学号：20002462 姓名：刘子言

实验二 数值积分实验

一. 实验目的

（1）熟悉数值积分与数值微分方法的基本思想，加深对数值积分与数值微分方法的理解。

（2）熟悉 Matlab 编程环境，利用 Matlab 实现具体的数值积分与数值微分方法。

二. 实验要求

用 Matlab 软件实现复化梯形方法、复化辛普森方法、龙贝格方法和高斯公式的相应算法，并用实例在计算机上计算。

三. 实验内容

1. 实验题目

编写用复化梯形法、复化辛普森法、龙贝格法、三点高斯法求积分$I=\frac{2}{\sqrt{\pi}}\int_0^1 e^{-x}dx$近似值的计算机程序，要求误差不超过 10^{-6}。

2. 设计思想

（1）复化求积法的设计思想

设将求积区间[a,b]划分为 n 等分，步长 h=(b-a)/n，等分点为 $x_i=a+ih$，i=0,1,...,n。复化求积法就是先用低阶求积公式，求得每个子段$[x_i,x_{i+1}]$上的积分值I_i，然后再将他们累加求和，用各段积分之和$\sum_{i=0}^{n-1}I_i$作为所求积分的近似值。

• 复化梯形法的计算公式设计：

$$T_n=\sum_{i=0}^{n-1}\frac{h}{2}\left[f(x_i)+f(x_{i+1})\right]=\frac{b-a}{2n}\left[f(a)+2\sum_{i=1}^{n-1}f(x_i)+f(b)\right]$$

• 复化辛普森法的计算公式设计：

$$S_n=\sum_{i=0}^{n-1}\frac{h}{6}\left[f(x_i)+4f\left(x_{i+\frac{1}{2}}\right)+f(x_{i+1})\right]$$

$$=\frac{b-a}{6n}[f(a)+4\sum_{i=0}^{n-1}f\left(x_{i+\frac{1}{2}}\right)+2\sum_{i=1}^{n-1}f(x_i)+f(b)]$$

（2）龙贝格法的设计思想

对复化辛普森公式进行加工，取$\omega=\frac{1}{15}$，可以得到复化 Cotes 公式：

$$C_n=\frac{4^2S_{2n}-S_n}{4^2-1}=\frac{16}{15}S_{2n}-\frac{1}{15}S_n$$

继续加工，取$\omega=\frac{1}{63}$时，就可以得到龙贝格公式：

$$R_n=\frac{4^3C_{2n}-C_n}{4^3-1}=\frac{64}{63}C_{2n}-\frac{1}{63}C_n$$

（3）三点高斯法的设计思想

首先取积分区域[a,b]，n+1 个结点的求积公式：$\int_a^b f(x)dx\approx(b-a)\sum_{i=0}^{n}\lambda_i f(x_i)$

根据公式$\sum_{i=0}^{n}\lambda_i x_i^k=\frac{1+(-1)^k}{2(k+1)},k=0,1,...,2n+1$，适当选取$x_i,\lambda_i$的值可使求积公式最高可具有 2n+1 阶精度。这样构造出的三点高斯公式为：

$$G_3=\frac{5}{9}f\left(-\sqrt{\frac{3}{5}}\right)+\frac{8}{9}f(0)+\frac{5}{9}f\left(\sqrt{\frac{3}{5}}\right)$$

3. 对应程序

（1）原积分 I 的函数代码如下：

```
function f = liuziyan_2_Integrand_function( x )
f = 2*exp(-x)/sqrt(pi);
```

（2）复化梯形法的程序如下：

```
function [S,S0,N] = liuziyan_2_1_FTrapezoid( f,a,b,eps )
% f--被积函数句柄；a,b--被积区间[a,b]；eps--输入精度
% S--用复化梯形公式求得的积分值；S0--积分的准确值；N--区间个数
% flag--判断精度是否达到要求，0 为不够，1 为达到
N = 0;
flag = 0;
S0 = 2*(1-exp(-1))/sqrt(pi);
while flag == 0
    N = N+1;
    h = (b-a)/N;
    fa = feval(f,a);
    fb = feval(f,b);
    S = fb + fa;
    x = a;
    for i = 1:N-1
        x = x + h;
        fx = feval(f,x);
```

```
        S = S + 2*fx;
    end
    S = S*h/2;
    if abs(S - S0) < eps
        flag = 1;
    end
end
```

（3）复化辛普森法的程序如下：

```
function [S,S0,N] = liuziyan_2_2_FSimpson( f,a,b,eps )
% f--被积函数句柄；a,b--被积区间[a,b]；eps--输入精度
% S--用复化 Simpson 公式求得的积分值；S0--积分的准确值；N--区间个数
% flag--判断精度是否达到要求，0 为不够，1 为达到
N = 0;
flag = 0;
S0 = 2*(1-exp(-1))/sqrt(pi);
while flag == 0
    N = N+1;
    h = (b-a)/N;
    fa = feval(f,a);
    fb = feval(f,b);
    S = fb + fa;
    x = a;
    for i = 1:N
        x = x + h/2;
        fx = feval(f,x);
        S = S + 4*fx;
        if i < N
            x = x + h/2;
            fx = feval(f,x);
            S = S + 2*fx;
        end
    end
    S = S*h/6;
    if abs(S - S0) < eps
        flag = 1;
    end
end
```

（4）龙贝格法的程序如下：

```
function [Ans,R] = liuziyan_2_3_Romberg( f,a,b,eps )
% f--被积函数句柄；a,b--被积区间[a,b]；R--Romberg 表
% Ans--用 Romberg 公式求得的积分值
```

```
% eps--输入精度；err--表示误差的估计
h = b-a;
R(1,1) = h*(feval(f,a)+feval(f,b))/2;
M = 1; J = 0; err = 1;
while err > eps
    J = J+1;
    h = h/2;
    S = 0;
    for p = 1:M
        x = a+h*(2*p-1);
        S = S+feval(f,x);
    end
    R(J+1,1) = R(J,1)/2+h*S;
    M = 2*M;
    for k = 1:J
        R(J+1,k+1) = R(J+1,k)+(R(J+1,k)-R(J,k))/(4^k-1);
    end
    err = abs(R(J+1,J)-R(J+1,J+1));
end
Ans = R(J+1,J+1);
```

（5）三点高斯法的程序如下：

```
function [G,eps] = liuziyan_2_4_ThreeGauss( f,a,b )
% f--被积函数句柄；a,b--被积区间[a,b]；
% G--用 Romberg 公式求得的积分值;G0--积分的准确值
% eps--输入精度
x1 = (a+b)/2 - sqrt(3/5) * (b-a)/2;
x2 = (a+b)/2 + sqrt(3/5) * (b-a)/2;
G = (b-a) * (5*feval(f,x1)/9 + 8*feval(f,(a+b)/2)/9 +...
    5*feval(f,x2)/9)/2;
G0 = 2*(1-exp(-1))/sqrt(pi);
eps = abs(G - G0);
end
```

4. 实验结果

本实验的运行结果截图如下：

（实验运行结果截图中函数的名称体现了姓名+题号+算法名称）

（1）复化梯形法的程序如下：

```
>> f=@liuziyan_2_Integrand_function;a=0;b=1;eps=10^(-6);
>> [S,S0,N] = liuziyan_2_1_FTrapezoid( f,a,b,eps )

S =

   0.713272668050509


S0 =

   0.713271669674918


N =

   244
```

（2）复化辛普森法的程序如下：

```
>> f=@liuziyan_2_Integrand_function;a=0;b=1;eps=10^(-6);
>> [S,S0,N] = liuziyan_2_2_FSimpson( f,a,b,eps )

S =

   0.713272635314936


S0 =

   0.713271669674918


N =

     4
```

（3）龙贝格法的程序如下：

```
>> f=@liuziyan_2_Integrand_function;a=0;b=1;eps=10^-6;
>> [Ans,R] = liuziyan_2_3_Romberg( f,a,b,eps )

Ans =

   0.713271669814180


R =

   0.771743332258054                   0                   0                   0
   0.728069946441243   0.713512151168973                   0                   0
   0.716982762290904   0.713287034240791   0.713272026445579                   0
   0.714200167058928   0.713272635314936   0.713271675386546   0.713271669814180
```

（4）三点高斯法的程序如下：

```
>> f=@liuziyan_2_Integrand_function;a=0;b=1;
>> [G,eps] = liuziyan_2_4_ThreeGauss( f,a,b )

G =

   0.713271327590424


eps =

     3.420844943979873e-07
```

四. 实验体会

对实验过程进行分析总结，对比不同方法的精度，指出每种算法的设计要点及应注意的事项，以及自己通过实验所获得的对数值积分方法的理解。

答：

1、对于外推方法得到的一系列算法公式

外推方法即用若干个积分近似值推算出更为精确的积分近似值的方法。

（1）由复化梯形公式的余项表达式加工，可以得到复化辛普森公式，再加工可以得到复化 Cotes 公式，继续加工就可以得到龙贝格公式；

（2）复化梯形公式：算法简单，无论 n 取多大都是数值稳定的，但收敛慢（N=244），精度低；它是低精度的方法，但对于光滑性较差的函数，有时比用高精度的方法能得到更好的效果。

（3）复化辛普森公式也是数值稳定的，且收敛比复化梯形公式要快（N=4）；

（4）龙贝格加速算法简单，效果明显，当节点加密提高积分近似程度时，前面的计算结果还可以为后面的计算中使用，因此，有利于减少计算量；它的代数精度为 m 的两倍，收敛阶高达 m+1 的两倍。

2、对于 Gauss 方法求数值积分

（1）一点高斯公式具有 1 阶精度，两点高斯公式具有 3 阶精度，三点高斯公式可以达到 5 阶精度；低阶高斯公式的代数精度较高，高斯系数≥0，故计算过程稳定；

（2）但由于增加节点时前面已经算出的函数值后面不能再用，而且更高阶的高斯公式的构造更为复杂，其实有使用价值的仅仅是低阶的高斯公式；

（3）Gauss 方法虽然计算过程比较麻烦，但精度高，特别是对计算无穷区间上的积分和广义积分，则是其他方法所不能比的。