对于下列积分式:

$$I_n = \frac{1}{e}\int_0^1 x^n \cdot e^x dx$$

我们有:

$$I_0 = \frac{1}{e}\int_0^1 e^x dx = 1 - \frac{1}{e},$$

$$I_n = \frac{1}{e}\int_0^1 x^n \cdot d(e^x) = \frac{1}{e}\left(x^n \cdot e^x|_0^1 - \int_0^1 e^x \cdot nx^{n-1}dx\right) = 1 - nI_{n-1}.$$

我们可以依据上述递推式以及初始值I_0,求得$I_i, (i = 1, 2, 3, \cdots, 20)$.

但是这样运行出来的程序,I_{20}是一个负值,这显然不可能,这时我们意识到这个算法存在精度问题;

如果我们将源程序中的double改换为long double,会发现结果仍然是负值.

这时我们来分析一下误差的大致增长情况,由递推式$I_n = 1 - nI_{n-1}$可以知道,虽然负号的存在会使得每次递推抵消掉一部分误差,但从大体上看,误差的增长率是阶乘级别的.事实上根据和准确值的对比,可以发现,从I_{12}开始就存在不少于10^{-6}的误差,那么到I_{20}误差的累积量可与$\frac{20!}{12!} \times 10^{-6}$同阶,这样的误差是不可以忽略的.

这里有一种非常经典的处理方式,即我们先通过数值积分的手段,求出I_{20}的近似值,误差记作δ_{20},这时我们反过来利用递推式,倒推出前面的各项:$I_{n-1} = \frac{1 - I_n}{n}$,可以证明:

$$\delta_n \le \frac{n!}{20!} \cdot \delta_{20} \le \delta_{20}, (0 \le n \le 20)$$

只要I_{20}的误差满足要求,则前面各项的误差都必然满足要求,因为倒推过程中,各项误差在不断减小.

计算I_{20}的方法:

一种是计算黎曼和,原理即为:

$$\int_0^1 f(x)dx = \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n} f(\frac{i}{n}) \approx \frac{1}{N}\sum_{i=1}^{N} f(\frac{i}{N})$$

另一种常用的高效手段是: 自适应辛普森积分

令$F(x) = \int_0^x f(x)dx$,则有$\int_a^b f(x)dx = F(b) - F(a)$.

记$h = \frac{b-a}{2}, m = \frac{a+b}{2}$, 首先我们考察下列泰勒展开式:

$$F(b) = F(m) + h \cdot f(m) + \frac{h^2}{2} \cdot f'(m) + \frac{h^3}{6}f''(m) + O(h^4) \quad (1)$$

$$F(a) = F(m) - h \cdot f(m) + \frac{h^2}{2} \cdot f'(m) - \frac{h^3}{6}f''(m) + O(h^4) \quad (2)$$

$$f(b) = f(m) + h \cdot f'(m) + \frac{h^2}{2} \cdot f''(m) + O(h^3) \quad (3)$$

$$f(a) = f(m) - h \cdot f'(m) + \frac{h^2}{2} \cdot f''(m) + O(h^3) \quad (4)$$

$3 \times ((1) - (2)) - h \times ((3) + (4))$:

$$3(F(b) - F(a)) - (f(a) + f(b)) \cdot h = 4h \cdot f(m) + O(h^4) \quad (5)$$

$$(5) \Rightarrow \int_a^b f(x)dx = F(b) - F(a) = \frac{f(a) + 4f(m) + f(b)}{3} \cdot h + O(h^4)$$

$$\therefore \int_a^b f(x)dx = \frac{f(a) + 4f(\frac{a+b}{2}) + f(b)}{6} \cdot (b - a) + O(h^4)$$

利用$\frac{1}{6}\left(f(a)+4f(\frac{a+b}{2})+f(b)\right)\cdot(b-a)$来近似计算$\int_a^b f(x)dx$精度要比利用$f(\frac{a+b}{2})\cdot(b-a)$好很多
这里再使用自适应的分治算法去计算区间$[L,R]$上的定积分,精度 ϵ :

1. 取$[L,R]$中点$M=\frac{L+R}{2}$;
2. 分别利用上述计算公式计算区间$[L,M]$上的积分近似值S_1, 区间$[M,R]$上的积分近似值S_2, 区间$[L,R]$上的积分近似值S;
3. 计算差值$\delta=|S_1+S_2-S|$;
4. 如果$\delta<\epsilon$,则返回 S作为结果;
5. 否则递归计算$[L,M]$,$[M,R]$上精度为$\frac{\epsilon}{2}$ 的积分近似值,相加后返回.

但这样其实还有一些不常见的特殊情况,比如说:考察函数: $f(x)=\sin^2\pi x$在区间$[0,4]$上的积分:
用上面的算法计算,会因为:$f(0)=f(1)=f(2)=f(3)=f(4)=0$而得出积分值等于0的荒谬结果.
为了避免这样的问题,我们可以有一个简单粗暴的办法,在上述的第4步中,额外加一条判断,限定区间长度$R-L$必须小于给定的值;因为误差值是区间长度的四阶小量,所以保证区间长度足够小,即可保证误差足够小.

当然这里还存在另一种调整方式:对于区间$[L,R]$取的分割点M不一定要是中点,这里如果加入随机化处理,就有利于避免被特殊数据针对.

但实际上,面对大多数情况,上面的自适应辛普森积分是可以高效地给出正确结果的.