龙格-库塔方法与亚当姆斯方法相结合求解偏微分方程

姓名：岳宇轩 学号：19020011038

专业：19慧与 序号：14 指导教师：高云

实验目的

1.实现龙格-库塔方法，并验证它的正确性；

2.先用龙格-库塔方法计算前3点，然后实现亚当姆斯预报-校正系统，并验证算法的正确性。

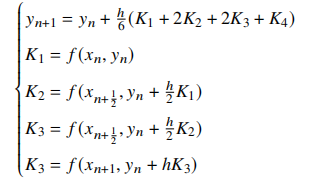
实验步骤

1.实现龙格-库塔方法，并用它启动

2.验证亚当姆斯预报-校正系统

实验原理

1. 四阶经典龙格-库塔方法：



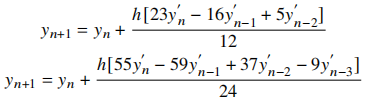
在 [xn,,xn+1] 区间上多预报几个点的斜率，然后将他们加权平均作为平均斜率，则可以构造出更高精度的公式，这就是龙格库塔方法的设计思想。

1. 亚当姆斯方法

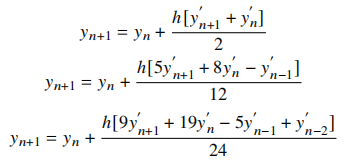
亚当姆斯方法的设计思想是充分利用计算yn+1之前已得到一系列结点xn.xn-1....上的斜率值来减少计算量。譬如，可以用xn,xn-1两点的斜率的加权平均作为区间[xn,xn+1]上的平均斜率，于是可以设计出如下二阶亚当姆斯格式:



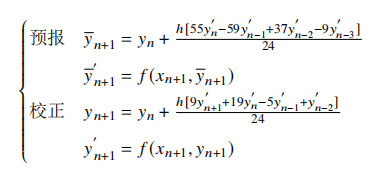
同理可得三、四阶亚当姆斯格式：



同样，也可导出如下隐式的二阶、三阶和四阶亚当姆斯格式：



将显式和隐式两种亚当姆斯格式相匹配，可构成下列亚当姆斯预报-校正系统



实验过程

①实现龙格库塔方法，验证正确性

选择微分方程  ，此方程有解析解 。

实现微分方程的代码如下：

double f(double x, double y) {

return y - 2 \* x / y;

}

参数x,y即为微分方程中的x,y

根据实验原理中四阶经典龙格-库塔公式，编写龙格-库塔方法函数如下：

double Runge\_Kutta(double x0, double x1, double y0) {

double K1, K2, K3, K4;

double h = x1 - x0;

K1 = f(x0, y0);

K2 = f(x0 + h / 2.0, y0 + h / 2.0 \* K1);

K3 = f(x0 + h / 2.0, y0 + h / 2.0 \* K2);

K4 = f(x1, y0 + h \* K3);

return y0 + h / 6.0 \* (K1 + 2 \* K2 + 2 \* K3 + K4);

}

参数x0,x1,y0即为公式中的xn,xn+1,yn

②先用龙格-库塔方法计算前3点，然后实现亚当姆斯预报-校正系统，并验证算法的正确性。

实现代码如下：

//x存储x0-xn，y存储预报值，real存储校正值，y\_dao存储预报导数值，real\_dao存储校正导数值，true\_value存储真值，loss存储校正值与真值的误差

double x[100], y[100], real[100], y\_dao[100], real\_dao[100], true\_value[100], loss[100];

void Adams(double a, double b, double y0, double h) {

//初始化

x[0] = a;

y[0] = y0;

real[0] = y0;

y\_dao[0] = 0;

real\_dao[0] = 0;

true\_value[0] = y0;

loss[0] = 0;

int n = (int)((b - a) / h); //定义n

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

{

x[i] = a + i \* h; //计算x

true\_value[i] = sqrt(1 + 2 \* x[i]); //通过解析解计算真值

if (i < 4)

{

//对于前三个结点，用龙格-库塔公式启动

y[i] = Runge\_Kutta(x[i - 1], x[i], real[i - 1]);

real[i] = y[i];

y\_dao[i] = f(x[i - 1], y[i - 1]);

real\_dao[i] = y\_dao[i];

}

else

{

//预报

y[i] = real[i - 1] + h / 24.0 \* (55 \* real\_dao[i - 1] - 59 \* real\_dao[i - 2] + 37 \* real\_dao[i - 3] - 9 \* real\_dao[i - 4]);

y\_dao[i] = f(x[i], y[i]);

//校正

real[i] = real[i - 1] + h / 24.0 \* (9 \* y\_dao[i] + 19 \* real\_dao[i - 1] - 5 \* real\_dao[i - 2] + real\_dao[i - 3]);

real\_dao[i] = f(x[i], real[i]);

}

loss[i] = abs(true\_value[i] - real[i]); //计算误差值

}

}

③主函数如下:

int main() {

double a = 0, b = 1.0, y0 = 1.0, h = 0.1;

Adams(a, b, y0, h);

cout << "Xn\t\t 预报\t\t 校正\t\t 真值\t\t 误差" << endl << endl;

int n = (int)((b - a) / h);

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

{

cout << fixed << setprecision(1) << x[i] << "\t\t";

if (i < 4)

cout << " \t\t";

else

cout << fixed << setprecision(8) << y[i] << "\t\t";

cout << fixed << setprecision(8) << real[i] << "\t\t" << true\_value[i] << "\t\t" << loss[i] << endl;

}

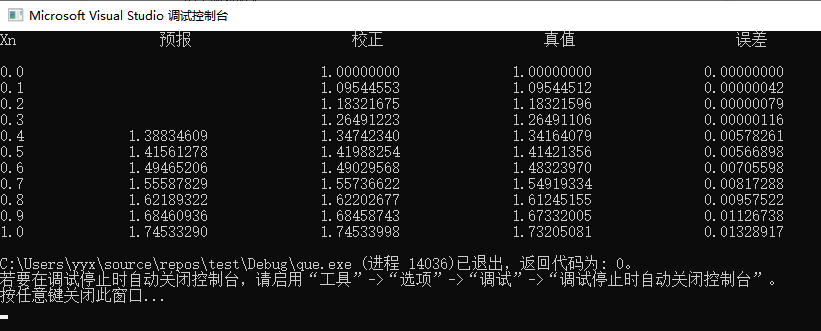
return 0;

}

这里设置区间为[0,1]，步长0.1

实验结果及分析

实验输出如图



分析：

①对于x0, x1, x2, x3来说，没有预报值，使用龙格-库塔启动得到的结果作为校正值进行存储。可以看到龙格-库塔公式预测结果的误差分别为0.00000042,0.00000079,0.00000116.误差较小，从而完成了实验目的1：实现龙格-库塔方法，并验证它的正确性；

②对于使用亚当姆斯预报-校正系统进行预测的结点x4-x10,误差分别为 0.00578261，0.00566898，0.00705598，0.00817288，0.00957522， 0.01126738，0.01328917。可以看到误差较小，从而完成了实验目的二：先用龙格-库塔方法计算前3点，然后实现亚当姆斯预报-校正系统，并验证算法的正确性。

实验体会：

在这次的实验过程中，利用c++编写了龙格-库塔算法和亚当姆斯算法的有关代码，增加了编程能力，也进一步理解了这两种常微分方程的计算方法。

龙格-库塔方法是显式的自开始方法，而且精度比较高，具有易于改变步长等优点。但使用龙格-库塔方法时，每一步需要多次计算函数f(x,y)的值，计算量大。而且由于龙格-库塔方法的导出基于泰勒展开，所以要求函数具有较高的光滑性。对于光滑性不太好的解，最好采用低阶算法而将步长h取小。

亚当姆斯方法的计算量比龙格-库塔方法少，却具有同样的精度，但必须用龙格库塔方法提供开头几个函数值。