

Chapter 4 数值积分

张亚楠*

问题提出

如何“简单、准确、高效地”计算上述定积分？

$$\int_a^b f(x) dx = \lim_{n \rightarrow \infty} \sum_{j=0}^n f(x_j) \Delta x_j.$$

“理想、完美”答案：Newton-Leibniz公式

$$\int_a^b f(x) dx = F(b) - F(a).$$

如果无法找到原函数或者原函数过于复杂不便计算，怎么办？

0.1 数值积分的几种思路

- 积分中值定理:

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

- 近似被积函数

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

- 积分的区域可加性：复化求积分公式

$$\int_a^b f(x) dx = \int_a^c f(x) dx + \int_c^b f(x) dx = \dots$$

- 进一步提高效率：自适应求积

例 1 左矩形，右矩形公式，中点公式，梯形公式

$$\int_a^b f(x) dx = f(\xi)(b-a) \approx \begin{cases} f(a)(b-a) \\ f(b)(b-a) \\ f(\frac{a+b}{2})(b-a) \\ \frac{f(a)+f(b)}{2}(b-a) \end{cases}$$

* ynzhang@suda.edu.cn 苏州大学数学科学学院

可利用Taylor展开推导梯形公式的截断误差

$$f(x) = f(a) + f'(a)(x-a) + f''(\xi_1)(x-a)^2$$

$$f(x) = f(b) + f'(b)(x-b) + f''(\xi_2)(x-b)^2$$

$$\begin{aligned} \text{Error} &= \int_a^b \frac{1}{2} \left[f'(a)(x-a) + f'(b)(x-b) + f''(\xi_1)(x-a)^2 + f''(\xi_2)(x-b)^2 \right] dx \\ &= \frac{1}{2} \left[(f'(a) - f'(b)) \frac{(b-a)^2}{2} + C_1 * (b-a)^3 \right] \\ &= C_2 * (b-a)^3 \end{aligned}$$

常数 C_2 与 $f(x)$ 二阶导数有关，利用微分中值定理和介值定理可精确估计。其它几个矩形公式的误差推导留作习题。

1 插值型求积和代数精度

对被积函数 $f(x)$ 考虑多项式 $p_n(x)$ 插值，进而对容易积分的多项式精确求解

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

定义 1 给定 $n+1$ 个节点 $\{x_j\}$ ，求积公式

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

A_k 称为求积系数， x_j 求积节点；如果

$$A_k = \int_a^b l_k(x) dx$$

则称求积公式是插值型的。

思考：如果 $f(x)$ 本身是次数不超过 n 的多项式，上述求积公式误差是多少？ **进一步思考：**

若一求积公式对所有次数不超过 n 的多项式精确成立， n 是否越大越好？如果是的话，如何描述“好的程度”？

定义 2 如果某个求积公式对所有次数不超过 m 的代数多项式精确成立，但是对至少一个 $m+1$ 阶多项式不精确成立，则称该公式有 m 次代数精度(Order of exactness)。

如何刻画求积公式对所有次数不超过 n 的代数多项式精确成立？

$$H_n = \text{span}\{1, x, x^2, \dots, x^n\}$$

只要数值积分公式对 H_n 中的一组基准确成立，则该积分公式至少具有 n 次代数精度。

验证：中矩形公式和梯形公式具有一次代数精度。梯形公式作为例题，中矩形公式作为课堂练习。李庆扬教材 p100, 例1

Theorem 1 形如(1)的求积公式具有至少 n 次代数精度 \Leftrightarrow 该公式是插值型的。

证明：充分性略；必要性：若求积公式

$$I_n = \sum_{k=0}^n A_k f(x_k) \quad (1)$$

具有至少 n 次代数精度，则对特殊的 n 次多项式 - Lagrange 插值基函数 $l_k(x)$ 精确成立。即，

$$\int_a^b l_k(x) dx = \sum_{j=0}^n A_j l_k(x_j) = A_k$$

上式用到 $l_k(x_j) = \delta_{kj}$ 果然是插值型的。

定义 3 若

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

其中 $h = \max_j |x_{j+1} - x_j|$ 则称该积分公式收敛。

定义 4 $\forall \varepsilon > 0$, 若 $\exists \delta > 0$, 只要 $|f(x_k) - \tilde{f}_k| < \delta$, 就有

$$|I_n(f) - I_n(\tilde{f})| = \left| \sum_k A_k [f(x_k) - \tilde{f}_k] \right| < \varepsilon$$

则称该积分公式是稳定的。

Theorem 2 若求积系数 A_k 非负，则该求积公式是稳定的。

简单证明并举例说明高阶 Newton-Cotes 公式不适合实际计算。李庆扬教材 p104 表 4-1

算法三要素——算法好坏的衡量标准

- 1) 计算复杂度(计算量)——computational complicity 或者 computational cost
- 2) 稳定性——输出数据对输入数据(的扰动)的依赖程度
- 3) 收敛性—— n (自由度个数，比如求积(插值)节点个数)趋于无穷时，数值解是否收敛到精确解

1.1 代数精度与积分余项之间的关系(略)

标注 1.1 若如下求积公式 具有 m 次代数精度, 则其求积公式余项必可表达为如下形式:

$$R[f] = \int_a^b f(x) dx - \sum_{k=0}^n A_k f(x_k) = K \cdot f^{(m+1)}(\eta)$$

其中 K 是与 f 无关的待定参数, $\eta \in (a, b)$.

说明: 该结论在李庆扬《数值分析》第五版第101页(1.8)式, 结论不对。刘长剑给出以下反例!

设 $m = n = 0$, 考查积分公式 (结点和系数待定)

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

因为该公式有零次代数精度, 也即 $f(x) = 1$ 时, 成立

$$\int_{-1}^1 1 dx = A \cdot 1 \Rightarrow A = 2.$$

即

$$\int_{-1}^1 f(x) dx \approx 2f(x_0)$$

若李的陈述成立, $\int_{-1}^1 f(x) dx - 2f(x_0) = K \cdot f'(\eta)$

令 $f(x) = x$, 计算 K

$$\int_{-1}^1 x dx - 2x_0 = K \cdot f'(\eta) \Rightarrow \underbrace{-2x_0 = K}$$

验证 $f(x) = x^2$,

$$\int_{-1}^1 x^2 dx - 2x_0^2 = (-2x_0) \cdot 2\eta \Rightarrow |2/3 - x_0^2| \leq |4x_0|$$

取 $x_0 = \varepsilon$ 可得上式不成立。矛盾!!

例 2 数值求积公式

$$\int_{-1}^1 f(x) dx \approx 2f(0.01)$$

具有0次代数精度, 且满足 $m = n = 0$. 则由李教材(1.8-1.9)结论

$$R(f) = \int_{-1}^1 f(x) dx - 2f(0.01) = -2(0.01)f'(\eta)$$

取 $f(x) = x^2$, 得

$$\frac{2}{3} - 0.0002 = -0.02 \cdot 2\eta, \quad \eta \in (-1, 1)$$

不存在满足条件的 η 使得上式成立, 矛盾!

2 Newton-Cotes公式及其误差

由插值性求积公式的定义可知: 一旦求积节点 x_j 给定, 则求积系数 A_k 唯一确定, 进而求积公式唯一确定。设计插值性求积公式的唯一目标即是: 选择合适的求积节点! Newton-cotes公式给出了最简单的节点选择方式: 在积分区间均匀选择 $(n+1)$ 个点 (包含端点)

$$\text{Newton-Cotes} \quad \int_a^b f(x) dx \approx \sum_{k=0}^n f(x_k) A_k = \sum_{k=0}^n f(x_k) \int_a^b \ell_k(x) dx, \quad x_k = a + k \frac{b-a}{n}$$

其中 $\ell_k(x)$ 是Lagrange插值基函数。较为常用的是 $n = 1, 2$, 此时也称为梯形公式和Simpson公式。

2.1 Newton-Cotes公式的求积系数

求积系数有显式表达式

$$A_k = \int_a^b \ell_k(x) dx = \int_a^b \frac{\prod_{j=0, j \neq k} (x - x_j)}{\prod_{j=0, j \neq k} (x_k - x_j)} dx \quad (2)$$

记 $x_j = a + jh$, $h = \frac{b-a}{n}$, 代入可得

$$A_k = \frac{1}{h^n \prod_{j=0, j \neq k} (k-j)} \int_a^b \prod_{j=0, j \neq k} (x - x_j) dx \quad (3)$$

写出以上被积函数的原函数仍然是麻烦的. 根据代数精度理论可知: 上述公式对次数不超过 n 的多项式精确成立.

$$\sum_{k=0}^n (x_k)^j A_k = \int_a^b x^j dx = \frac{b^{j+1} - a^{j+1}}{j+1}, \quad j = 0, 1, \dots, n$$

求解上述线性方程组, 可得 A_k .

```
1 function [x,w] = newtonquad(n,a,b)
2 if nargin<2, a = -1; b = 1; end
3 x = linspace(a,b,n+1)'; A = x.^(0:n);
4 j1 = (1:n+1)'; r = (b.^j1 - a.^j1)./(j1);
5 w = A' \ r;
```

可以检验当 $n \leq 7$ 时, 上述求积系数为正. 当 $n \geq 8$ 时求积系数出现负数, 此时求积公式不稳定. 上述利用代数精度计算求积系数的方法适用于任何插值型公式. 但是需要注意: 若 $n \gg 10$ 时, 所求线性方程组的系数矩阵Vandermonde矩阵时高度病态的. 应当使用正交多项式做测试函数(见后文Gauss积分公式的系数计算).

例 3 检验梯形公式($n=1$ 时的Newton-Cotes公式, $x_0=a$, $x_1=b$)的求积系数

$$A_0 = A_1 = \frac{b-a}{2}$$

并检验其具有1次代数精度

例 4 取 $x_0=a$, $x_1=\frac{a+b}{2}$, $x_2=b$, (三点等距插值 $n=2$)

1) 推导Simpson公式

$$\int_a^b f(x) dx \approx \sum_{k=0}^2 f(x_k) A_k$$

求积系数

$$A_0 = A_2 = \frac{b-a}{6}, \quad A_1 = \frac{2}{3}(b-a)$$

2) 验证其代数精度为3

留作课堂练习或研究生作业.

2.2 Newton-Cotes积分公式的误差估计

插值型求积公式的余项

$$R[f] = \int_a^b f(x)dx - \int_a^b p_n(x)dx = \int_a^b \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{j=0}^n (x-x_j)dx$$

n = 1时对应梯形公式，根据插值余项估计

$$\begin{aligned} I(f) - T(f) &= \int_a^b \frac{f''(\xi)}{2} (x-a)(x-b)dx \\ &= \frac{f''(\eta)}{2} \int_a^b (x-a)(x-b)dx = -\frac{f''(\eta)}{2} \frac{(b-a)^3}{6} \end{aligned}$$

梯形公式误差

$$I(f) - T(f) = -\frac{f''(\eta)}{12} (b-a)^3$$

思考：

- 1) 二次插值所得积分公式为何对三次多项式精确成立？
- 2) Simpson公式对所有三次多项式精确成立；却只有三个插值节点？是否矛盾？
- 3) 若不矛盾，如何解释比较精细地分析其误差？

Hint: 考虑 Hermite插值及其余项 记

$$a, x_1 = (a+b)/2, b, x_1$$

为插值节点，其中 x_1 是重节点，则积分公式的Newton型表达式如下

$$\int_a^b \left[f(a) + f[a, x_1](x-a) + f[a, x_1, b](x-a)(x-x_1) + f[a, x_1, b, x_1](x-a)(x-x_1)(x-b) \right] dx$$

上式红色部分的积分为零，即三次多项式所得积分公式与Simpson公式一样。

Hermite插值余项有如下形式

$$\frac{f^{(4)}(\xi)}{4!} (x-a)(x-\frac{a+b}{2})^2(x-b)$$

划线部分不变号，类似梯形公式可用积分中值定理估计。

$$\begin{aligned} I(f) - I(S) &= \int_a^b \left[\frac{f^{(4)}(\xi)}{4!} (x-a)(x-\frac{a+b}{2})^2(x-b) \right] dx = \\ &= \frac{f^{(4)}(\zeta)}{4!} \int_a^b (x-a)(x-\frac{a+b}{2})^2(x-b)dx = \frac{f^{(4)}(\zeta)}{4!} \frac{-1}{120} (b-a)^5 \end{aligned}$$

Simpson 公式误差

$$I(f) - I(S) = \frac{-f^{(4)}(\zeta)}{2880}(b-a)^5$$

标注 2.1 若考虑三点做二次插值, 余项估计比较麻烦

$$\frac{f^{(3)}(\eta)}{3!} \frac{(x-a)(x-\frac{a+b}{2})(x-b)}{1}$$

横线部分在 $[a, b]$ 积分为零, 无法使用积分中值定理, 整体积分无法估计。

Theorem 3 当 n 为偶数时, Newton-Cotes 公式对 $f(x) = x^{n+1}$ 精确成立

证明:

$$R[f] = \int_a^b \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{j=0}^n (x-x_j) dx = \int_a^b \prod_{j=0}^n (x-x_j) dx = 0$$

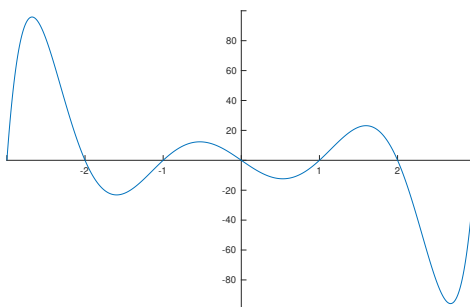


Figure 1: 以偶数次 Newton-Cotes 公式积分节点为零点的多项式是对称区间上的奇函数, 积分为零

标注 2.2 Newton-Cotes 公式当 $n \rightarrow \infty$ 时发散, 实际应用中当 $n \leq 6$ 可结合复合求积公式使用, 效果佳. $n \geq 8$ 不稳定, 不可用.

3 复化求积公式

3.1 复合梯形公式

将 $[a, b]$ 进行 n 等分, 记

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

$$I(f) = \int_a^b f(x) dx = \sum_{j=0}^{n-1} \int_{x_j}^{x_{j+1}} f(x) dx = \frac{h}{2} \sum_{j=0}^{n-1} [f(x_j) + f(x_{j+1})] + R_n(f)$$

记

$$T_n = \frac{f(a) + f(b)}{2} h + h \sum_{j=1}^{n-1} f(x_j)$$

为复合梯形公式, 余项

$$R_n(f) = \sum_{j=0}^{n-1} \frac{-h^3}{12} f''(\eta_j)$$

由介值定理 $\exists \eta$, s.t.,

$$f''(\eta) = \frac{1}{n} \sum_{j=0}^{n-1} f''(\eta_j)$$

则

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

思考：复合梯形公式的计算复杂度、稳定性、收敛性如何？

3.2 复合Simpson公式

将区间[a,b]进行2n等分，记 $x_j = a + jh$

$$\int_a^b f(x) dx = \sum_{j=0}^{n-1} \int_{x_{2j}}^{x_{2j+2}} f(x) dx \approx \frac{h}{3} \sum_{j=0}^{n-1} [f(x_{2j}) + 4f(x_{2j+1}) + f(x_{2j+2})]$$

得复合Simpson公式

$$S(f) = \frac{h}{3} \left[f(a) + f(b) + 2 \sum_{j=0}^{n-1} f(x_{2j}) + 4 \sum_{j=0}^{n-1} f(x_{2j+1}) \right]$$

相应误差分析

$$R_n(S) = \sum_{j=0}^{n-1} \frac{f^{(4)}(\zeta_j)}{4!} \frac{-1}{120} (2h)^5 = \frac{-1}{120 * 4!} (2h)^5 * n * f^{(4)}(\zeta) = \frac{-1}{120 * 4!} (2h)^4 * (b-a) * f^{(4)}(\zeta)$$

即

$$R_n(S) = I(f) - S_n(f) = \frac{-(b-a)}{180} f^{(4)}(\zeta) * h^4$$

思考：复合Simpson公式的计算量、稳定性和收敛性如何？

例3 教材108页：Simpson 公式具有较高的精度！

标注 3.1 对于光滑函数，Simpson公式的误差是 $O(h^4)$ 。习惯上，我们称这类具有4阶精度算法为高精度算法。例如： $h=1/100$ 时，可以得到 10^{-8} 量级的误差。由于其简单、有效，复合Simpson公式一直很受欢迎。

3.3 复合梯形公式的超收敛性

利用复合梯形公式计算光滑的周期函数可得高精度数值解¹。这一结论可由Euler-Maclaurin公式给出。关于E-M公式与梯形公式的关系，我们放在本节后面²，仅供读者参考。

我们可以从三角插值的观点来看待梯形公式的超收敛性，首先给出一个结论

¹Trefethen, The Exponentially Convergent Trapezoidal Rule, 2014 SIREV(56)385 – 458

²关于E-M公式，可参考谢惠民老师的《数学分析讲义》及《数学分析习题课讲义》；另：关于谢老师这两本著名的、难度很大的书籍，建议：学习特别有余力的同学阅读。自视不高的同学建议先打好基础。

例 5 将 $[0, 2\pi)$ 进行 m 等分, 记 $h = 2\pi/m$, $x_j = j \cdot h$. 则对任意正整数 $k \leq m$, 成立

$$\sum_{j=0}^m e^{\pm i k x_j} = 0, \quad m \geq k \geq 1$$

进而有

$$\sum_{j=0}^m \sin(k x_j) = \sum_{j=0}^m \cos(k x_j) = 0$$

考虑对周期函数 $f(x)$ 进行三角插值

$$f(x) \approx P_n(x) := a_0 + \sum_{k=1}^n a_k \sin kx + b_k \cos kx = \sum_{k=-n}^n c_k e^{i k x}$$

$$\int_0^{2\pi} f(x) dx \approx \int_0^{2\pi} P_n(x) dx = 2\pi \cdot c_0 = \frac{2\pi}{2n+1} \sum_{j=0}^{2n} f(x_j) = h \sum_{j=0}^{2n} f(x_j)$$

倒数第二个等号用到了三角插值系数表达式

$$c_k = \frac{1}{2n+1} \sum_{j=0}^{2n} f(x_j) e^{-i k j \frac{2\pi}{2n+1}}$$

标注 3.2 以上说明: 对周期函数 $f(x)$ 作用梯形公式(或中点公式), 等价于先对其进行三角插值得到 $P_n(x)$, 再对 $P_n(x)$ 精确计算。同时, 上例还说明, 梯形公式对于 $\sin kx, \cos kx$ 在 $[0, 2\pi)$ 内是精确求积的。

Theorem 4 (Euler-Maclaurin 公式) For $g \in C^{2m+2}[0, N]$ (g is sufficiently many times continuously differentiable over $[0, N]$):

$$\begin{aligned} & \frac{1}{2}g(0) + g(1) + \cdots + g(N-1) + \frac{1}{2}g(N) \\ &= \int_0^N g(t) dt + \sum_{l=1}^m \frac{B_{2l}}{(2l)!} \left(g^{(2l-1)}(N) - g^{(2l-1)}(0) \right) + \frac{B_{2m+2}}{(2m+2)!} N g^{(2m+2)}(\alpha), \quad \alpha \in [0, N] \end{aligned} \quad (1)$$

To see the connection with the composite trapezoidal rule, we make a change of coordinates:

$$[0, N] \rightarrow [a, b]: t \mapsto x = a + ht, \quad h = \frac{b-a}{N}, \quad dx = h dt, \quad \text{so } \int_0^N g(t) dt = \int_a^b f(x) \frac{1}{h} dx \quad (2)$$

To replace the derivatives of $g(t)$ in (1), we observe $g(t) = f(a + ht) = f(x)$ and apply the chain rule:

$$g'(t) = f'(x)h \quad \text{and for any } l: g^{(l)}(t) = f^{(l)}(x)h^l$$

After executing the coordinate change (2), formula (1) turns into

$$\begin{aligned} & \frac{1}{2}f(a) + f(a+h) + \cdots + f(b-h) + \frac{1}{2}f(b) = \int_a^b f(x) \frac{1}{h} dx \\ & + \sum_{l=1}^m \frac{B_{2l}}{(2l)!} h^{2l-1} \left(f^{(2l-1)}(b) - f^{(2l-1)}(a) \right) + \frac{B_{2m+2}}{(2m+2)!} N h^{2m+2} f^{(2m+2)}(\beta), \quad \beta \in [a, b] \end{aligned} \quad (4)$$

Multiplying (4) by h , we obtain the error formula for the composite trapezoidal rule $T(h)$:

$$T(h) = \int_a^b f(x) dx + \sum_{l=1}^m \frac{B_{2l}}{(2l)!} h^{2l} \left(f^{(2l-1)}(b) - f^{(2l-1)}(a) \right) + \frac{B_{2m+2}}{(2m+2)!} (Nh) h^{2m+2} f^{(2m+2)}(\beta)$$

4 Romberg 求积公式

作为Richardson外推法的一个重要例子，Romberg积分在计算光滑函数积分时可以达到极好的效果（大优于Simpson）。由于其容易理解，算法实现简单，Romberg积分也是广泛使用的一个方法。本质上，Romberg积分公式是复合4阶Newton-Cotes公式。教材单列一节，另一个目的是为了引出数值计算中的一个重要技巧——Richardson外推法。

复合梯形公式有如下误差表达式（参考E-M公式）

$$I(f) - T_n(f) = \beta_1 h^2 + \beta_2 h^4 + \beta_3 h^6 + \beta_4 h^8 + \dots$$

思考：能否利用上式，结合 T_n, T_{2n} 的结果得到更为精确的积分值？

例 6 检验Simpson公式可由梯形公式组合得到

$$S(f) = \frac{4}{3} T_2(f) - \frac{1}{3} T_1(f)$$

进而得到Simpson公式的误差估计式

$$I(f) - S_n(f) = \gamma_1 h^4 + \gamma_2 h^6 + \gamma_3 h^8 + \dots$$

思考：能否利用类似外推思想，借助于Simpson公式构造更高精度的求积公式？

Romberg积分

$$R_n(f) = \frac{16}{15} S_{2n}(f) - \frac{1}{15} S_n(f) = S_{2n}(f) + \frac{1}{15} [S_{2n}(f) - S_n(f)]$$

为了避免重复计算，Romberg积分的算法通常以如下过程实现

$$\begin{cases} T_1(f) \\ T_2(f) \\ T_3(f) \end{cases} \begin{matrix} \swarrow \\ \swarrow \\ \swarrow \end{matrix} \Rightarrow \begin{cases} S_1(f) \\ S_2(f) \end{cases} \Rightarrow R(f)$$

```
1 % Romberg integral by trapezoid 2021 11
2 function [R, S2,T3] = myromberg(f,a,b,n)
3 %% the number of nodes is 4n
4 %% === Trapezoidal rule
5 h = (b-a)/n; x = a:h:b;
6 fx = f(x); T1 = h/2*(fx(1) + fx(n+1)) + h*sum(fx(2:n));
7 %% ===== refine
8 h = h / 2; x = a + h: 2*h : b - h;
9 fx = f(x); T2 = T1 / 2 + sum(fx)*h;
10 %% ===== refine twice
```

```

11 h = h / 2; x = a + h: 2*h : b - h;
12 fx = f(x); T3 = T2 / 2 + sum(fx)*h;
13 % % === simpson
14 S1 = T2 + (T2 - T1)/3; S2 = T3 + (T3 - T2)/3;
15 % % === romberg
16 R = S2 + (S2 - S1)/15;
17 end

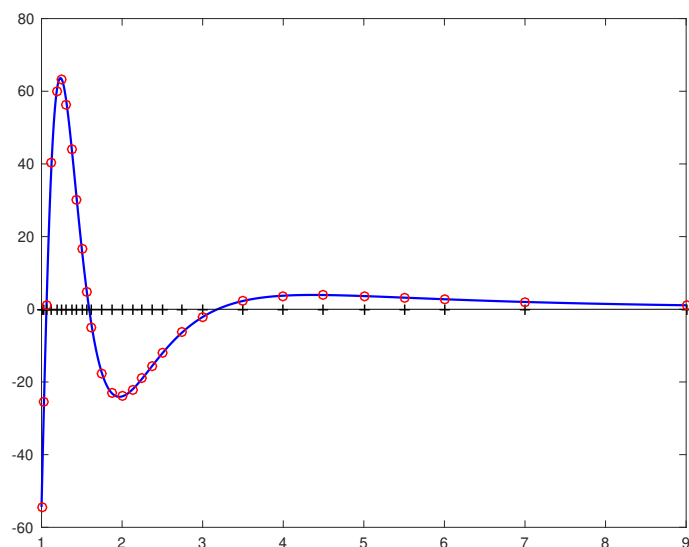
```

5 自适应求积

例 7 考查积分

$$\int_a^b \frac{100}{x^2} \sin\left(\frac{10}{x}\right) dx$$

被积函数如图所示



- (1) 怎样合理的分布求积节点?
- (2) 图中的节点如何自动选取?

自适应算法在现代计算科学中非常重要，它以较少的计算节点实现了复杂问题的计算。避免了由于节点过多带来的灾难性计算困难。国内学者和一些研究机构在自适应算法如自适应有限元等领域有较多的研究 (***)。自适应积分的算法实现和理论分析无疑是相对复杂的、困难的，但是其算法思想确实容易理解的。本节的目标是介绍自适应数值积分的算法实现和指出其关键步骤。

5.1 自适应Simpson公式

思考：在构造自适应积分时，哪一步是关键？后验误差！如何快速有效的计算后验误差？以Simpson公式为例：

$$I(f) - S_n(f) = \alpha_1 h^4 + \alpha_2 h^6 + \mathcal{O}(h^8)$$

$$I(f) - S_{2n}(f) = \alpha_1 \left(\frac{h}{2}\right)^4 + \alpha_2 \left(\frac{h}{2}\right)^6 + \mathcal{O}(h^8)$$

第二式乘以16减去第一式得到：

$$15 \left[I(f) - S_{2n}(f) \right] - \left[S_{2n}(f) - S_n(f) \right] = \mathcal{O}(h^6)$$

则 $\frac{1}{15} [S_{2n}(f) - S_n(f)]$ 可作为 $S_{2n}(f)$ 的后验误差。我们给出自适应 Simpson 求积公式的测试代码。读者可以检验，对于本节给出的例子，自适应算法比Romberg积分有明显优势。

```
1 function [Q,fcount,PTS] = myadapt_simp(F,a,b,tol)
2 % a test code for adaptive numerical integral
3 % for example:
4 % f = @(xx) 100./xx.^2.*sin(10./xx); a = 0.7; b = 5; Tol = 1e-2;
5 % [Qs,fcount,PTS] = myadapt_simp(f,a,b,Tol) ;
6 % figure(1); fplot(f,[a,b],'b-','LineWidth',1);
7 % hold on, plot([a;b],[a;b]*0,'k-','LineWidth',1)
8 % plot(PTS,f(PTS),'ro'); plot(PTS,PTS*0,'k*'); hold off
9
10 % Recursive subfunction used
11 Xnodes = linspace(a,b,5)'; fX = F(Xnodes);
12 W1 = (b-a)/6*[1,4,1]; W2 = (b-a)/12*[1,4,2,4,1];
13 Q1 = W1*fX([1,3,5]); Q2 = W2*fX;
14 PTS = Xnodes; c = Xnodes(3);
15 if abs(Q2 - Q1) <= 15*tol % || abs(b-a) < 1e-4
16 Q = Q2 + (Q2-Q1)/15; fcount = 4;
17 else
18 [Qa,ka,PTS1] = myadapt_simp(F,a,c,tol);
19 [Qb,kb,PTS2] = myadapt_simp(F,c,b,tol);
20 Q = Qa + Qb; fcount = ka + kb;
21 PTS = [PTS; PTS1; PTS2]; PTS = uniquetol(PTS,1e-11);
22 end
```

6 Gauss 求积

前几节关于数值积分的思路是基于求积节点均匀分布在积分区间上的，包括自适应Simpson公式（分段均匀分布）。实际上，关于数值积分的现代科学计算方法都是基于Gauss型的。作为插值性求积公式，Gauss型求积几乎具有算法的所有优点：计算简单、稳定、收敛、精度高(误差小)、代数精度最高；是数值积分的最优算法。本节目标：了解Gauss积分的思想，掌握构造一

般带权积分的Gauss积分公式的算法。会使用教材中几类特殊的求积公式（包括奇异积分，无穷区间积分），并会检验查表得到数据的准确性(教材中有 错误且所给数据精度不高)。

6.1 Gauss求积的一般原理

给定如下数值积分公式

$$\int_a^b f(x)dx \approx \sum_{k=0}^n A_k f(x_k) \quad (4)$$

求积节点和求积系数待定；针对 $(2n+2)$ 个待定参数，可以期望该公式有 $(2n+1)$ 次代数精度；即对

$$f(x) = \{1, x, x^2, \dots, x^{2n+1}\}$$

精确成立。

可以证明：代数精度不会达到 $(2n+2)$ ； $f(x)$ 取如下 $(2n+2)$ 阶多项式

$$f(x) = \prod_{j=0}^n (x - x_j)^2$$

则

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

一般的带权积分的数值积分公式

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

定义 5 如果上述求积公式具有 $(2n+1)$ 次代数精度，则称其节点 x_k 为Gauss点，相应公式称为高斯型求积公式。

标注 6.1 积分公式(4)最高具有 $(2n+1)$ 次代数精度。从定义出发推导高斯求积公式比较复杂；例如直接推导两点高斯公式涉及非线性方程组的求解，技巧性强，计算并不容易。

若求积节点 x_j 是积分区间 $[a, b]$ 上的正交多项式的零点，则

$$\omega(x) = \prod_{j=0}^n (x - x_j)$$

$\omega(x)$ 是区间 $[a, b]$ 上的首项系数为1的 $(n+1)$ 次带权 $\rho(x)$ 的正交多项式。由插值余项可知

$$\begin{aligned} E(f) &= \int_a^b f(x)\rho(x)dx - \sum_{k=0}^n A_k f(x_k) \\ &= \frac{1}{(n+1)!} \int_a^b f^{(n+1)}(\xi(x))\omega(x)\rho(x)dx \\ &= \int_a^b \rho(x)f[x_0, x_1, \dots, x_n, x]\omega(x)dx \end{aligned}$$

观察上式右端, 当

$$f(x) = x^m, m = 0, 1, \dots, n; \Rightarrow f^{(n+1)} = 0$$

求积公式精确成立. 当

$$m = n+1, n+2, \dots, 2n+1$$

$f[x_0, x_1, \dots, x_n, x]$ 是次数不超过你 n 的多项式; 与 $\omega(x)$ 带权正交; 则 $E(f)$ 为零。

Theorem 5 插值型求积公式的求积节点是高斯点的充要条件是 $\omega(x)$ 与任何次数不超过 n 的多项式带权 $\rho(x)$ 正交. 此时所得积分公式称之为 Gauss 求积公式, 且其具有 $(2n+1)$ 代数精度

例 8 确定求积公式

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

的系数及节点使其具有最高代数精度.

解答: 根据定理结论, 需要构造 Gauss 型积分公式, 权函数 $\rho(x) = x^{\frac{1}{2}}$ 令

$$\omega(x) = (x - x_0)(x - x_1) = x^2 + bx + c$$

则 $\omega(x)$ 与 $1, x$ 带权正交, 也即:

$$\begin{aligned} \int_0^1 \sqrt{x} \cdot 1 \cdot \omega(x) dx &= \int_0^1 \sqrt{x} \cdot 1 \cdot (x^2 + bx + c) dx = 0 \\ \int_0^1 \sqrt{x} \cdot x \cdot \omega(x) dx &= 0 \end{aligned}$$

解得:

$$b = -10/9, \quad c = 5/21$$

进而求出: x_0, x_1 ;

利用该公式有 3 次代数精度, 对 $f(x) = 1, x$ 是精确的

$$\begin{cases} A_0 + A_1 = \int_a^b \sqrt{x} \cdot 1 dx \\ A_0 x_0 + A_1 x_1 = \int_a^b \sqrt{x} \cdot x dx \end{cases}$$

进而求出 A_k 即可.

标注 6.2 读者应当注意: 该例子不仅仅给出了 $[0, 1]$ 区间上带权 $\rho(x) = \sqrt{x}$ 的两点 Gauss 公式, 同时也给出任意区间, 带任意权函数的、有 n 个节点的一般算法。

6.2 Gauss 公式的误差估计、稳定性

利用 $f(x)$ 在结点 x_j 的函数值及其导数值, 构造 Hermite 插值 $H_{2n+1}(x)$, 满足:

$$H(x_j) = f(x_j), \quad H'(x_j) = f'(x_j), \quad 0 \leq j \leq n$$

由插值余项可得:

$$f(x) - H(x) = \frac{f^{(2n+2)}(\xi_x)}{(2n+2)!} \omega^2(x)$$

其中

$$\omega(x) = \prod_{j=0}^n (x - x_j).$$

上式乘以权函数并积分可得积分余项

$$R_G(f) = \int_a^b \rho(x) f(x) dx - \int_a^b \rho(x) H(x) dx = \int_a^b \frac{f^{(2n+2)}(\xi_x)}{(2n+2)!} \rho(x) \omega^2(x) dx = \frac{f^{(2n+2)}(\eta)}{(2n+2)!} \int_a^b \rho(x) \omega^2(x) dx$$

一般情况下 $\omega(x)$ 的最大值很小, 例如Legendre和Chebyshev多项式零点 x_j 满足

$$\|\omega(x)\|_{\infty} \approx 2^{-n}$$

Gauss积分余项表明: Gauss求积公式误差小, 收敛快!

标注 6.3 Gauss 公式都是收敛的: $R(f) \rightarrow 0$ as $n \rightarrow \infty$. 注意: 此时Newton-Cotes公式是发散的。

Theorem 6 Gauss求积公式都是稳定的.

证明: 只需要说明求积系数是非负的即可. 取Lagrange插值基函数

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

则 $\ell_k^2(x)$ 是次数为 $2n$ 的多项式, 进而

$$0 < \int_a^b \rho(x) \ell_k^2(x) dx = \sum_{j=0}^n A_j \ell_k^2(x_j) = A_k$$

6.3 Legendre - Gauss 积分公式

当权函数 $\rho(x) = 1$ 时, 利用Legendre 多项式在 $[-1, 1]$ 区间的正交性可知: 取求积节点 x_k 为 Legendre多项式 $P_{n+1}(x)$ 的零点, 求积系数

$$A_k = \int_{-1}^1 \ell_k(x) dx = \int_{-1}^1 \frac{\prod_{j \neq k} (x - x_j)}{\prod_{j \neq k} (x_k - x_j)} dx$$

即可得到Gauss积分公式:

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

回顾³: 正交多项式的根可通过计算对称三对角矩阵的特征值得到。一旦得到求积节点, 则求积系数 A_k 唯一确定。

计算求积节点和求积系数可分以下两步完成

1. 给定节点个数, 计算相应Legendre 多项式的根 (T的特征值); 得到 x_k ,

$$T = \begin{bmatrix} 0 & \gamma_1 & 0 & 0 & 0 \\ \gamma_1 & 0 & \gamma_2 & 0 & 0 \\ 0 & \gamma_2 & 0 & \gamma_3 & 0 \\ 0 & 0 & \gamma_3 & 0 & \gamma_4 \\ 0 & 0 & 0 & \gamma_4 & 0 \end{bmatrix}, \quad \gamma_j = \sqrt{\frac{1}{4 - 1/j^2}}$$

³见第三章讲义

2. 计算求积系数 A_k （借助于表达式或者利用其具有至少 n 次代数精度求解对应线性方程组）

$$\sum_{k=0}^n (x_k)^j A_k = \int_{-1}^1 x^j dx = \begin{cases} 2/(j+1), & \text{even} \\ 0, & \text{odd} \end{cases}, \quad j = 0, 1, \dots, n$$

注意利用上式求解 A_k 的线性方程组其系数矩阵是Vandermonde矩阵，坏条件数的；当 n 较大时， A_k 计算不准确。此时，应当选择 Legendre多项式 $P_j(x)$ 检验代数精度，即令 A_k 满足：

$$\sum_{k=0}^n P_j(x_k) A_k = \int_{-1}^1 P_j(x) dx = \begin{cases} 2, & j = 0 \\ 0, & \text{otherwise} \end{cases}, \quad j = 0, 1, \dots, n$$

系数矩阵的第 j 行对应 $P_j(x)$ 在积分节点处的取值； $P_j(x)$ 在连续意义下正交，离散情形下近似正交，条件数很小， A_k 计算准确。 $n = 1000$ 都没问题！！

```
1 function [x,w] = gauss(N)
2 % legendre gauss quad: sum f(x)*w
3 J1 = (1:N-1); gamma = sqrt(1./(4 - 1./(J1).^2));
4 S = diag(gamma,-1) + diag(gamma,1); % tridiag matrix
5 x = eig(S); % gauss quad points
6 % sum P_j(x)*w = int P_j(x), % j = 0:N-1;
7 P = ones(N,N); P(:,2) = x;
8 for k = 2:N-1
9     P(:,k+1) = ((2*k-1)*x.*P(:,k) - (k-1)*P(:,k-1))/k;
10 end
11 r = [2; zeros(N-1,1)]; w = P' \ r; % cond(P)
12 end
```

Table 1: Abscissae and weight factors for Gauss – Legendre quadrature

	x_i	A_i
n = 3	0	0.88888 88888 88889
	± 0.774596669241483	0.55555 55555 55556
n = 4	± 0.339981043584856	0.65214 51548 62546
	± 0.861136311594053	0.34785 48451 37454

标注 6.4 Gauss–Legendre quadrature was derived by Gauss in 1814 using a continued fraction expansion. In 1826 Jacobi showed that the nodes were the zeros of the Legendre polynomials and that they were real, simple, and in $[1,1]$. The convergence of Gaussian quadrature methods was first studied by Stieltjes in 1884. Recent results by Trefethen suggest that the Clenshaw–Curtis rule may be as accurate as Gauss–Legendre quadrature with an equal number of nodes.

标注 6.5 对于光滑函数的积分，复合Simpson，Romberg，复合6阶Newton-Cotes公式，Legendre-Gauss 计算效果依次增强。在求解微分方程的谱方法中，更多选择端点固定的Legendre-Gauss-Lobatto公式。此时 $\{x_j\}$ 选择为 $(1-x^2)P'_n(x)$ 的零点

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

Greg von Winckel 基于Newton法给了一个稳定性极好的Matlab代码 `lglnodes.m`，见第三章讲义。

7 Clenshaw-Curtis求积公式

回顾：Chebyshev点插值能有效避免Runge现象，对光滑函数的近似效果极好，近似于最佳一致逼近！猜想：选择Chebyshev点作为求积节点，效果应该也不错吧？！

标注 7.1 The idea of Clenshaw-Curtis quadrature is to use Chebyshev points instead of optimal nodes(Gauss nodes). There are three main variations on this theme ⁴

- Chebyshev roots in $(-1,1)$: Fejér 's "first rule" (1933)
- Chebyshev extrema in $(-1,1)$: Fejér 's "second rule" (1933)
- Chebyshev extrema in $[-1,1]$: Clenshaw and Curtis (1960)

Clenshaw and Curtis call it “the Chebyshev formula”—but the term “Clenshaw–Curtis” is standard. Clenshaw and Curtis published their paper in 1960, before the introduction of the FFT in 1965. Soon afterwards, the connection with the FFT was pointed out by Gentleman in 1972. Trefethen 指出(SIREV2008) CC积分方法和Legendre-Gauss（最优“best”）具有几乎一样的精度，同时是稳定的、收敛的。但是，CC公式的只具有 n 次代数精度，远远低于Gauss公式的“最高次 $(2n+1)$ 代数精度”。此处提醒读者注意：代数精度和积分余项都是衡量数值积分好坏的标准，但二者并不是完全一致的、或者简单等同的。

```

1 function [x,w] = clenshaw_curtis(n)
2 x = cos(pi*(0:n)'/n); x = sort(x); % C-C formula
3 % x = cos(pi*(1:2:2*n+1)/(2*n+2)); % Fejer's rule
4 % vander lead to ill condition (bad when n>10)
5 % S = x.^(0:n); S = S.'; r = 2./(1:n+1); r(2:2:end) = 0;
6 % w = S \ r(:); cond(S),
7 % legendre poly is stable
8 N = n+1; P = ones(N,N); P(:,2) = x;
9 for k = 2:N-1
10     P(:,k+1)=( (2*k-1)*x.*P(:,k)-(k-1)*P(:,k-1) )/k;
11 end
12 r = [2; zeros(N-1,1)]; % cond(P)
13 w = P' \ r;
```

⁴Trefethen, Is Gauss Quadrature Better than Clenshaw – Curtis? SIREV(2008)

标注 7.2 LG, LGL, CC 的求积系数均有显示表达式, 可查阅《谱方法》专著。本讲义利用数值积分一般原理给出, 需要求解线性方程组, 是低效率的, 特别是计算时应当关注计算求积系数所涉及的矩阵条件数。优点是不涉及过多的正交多项式理论, 容易理解! 实际上, 求积公式一旦给出, 系数和节点就存好备用; 这是一劳永逸的事情, 不必过于关心计算求积系数的代价! (实际代价也很小!) 另外, CC公式和FFT的关系被发现之后, 计算效率进一步提高; 计算求积系数不需要求解上述线性方程组 (Trefethen, 2008SIREV)。

7.1 Clenshaw-Curtis与余弦变换

本节我们换个观点来看待和推导CC数值积分公式. 已知CC公式的求积节点 $x_k = \cos(\frac{k\pi}{N})$, 记 $I_N f$ 是以 x_k 为插值节点的 N 次插值多项式, 则由插值性求积公式知道

$$I(f) = \int_{-1}^1 f(x) dx \approx \int_{-1}^1 I_N f(x) dx = \sum_{k=0}^N \omega_k f(x_k)$$

我们推导上述求积公式的快速计算. 由于 $I_N f$ 是多项式, 则必可写为 $T_k(x)$ 的线性组合

$$I_N f(x) = \sum_{k=0}^N \hat{f}_k T_k(x)$$

\hat{f}_k 由插值条件确定, 即

$$I_N f(x_j) = f(x_j) = \sum_{k=0}^N \hat{f}_k T_k(x_j), \quad x_j = \cos \frac{j\pi}{N}, \quad 0 \leq j \leq N.$$

以上结果就是Chebyshev点插值, 且系数 \hat{f}_k 可由余弦逆变换计算得到. 于是

$$\int_{-1}^1 I_N f(x) dx = \int_{-1}^1 \sum_{k=0}^N \hat{f}_k T_k(x) dx = \sum_{k=0}^N \hat{f}_k \left(\int_{-1}^1 T_k(x) dx \right) = \sum_{k=0}^N \hat{f}_k d_k$$

例 9 记 $d_k = \int_{-1}^1 T_k(x) dx$, 检验 $d_0 = 2$,

$$d_k = \begin{cases} \frac{1}{k+1} - \frac{1}{k-1} & k \geq 2 \text{ even} \\ 0 & k \text{ odd} \end{cases} \quad (5)$$

提示: 代入 $T_k(x)$ 表达式, 利用积分换元法和三角函数积分

由上例可得 k 是偶数时, 系数 d_k 可记为 $\frac{2}{1-k^2}$. 于是CC公式求积分可分两步:

(1) 对 $f(x_j)$ 做余弦逆变换得到 \hat{f}_k ;

(2) \hat{f}_k 与 d_k 做内积

记 C 为余弦矩阵, 则

$$I(f) \approx \vec{d}^T \cdot (C^{-1} \vec{f})$$

```
1 function I = clenshaw_curtis_dct(f,n)
2 x1 = cos(pi*(0:n)'/n); f1 = f(x1); fk = imydct0(f1(:));
3 dk = 0*x1'; dk(1:2:n+1) = 2./(1-(0:2:n).^2); I = dk*fk;
```

函数 `imydct0.m` 是余弦逆变换⁵。上述代码计算积分只要 $(N \log N + N)$ 次运算, 和复合梯形、Simpson、Newton 几乎一样的计算复杂度, 但是却可达到与 Gauss 公式类似的精度⁶。若积分只要计算一次, 上述代码是合适的。但若是需要多次调用, 带求积系数的表达式使用更方便。求积系数 ω_k 可由代数精度来确定, 但是需要求解线性方程组, 当 N 较大时效率低; 另一方面, 求解线性方程组得到 ω_k 不能直接说明 ω_k 的非负性及求积公式的稳定性。我们采用另一方法给出求积系数的显示表达式并证明其非负性, 进而说明 CC 公式是稳定性。由上一算法可知 $I(f) \approx (\vec{d})^T * C^{-1} * \vec{f} = (\vec{f})^T * C^{-1} * \vec{d}$, 则 $\omega = C^{-1} * \vec{d}$ 。

```
1 function [x,w,If] = clenshaw_curtis2(n,a,b,f)
2 % If = int_{-1}^{1} f(x) dx // CC method pts x & weight w
3 x = -cos(pi*(0:n)'/n); w = 0*x; j0 = (0:2:n)'; % (n+1) chebpts
4 w(1:2:n+1) = 2./(1-j0.^2); w = imydct0(w); w = w';
5 % % trans [-1,1] to [a,b]
6 if (nargin > 1), x = (b-a)*x/2 + (a+b)/2; w = (b-a)/2.*w; end
7 if (nargin >= 4), If = w*f(x); end
```

例 10 检验 CC 公式的求积权重 ω_k 满足 $\vec{\omega} = C^{-1} * \vec{d}$, 其中

$$C_{jk} = \cos\left(\frac{jk\pi}{N}\right), \quad 0 \leq j, k \leq N$$

是余弦矩阵, \vec{d} 由 (5) 定义。进一步根据矩阵乘法规则检验 $\omega_k > 0$ 。

记

$$C = \left(C_{jk} \right)_{j,k=0}^N = \cos\left(\frac{jk\pi}{N}\right), \quad D = \begin{bmatrix} 1/2 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \\ & & & & 1/2 \end{bmatrix},$$

则求积系数 $\vec{\omega} = \frac{2}{N} D * C * D * \vec{d}$ 。下面说明求积系数的非负性, 注意 $d_0 = 2, d_k \leq 0, k \geq 2$

$$v_j = \left(C * D * \vec{d} \right)_j = \sum_{k=0}^N \cos\left(\frac{jk\pi}{N}\right) * (D * \vec{d})_k = 0.5 * d_0 - \sum_{k=1}^N \cos\left(\frac{jk\pi}{N}\right) * d_k \geq 1 - \sum_{k=2}^N |d_k|,$$

代入表达式

$$d_k = \begin{cases} 0, & k \text{ is odd} \\ \frac{1}{k+1} - \frac{1}{k-1} & k \text{ is even} \end{cases}$$

得到

$$\sum_{k=2}^N |d_k| = \begin{cases} 1 - \frac{1}{N+1}, & N \text{ is odd} \\ 1 - \frac{1}{N}, & N \text{ is even} \end{cases}$$

⁵代码见第三章讲义

⁶Lloyd N. Trefethen, Is Gauss Quadrature Better than Clenshaw – Curtis? SIREV 2008

进而 $v_j \geq \frac{1}{N+1} > 0$, 求积系数

$$\omega_j = \begin{cases} \frac{2v_j}{N}, & j = 2:N-1, \\ \frac{v_j}{N}, & j = 0, N \end{cases}$$

因此, Clenshaw-Curtis公式的求积系数非负, 求积公式是稳定的!

8 带权Gauss求积公式简介 (略)

8.1 Chebyshev-Gauss

当权函数 $\rho(x) = \frac{1}{\sqrt{1-x^2}}$, 对应高斯公式:

$$\int_{-1}^1 \frac{f(x)}{\sqrt{1-x^2}} dx \approx \frac{\pi}{n} \sum_{k=1}^n f(x_k), \quad x_k = \cos \frac{2k-1}{2n} \pi$$

$$\int_{-1}^1 \frac{f(x)}{\sqrt{1-x^2}} dx \approx \frac{\pi}{n} \sum_{k=0}^{n-1} f(x_k), \quad x_k = \cos \frac{k\pi}{n}$$

其中 \sum'' 表示端点求和取一半权重。注意被积函数是奇异积分, Simpson公式无法使用, Legendre-Gauss精度很差, 自适应高斯积分勉强可用。此时Chebyshev-Gauss简单、高效。但是要注意该公式只对权函数 $\rho(x) = \frac{1}{\sqrt{1-x^2}}$ 。实际上, 形如上式的积分, 可通过积分换元法转化为周期函数的积分, 再利用梯形公式计算。

```
1 f = @(x) exp(x);
2 g = @(x) (1-x.^2).^(1/2).*f(x);
3 figure; fplot(g,[-1,1]); % singular function
4 a = -1; b = 1; I = integral(g,a,b); % exact solution by matlab
5 % % gauss cheby1
6 n = 7; x1 = cos((0:n)'*pi/n);
7 f1 = f(x1); f1([1,end]) = 0.5*f1([1, end]);
8 Q = sum(f1)*pi/n; err = abs(I-Q),
```

8.2 Chebyshev - Gauss 2

当权函数 $\rho(x) = \sqrt{1-x^2}$, Gauss 公式

$$\int_{-1}^1 \sqrt{1-x^2} \cdot g(x) dx \approx \sum_{j=1}^n A_j g(x_j)$$

其中

$$x_j = \cos \frac{j\pi}{n+1}, \quad A_j = \frac{\pi}{n+1} \sin^2 \frac{j\pi}{n+1}$$

这个积分没有明显奇异性, Romberg, Gauss, CC都可以直接使用。但是分析发现该被积函数的高阶导数可能不存在, Romberg、Gauss、CC很难得到高精度的结果。而Cheb-Gauss2只要极少的点即可实现高精度。

```
1 f = @(x) exp(x);
2 g = @(x) (1-x.^2).^(1/2).*f(x);
```

```

3 figure; fplot(g,[-1,1]);
4 a = -1; b = 1; I = integral(g,a,b);
5 %% gauss cheby2
6 n = 8;
7 x1 = cos((1:n)'*pi/(n+1));
8 w = (1 - x1.^2)*pi/(n+1);
9 Q = dot(w,f(x1));
10 %% gauss legendre
11 [x2,w2] = gauss(n); Q2 = dot(g(x2),w2);
12 err = abs(I-Q), err2 = abs(I - Q2)

```

8.3 Gauss – Laguerre quadrature

无穷区间上的积分在理论和计算上都是困难的，常用的两类Gauss积分公式分别关联于Laguerre和Hermite多项式。考查以下两类积分

$$\int_0^{\infty} e^{-x} f(x) dx, \quad \int_{-\infty}^{\infty} e^{-x^2} f(x) dx$$

李庆扬教材（p124页）给出了6点高斯公式的求积节点和求积系数，供读者查表使用。所给Gauss-Laguerre求积系数表达式是错误的！！正确表达式如下

$$\text{Laguerre求积系数} \quad A_k = \frac{x_k}{[L_n(x_k) * (n+1)]^2}, \quad k = 0, 1, \dots, n$$

其中 x_k 是 $L_{n+1}(x)$ 的根. Gauss积分的求积节点(求积系数)和正交多项式理论密切相关，求积系数一般都有显式表达式。本讲义只给出一般的计算方法：基于求解代数精度满足的线性方程组。广义积分本身求解困难，在应用于谱方法中，会对这两类多项式做进一步改进⁷。

Laguerre多项式满足如下循环式

$$\begin{cases} (n+1)L_{n+1}(x) = (2n+1-x)L_n(x) - nL_{n-1}(x), & n = 1, 2, \dots \\ L_0(x) = 1, & L_1(x) = 1 - x \end{cases}$$

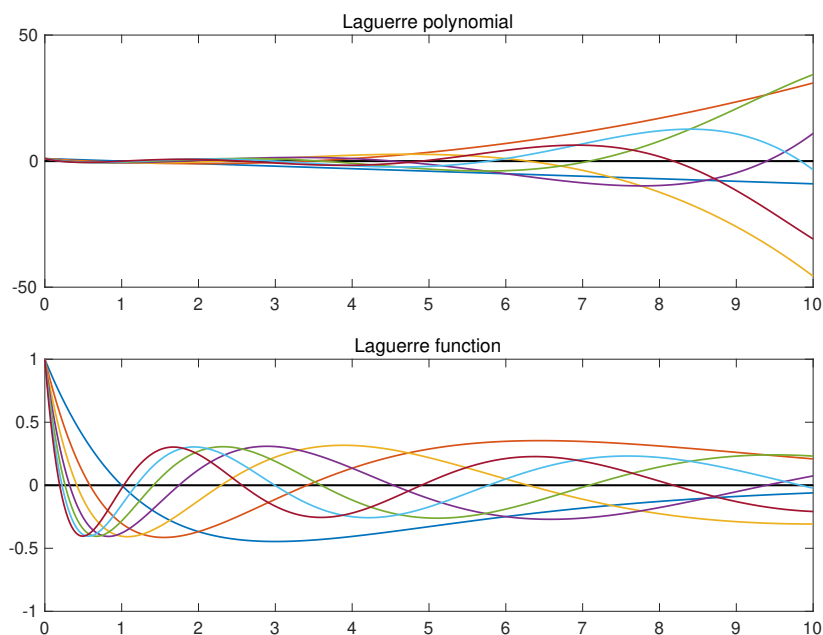
且 $L_n(x)$ 是 $[0, \infty)$ 上带权 e^{-x} 的正交多项式

$$(L_j, L_k)_\rho = \int_0^{+\infty} e^{-x} L_j(x) L_k(x) dx = \delta_{jk}.$$

测试可发现当 $n \rightarrow \infty, x \rightarrow \infty$ 时， $L_n(x)$ 很快趋于无穷大，直接使用会带来数值不稳定. 实际使用时多使用Laguerre函数 $\hat{L}_n(x) = e^{x/2} L_n(x)$ ，可知

$$(\hat{L}_j, \hat{L}_k) = \int_0^{+\infty} \hat{L}_j(x) \hat{L}_k(x) dx = \int_0^{+\infty} e^{-x} L_j(x) L_k(x) dx = \delta_{jk}$$

⁷ 参考书：Tang Tao & Shen Jie, Spectral and High-Order Methods with Applications



例 11 试给出 Gauss - Laguerre 公式

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

的一个计算方法。编写代码计算 ($n=10, 15, 20$) 时的求积节点和求积系数, 并取 $f(x) = \sin(x)$ 检验公式精度, 已知 $\int_0^{\infty} e^{-x} \sin x dx = \frac{1}{2}$.

该例子留作研究生作业。提示: 先计算求积节点 (Laguerre 多项式的零点/三对角矩阵特征值); 再根据代数精度确定求积系数。chebfun/lagpts.m 可计算相应的求积节点和求积系数, 且可认为结果是精确的。

8.4 Gauss - Hermite quadrature

在区间 $(-\infty, \infty)$ 上带权 e^{-x^2} 的正交多项式, 称为 Hermite 多项式, 满足

$$\int_{-\infty}^{+\infty} e^{-x^2} H_m(x) H_n(x) dx = 2^n n! \sqrt{\pi} \delta_{mn}$$

以及循环式

$$\begin{cases} H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x) \\ H_0(x) = 1; \quad H_1(x) = 2x \end{cases}$$

注意到 H_n 是首项系数为 2^n 的 n 次多项式, 当 $n, x \rightarrow \infty$ 时, $H_n(x)$ 快速趋于无穷大, 因此常用的Hermite 函数

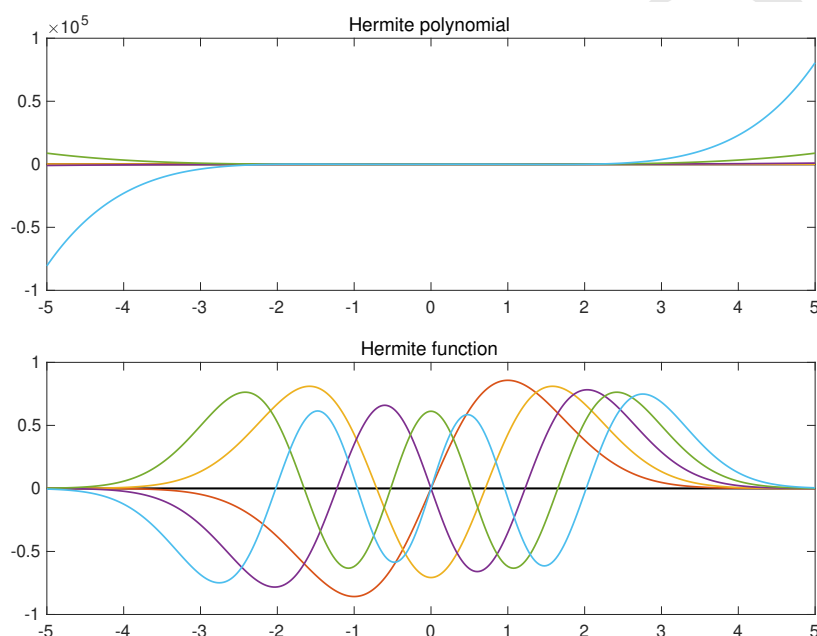
$$\hat{H}_n(x) = \frac{1}{\sqrt{2^n n!}} e^{-x^2/2} H_n(x), \quad n \geq 0, \quad x \in \mathbb{R}$$

是对多项式的修正, 且满足如下 L^2 正交性

$$\int_{\mathbb{R}} \hat{H}_n(x) \hat{H}_m(x) dx = \sqrt{\pi} \delta_{mn}.$$

以及循环式

$$\begin{cases} \hat{H}_{n+1}(x) = x\sqrt{\frac{2}{n+1}} \hat{H}_n(x) - \sqrt{\frac{n}{n+1}} \hat{H}_{n-1}(x), & n \geq 1, \\ \hat{H}_0(x) = e^{-x^2/2}, \quad \hat{H}_1(x) = \sqrt{2}xe^{-x^2/2} \end{cases}$$



标注 8.1 定积分的计算门道很多, 好的被积函数各种方法都行, 性质不好的函数需要仔细分析, 最好能利用数学知识化繁为简. 例如:

$$I = \int_1^{\infty} x^{-\frac{3}{2}} \sin \frac{1}{x} dx$$

如果直接在区间 $[1, 100]$ 积分, 剩余做截断, 误差很大(约 10^{-2}). 化简后可改善计算效果. 令 $x = 1/t$, 得到

$$I = \int_1^0 t^{3/2} \sin t \cdot d\left(\frac{1}{t}\right) = \int_0^1 \frac{\sin t}{\sqrt{t}} dt$$

上述被积函数 $f(x)$ 无奇性, 可用复合中点公式、带权 $\rho(x) = \frac{1}{\sqrt{x}}$ 的Gauss积分、或者Legendre-Gauss、CC计算.

还可进一步做改进

$$I = \int_0^1 \frac{\sin t - t + t^3/6}{\sqrt{t}} dt + \int_0^1 \frac{t - t^3/6}{\sqrt{t}} dt$$

上式第一项积分被积函数更加平滑，效率更高；第二项直接求出即可。

9 重积分 (略)

关于一般区域上的二重积分，计算较为困难。在精度要求不高时，可以对区域进行三角形网格化分⁸，再计算每个三角形积分后求和。

Theorem 7 设 T 是以 p_1, p_2, p_3 为顶点的三角形，则积分公式

$$\int_T u(x, y) dx dy = \frac{Y}{3} \left(u\left(\frac{1}{2}(p_1 + p_2)\right) + u\left(\frac{1}{2}(p_1 + p_3)\right) + u\left(\frac{1}{2}(p_2 + p_3)\right) \right)$$

对所有二次函数精确成立，其中 Y 是三角形面积。

标注 9.1 一般区域上的高精度积分公式很难直接得到。对于矩形区域上的二重积分，前文给出的复合梯形、Simpson、复合Gauss都是可以直接应用的。

9.1 矩形区域上的Newton-Cotes积分公式

考查二重积分

$$I(f) = \iint_{\Omega} f(x, y) d\Omega = \int_a^b \int_c^d f(x, y) dx dy$$

将积分区域均匀分割成 $M \times N$ 个小矩形，记网格节点 (x_i, y_j) 以及小矩形中心 $(x_{i+1/2}, y_{j+1/2})$ ，记 $f_{ij}, f_{i+1/2, j+1/2}$ 为相应的函数值。

中点公式

$$I(f) \approx \frac{(b-a)(d-c)}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f_{i+1/2, j+1/2}$$

梯形公式

$$I(f) \approx \frac{(b-a)(d-c)}{4MN} \left[\sum_{corner} f_{ij} + 2 \sum_{Edge} f_{ij} + 4 \sum_{inner} f_{ij} \right]$$

如果被积函数是周期函数且 $(b-a), (d-c)$ 是相应的周期，则中点公式和梯形公式是指数收敛的。如果对于一般非周期函数，且网格节点必须选择等距的矩阵网格时，复合Newton-Cotes公式可得到更高精度的数值解。

$$I(f) = \int_a^b \left[\sum_{j=0}^N f(x, y_j) B_j \right] dx = \sum_{i=0}^M f(x_i, y_j) A_i B_j$$

其中 A_i 为复合Newton-Cotes公式在 $[a, b]$ 上 M 等分的求积系数。 B 类似定义。我们选择光滑函数测试精度，结果显示Newton-Cotes公式精度比Matlab自带函数integral2.m精度更高。我们调用工具包chebfun2并认为其计算结果是精确的。

```
1 function If = newton_cotes_2d(m1, m2, f, a, b, c, d, nd)
2 % % m1 and m2 must divided by 4 / 6
3 % % comosition integral by newtoncotes for n = 4//6
```

⁸Per-Olof Persson & Gilbert Strang, A Simple Mesh Generator In Matlab, SIREV2004, Persson教授主页上(UC Berkeley)有软件包供下载


```

4 h1 = (b-a)/m1; h2 = (d-c)/m2; x1 = a:h1:b; x2 = c:h2:d;
5 if (nargin<8), nd = 4; end % newton cotes 5 pts
6 [~, w0] = newtonquad(nd); w1 = w0*(h1*nd/2); w2 = w0*(h2*nd/2);
7 %
8 mj = repmat(w1, 1, m1/nd );
9 mj(1, 2:end) = 2*w1(1); mj(end,:) = [];
10 w1 = [mj(:); w1(1)];
11 %
12 mj = repmat(w2, 1, m2/nd );
13 mj(1, 2:end) = 2*w2(1); mj(end,:) = [];
14 w2 = [mj(:); w2(1)];
15 %
16 w = w1*(w2'); w = w(:);
17 [x2,x1] = meshgrid(x2,x1); Z = f(x1,x2); Z = Z(:);
18 Z(isinf(Z)) = 0; Z(isnan(Z)) = 0;
19 If = w'*Z;

```

```

1 f = @(x,y) exp(x).*sin(y);
2 a = -1; b = 1; c = 0; d = 3;
3 f_cheb = chebfun2(f,[a,b,c,d]);
4 figure; fig1 = surf(f_cheb); fig1.FaceAlpha = 0.8;
5 Q = sum2(f_cheb); % exact soln // via chebfun
6 %
7 Q1 = integral2(f,a,b,c,d); % // matlab
8 e1 = abs(Q-Q1),
9 % composite newton cotes for n = 6
10 m1 = 54; m2 = 60; nd = 6;
11 If = newton_cotes_2d(m1, m2, f, a, b, c, d, 6);
12 e2 = abs(Q - If),

```

9.2 矩形区域上的Gauss型求积公式

考查如下积分

$$\int_a^b \int_c^d f(x,y) dx dy = \frac{(b-a)(d-c)}{4} \int_{-1}^1 \int_{-1}^1 g(s,t) ds dt$$

1. 计算Gauss点 $x = \{x_0, \dots, x_m\}$, $y = \{y_0, \dots, y_n\}$ 和求积权 w_1, w_2 , 由坐标变换

$$t = [(b-a) * x + a + b]/2, \quad s = [(d-c) * y + d + c]/2,$$

meshgrid(s,t) 生成二维Gauss坐标; 提取节点处函数值 f_{ij}

2. 由 w_1, w_2 生成二维求积权重矩阵 W

3. 对 f 和 W 求离散内积(对应相乘相加)即可.

可通过测试上一代码中的算例发现：二维Gauss求积公式对于光滑函数效果良好，只要极少数点就可达到高精度数值解。

```

1 function If = gauss_2d(f,a,b,c,d,m1,m2)
2 % output: If = int_a^b int_c^d f(x,y) dx dy
3 % input: m1 is number of gauss pts in(a,b) // m2 -->(c,d)
4 [x1,w1] = gauss(m1);
5 x1 = (b-a)*x1/2 + (a+b)/2; w1 = (b-a)/2.*w1;
6 [x2,w2] = gauss(m2);
7 x2 = (d-c)*x2/2 + (c+d)/2; w2 = (d-c)/2.*w2;
8 [t,s] = meshgrid(x2,x1); ff = f(s,t);
9 ww = w1*w2'; ww = ww(:); If = ww'*ff(:);

```

习题和上机练习

(1) 考查积分

$$\int_0^1 \sqrt{x} f(x) dx, \quad \int_0^1 \frac{f(x)}{\sqrt{x}} dx$$

试建立2点，4点Gauss公式

(2) 取 $n = 8, 16$ ；用复合梯形公式，Simpson公式，Romberg积分计算

$$\int_{0.5}^5 \frac{\sin x}{x} dx$$

(3) 利用Romberg积分公式和自适应积分公式计算积分

$$\int_{0.5}^5 \frac{100}{x^2} \sin \frac{10}{x} dx$$

可选择自适应Simpson，自适应中点公式，或者自适应两点Gauss公式。

测试函数mittag-leffler 函数

对RL导数成立

$${}_0D_t^\gamma \left(t^{\alpha k + \beta - 1} E_{\alpha, \beta}^{(k)}(\lambda t^\alpha) \right) = t^{\alpha k + \beta - \gamma - 1} E_{\alpha, \beta - \gamma}^{(k)}(\lambda t^\alpha)$$

其中

$$E_{\alpha, \beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)}, \quad E_{1,1}(z) = e^z$$

特殊情况如 $k = 0, \alpha = \beta = \lambda = 1$ ，可以检验指数函数的分数阶导数

$${}_0D_t^\gamma e^t = \frac{t^{-1/2}}{\Gamma(1/2)} + \frac{1}{\Gamma(1/2)} \int_0^t \frac{e^s}{(t-s)^{1/2}} ds = t^{-\frac{1}{2}} E_{1,1/2}(t)$$