

实验报告 2

姓名：徐煜森 学号： PB16110173

1. 算法分析

本次实验中分别使用复化梯形积分和复化 Simpson 积分计算积分 $\int_1^6 \sin(x) dx$ ，取等距节点 $\{x_i, i = 0, \dots, N\}$ ，其中 N 为 $\{2^k, k = 0, 1, \dots, 12\}$ ，对于 Simpson 积分 N 为 $\{2^k, k = 1, \dots, 12\}$ 。

为了程序的通用性，使用函数指针传递需要积分的函数、积分区间和节点数。

程序主体分为两个循环，循环变量为 k ，将参数分别传入计算复化梯形积分和复化 Simpson 积分的函数，返回值为计算结果。

其中复化梯形积分代码：

```
double Trapezoid(double(*f)(double), double low, double high, int n) {
    double result = 0.0;
    double h = (high - low) / n;
    for (int i = 1; i < n; i++) {
        result += (*f)(low + i * h);
    }
    return h * (result + (*f)(low) / 2.0 + (*f)(high) / 2.0);
}
```

复化 Simpson 积分代码：

```
double Simpson(double(*f)(double), double low, double high, int n) {
    double result = 0.0;
    double h = (high - low) / n;
    int m = n / 2;
    result += 4 * (*f)(low + h);
    for (int i = 1; i < m; i++) {
        result += 4 * (*f)(low + (2 * i + 1) * h) + 2 * (*f)(low + 2 * i * h);
    }
    return h / 3.0 * ((*f)(low) + result + (*f)(high));
}
```

计算误差与误差阶：

```
void cal(double(*f)(double), int low, int high) {
    double true_ans = cos(low) - cos(high);
    printf("true answer: %7lf\n", true_ans);

    int n;
    double d = 0.0;
    double e = 0., e2 = 0.;
    double res = 0.;
    cout << "Trapezoid" << endl;
    for (int k = 0; k <= N; k++) {
        printf("k = %d\t\t", k);
        res = Trapezoid(f, low, high, pow(2, k));
        e2 = true_ans - res;
        printf("e = %.12e\t\t", e2);
        if (k > 0) {
            d = -log(e2 / e) / log(2);
            printf("d = %.6lf", d);
        }
        e = e2;
        cout << endl;
    }
}
```

```
    cout << "Simpson" << endl;
    for (int k = 1; k <= N; k++) {
        printf("k = %d\t\t", k);
        res = Simpson(f, low, high, pow(2, k));
        e2 = true_ans - res;
        printf("e = %.12e\t\t", e2);
        if (k > 1) {
            d = -log(e2 / e) / log(2);
            printf("d = %.6lf", d);
        }
        e = e2;
        cout << endl;
    }
}
```

2. 计算结果

复化梯形积分

使用复化梯形积分计算得的误差和误差阶如下表所示：

表 1 复化梯形积分

K	误差	误差阶
0	-1.825006697305e+00	
1	-2.454792698194e-01	2.894229
2	-5.614913519149e-02	2.128265
3	-1.375739486821e-02	2.029056
4	-3.422468688994e-03	2.007098
5	-8.545713803539e-04	2.001764
6	-2.135776256161e-04	2.000440
7	-5.339033240903e-05	2.000110
8	-1.334732851249e-05	2.000028
9	-3.336816218225e-06	2.000007
10	-8.342030577979e-07	2.000002
11	-2.085507060379e-07	2.000000
12	-5.213767340084e-08	2.000000

复化 Simpson 积分

表 2 复化 Simpson 积分

K	误差	误差阶
1	2.810298726757e-01	
2	6.960909684481e-03	5.335304
3	3.731852395463e-04	4.221312

4	2.250670407800e-05	4.051465
5	1.394389193332e-06	4.012650
6	8.695929759606e-08	4.003149
7	5.431993799210e-09	4.000787
8	3.394540759594e-10	4.000194
9	2.121447462144e-11	4.000096
10	1.327493670544e-12	3.998272
11	8.237854842719e-14	4.010292
12	8.770761894539e-15	3.231495

3. 结果分析与对比

3.1 容易看出,随着 K 不断增大,两种积分算法的误差都在逐渐减小,同时误差阶也逐渐趋于稳定。查阅资料知道,复化梯形积分的误差阶理论值是 2,复化 Simpson 积分的误差阶理论值是 4。从实验结果可以看出, K 较大时复化梯形积分的误差阶几乎等于理论值,而复化 Simpson 积分的误差阶则在 $K=12$ 时出现了抖动。与同学讨论后得到的结论是,在计算复化 Simpson 积分时

$$\frac{h}{3} [f(a) + 4 \sum_{i=0}^{m-1} f(x_{2i+1}) + 2 \sum_{i=1}^{m-1} f(x_{2i}) + f(b)]$$

如果将 $\frac{h}{3}$ 乘入中括号内计算,则结果会偏大,如果在中括号内算式计算结束后再乘 $\frac{h}{3}$,计算结果会偏小。推测原因可能是 $K=12$

时积分结果与真实值十分接近，在计算误差时相近两数相减，产生较大误差。

3.2 通过比较 k 值相同时两种积分公式的误差，可以发现复化 Simpson 积分的误差小于复化梯形积分的误差，复化 Simpson 积分的误差阶为 4，而复化梯形积分的误差阶稳定于 2。因为误差阶理论上应等于代数精度+1，因此可以看出，复化 Simpson 积分的代数精度高于复化梯形积分。

3.3 综合上述两点，复化 Simpson 积分优于复化梯形积分，因计算误差小且代数精度高。

实验小结

本次实验中使用复化 Simpson 积分和复化梯形积分计算 $\int_1^6 \sin x dx$ ，通过对比可以看出，复化 Simpson 积分的误差远小于复化梯形公式，且其误差阶为 4，大于复化梯形积分误差阶 2，说明复化 Simpson 积分的优于复化梯形公式。在实际应用中，可以选取复化 Simpson 积分来计算无法表示或难以积分的函数的数值积分。