**Runge-kutta法上机报告**

**计试81 白思雨 2186123935**

**⼀、算法原理**

利用泰勒展开可以导出龙格-库塔法。级龙格-库塔法的一般形式为：



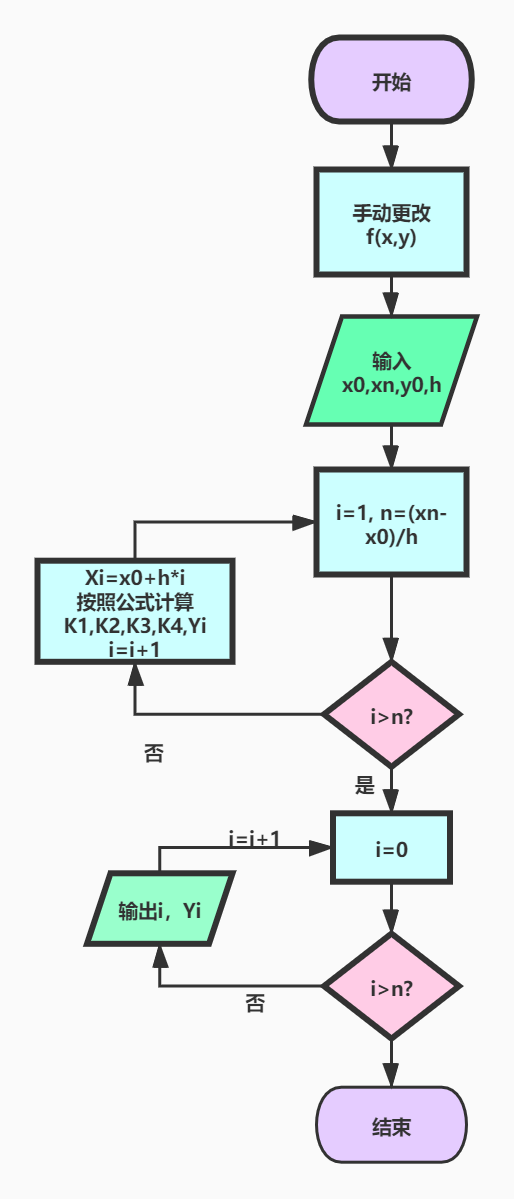
其中均为常数，由待定系数法确定，确定的原则则是将局部截断误差在处泰勒展开，适当选取的系数，使得局部截断误差的阶尽可能⾼。



经典（标准）级阶法：



**二、程序框图**



**三、程序及使用说明(cpp)**

**#include<iostream>**

**#include<cmath>**

**using namespace std;**

**double func1(double x, double y)**

**{**

**return (-0.9 \* y / (1 + 2 \* x)); //这是被积函数，若计算其他函数积分则在此更换函数**

**}**

**double Y[1000],X[1000];**

**double x0, xn, yo, h;**

**int main()**

**{**

**//输入基本条件，x0,xn,yo,h**

**cout << "Please input x0, xn, yo and h:";**

**cin >> x0 >> xn >> yo >> h;**

**//计算分点个数，n为分点的个数-1；**

**int n = (xn - x0) / h;**

**Y[0] = yo;**

**X[0] = x0;**

**//计算yi(i=0,1...n)，并将其储存到Y[1000]数组中**

**for(int i = 1; i <= n; ++i)**

**{**

**X[i] = x0 + h \* i;**

**double K1 = h \* func1(X[i - 1], Y[i - 1]);**

**double K2 = h \* func1(X[i - 1] + 0.5 \* h, Y[i - 1] + 0.5 \* K1);**

**double K3 = h \* func1(X[i - 1] + 0.5 \* h, Y[i - 1] + 0.5 \* K2);**

**double K4 = h \* func1(X[i - 1] + h, Y[i - 1] + K3);**

**Y[i] = Y[i - 1] + (K1 + 2 \* K2 + 2 \* K3 + K4) / 6;**

**}**

**//输出yi(i = 0,1...n)**

**for(int i = 0; i <= n; ++i)**

**{**

**cout << "y" << i << ':' << Y[i] << endl;**

**}**

**return 0;**

**}**

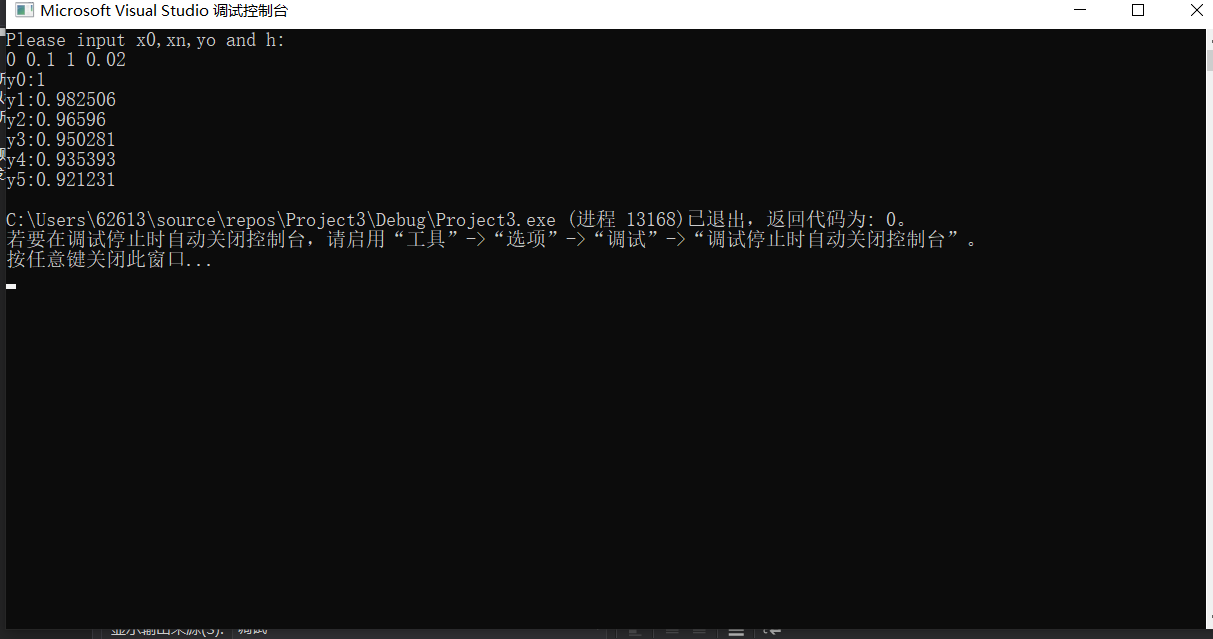
（说明：相关程序说明已在代码中注释写出，对于不同的函数积分需要更改代码中 func1.）

**四、算例及计算结果**

算例：例题9.1.1:



计算结果：



与原题结果相符，实验正确。