数值分析第六次上机练习实验报告

——数值积分与数值微分

huangyf15

一、问题描述

试用不同数值积分方法计算$I(f)=\int_1^3 f(x)dx$的近似值，其中$f(x)=\frac{1}{x^2}sin\frac{2\pi}{x}$. 注：$I(f)=-0.2387324146478436\cdots$.

(1)把[1,3]分成 4 个子区间，用五点 Gauss-Legendre 求积公式的复合求积公式计算.

(2)用 Romberg 求积算法计算积分，取$\varepsilon=10^{-7}$，并与第一种办法比较.

二、方法描述——复化 Gauss-Legendre 方法与 Romberg 求积算法

数值积分即用数值方法求定积分的值，具体地说，就是研究如何利用被积函数的离散取值列表来计算其在连续区间上的积分．它的最基本思路是，通过设置求积节点并计算各节点处的求积系数作为权重，然后将被积函数在各节点处的函数值进行加权求和，从而得到近似积分值．当然，我们期望随着节点个数的增加，上述求积公式求出的近似积分值能够最终收敛于待求积分；并且被积函数在节点处的微小扰动不会造成求积公式计算结果的大幅度变动，即求积公式应该是稳定的.

作为函数逼近的一个重要应用，我们往往用被积函数对应的逼近多项式的积分来做近似计算，而获得逼近多项式的最常用方法就是插值法，这也就是插值型求积公式的思想．实际上，若节点数固定为$\mathrm{n}+1$，则凡是代数精度不小于 n 的求积公式必定是插值型求积公式．从节点的选取方式上看，插值型求积公式分为等距节点情形的 Newton-Cotes 公式与不等距节点情形的 Gauss 型求积公式．前者优点是算法简便，但不足之处也比较明显：由于节点固定使得可选取参数的个数减少，n 阶 Newton-Cotes 公式能达到的最高代数精度为$\mathrm{n}+1$．而后者由于参数的选取更加自由，n 阶 Gauss 型求积公式的代数精度可达$2\mathrm{n}+1$，此时的求积节点就是给定权函数下$\mathrm{n}+1$次正交多项式的零点.

从稳定性上考虑，等距节点高次插值的不稳定性即 Runge 现象使得高阶 Newton-Cotes 公式的算法是不稳定的．与插值法相类似地，我们可以采取将被积区间分段为多个子区间分而治之的策略，这就是求积公式的复化形式．较为常用的复化求积公式有复化梯形公式(即 1 阶 Newton-Cotes 公式)，复化 Simpson 公式(即 2 阶 Newton-Cotes 公式)，复化 Hermite 公式等．从收敛性上考虑，为了提高算法的收敛阶数，我们常常采用迭代加速的策略．与 Aitken 加速思想相类似地，可以将 Richardson 外推思想用于等距网格减半加密的复化梯形公式得到 Romberg 求积算法，其主要思想是通过消去函数在 0 附近 Taylor 展开的低阶项来获得更快的收敛速度.

本次上机实习题中涉及的复化 Gauss-Legendre 方法与 Romberg 求积算法便是综合了前述思想方法的典例．由于待求积分为常义积分，因此不必考虑区间截断、变量替换、奇点分离或转化为 Gauss 型积分等技巧，只需按照常规步骤进行计算即可．

1.复化 Gauss-Legendre 方法

Step1：将积分区间[a,b]均分为 l 个子区间，并通过变量替换$x = x(t)$及$f(x) = g(t)$将各个子区间上的积分变换至区间$[-1,1]$.

Step2：应用 n 点 Gauss-Legendre 求积公式计算各个子区间上的积分．

(a)求出 n 阶 Legendre 多项式$P_n(x)$，并计算其零点$\{x_k\}_{k=0}^{n}$作为求积节点．

(b)计算求积系数$\{A_k\}_{k=0}^{n}$，并与节点处的被积函数值$\{g(x_k)\}_{k=0}^{n}$加权求和，得到各子区间上的积分值．

$$A_k = \frac{2}{n+1}\frac{1}{P_n(x_k)P'_{n+1}(x_k)}.$$

Step3：将各个子区间上的积分值求和，得到原待求积分的近似计算值．

2.Romberg 求积算法

两个关键步骤为：

(1)应用等距网格减半加密的复化梯形公式计算初始序列$\{T(k,1)\}_{k=0}^{\infty}$.

$$T(k+1,1) = \frac{1}{2}\big(T(k,1) + h(k)H(k)\big), k = 0,1 \ldots$$

$$h(k) = \frac{b-a}{2^k}, k = 0,1 \ldots$$

$$H(k) = \sum_{l=1}^{2^k} f(a + \left(l - \frac{1}{2}\right)h(k)), k = 0,1 \ldots$$

其中，初值$T(0,1)$由下式给出

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

(2)利用 Richardson 外推思想加速得到优化序列$\{T(k,k+1)\}_{k=1}^{\infty}$.

$$T(k+1,j+1) = \frac{4^j T(k+1,j) - T(k,j)}{4^j - 1}, j = 1,2,\ldots,k+1$$

在实际应用中往往采取如下策略：

Step1：由上面各式依次计算得到$T(0,1)$以及 $h(0)$、$H(0)$和$T(1,2)$．若成立$|\mathrm{T}(0,1)-\mathrm{T}(1,2)|<\varepsilon$ (ε为给定的计算精度)，则程序停止，并输出$\mathrm{T}(1,2)$；否则置$k=1$，进入 Step2．

Step2：由以上各式计算得到$h(k)$、$H(k)$和$T(k+1,k+2)$．

Step3：判断$|T(k,k+1)-T(k+1,k+2)|<\varepsilon$？若是，则停止程序，并输出$T(k+1,k+2)$；否则令$k=k+1$，返回 Step2．

三、方案设计

我们通过编写 Matlab 程序来实现任意给定被积函数在给定积分区间上的常义积分计算．两个主程序文件分别为 quad_GaussLegendre.m，quad_Romberg.m；每个文件的主程序输入变量有被积函数和积分区间，前一程序还包括节点数量(默认为 5)和复化个数(默认为 4)，后一程序还包括给定精度(默认为10^{-7})；每个程序的输出变量有数值积分值、精确积分值、绝对误差及其数量级以及程序运行时间，后一程序还包括 Richardson 外推的迭代次数．需要说明的是，前一程序还适用于区间端点为被积函数瑕点的广义积分计算．

由于算法非常明确，且在“方法描述”一节中已做过详细说明，因此这里不再赘述．具体程序的实现可参见所给程序的相关注释．

四、计算结果及其分析

通过运行 Matlab 程序，我们得到了$I(f)=\int_1^3 \frac{1}{x^2} sin\frac{2\pi}{x}dx$的准确值：

$$I_0=-0.2387324146378430045345453436311 8\ldots.$$

以及题干要求的积分近似值：

$$I_{G-L}^{(5,4)}=-0.\underline{23873234}0343\ldots\ (\mathrm{t}=2.69\mathrm{s})$$

$$I_R^{(10^{-7})}=-0.\underline{238732414}267\ldots\ (\mathrm{t}=0.14\mathrm{s})$$

由上面的运行结果，我们似乎可以认为 Romberg 算法无论在计算精度上还是在运算用时上都要优于复化 Gauss-Legendre 方法．为了对上述两种算法的性能作出置信度更高的判断，也为了更深入地了解各个参数对计算结果的影响，我们决定对涉及的参数作更为全面的考察．通过不断调整相关参数，我们获得了如下的实验结果：

表格 1-1：复化 Gauss-Legendre 方法的运行结果绝对误差

分段数 l\节点数 n	5	10	20	40
4	7.43E-07	1.22E-14	5.46E-16	8.81E-18
6	1.32E-08	1.66E-15	5.64E-16	5.46E-16
10	3.98E-10	8.81E-18	8.41E-16	5.64E-16
20	6.52E-13	5.46E-16	8.81E-18	5.46E-16
40	8.24E-16	1.38E-15	1.38E-15	8.24E-16

表格 1-2：复化 Gauss-Legendre 方法的计算过程用时（单位：s）

分段数 l\节点数 n	5	10	20	40
4	2.69	4.65	9.09	21.29
6	3.47	5.19	9.92	24.32
10	4.75	7.06	12.86	27.14
20	7.68	12.09	21.30	39.82
40	14.69	18.87	33.19	60.30

表格 2：Romberg 求积算法的运行结果

给定精度 eps	$\boldsymbol{Err\ of\ T(k,1)}$	$\boldsymbol{Err\ of\ T(k,k+1)}$	迭代次数 k	运行时间 tic-toc/s
1.00E-07	5.06E-03	3.70E-09	6	0.14
1.00E-09	5.09E-04	3.70E-09	6	0.14
1.00E-10	1.27E-04	1.62E-10	7	0.14
1.00E-11	3.18E-05	9.71E-14	8	0.14
1.00E-15	7.96E-06	1.12E-15	9	0.14
1.00E-16	>1.00E-12	>1.00E-12	>30	\

注：1. 上述表格中，带有下划线的参数为自变量，其余均为因变量.

2. 表格 2 中有部分采样数据未列出，相关结果可见于下面的数据分析部分.

对上述结果，我们可以作出如下的分析：

(1)表 1-1 和 1-2 给出了复化 Gauss-Legendre 方法在指定参数下的误差与运算效率. 从表 1-1 中可以注意到，当每段区间的节点个数较少时，增加区间分段数对减小误差的贡献还是比较明显的；然而当每段区间的节点数继续增大时，绝对误差几乎全部稳定在10^{-16}的量级，有时甚至反而会随节点增多而增加. 容易看出，舍入误差是这一情况下的主要影响因素. 结合表 1-2，我们也可以看到，无论是分段数加密还是节点数加密，都会使复化 Gauss-Legendre 方法的计算时间呈代数增长.

(2)表 2 给出了 Romberg 求积算法在给定不同精度下的误差与运算效率. 可以清晰地看到，Romberg 算法在迭代次数较少时的加速效果是非常明显的，并且由于迭代只涉及线性运算，因此运算效率也非常高. 然而随着区间分段数的继续增加，利用复化梯形公式得到的迭代初值$T(k,1)$的绝对误差将在$k=25$左右达到极小值(10^{-14}量级)，而利用 Richardson 外推算法得到的加速序列$T(k,k+1)$的绝对误差将在$k=15$左右达到极小值

(10^{-15}量级)；在经过各自极小值点后，$T(k,1)$和$T(k,k+1)$与精确值之间的绝对误差均将随着舍入误差的累积而不断增大，从而始终无法达到误差$\varepsilon \sim 10^{-16}$的精度要求．另一方面，通过对算法的分析容易知道，由于等距网格的节点加密过程需要计算被积函数在新增的2^k个节点处函数值之和，因此 Romberg 求积算法的计算时间将随着迭代次数的增加而呈现几何增长．举例来说，从程序开始运行至迭代 32 次止，总共花费了约 1h 的时间，其中上述的求和过程占用了全部时间的 93.6%，并且其结果误差已经不小于10^{-11}量级．

五、结论

在本次上机实验中，通过比较分析两种数值积分算法的结果，我们对复化 Gauss-Legendre 方法与 Romberg 求积算法有了更感性的认识．相比之下，在同样计算机精度的限制下，前者由于代数精度更高，其能达到的计算精度略高一些；而后者在精度要求不高的情况下，其运算效率更为优越．具体来说，由于 Romberg 求积算法的计算用时是随减半加密次数呈几何增长的，因此从可操作性的层面来说，该算法几乎在精度要求不高的时候具有较大的应用价值

从本次实验中，我们可以获得的一个重要经验是，由于舍入误差与计算效率等重要因素的制约，试图通过一味增加区间分段数或求积节点个数的方法来提高精度并非可取的选择．