|  |  |
| --- | --- |
| 姓名 |  |
| 评分 |  |

* 实验报告

实验报告

课程名称： 数值分析

课题名称： 龙贝格数值积分算法的实验与分析

专 业： 地球物理

姓 名： 王锴

班 级： 201145

完成日期： 2016 年 12 月 8 日

**目录**

[**一、实验名称** 1](#_Toc468984696)

[**二、实验目的** 1](#_Toc468984697)

[**三、实验要求** 1](#_Toc468984698)

[**四、实验原理** 1](#_Toc468984699)

[**五、实验题目** 1](#_Toc468984700)

[**六、实验步骤** 1](#_Toc468984701)

[**七、实验整体流程图** 2](#_Toc468984702)

[**八、程序代码** 3](#_Toc468984703)

[**九、运行结果及实验结果分析** 5](#_Toc468984704)

[**十、拓展：使用GUI实现龙贝格积分算法** 5](#_Toc468984705)

[**十一、实验体会** 7](#_Toc468984706)

**实验报告**

* **一、实验名称**

龙贝格数值积分算法的实验与分析。

* **二、实验目的**

1. 掌握龙贝格数值积分算法的原理与方法；
2. 培养编程与上机调试能力；
3. 熟悉matlab软件环境。

* **三、实验要求**

1. 通过编程实现龙贝格积分算法；
2. 分析龙贝格积分算法的优缺点。

* **四、实验原理**

在求解定积分的值的问题中,Newton-Leibnitz公式无论在理论上还是在解决实际问题上都起了很大作用，但它并不能完全解决定积分的计算问题，因为积分学涉及的实际问题极为广泛，而且极其复杂，在实际计算中经常遇到以下三种情况：①被积函数f(x)并不一定能够找到用初等函数的有限形式表示的原函数F(x)；②被积函数f(x)的原函数能用初等函数表示，并不复杂，但表达式太复杂；③被积函数f(x)没有具体的解析表达式，其函数关系由表格或图形表示。对于这些情况，就需要用数值解法来建立积分的近似计算方法。

龙贝格数值积分法是建立在变步长梯形公式和理查德加速算法的基础上一种算法，它具有精度高、收敛快和步长可自适应选取的优点，其计算过程可由以下方程表示：

* **五、实验题目**

编程实现一种积分算法并分析其优缺点。

* **六、实验步骤**

1. 用梯形公式计算积分近似值：
2. 将区间逐步分半，按变步长梯形公式计算积分近似值：
3. 按加速算法求加速值：
4. 精度控制，直到相邻两次积分值

（其中ε为允许的误差限）则终止计算并取作为积分近似值，否则将区间再对分，重复步骤（1）-（3），直到满足要求。

* **七、实验整体流程图**

开始

输入被积函数f、被积变量x、积分上下限a、b，误差限e

判断f、x是否为字符型，a、b、e是否为单值，否则报错

T(1,1)=(b-a)/2\*(subs(f,x,a)+subs(f,x,b)); n=2; h=b-a;

T(2,1)=T(1,1)/2+h/2\*double(subs(f,x,a+h/2));

T(2,2)=4/3\*T(2,1)-1/3\*T(1,1); d=abs(T(2,2)-T(1,1));

n=n+1; h=h/2; T(n,1)=T(n-1,1)/2; i=1

T(n,1)=T(n,1)+h/2\*double(subs(f,x,a+(i-1/2)\*h));

i=i+1

N

i>2^(n-2)?

Y

i=2

k=4^(i-1);

T(n,i)=k/(k-1)\*T(n,i-1)-1/(k-1)\*T(n-1,i-1);

i=i+1

N

i>n?

Y

d=abs(T(n,n)-T(n-1,n-1));

Y

d>e?

N

输出积分估计值 I=T(n,n)

结束

* **八、程序代码**

1. **龙贝格积分函数：**

function I=Romberg(f,x,a,b,e)

% Romberg 求积函数

% I=Romberg(f,x,a,b,e)

% f为字符型，被积函数

% x为字符型，被积变量

% a为单值，积分下限

% b为单值，积分上限

% e为单值，精度要求

% I为积分的估计值

if ischar(f)==0||ischar(x)==0 %f、x应为字符型

error('f and x must be character!'); %报错提示

return;

end

if max(size(a))>1||max(size(b))>1||max(size(e))>1 %a、b、e应为单值

error('The size of a, b and e must be 1!'); %报错提示

return;

end

T(1,1)=(b-a)/2\*(subs(f,x,a)+subs(f,x,b)); %用梯形公式计算T(1,1)

T=double(T); %转换为数值型

n=2; %n表示T矩阵行号

h=b-a; %步长赋值

T(2,1)=T(1,1)/2+h/2\*double(subs(f,x,a+h/2)); %用变步长梯形公式计算T(2,1)

T(2,2)=4/3\*T(2,1)-1/3\*T(1,1); %用梯形加速公式计算T(2,2)

d=abs(T(2,2)-T(1,1)); %估计T(2,2)与真实值I的误差

while d>e %误差是否达到误差限

n=n+1; %行号加1

h=h/2; %步长赋值

%%%%%%%%%%%% 用变步长梯形公式计算T(n,1) %%%%%%%%%%%%

T(n,1)=T(n-1,1)/2; %计算首项

for i=1:2^(n-2)

T(n,1)=T(n,1)+h/2\*double(subs(f,x,a+(i-1/2)\*h)); %计算求和项

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=2:n

k=4^(i-1);

T(n,i)=k/(k-1)\*T(n,i-1)-1/(k-1)\*T(n-1,i-1); %用加速公式计算T(n,i)

end

d=abs(T(n,n)-T(n-1,n-1)); %估计T(n,n)与真实值I的误差

end

I=T(n,n); %求得积分估计值I

1. **调用龙贝格积分函数：**

f='4/(1+x^2)'; %被积函数

x='x'; %被积变量

a=0; %积分下限

b=1; %积分上限

e=1e-5; %误差限

I=Romberg(f,x,a,b,e) %输出积分近似值

* **九、运行结果及实验结果分析**

|  |  |  |  |
| --- | --- | --- | --- |
| （a） | | （b） | |
| （c） | （d） | |

图9.1 龙贝格积分算法实验结果

四次数值实验均有较好的积分结果，经检验，积分结果均在误差限内，有些甚至比误差限精确两个数量级，对于被积函数也具有普适性，误差可以达到任意想要的精度，满足高精度的需要，同时在加速算法的应用下，计算次数被大大压缩，几乎每计算一行T矩阵，积分近似值的精度就可以提高两个数量级，计算效率大大提高。

* **十、拓展：使用GUI实现龙贝格积分算法**

为了更加灵活地运用龙贝格积分算法，我将其用GUI实现，其操作更加简便。核心代码如下：

function pushbutton1\_Callback(hObject, eventdata, handles)%“计算”按钮的响应函数

f=get(handles.edit3,'String'); %读取被积函数

x=get(handles.edit4,'String'); %读取被积变量

a=str2double(get(handles.edit2,'String')); %读取积分下限

b=str2double(get(handles.edit1,'String')); %读取积分上限

e=str2double(get(handles.edit6,'String')); %读取误差限

T(1,1)=(b-a)/2\*(subs(f,x,a)+subs(f,x,b)); %用梯形公式计算T(1,1)

T=double(T); %转换为数值型

n=2; %n表示T矩阵行号

h=b-a; %步长赋值

T(2,1)=T(1,1)/2+h/2\*double(subs(f,x,a+h/2)); %用变步长梯形公式计算T(2,1)

T(2,2)=4/3\*T(2,1)-1/3\*T(1,1); %用梯形加速公式计算T(2,2)

d=abs(T(2,2)-T(1,1)); %估计T(2,2)与真实值I的误差

while d>e %误差是否达到误差限

n=n+1; %行号加1

h=h/2; %步长赋值

%%%%%%%%%%%% 用变步长梯形公式计算T(n,1) %%%%%%%%%%%%

T(n,1)=T(n-1,1)/2; %计算首项

for i=1:2^(n-2)

T(n,1)=T(n,1)+h/2\*double(subs(f,x,a+(i-1/2)\*h)); %计算求和项

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=2:n

k=4^(i-1);

T(n,i)=k/(k-1)\*T(n,i-1)-1/(k-1)\*T(n-1,i-1); %用加速公式计算T(n,i)

end

d=abs(T(n,n)-T(n-1,n-1)); %估计T(n,n)与真实值I的误差

end

I=T(n,n); %求得积分估计值I

en=fix(-(log10(e)))+1; %计算结果显示位数

set(handles.edit5,'String',num2str(I,['%.' num2str(en) 'f'])); %输出积分近似值

运行结果如下：

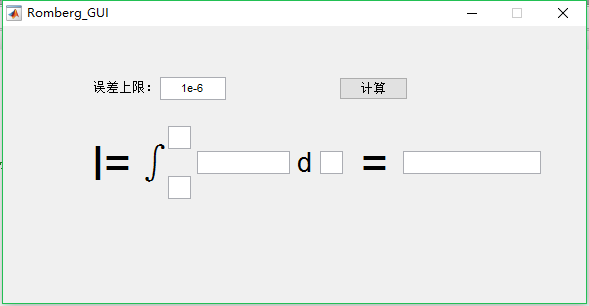


图10.1 初始界面（误差限默认为）

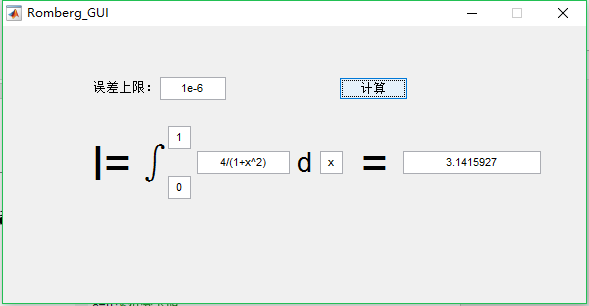


图10.2 实验1

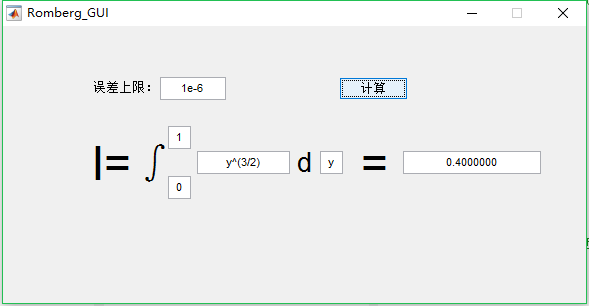


图10.3 实验2

龙贝格算法使用GUI实现之后，其使用更加灵活方便，可以作为计算器使用。

* **十一、实验体会**

本次实验中，我掌握了龙贝格数值积分算法的原理与计算过程，并通过编程实现了定积分的求解，在分析结果的过程中，我主要得到了以下结论：①龙贝格算法可以达到任意精度；②计算效率高，几乎每计算一行T矩阵，积分近似值的精度就可以提高两个数量级；③具有普适性，可以求解几乎所有函数的定积分。

但是，从代数精度的角度来考虑龙贝格积分算法，可以发现它的代数精度受到节点个数的限制，这是它的主要缺陷，高斯型求积则较好地解决了这个问题，但同时，高斯型求积不如龙贝格求积算法简便。

最后，我要感谢付老师的谆谆教诲与同学们的帮助。