epsilon_2.$$

则

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h}, \tilde{f}'(x_i) = \frac{\tilde{f}(x_{i+1}) - \tilde{f}(x_{i-1})}{2h},$$

忽略h的舍入误差，则由舍入误差带来的计算结果的误差

$$\delta(f'(x_i)) = |f'(x_i) - \tilde{f}'(x_i)| \le \frac{|\epsilon_1| + |\epsilon_2|}{2h} \le \frac{\epsilon}{h}.$$

数值微分公式的稳定性较差，截断误差与舍入误差的总和：

$$E(h) = M\frac{h^2}{6} + \frac{\epsilon}{h},\ M = \max|f'''|, h = h^* = \sqrt[3]{\frac{3\epsilon}{M}}\text{时达到极小}$$

隐式数值微分格式

在(0.14)和(0.15)中取$k=6$并把两式相加就有

$$\frac{f(x_{i+1})-2f(x_i)+f(x_{i-1})}{h^2}=f''(x_i)+\frac{1}{12}h^2f^{(4)}(x_i)+O(h^4). \tag{0.16}$$

在(0.14)和(0.15)中取$k=7$并把两式相减可得

$$f'(x_i)=\frac{f(x_{i+1})-f(x_{i-1})}{2h}-\frac{h^2}{6}f'''(x_i)-\frac{h^4}{120}f^{(5)}(x_i)+O(h^6). \tag{0.17}$$

由(0.16)可得

$$f'''(x_i)=\frac{f'(x_{i+1})-2f'(x_i)+f'(x_{i-1})}{h^2}-\frac{1}{12}h^2f^{(5)}(x_i)+O(h^4),$$

把上式代入(0.17)有

$$f'(x_i)=\frac{f(x_{i+1})-f(x_{i-1})}{2h}-\frac{h^2}{6}\cdot\frac{f'(x_{i+1})-2f'(x_i)+f'(x_{i-1})}{h^2}+O(h^4).$$

略去$O(h^4)$项，对$i=1,\cdots,n-1$，记$f'(x_i)\approx m_i, f_i=f(x_i)$，则

$$m_i=\frac{f_{i+1}-f_{i-1}}{2h}-\frac{1}{6}(m_{i+1}-2m_i+m_{i-1}),\ i=1,2,\cdots,n-1.$$

还需补充两个边界条件，假设$m_0=f'(x_0), m_n=f'(x_n)$，则

$$\begin{cases} m_{i-1}+4m_i+m_{i+1}=\frac{3}{h}(f_{i+1}-f_{i-1}), i=1,2,\cdots,n-1, \\ m_0=f'(x_0),\ m_n=f'(x_n). \end{cases}$$

$$\begin{bmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_{n-2} \\ m_{n-1} \end{bmatrix} = \begin{bmatrix} \frac{3}{h}[f(x_2)-f(x_0)]-f'(x_0) \\ \frac{3}{h}[f(x_3)-f(x_1)] \\ \vdots \\ \frac{3}{h}[f(x_{n-1})-f(x_{n-3})] \\ \frac{3}{h}[f(x_n)-f(x_{n-2})]-f'(x_n) \end{bmatrix}$$

可一次求出所有点的导数，精度高，数值稳定性好。

插值型微分公式

设$f(x)$的n次插值多项式为$L_n(x)$

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

$\omega_{n+1}(x) = \prod_{i=0}^{n}(x - x_i)$，用$L_n'(x)$近似$f'(x)$，即

$$f'(x) = L_n'(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!}\omega_{n+1}'(x) + \frac{1}{(n+1)!} \cdot \frac{\mathrm{d}}{\mathrm{d}x} f^{(n+1)}(\xi(x)) \cdot \omega_{n+1}(x).$$

当$x = x_i$时有

$$f'(x_i) = L_n'(x_i) + \frac{f^{(n+1)}(\xi(x_i))}{(n+1)!}\omega_{n+1}'(x_i),$$

可以用$L_n'(x_i)$近似$f'(x_i)$。

$n=1$,等距节点

$$\begin{aligned} f(x) &= L_1(x)+R_1(x) \\ &= \frac{x-x_1}{x_0-x_1}f(x_0)+\frac{x-x_0}{x_1-x_0}f(x_1)+\frac{f''(\xi(x))}{2!}(x-x_0)(x-x_1), \end{aligned}$$

则

$$\begin{aligned} f'(x_0) &= \frac{f(x_1)-f(x_0)}{x_1-x_0}-\frac{f''(\xi(x_0))}{2!}h, \\ f'(x_1) &= \frac{f(x_1)-f(x_0)}{x_1-x_0}+\frac{f''(\xi(x_1))}{2!}h, \end{aligned}$$

亦即

$$\begin{aligned} \text{向前差商,}\quad f'(x_0) &\approx \frac{f(x_1)-f(x_0)}{x_1-x_0}, \\ \text{向后差商,}\quad f'(x_1) &\approx \frac{f(x_1)-f(x_0)}{x_1-x_0}, \end{aligned}$$

当$n=2$时有

$$\begin{aligned}
L_2(x) &= \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}f(x_0)+\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}f(x_1)\\
&\quad+\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}f(x_2),\\
L_2'(x) &= \frac{2x-x_1-x_2}{2h^2}f(x_0)-\frac{2x-x_0-x_2}{h^2}f(x_1)\\
&\quad+\frac{2x-x_0-x_1}{2h^2}f(x_2),\\
f'(x_i) &= L_2'(x_i)+\frac{1}{3}f^{(3)}(\xi(x_i))\omega_3'(x_i),\\
f'(x_0) &= \frac{-3f(x_0)+4f(x_1)-f(x_2)}{2h}+\frac{2}{3}f^{(3)}(\xi(x_0))h^2,\\
f'(x_1) &= \frac{f(x_2)-f(x_0)}{2h}-\frac{1}{3!}f^{(3)}(\xi(x_1))h^2,\\
f'(x_2) &= \frac{f(x_0)-4f(x_1)+3f(x_2)}{2h}+\frac{2}{3!}f^{(3)}(\xi(x_2))h^2,
\end{aligned}$$

用数值积分求导

设f充分光滑的函数，待求的导数为φ，则

$$f(x_{i+1}) = f(x_{i-1}) + \int_{x_{i-1}}^{x_{i+1}} \varphi(t)dt,\ i = 1, 2, \cdots, n-1.$$

对积分项采用不同的求积公式可得不同的数值微分公式。
对积分项采用中点公式有

$$\int_{x_{i-1}}^{x_{i+1}} \varphi(t)dt = 2h\varphi(x_i) + \frac{1}{24}(2h)^3\varphi''(\xi_i), \xi_i \in (x_{i-1}, x_{i+1}).$$

由此可得中点微分公式

$$f'(x_i) = \varphi(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} - \frac{h^2}{6}f'''(\xi_i).$$

采用Simpson求积公式有

$$\int_{x_{i-1}}^{x_{i+1}} \varphi(t)dt = \frac{h}{3}[\varphi(x_{i-1}) + 4\varphi(x_i) + \varphi(x_{i+1})] - \frac{h^5}{90}f^{(5)}(\xi_i).$$

略去高阶项，记$m_i = \varphi(x_i) = f'(x_i)$，则可得Simpson数值微分公式

$$m_{i-1} + 4m_i + m_{i+1} = \frac{3}{h}[f(x_{i+1}) - f(x_{i-1})],\ i = 1, 2, \cdots, n-1.$$

若在端点给定$m_0 = f'(x_0), m_n = f'(x_n)$，即得前面推导的隐式微分公式。若端点的导数值未知，对m_1和m_{n-1}用中点微分公式近似就有

$$\begin{cases} m_1 = \frac{1}{2h}[f(x_2) - f(x_0)], \\ m_{i-1} + 4m_i + m_{i+1} = \frac{3}{h}[f(x_{i+1}) - f(x_{i-1})], i = 2, 3, \cdots, n-2, \\ m_{n-1} = \frac{1}{2h}[f(x_n) - f(x_{n-2})]. \end{cases}$$

中点微分公式

$$f'(x) \approx G(h) = \frac{1}{2h}[f(x+h) - f(x-h)],$$

利用Taylor展开有

$$f'(x) - G(h) = \alpha_1 h^2 + \alpha_2 h^4 + \cdots,$$

利用Richardson外推公式可得

$$\begin{cases} G_1(h) = G(h), \\ G_{k+1}(h) = \dfrac{4^k G_k(\frac{h}{2}) - G_k(h)}{4^k - 1}, k = 1, 2, \cdots. \end{cases}$$

Richarson外推方法的误差为

$$f'(x) - G_k(h) = O(h^{2(k+1)}).$$