实验三 数值积分(2 课时)

一、实验目的

1. 了解数值积分的基本原理和方法。
2. 掌握复合梯形公式。
3. 了解求积公式外推思想、Romberg 公式及 Romberg 积分法。

二、实验要求

1. 编写定步长复合梯形公式
2. 编写变步长复合梯形公式。
3. 进一步加深对数值积分的理解。

三、实验原理

(一)定步长复合梯形公式

1. 公式

将积分区间$[a,b]$ n等分，分点为$x_i = a + ih$，$i = 0,1,\cdots,n$，其中$h = \frac{b-a}{n}$称为积分步长。

$$\int_a^b f(x)dx = T_n = \frac{h}{2}\left[f(a) + 2\sum_{k=1}^{n-1} f(x_k) + f(b)\right]$$

2. 例子

用复合梯形公式求积分$\pi = \int_0^1 \frac{4}{1+x^2}dx$的近似值。(取 8 位小数，精确解为 3.14159265)

$$\pi \approx T_8 = \frac{1}{16}\{f(0) + 2[f(\frac{1}{8}) + f(\frac{1}{4}) + f(\frac{3}{8}) + f(\frac{1}{2}) + f(\frac{5}{8}) + f(\frac{3}{4}) + f(\frac{7}{8})] + f(1)\} = 3.138988$$

(二) 变步长复合梯形公式

1. 公式

$$T_{2n}=\frac{1}{2}T_n+\frac{b-a}{2n}\sum_{k=1}^{n}f\left[a+(2k-1)\frac{b-a}{2n}\right]$$

递推公式：

$$\begin{cases} T_1=\dfrac{b-a}{2}\left[f(a)+f(b)\right] \\ T_{2^k}=\dfrac{1}{2}T_{2^{k-1}}+\dfrac{b-a}{2^k}\displaystyle\sum_{i=1}^{2^{k-1}}f\left[a+(2i-1)\dfrac{b-a}{2^k}\right] \end{cases} \qquad (k=1,2,3,\cdots)$$

2. 例子

用递推公式求积分 $\pi=\int_0^1\frac{4}{1+x^2}dx$ 的近似值，使误差不超过 10^{-6} 。

$$T_1=\frac{1}{2}\left[f(0)+f(1)\right]=\frac{1}{2}(4+2)=3$$

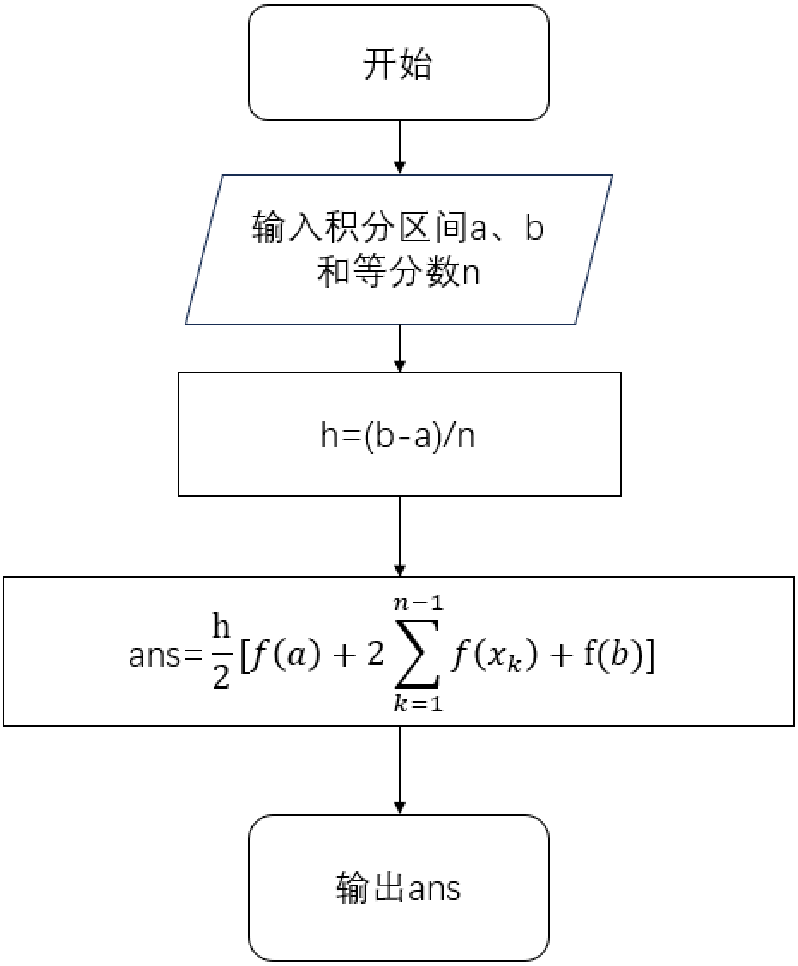
$$T_2=\frac{1}{2}T_1+\frac{1}{2}f(\frac{1}{2})=3.1$$

$$T_4=\frac{1}{2}T_2+\frac{1}{4}\left[f(\frac{1}{4})+f(\frac{3}{4})\right]=3.13117647$$

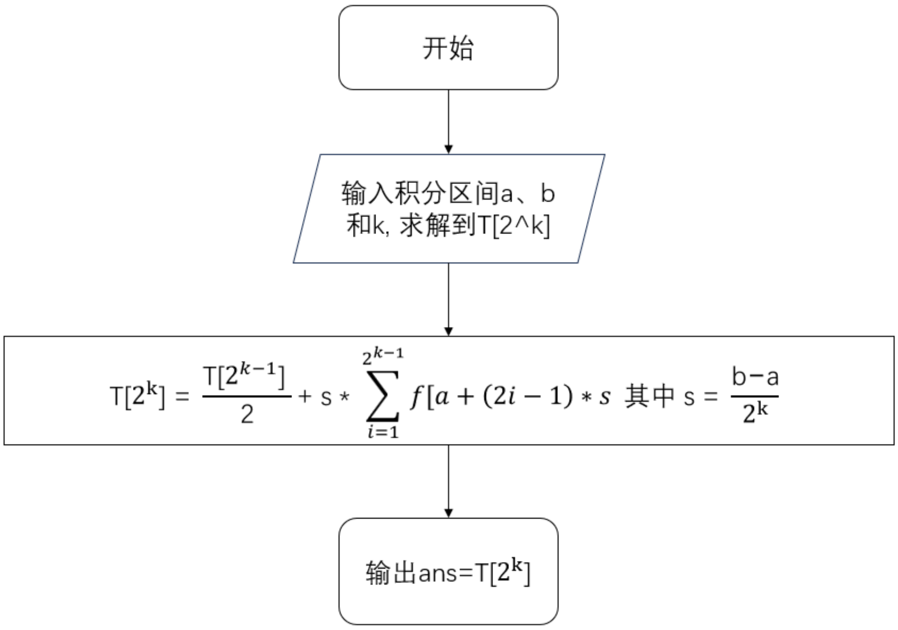
四、实验内容

（一）算法流程图

1. 定步长复合梯形算法流程图



2. 变步长复合梯形算法流程图



（二）编程作业

求 $\Pi = \int_0^1 \frac{4}{1 + x^2}$ 的近似值。

(1)编写定步长复合梯形程序求解上式;

(2)编写变步长复合梯形程序求解上式，使误差不超过 10-6.

【提示】请根据前面的算法流程图进行编写程序。

1.定步长复合梯形程序求解

```cpp
#include <bits/stdc++.h>

using namespace std;

double f(double x) {
    return 4 / (1 + x * x);
}

int main() {
    cout << "输入积分区间:";
    double a, b;
    int n;
    cin >> a >> b;
    cout << "输入等分数:";
    cin >> n;
    double h = (b - a) / n;
    double ans = 0;
    ans += f(a) + f(b);
    for (int k = 1; k <= n - 1; k++) {
        ans += 2 * f(a + k * h);
    }
    ans *= h / 2;
    cout << "积分结果为:" << ans << endl;
    double pi = 3.14159265;
    cout << "与精确值(" << pi << ")的误差为:" << ans - pi << endl;
}
```

```
输入积分区间:0 1
输入等分数:100
积分结果为:3.14158
与精确值(3.14159)的误差为:-1.66631e-05

进程已结束，退出代码为 0
```

2. 变步长复合梯形程序求解

```
#include <bits/stdc++.h>

using namespace
        std;

double f(double x) {
    return 4 / (1 + x * x);
}

double getTk1(int k, double a, double b) {
    double T[(1 << k) + 1];
    T[1] = (f(a) + f(b)) / 2;
    for (int i = 2; i <= (1 << k); i <<= 1) {
        // T[2^k] = T[2^(k-1)]/2 + s * sum{ f[a+(2i-1)*s] |
i=1->2^(k-1) } 其中 s = (b-a)/2^k
        T[i] = T[i / 2] / 2;
        double s = (b - a) / i;
        double t = 0;// t = sum{ f[a+(2i-1)*s] | i=1->2^(k-1) }
        for (int j = 1; j <= i / 2; j++) {
            double x = a + (2 * j - 1) * s;
            t += f(x);
        }
        T[i] += s * t;
    }
    return T[1 << k];
}

double getTk2(int k, double a, double b) {
    // 压缩 T 数组, 2^k -> k
    double T[k + 1];// T[k] <--> T[2^k]
    T[0] = (f(a) + f(b)) / 2;
    for (int i = 1; i <= k; i++) {
        // T[2^k] = T[2^(k-1)]/2 + s * sum{ f[a+(2i-1)*s] |
i=1->2^(k-1) } 其中 s = (b-a)/2^k
        // T[i] = T[i-1]/2 + s * sum{ f[a+(2j-1)*s] | j=1->2^(i-1) }
其中 s = (b-a)/2^i
        T[i] = T[i - 1] / 2;
        double s = (b - a) / (1 << i);
        double t = 0;// t = sum{ f[a+(2i-1)*s] | i=1->2^(k-1) }
        for (int j = 1; j <= (1 << (i - 1)); j++) {
            double x = a + (2 * j - 1) * s;
            t += f(x);
        }
```

```cpp
            T[i] += s * t;
        }
    return T[k];
}

double getTk3(int k, double a, double b) {
    // T 的后一项只依靠前一项,所以仅使用一个变量存储即可
    double T = (f(a) + f(b)) / 2;
    for (int i = 1; i <= k; i++) {
        // T[2^k] = T[2^(k-1)]/2 + s * sum{ f[a+(2i-1)*s] | i=1->2^(k-1) } 其中 s = (b-a)/2^k
        // T[i] = T[i-1]/2 + s * sum{ f[a+(2j-1)*s] | j=1->2^(i-1) } 其中 s = (b-a)/2^i
        T /= 2;
        double s = (b - a) / (1 << i);
        double t = 0;// t = sum{ f[a+(2i-1)*s] | i=1->2^(k-1) }
        for (int j = 1; j <= (1 << (i - 1)); j++) {
            double x = a + (2 * j - 1) * s;
            t += f(x);
        }
        T += s * t;
    }
    return T;
}

int main() {
    cout << "输入积分区间:";
    double a, b;
    int k;
    cin >> a >> b;
    cout << "输入 k(求到 T[2^k]):";
    cin >> k;

    double ans = getTk3(k, a, b);
    cout << "积分结果为:" << ans << endl;
    double pi = 3.14159265;
    cout << "与精确值(" << pi << ")的误差为:" << ans - pi << endl;
}
```

```
输入积分区间:0 1
输入k(求到T[2^k]):10
积分结果为:3.14159
与精确值(3.14159)的误差为:-1.55356e-07

进程已结束，退出代码为 0
```