Numerical Analysis Homework7

Zhang Jiyao,PB20000204

2023 年 4 月 20 日

1 Introduction

编写一个子程序,用来执行定义在任意区间$[a,b]$上的函数f的龙贝格算法。用户要具体指定阵列中所计算的行数,并且当计算完成后要看到整个阵列。编写一个主程序并且用下列积分测试这个子程序。

$$\int_0^1 \frac{sinx}{x}dx \tag{1}$$

$$\int_{-1}^1 \frac{cosx - e^x}{sinx}dx \tag{2}$$

$$\int_1^\infty (xe^x)^{-1}dx \tag{3}$$

2 Method

考虑龙贝格算法：对于任意区间$[a,b]$,设$h_0 = b-a$,$h_n = \frac{h_{n-1}}{2}$ $(n \geq 1)$

使用上述公式,我们有

$$R(0,0) = \frac{1}{2}(b-a)[f(a)+f(b)]$$

$$R(n,0) = R(n-1,0) + h_n \sum_{i=1}^{2^{n-1}} f(a+(2i-1)h_n)$$

则对于一个适度的M值,计算出$R(0,0), R(1,0), R(2,0), ..., R(M,0)$的值。然后不需要依赖被积函数$f$的值,根据公式

$$R(n,m) = R(n,m-1) + \frac{1}{4^m-1}[R(n,m-1) - R(n-1,m-1)]$$

可以求得全部的阵列。

编写对应的子程序Romberg.m。用户在调用时需要自行输出对应的区间,被积函数,以及输出阵列的行数M。

注意对于本题的三个积分,我们不能直接代入区间计算。因为它们是反常积分。我们可以取一个很小的ϵ,例如对第一个积分,在区间$[\epsilon,1]$上进行积分。

对于第一个积分,注意到$\lim\limits_{x\to 0^+} \frac{sinx}{x} = 1$,因此取充分小的$\epsilon$,在区间$[0,\epsilon]$上积分的贡献是小于$\epsilon$的,因此可以忽略不计。

对于第二个积分,注意到0是一个间断点,因此在区间$[-1,-\epsilon]$以及$[\epsilon,1]$上分别来讨论即可。也是注意到极限$\lim\limits_{x\to 0}\frac{cosx-e^x}{sinx}=1$

对于第三个积分,进行变量替换$x=\frac{1}{t}$,原式化为

$$\int_0^1 \frac{1}{te^{\frac{1}{t}}}dt$$

并且注意到极限$\lim\limits_{x\to 0^+}\frac{1}{xe^{\frac{1}{x}}}=0$方法同第一个。

3 Results

求得的结果如下图所示。依照次序,分别是三个积分求得的值。

0.920735	0	0	0	0	0	0
0.939793	0.946146	0	0	0	0	0
0.944514	0.946087	0.946083	0	0	0	0
0.945691	0.946083	0.946083	0.946083	0	0	0
0.945985	0.946083	0.946083	0.946083	0.946083	0	0
0.946059	0.946083	0.946083	0.946083	0.946083	0.946083	0
0.946077	0.946083	0.946083	0.946083	0.946083	0.946083	0.946083

-1.95171	0	0	0	0	0	0
-2.06277	-2.09979	0	0	0	0	0
-2.1451	-2.17255	-2.1774	0	0	0	0
-2.19342	-2.20952	-2.21199	-2.21254	0	0	0
-2.2194	-2.22806	-2.22929	-2.22957	-2.22963	0	0
-2.23284	-2.23732	-2.23794	-2.23808	-2.23811	-2.23812	0
-2.23968	-2.24196	-2.24227	-2.24234	-2.24235	-2.24236	-2.24236

0.18394	0	0	0	0	0	0
0.227305	0.24176	0	0	0	0	0
0.219834	0.217344	0.215716	0	0	0	0
0.219351	0.21919	0.219313	0.21937	0	0	0
0.219384	0.219394	0.219408	0.21941	0.21941	0	0
0.219384	0.219384	0.219383	0.219383	0.219383	0.219383	0
0.219384	0.219384	0.219383	0.219383	0.219383	0.219383	0.219383

4 Discussion

这三个积分我们都可以求得准确值。对比误差可知,Romberg积分求得的结果还是较为准确的,收敛速度也较快。

5 Computer Code

```
function [Dy,dy,n] =Romberg(f,a,b,M)
h=b-a;
Dy(1,1)=0.5*h*(f(a)+f(b));



for n=2:M+1
    h=h/2;
    Dy(n,1)=0.5*Dy(n-1,1);
    for i=1:2^(n-2)
        Dy(n,1)=Dy(n,1)+h*f(a+(2*i-1)*h);
    end

    for m=2:n

        Dy(n,m)=Dy(n,m-1)+(Dy(n,m-1)-Dy(n-1,m-1))/(4^(m-1)-1);

    end


end


[k,k]=size(Dy);
dy=Dy(k,k);
f=@(x)(sin(x)/x);
g=@(x)((cos(x)-exp(x))/sin(x));
h=@(x)1/(x*exp(1/x));

a=h(1);

D1=Romberg(f,1e-16,1,6);
D2=Romberg(g,-1,-1*1e-16,6)+Romberg(g,1e-16,1,6);
D3=Romberg(h,1e-16,1,6);

digits(6);

disp(vpa(D1));
disp(vpa(D2));
```

```
disp(vpa(D3));
```