

本篇也比较摸，因为突然喜欢上了东京爱乐乐团(这个团的德九味儿是真的正！)和武満徹老师的管弦作品. 听了好长时间，忘记了数学.

推荐武満徹先生的交响-念白作品「系図 -若い人たちのための音楽詩-」，一说20世纪严肃音乐始于斯特拉文斯基“春之祭”，终于武満徹“族谱 -为年轻的人们而作的音乐诗-”，系図的这种无调性-调性语言交织的音乐语言体系提供了相当充足的感染力和戏剧性，以支持和传达其中的充盈的情感和完整的叙事故事. 为符合年轻人们的受众定位(现在年轻人们很少能听明白严肃音乐了)，作曲家特地加上了富有日本近现代文学色彩的念白(我是国内的第一个翻译者-.). 羡慕，只可惜没考上央音作曲系，也没交响现场可以听，只能被迫听唐山研究院难听到死的不知道基于什么通俗流行改编的上课铃(看不起流行音乐属于是我的传统艺能了).

废话说了. 本篇讨论数值微积分.

数值微分

数值微分问题求一函数 $f(x)$ 在其解析域内的某点 x_0 处的近似一阶导数 $f'(x_0)$ ，常用的方法是用一简单插值函数来代替 $f(x)$ 求导

插值法数值微分

我们在原函数 $f(x)$ 上找到 $n > 1$ 个点，据其构造一插值函数 $\phi(x)$ ，以 $\phi'(x_0)$ 来代替 $f'(x_0)$

误差分析：在点集上的 n 次插值余项可写作：

$$R_n(x) = f(x) - \phi(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega_{n+1}(x) \tag{4.7}$$

其中 $\omega_{n+1}(x) = \prod_{i=0}^n (x - x_i)$, $\xi \in (a, b)$

对上式求导并将近似求导点 $x = x_i$ 代入其中：

$$f'(x_i) - \phi'(x_i) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{j=0, j \neq i}^n (x_i - x_j) \tag{6.1}$$

我们以 $n = 2, 3$ 时的情形简要分析：

两点公式

$n = 2$ 时的情况. 取插值锚点 x_0, x_1 ，要求 $f(x)$ 在 (x_0, x_1) 上至少1阶可导，使用直线 $\phi(x) = f(x_0) + \frac{f(x_1)-f(x_0)}{x_1-x_0}(x - x_0)$ 代替求导，得出结果为

$$f'^*(x_0) = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad (6.2)$$

x_1 不一定在 x_0 的左或者右，因此我们可以统一化书本上的“向前差商”和“向后差商”为上述一个式子。

误差分析，对 $f(x)$ 在 x_0 处执行级数展开，有

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + o(x - x_0)$$

分别代入 $x = x_0, x_1$ 得：

$$f'^*(x_0) = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = f'(x_0) + o(x - x_0)$$

即，两点公式产生一 $o(x - x_0)$ 误差。若 $f(x)$ 在该邻域内至少二阶可导，则展开式可拥有一Lagrange余项：

$$f'^*(x_0) = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = f'(x_0) + f''(\xi)(x_1 - x_0), \xi \in (x_0, x_1) \text{ or } (x_1, x_0)$$

即，产生一 $f''(\xi)(x_1 - x_0)$ 误差。

三点公式

三点公式之“三点”的选择可能出现不同的位置，故为了方便讨论，我们不妨假设选择的三个点距离相同，且仍记为 $x_0, x_1 = x_0 + h, x_2 = x_0 + 2h (h > 0)$ ，后我们在不同的位置求导。

容易得知，上述三点的Lagrange插值多项式是

$$\phi(x) = f(x_0) \frac{(x - x_1)(x - x_2)}{2h^2} + f(x_1) \frac{(x - x_0)(x - x_2)}{-h^2} + f(x_2) \frac{(x - x_0)(x - x_1)}{2h^2}$$

对其求导有

$$\phi'(x) = f(x_0) \frac{2x - x_1 - x_2}{2h^2} + f(x_1) \frac{2x - x_0 - x_2}{-h^2} + f(x_2) \frac{2x - x_0 - x_1}{2h^2}$$

对于不同的数值微分位置，有：

1. $x = x_0, \phi'(x) = \frac{1}{2h}[-3f(x_0) + 4f(x_1) - f(x_2)]$
2. $x = x_1, \phi'(x) = \frac{1}{2h}[-f(x_0) + f(x_2)]$, 中心差商公式
3. $x = x_2, \phi'(x) = \frac{1}{2h}[f(x_0) - 4f(x_1) + 3f(x_2)]$

其截断误差可以由(6.2)表示出。

除此之外，样条也可以用作估计数值微分。

数值积分

数值积分求一函数在其连续域 $[a, b]$ 内的积分近似值.

矩形近似

正如任何哲学家或科学家在千百年来所作之艰苦卓绝的探索那样, 我们首先将我们不会求解的问题抽象为简单的问题——即使这种抽象是不精确的, 因此我们想到在之前提出的积分中值定理:

$$\int_b^a f(x)dx = f(\xi)(a - b), \xi \in [b, a]$$

我们可以在 $[a, b]$ 内自由选择 ξ 的值以近似估计, 这一式子实际上是用一矩形来近似代替积分, 因此称为矩形公式. 特殊地, 若 $\xi = a, b, \frac{a+b}{2}$ 时, 上式被称为左矩形公式、右矩形公式和中矩形公式.

矩形近似本质上是用零次插值函数 $\phi(x) = c$ 来代替 $f(x)$. 更一般的讨论:

插值近似

我们使用对 $f(x)$ 上的若干点进行 n 次插值, 得到插值函数 $\phi(x)$, 以 $\int_b^a \phi(x)dx$ 来代替 $\int_b^a f(x)dx$.

易知 n 次插值公式的截断误差为 $\int_b^a \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega_{n+1} dx$

特别地, $n = 1$ 时, 是以 x 轴、 $x = a$ 、 $x = b$ 、直线 $(a, f(a)), (b, f(b))$ 围出的梯形面积代替积分值, 写作:

$$\int_b^a f(x)dx \approx \int_b^a \phi(x)dx = \frac{a-b}{2}[f(a) + f(b)] \quad (6.3)$$

式子6.3被称为**梯形公式**.

继续讨论: $n = 2$ 是, 是以 x 轴、 $x = a$ 、 $x = b$ 、抛物线 $\phi(x)$ 围成的曲边梯形面积代替积分值, 而抛物线 $\phi(x)$ 由点 $(b, f(b)), (\frac{a+b}{2}, f(\frac{a+b}{2})), (a, f(a))$ 确定, 由Lagrange插值法可以写成:

$$\phi(x) = f(b)l_0(x) + f(\frac{a+b}{2})l_1(x) + f(a)l_2(x)$$

其中 $l_i(x)$ 是第 i 点的Lagrange基.

积分易得:

$$\int_b^a \phi(x)dx \approx \frac{1}{6}[f(b) + 4f(\frac{f(a) + f(b)}{2}) + f(a)](a - b) \quad (6.3)$$

式子6.4被称为**抛物线公式**(Simpson公式).

彻底一般化, 我们用 n 次插值公式来近似, 并将积分区间 $[a, b]$ n 等分, 得到 $(n + 1)$ 个等分点 $x_i, i \in [0, n]$, 则近似的积分式有如下形式:

$$\int_b^a \phi(x)dx \approx (a - b) \sum_{i=0}^n C_i f(x_i)$$

上述公式被称为 n 阶Newton-Cotes积分公式, C_i 是一 $(n + 1)$ 项的系数族, 每一族的总和为1, 其系数只与阶数 n 有关.

1. $n = 1$ 时, $C = (\frac{1}{2}, \frac{1}{2})$
2. $n = 2$ 时, $C = (\frac{1}{6}, \frac{4}{6}, \frac{1}{6})$
3. $n = 3$ 时, $C = (\frac{1}{8}, \frac{3}{8}, \frac{3}{8}, \frac{1}{8})$, Simpson' $\frac{3}{8}$ 公式
4. $n = 4$ 时, $C = (\frac{7}{90}, \frac{32}{90}, \frac{12}{90}, \frac{32}{90}, \frac{7}{90})$, Cotes公式

$N \leq 7$ 时, Newton-Cotes积分公式数值稳定.

误差估计

代数精确度

在上述的讨论中, 我们使用插值函数 $\phi(x)$ 来近似 $f(x)$ 以求得积分. 不难想象, n 次插值函数可以完全精确刻画任意 n 次的原函数 $f(x)$, 因此在这种情况下, 由这个插值函数产生的积分估算方法 $S(a, b, f)$ 将能精确求出任意 n 次的原函数 $f(x)$ 的任意定积分, 故我们称这种估算方法 $S(a, b, f)$ 具有 n 次的代数精确度.

有时我们需要评价一积分估算方法 $S(a, b, f)$ 的代数精确度, 即可以令 $f_i(x) = 1, x, x^2, x^3, \dots$, 对于逐个 f_i , 判断 $S(a, b, f_i)$ 是否严格与 $\int_b^a f(x)dx$ 相等, 能够相等的最高 x 次数即为估算方法 $S(a, b, f_i)$ 的代数精确度.

易知, n 阶的Newton-Cotes积分公式至少有 n 次代数精确度. 且可以证明, $2k$ 阶的Newton-Cotes积分公式至少有 $(2k + 1)$ 阶代数精度(证明提示: $y = x^{2n+1}$ 具有中心对称性).

Newton-Cotes积分方法的误差估计

Newton-Cotes积分方法基于插值法, 而插值多项式具有余项:

$$R_n(x) = f(x) - \phi(x) = \frac{f^{(n+1)}(\xi)}{(n + 1)!} \omega_{n+1}(x) \quad (4.7)$$

其中 $\omega_{n+1}(x) = \prod_{i=0}^n (x - x_i)$, $\xi \in (a, b)$

式4.7两侧同时积分:

$$\int_b^a f(x)dx - \int_b^a \phi(x)dx = \int_b^a \frac{f^{(n+1)}(\xi)}{(n + 1)!} \omega_{n+1}(x)dx$$

需要注意，此处的 $f^{(n+1)}(\xi)$ 属与 x 相关的量，不可直接作为常数提出积分表达式外. 因此对于 $n = 1$ 的梯形公式情况，由积分中值：

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

此为梯形公式的误差

使用更为复杂的方式可以得到Simpson公式的误差：

$$R(a, b, f) = -\frac{(a-b)^5}{2880} f^{(4)}(\eta), \eta \in [b, a]$$

在对Newton-Cotes积分方法的误差估计中，我们发现：即使是最简单的梯形公式之误差，也与积分区间长度 $b-a$ 的三次方成正比，故我们想到：是否可以通过缩减单次积分的长度，将整个区间上的积分化为若干小区间上积分之和来解决问题呢？

复化积分

经过上述的启发，我们将函数 $f(x)$ 的积分域 $[a, b]$ 等分为 n 个小区间，各区间长度为 h ，在每一段上分别使用插值法求积，再将其加和.

复化梯形公式

对于复化区间 $[x_{n-1}, x_n]$ ，梯形公式写作：

$$\int_{x_{n-1}}^{x_n} f(x)dx \approx \frac{1}{2}(x_n - x_{n-1})[f(x_n) + f(x_{n-1})] = \frac{h}{2}[f(x_n) + f(x_{n-1})]$$

则全域的积分可以表示为：

$$\int_b^a f(x)dx = \frac{h}{2}[f(x_0) + f(x_n) + 2 \sum_{i=1}^{n-1} f(x_i)] \quad (6.4)$$

上式称为复化梯形公式. 误差分析：

在每一段上，复化梯形公式产生误差：

$$R_i = -\frac{h^3}{12} f''(\xi_i), \xi_i \in (x_{i-1}, x_i)$$

则总的误差为各段误差之和：

$$R = \sum_{i=1}^n \left[-\frac{h^3}{12} f''(\xi_i) \right] = \sum_{i=1}^n \left[-\frac{nh^3}{12} \frac{f''(\xi_i)}{n} \right]$$

由离散的中值定理(均值公式):

$$R = -\frac{h^2}{12}(a-b)f''(\eta), \eta \in (b, a)$$

复化Simpson公式

对于内部有一额外点 x'_n 的复化区间 $[x_{n-1}, x_n]$, Simpson公式写作:

$$\int_{x_{n-1}}^{x_n} f(x)dx \approx \frac{1}{6}[f(x_{n-1}) + 4f(x'_n) + f(x_n)](a-b)$$

故全域的积分可以写作

$$\int_b^a f(x)dx = \frac{h}{2}[f(x_0) + f(x_n) + 2\sum_{i=1}^{n-1} f(x_i) + 4\sum_{i=1}^n f(x'_i)] \quad (6.5)$$

类似地, 复化Simpson公式拥有误差

$$R = -\frac{h^4}{180}(a-b)f^{(4)}(\eta), \eta \in (b, a)$$

复化积分的逐次分半算法

假设在实际计算中, 我们需要将误差控制至某一范围, 我们又知道, 区间越细分则误差越小, 故我们可以逐个增多区间数 n 来搜索出满足误差要求的答案. 但容易知道, 在 n 足够大时, n 每增加1所导致的 h 减小之幅度很小, 故我们考虑每次将 n 翻一倍.

而目前为止, 每一次取到新的 n , 我们都必须重新计算一遍复化公式, 这是很低效的, 因此我们以梯形公式为例, 考察 n 与 $2n$ 等分的梯形公式 T_n 和 T_{2n} 的关系:

$$T_n = \frac{h}{2}[f(x_0) + f(x_n) + 2\sum_{i=1}^{n-1} f(x_i)]$$
$$T_{2n} = \frac{h}{4}[f(x_0) + f(x_n) + 2\sum_{i=1}^{2n-1} f(x_i)]$$

作如下计算可得:

$$T_{2n} - \frac{1}{2}T_n = \frac{h}{2}\sum_{x \in A} f(x)$$

其中 A 为所有新增点的点集.

如此可以方便求出分半后的新梯形公式值:

$$T_{2n} = \frac{1}{2}T_n + \frac{h}{2}D$$

其中 D 为所有新增点函数值之和.

逐次分半法有另一好处, 即可以仅凭借估计值求得误差. 故这种误差估计被称为**后验误差估计**.

逐次分半算法-后验误差估计

我们有一种直觉: 将 n 翻一倍时, 我们距离真实值的误差会定量变化, 因此考虑 n 与 $2n$ 等分的梯形公式 T_n 和 T_{2n} 所产生的误差 (此处为方便, 记真实的积分值为 I):

$$R_n = I - T_n = -h^2 \frac{a-b}{2} f''(\eta_1), \eta_1 \in (a, b)$$

$$R_{2n} = I - T_{2n} = -\left(\frac{h}{2}\right)^2 \frac{a-b}{2} f''(\eta_2), \eta_2 \in (a, b)$$

我们回忆 $f''(\eta)$ 是由均值公式而来, 而当 n 充分大时, $f''(\eta)$ 可以被认为能较为精确刻画 $f''(x)$ 在 $[b, a]$ 上的均值, 因此 $f''(\eta_1) \approx f''(\eta_2)$:

$$R_n \approx 4R_{2n}$$

即:

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

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

如此得到了可以在实际中使用的误差估计法.

同样, Simpson公式也有后验的误差序列:

$$I - S_{2n} \approx \frac{1}{15}(S_{2n} - S_n)$$

逐次分半加速, Romberg求积公式

由梯形公式的后验误差估计 $I - T_{2n} \approx \frac{1}{3}(T_{2n} - T_n)$, 我们可以得到 $I \approx \frac{1}{3}(4T_{2n} - T_n)$

这提示我们, 通过计算 $\frac{1}{3}(4T_{2n} - T_n)$ 可能会得出更精确的解.

我们以 $n = 4$ 的情形观察:

$$T_8 = \frac{a-b}{16}[y_0 + y_8 + 2(y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7)]$$

$$T_4 = \frac{a-b}{8}[y_0 + y_8 + 2(y_2 + y_4 + y_6)]$$

$$\text{则 } I \approx \frac{1}{3}(4T_8 - T_4) = \frac{a-b}{8}[y_0 + y_8 + 4(y_1 + y_3 + y_5 + y_7) + 2(f_2 + f_4 + f_6)]$$

注意到上式等号右侧恰为 $n = 8$ 情形下的复化Simpson公式 S_8 .

实际上, 可以证明 $S_{2n} = \frac{1}{3}(4T_{2n} - T_n)$, 即 $2n$ 和 n 分割的两复化梯形值可以通过线性组合转化为复化Simpson值. 而这个转化过程将误差由 $O(h^2)$ 缩小到 $O(h^4)$

同样地, 有复化Cotes值: $C_{2n} = \frac{1}{15}(16S_{2n} - S_n)$, 误差 $O(h^8)$, 复化Romberg值 $R_{2n} = \frac{1}{63}(64C_{2n} - C_n)$, $O(h^{16})$. 如此可无限外推.

这一过程提示我们, 可以通过组合两"较低级的"复化估计值来生成一"较高级的"的复化估计值, 以缩减误差.

Gauss求积公式

以上, 我们完成了对基于等距节点插值多项式的数值积分的讨论, 我们发现, 以 n 次插值多项式拟合积分, 最多能提供 $(n + 1)$ 次代数精度, 直觉告诉我们这个代数精度是不令人满意的, 因此我们试图找到一更高代数精度是估计方法.

对于某带权函数(不带权也无所谓)的代数精确条件可以一般化描述如下:

$$\int_b^a \rho(x)f(x)dx = \sum_{i=1}^n A_i f(x_i) \quad (6.6)$$

对于 n 个插值点, 上式有 $2n$ 个待定系数(n 个 A_i , n 个 x_i), 因此分别取 $f(x) = \{1, x, x^2, \dots, x^{2n-1}\}$ 代入其中, 得到方程组(注意: 这一方程组并非简单的线性方程组, 写为矩阵形式仅为表示方便)(实际上还有基于算符的表示, 但是理解起来不那么直观):

$$\begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ x_1 & x_2 & x_3 & \dots & x_n \\ x_1^2 & x_2^2 & x_3^2 & \dots & x_n^2 \\ \vdots & \vdots & \vdots & & \vdots \\ x_1^{2n-1} & x_2^{2n-1} & x_3^{2n-1} & \dots & x_n^{2n-1} \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \\ A_3 \\ \vdots \\ A_n \end{pmatrix} = \begin{pmatrix} \int_b^a \rho(x)dx \\ \int_b^a \rho(x)x dx \\ \int_b^a \rho(x)x^2 dx \\ \vdots \\ \int_b^a \rho(x)x^{2n-1} dx \end{pmatrix} \quad (6.7)$$

$2n$ 条关系和 $2n$ 个自由度, 一定能得到一组 x_i 和 A_i 的解. 这表明 n 次多项式最高可以拥有 $(2n - 1)$ 次代数精度.

解出的这组 x_i 称为 **Gauss点**, 将 x_i 和 A_i 代入(6.6)得到的公式被称为 **Gauss求积公式**, Gauss求积公式的求法就是解上述的方程(6.7).

但这样的方程是非线性方程, 而且体量庞大, 故我们希望得到更简单的求解法, 至少可以求出Gauss点集.

正交多项式求法

可以证明, 对于估算方法 $\sum_{i=0}^n A_i f(x_i)$, 其节点族 $\{x_i\}$ 是Gauss点当且仅当 $\omega(x) = \prod_{i=1}^n (x - x_i)$ 与任意 $(n - 1)$ 次多项式在 $[a, b]$ 上关于权函数正交.

因此我们可以构造(可用正交化方法)一个与任意 $(n - 1)$ 次多项式正交的 n 次多项式 $\omega(x)$, 解出 $\omega(x) = 0$ 的 n 个根, 即为Gauss点集.

在Gauss点集求出后, 于方程(6.7)中抽取若干 n 行求出系数族 A_i

特殊的Gauss求积公式

(考这一部分多少有点不人道主义)

Gauss-Legendre求积公式

考察在 $[-1, 1]$ 上的任意权为1、锚点数为 n 的积分:

$$\int_{-1}^1 f(x)dx = \sum_{i=1}^n A_i f(x_i)$$

1. $n = 1$ 时, $x_1 = 0$; $A_1 = 2$, 则 $\int_{-1}^1 f(x)dx \approx 2f(0)$
2. $n = 2$ 时, $x = \pm \frac{1}{\sqrt{3}}$; $A_1 = A_2 = 1$, 则 $\int_{-1}^1 f(x)dx \approx f(-\frac{1}{\sqrt{3}}) + f(\frac{1}{\sqrt{3}})$
3. $n = 3$ 时, $x_{1,3} = \pm \sqrt{\frac{3}{5}}$, $x_2 = 0$; $A_1 = A_3 = \frac{5}{9}$, $A_2 = \frac{8}{9}$

上述公式系列被称为Gauss-Legendre求积公式, 更高次数的Gauss-Legendre求积公式参数可以查表得知.

Gauss-Chebyshev求积公式

考察在 $[-1, 1]$ 上的任意权为 $\rho(x) = \frac{1}{\sqrt{1-x^2}}$ 、锚点数为 n 的积分:

$$\int_{-1}^1 \frac{1}{\sqrt{1-x^2}} f(x)dx = \sum_{i=1}^n A_i f(x_i)$$

其Gauss点系为 $x_i = \cos \frac{2i-1}{2n} \pi$, 求积系数恒为 $A = \frac{\pi}{n}$

Gauss-Laguerre求积公式

考察在 $[0, +\infty)$ 上的任意权为 $\rho(x) = e^{-x}$ 、锚点数为 n 的积分,

其Gauss点系为 $L(x) = e^x \frac{d^n}{dx^n} (e^{-x} x^n)$ 的零点, 求积系数为 $A_i = \frac{(n!)^2}{x_i [L'_n(x_i)]^2}$

Gauss - Hermite求积公式

考察在 $(-\infty, +\infty)$ 上的任意权为 $\rho(x) = e^{-x^2}$ 、锚点数为 n 的积分,

其Gauss点系为 $H(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n}(e^{-x^2})$ 之零点, 求积系数为 $A_i = \frac{2^{n+1} n! \sqrt{\pi}}{[H'_n(x_i)]^2}$