计　算　方　法

实验三 Romberg 积分

姓名　孙骁

学号 1180300811

院系 计算机学院

专业　计算机系

哈尔滨工业大学

实验报告三 Romberg 积分

题目

Romberg 积分

摘要

利用复化梯形求积公式的误差估计式计算积分 $\int_a^b f(x)dx$. 记 $h=\frac{b-a}{n}$ $x_k=a+k\cdot h$， $k=0,1,\cdots,n$，其计算公式：

$$T_n=\frac{h}{2}\sum_{k=1}^{n}[f(x_{k-1})+f(x_k)]$$

复化梯形求积公式的误差，若 $f(x)\in C^2[a,b]$，则：

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

前言（目的和意义）

目的：

利用 Romberg 积分法计算积分 $I(f)=\int_a^b f(x)dx$.

意义：

通过此次实验，使用编程语言实现 Romberg 积分法，学会使用 Romberg 积分法求 $f(x)$ 在给定区间上的积分，以解决其他科学实验中的函数求根计算问题.

数学原理

利用复化梯形求积公式的误差估计式计算积分 $\int_a^b f(x)dx$.记 $h=\frac{b-a}{n}$

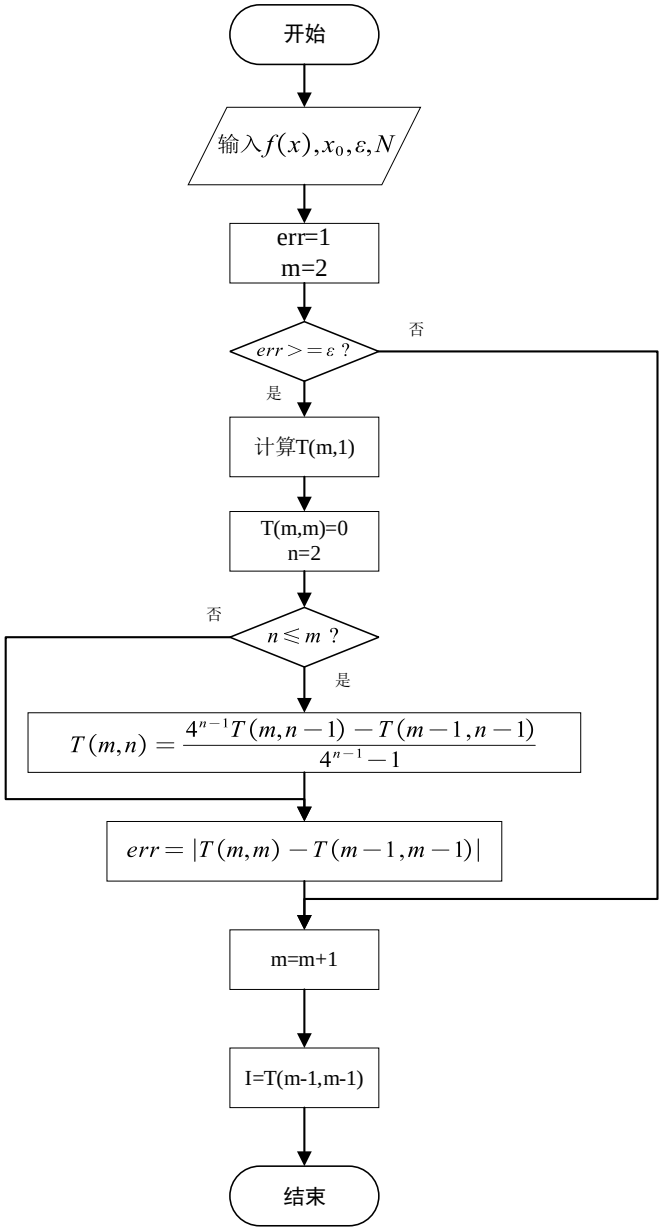
$x_k=a+k\cdot h$， $k=0,1,\cdots,n$，其计算公式：

$$T_n=\frac{h}{2}\sum_{k=1}^{n}[f(x_{k-1})+f(x_k)]$$

复化梯形求积公式的误差，若 $f(x)\in C^2[a,b]$，则：

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

程序设计流程



实验结果、结论与讨论

问题 1：

（1）

```
>> I = Romberg(inline('x.^2.*exp(x)'),0,1,25,1e-6);
>> I

I =

   0.718281829586945
```

（2）

```
>> I = Romberg(inline('exp(x).*sin(x)'),1,3,25,1e-6);
>> I

I =

  10.950170314696752
```

（3）

```
>> I = Romberg(inline('4./(1+x.^2)'),0,1,25,1e-6);
>> I

I =

   3.141592653313126
```

（4）

```
>> I = Romberg(inline('1./(1+x)'),0,1,25,1e-6);
>> I

I =

   0.693147180663664
```

思考题

（1）

输入的参数 N 越大，由复化梯形公式的误差公式$E=-\frac{b-a}{12}h^2 f''(\eta)$可知，在有限区间上分段越小，计算精度越高.

（2）

二分次数越多，计算精度越高。因为每一次二分，N 变为原来的二倍，由复化梯形公式的误差$E=-\frac{b-a}{12}h^2 f''(\eta)$，二分后误差约为二分前的四分之一.

程序代码

Romberg.m

```
function I = Romberg(fun, a, b, npanel, tol)
% RombergInterg  用Romberg方法求积分
%
% Synopsis:  I = Romberg(fun,a,b,n,tol)
%
% Input:     fun    = (string) 被积函数的函数名
%            a, b   = 积分下限和积分上限
%            npanel = (optional) 将积分区间平分的段数
%            tol    = (optional) 计算误差上限
%
% Output:    I = 通过Romberg方法求积分的近似值
T(1,1) = Trapezoid(fun, a, b, npanel);
err = 1;
m = 2;
while err >= tol
   T(m,1) = Trapezoid(fun, a, b, 2^m*npanel);
   T(m,m) = 0;
   for n = 2:m
       T(m,n) = ( 4^(n-1)*T(m,n-1) - T(m-1,n-1)) / (4^(n-1) - 1);
   end
   err = abs( T(m,m) - T(m-1,m-1) );
   m = m + 1;
end
I = T(m-1,m-1);
end
```

TrapezoidInteg.m

```
function  I = Trapezoid(fun, a, b, npanel)
% TrapezoidInteg  用复化梯形公式求积分
%
% Synopsis:  I = Trapezoid(fun,a,b,n)
%
% Input:     fun    = (string) 被积函数的函数名
%            a, b   = 积分下限和积分上限
%            npanel = (optional) 将积分区间平分的段数，默认为25
%
% Output:    I = 通过复化梯形公式求积分的近似值
```

```
if nargin < 4
    npanel = 25;
end
nnode = npanel + 1;
h = (b-a)/(nnode-1);
x = a:h:b;
f = feval(fun,x);
I = 0.5 * h * ( f(1) + 2 * sum(f(2:nnode-1)) + f(nnode) );
```