**吉林大学数学实验中心实验报告**

**2021年 11月 3日**

|  |
| --- |
| **课程名称：科学计算方法实验实验 题目： Romberg方法与复合Gauss求积公式**  **姓名：田泽禹 年级专业：2019**  **指导教师：王双**  **算法描述：使用Romberg方法计算积分；使用2点Gauss型求积公式推导相应的复合Gauss求积公式，并计算积分**  **相关内容如下**  代码0 (Romberg)：  function [integral]=Romberg(m,a,b)  Rom=(2^6\*Cotes(2\*m,a,b)-Cotes(m,a,b))/(2^6-1);  integral=Rom;  end    function [f]=Infunc(x) %evaluate  f=exp(x).\*sin(x);  end    function [Q1]=Trapezoid(m,a,b)  h=(b-a)/m;  x=linspace(a,b,m+1);  y=Infunc(x);  Q1=h/2\*(2\*sum(y)-y(1)-y(m+1));  end    function [Q2]=Simpson(m,a,b) %2m  Q2=(4\*Trapezoid(2\*m,a,b)-Trapezoid(m,a,b))/(4-1);  end    function [Q4]=Cotes(m,a,b)  Q4=(2^4\*Simpson(2\*m,a,b)-Simpson(m,a,b))/(2^4-1);  end  运行结果0：首先计算真值      代码2 (Gauss MATLAB)：  function [integral]=CompGauss(m)  x=linspace(1/m,2-1/m,m);  S1=(sum(Infunc(x+1/(sqrt(3)\*m))))/m;  S2=(sum(Infunc(x-1/(sqrt(3)\*m))))/m;  integral=S1+S2;  end    function [y]=Infunc(x)  y=4./(1+x.^2);  end  另一个文件：  function [m]=testGauss()  m=1;  S1=CompGauss(m);  S2=CompGauss(m+1);  while(abs(S1-S2)>1e-5)  m=m+1;  S1=CompGauss(m);  S2=CompGauss(m+1);  end  end  运行结果2；      代码3 (Gauss C):  #include<stdio.h>  #include<stdlib.h>  #include<math.h>  double sum(double a[],int m);  double \*Infunc(double x[],int m,double trans);  int main()  {  double m;  double a,b;  double integral;  scanf("%lf",&m);  scanf("%lf%lf",&a,&b);    double \*x;  x=(double \*)calloc(m,sizeof(double));  for(int k=0;k<m;k++)  {  x[k]=a+k\*(b-a)/m+1/m;  }    double S1,S2;  S1=sum(Infunc(x,m,1/(m\*sqrt(3))),m)/m;  S2=sum(Infunc(x,m,-1/(m\*sqrt(3))),m)/m;  integral=S1+S2;    printf("%lf",integral);  return 0;  }  double sum(double a[],int m)  {  double sum=0;  for(int k=0;k<m;k++)  {  sum+=a[k];  }  return sum;  }  double \*Infunc(double x[],int m,double trans)  {  double \*Infunc;  Infunc=(double \*)calloc(m,sizeof(double));  for(int k=0;k<m;k++)  {  Infunc[k]=4/(1+(x[k]+trans)\*(x[k]+trans));  }    return Infunc;  }运行结果3 (Gauss C):    我们带入上一问中的结果，在C语言中计算  总结：Romberg方法收敛性好，其消去法将误差项严格地消去了，使得收敛速度快。Gauss方法优于其它的插值逼近，且代数精确度较高，使用分区间段的近似可以提高精度，但整体上收敛速度很快。 |