Лабораторная работа №7

Тужилкина Н.Г., БПМ-18-2, вариант 23.

# Численное решение дифференциального уравнения методом Рунге-Кутта

Дано дифференциальное уравнение первого порядка

Представим его в виде :

Общее решение данного уравнения:

## Цель работы:

Решить заданное уравнение методом Рунге-Кутта с точностью 0.01.

## Метод Рунге-Кутта

### Изложение метода

Необходимо найти интервал длины 1 на оси абсцисс, и параметр , так чтобы для решение продолжалось бы до или до точки , максимально удалённой от , хотя и меньшей .

Положим интервал равным , а переменную интегрирования .

Начальное условие :

Таким образом, имеем ДУ первого порядка

с начальным условием

Так как а , шаг .

Пусть

Согласно методу Рунге-Кутта последовательные значения искомой функции определяются по формуле

где

Найденные на шаге оценки , сравниваются со значениями аналитического решения , подсчитанного в точках .

Если максимальная разница между оценками и известными точными значениями окажется больше, чем заданная точность , надо будет уменьшить шаг *.*

### Текст программы

#include <iostream>

#include <cmath>

#include <vector>

using namespace std;

double y\_orig(const double& x, const double& y) {

return x \* y \* y - log(x \* x) + log(y);

}

double secant(const double& x, double x1, double x2, const double& eps) {

double x3(0);

while (abs(x2 - x1) > eps) {

x3 = x2 - y\_orig(x, x2) \* (x2 - x1) / (y\_orig(x, x2) - y\_orig(x, x1));

x1 = x2;

x2 = x3;

}

return x3;

}

double f(const double& x, const double& y) {

return (2 \* y - x \* y \* y \* y) / (x + 2 \* x \* x \* y \* y);

}

void RK(const double& a, const double& b, const double& h, const double& eps) {

vector<double> x, y, k1, k2, k3, k4;

double n((b - a) / h);

x.push\_back(a);

y.push\_back(secant(a, a, b, 0.1 \* eps));

printf("\n n = %2d, x = %.1f: ", 0, x.back());

printf("|%.3f - %.3f| = %.3f\n", y.back(), y.back(), abs(y.back() - y.back()));

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

x.push\_back(x[0] + i \* h);

k1.push\_back(h \* f(x.back(), y.back()));

k2.push\_back(h \* f(x.back() + h / 2, y.back() + h \* k1.back() / 2));

k3.push\_back(h \* f(x.back() + h / 2, y.back() + h \* k2.back() / 2));

k4.push\_back(h \* f(x.back() + h, y.back() + k3.back()));

y.push\_back(y.back() + (k1.back() + 2 \* k2.back() + 2 \* k3.back() + k4.back()) / 6);

double y0(secant(x.back(), a, b, 0.1 \* eps));

printf(" n = %2d, x = %.1f: ", i, x.back());

printf("|%.3f - %.3f| = %.3f\n", y.back(), y0, abs(y.back() - y0));

}

}

int main() {

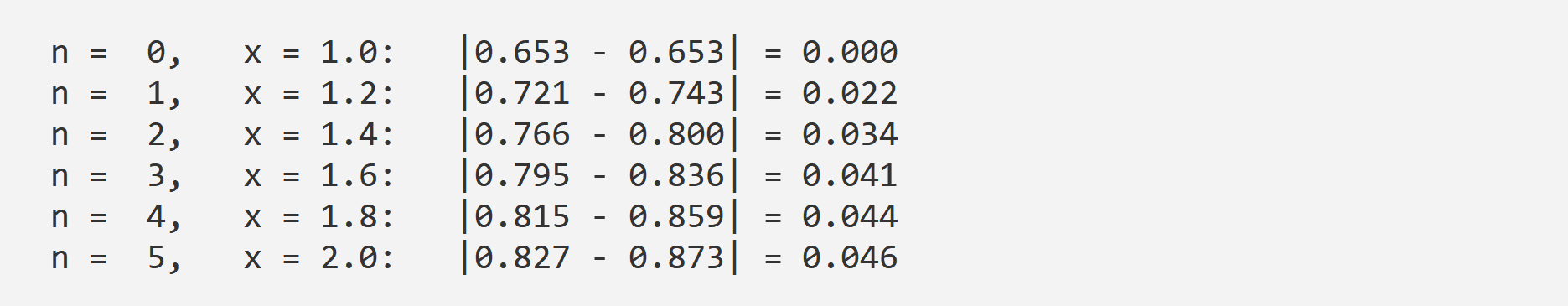
double a(1), b(2), h(0.05);

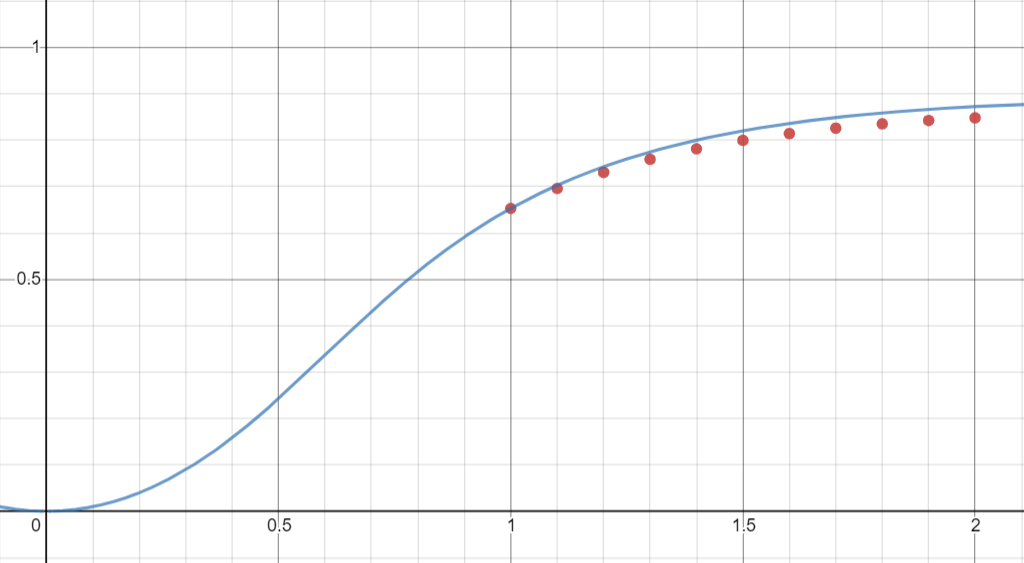
RK(a, b, h, 0.01);

### }

### Представление результатов

Результат при :





Уменьшим шаг до :

