南京大學

第 1 次实习报告

学　号　*161xx1yyy*

学生姓名　*Yvonne Wong*

提交日期　2018 年 12 月 6 日

实验一　数值方法求解定积分：用不同求积方法计算定积分值

(一)　题干

用不同积分方法计算如下定积分值

$$\int_0^1 \sqrt{x}\ln x\,dx$$

算法 1 取不同的步长 h。分别用复化梯形及复化辛普森求积公式计算积分，

算法 2 用龙贝格求积计算完成问题 1，使其精度达到10^{-4}。

(二)　分析

算法 1 取不同步长，使用复化梯形及复化辛普森求积公式计算定积分值.

算法 2 总结一下，可以得到龙贝格算法的过程如下：

$$\begin{cases}
T_1 = \dfrac{b-a}{2}[f(a)+f(b)] \\
T_{2^k} = \dfrac{1}{2}T_{2^{k-1}} + \dfrac{b-a}{2^k}\sum_{j=0}^{2^{k-1}-1} f(a+(2j+1)\dfrac{b-a}{2^k}) \quad k=1,2,\ldots \\
S_{2^{k-1}} = \dfrac{4}{3}T_{2^k} - \dfrac{1}{3}T_{2^{k-1}} \\
C_{2^{k-1}} = \dfrac{16}{15}S_{2^k} - \dfrac{1}{15}S_{2^{k-1}} \\
R_{2^{k-1}} = \dfrac{64}{63}C_{2^k} - \dfrac{1}{63}C_{2^{k-1}} \\
T_{m+1}(h) = \dfrac{4^m T_m(h/2)}{4^m-1} - \dfrac{T_m(h)}{4^m-1}
\end{cases}$$

其中 $T_m(h)$ 指步长为 *h*的 *2m-2*阶 *Newton-Cotes*公式计算结果，$T_m(\frac{h}{2})$ 指步长为 $\frac{h}{2}$ 的 *2m-2*阶 *Newton-Cotes*公式计算结果。

(三)　程序流程图

随便来张图：

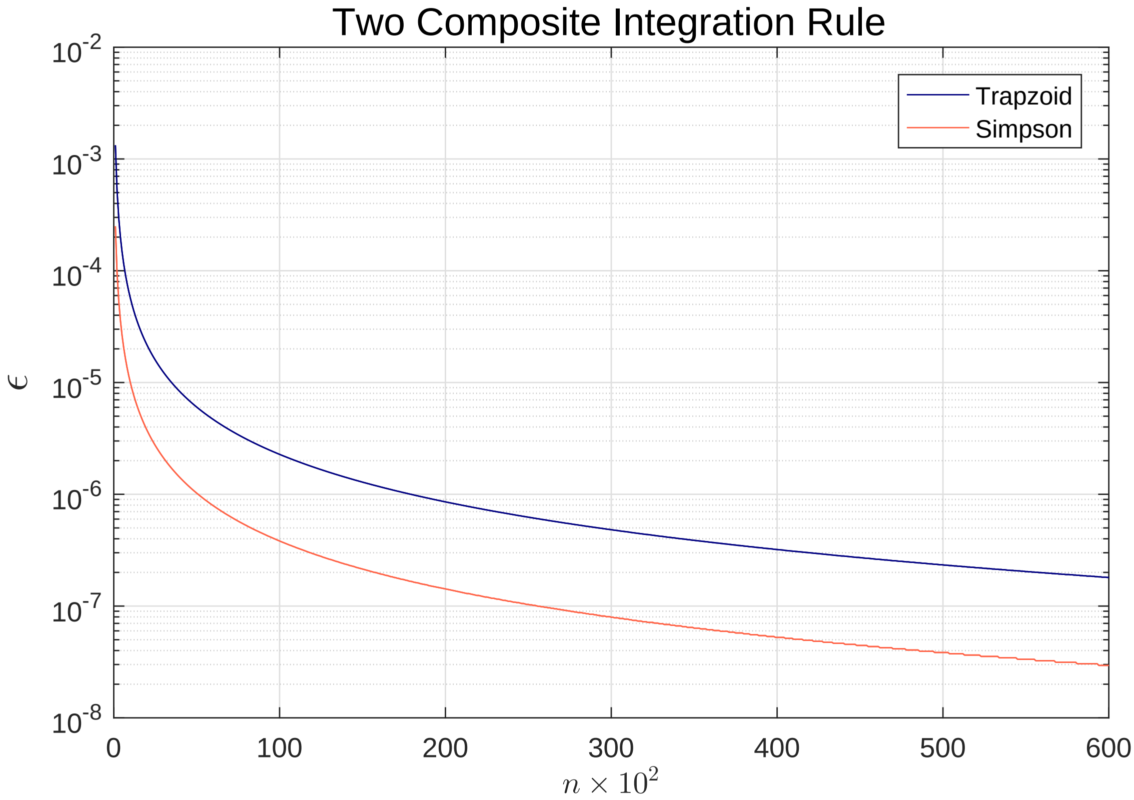

图 1: 用复化梯形及复化辛普森求积公式计算积分

(四) 运行结果

$0 \neq 1$

(五) 结果分析

算法 1 探究复化求积公式的误差，根据其余项公式可得到：

复化梯形公式的余项是区间步长 h的 2 阶无穷小量:

$$R_n(f) = -\frac{b-a}{12}h^2 f''(\eta) \quad \eta \in [0,1]$$

算法 2 龙贝格求积方法可总结为

$$T_{m+1}(h) = \frac{4^m T_m(h/2)}{4^m - 1} - \frac{T_m(h)}{4^m - 1}$$

其中 $T_m(h)$ 指步长为 h的 *2m-2*阶 *Newton-Cotes*公式计算结果，$T_m(\frac{h}{2})$ 指步长为 $\frac{h}{2}$ 的 *2m-2*阶 *Newton-Cotes*公式计算结果。在实际计算中，一直计算到 $|T_{m+1}(h) - T_m(h)| < \varepsilon = 1 \times 10^{-4}$ 为止。

采用这种计算方法，可得到 $m = 4$, 龙贝格积分值为-0.43760383679; 若采用 $|R - I| < \varepsilon = 1 \times 10^{-4}$ 为终止计算条件，可以得到不同的结果：$m = 9$, 龙贝格积分值为-0.44438622012.

(六) 编程总结

1. 处理数据时，可简单定义一个量以量化指示某种需要观察得到的变化，可通过编程简单计算再结合画图直观呈现的方式获得更好的理解。

2. 递归函数难度较大，常需要结合分治的思想，但一些可以借用循环完成的过程，不必要引入递归函数的操作

(七)　源代码

```
module integration
   implicit none
   real(kind=8) :: a=0.0,b=1.0
contains
   real(kind=8) function f(x)
   implicit none
   real(kind=8) x
   f=sqrt(x)*log(x)
end
end
program work6_1
   use integration
   implicit none
   real(kind=8):: T,S,h
   external T,S
   integer n
   open(10,file=”..//Result//6_1.csv”)
   do n=100,60000,100
      h=(b-a)/real(n);
      WRITE(*,”(’h=’,F12.9,4x,’T=’,F12.9,4x,’S=’,F12.9)”) h,T(n),S(n)
      WRITE(10,”(F12.9,’,’,F12.9,’,’,F12.9)”) h,T(n),S(n)
   enddo
   call Romberg()
end program
!T,S,C,R stands for integration(using different methods)
!using trapzoid integration
real(kind=8) function T(n)
   use integration
   implicit none
   real(kind=8):: h
   integer k,n
   h=(b-a)/real(n);T=f(b)
   do k=1,n-1
   T=T+2.0*f(a+k*h)
   enddo
   T=h*T/2.0
end
!using simpson(h)
real(kind=8) function S(n)
   use integration
   implicit none
   real(kind=8):: h
   integer k,n
   h=(b-a)/real(n);S=f(b)
   S=S+4*f(a+h/2.0)
   do k=1,n-1
   S=S+4.0*f(a+k*h+h/2.0)+2.0*f(a+k*h)
   enddo
   S=h*S/6.0
end
!using Romberg
subroutine Romberg()
   use integration
   implicit none
   real(kind=8):: h,T(10)=0.0,TT,s,q
   real(kind=8),parameter :: e=1e-4
   integer j,k,m
   T(1)=(b-a)/2.0*f(b);k=1;m=1;h=b-a
```

```
    do while(.true.)
       TT=0.0
       do j=0,k-1
       TT=TT+f(a+(j+0.5)*h)
       enddo
       TT=(TT*h+T(1))/2.0
       s=1.0
       do j=1,m
          s=4*s;q=(s*TT-T(j))/(s-1)!linear combination
          T(j)=TT;TT=q
       enddo
       write(*,"('m = ',I2,2x,'T(m+1)= ',F14.11,' T(m)=',F14.11,' difference = ',F14.11)") m,q,T(m),q-T(m)
       !if(abs(q-T(m))<=e) exit
       if(abs(q+4/9)<=e) exit
       m=m+1;T(m)=q;k=2*k
       h=h/2.0
    enddo
    write(*,"('m = ',I2,2x,'Romberg method = ',F14.11)") m,q
end
```