

Министерство образования и науки Российской Федерации  
Санкт-Петербургский политехнический университет Петра  
Великого

—  
Институт компьютерных наук и технологий  
Высшая школа программной инженерии

**Лабораторная работа №3**  
**по дисциплине «Вычислительная математика»**

Выполнил  
Студент группы 5130904/20004

Шелковников Д.С.

Преподаватель

Устинов С.М.

# Оглавление

Задание .....	2
Результаты.....	3
Выводы .....	5
Код программы.....	6
<DIR>/computational_mathematics/third_lab/Rkf45.cpp .....	6
<DIR>/computational_mathematics/third_lab/Rkf45.h .....	6
<DIR>/computational_mathematics/third_lab/main.cpp .....	7

## Задание

**ВАРИАНТ N 11**

Решить систему дифференциальных уравнений:

$$\frac{dx_1}{dt} = -71x_1 - 70x_2 + e^{1-t^2}; \quad \frac{dx_2}{dt} = x_1 + \sin(1-t);$$
$$x_1(0) = 0, \quad x_2(0) = 1; \quad t \in [0, 4]$$

следующими способами с одним и тем же шагом печати  $h_{print} = 0.2$  :

I) по программе **RKF45** с  $EPS=0.0001$ ;

II) методом семейства Рунге-Кутты

$$z_{n+1} = z_n + (23k_1 + 125k_3 - 81k_5 + 125k_6)/192; \quad k_1 = hf(t_n, z_n);$$
$$k_2 = hf(t_n + h/3, z_n + k_1/3); \quad k_3 = hf(t_n + 0.4h, z_n + 0.16k_1 + 0.24k_2);$$
$$k_4 = hf(t_n + h, z_n + 0.25k_1 - 3k_2 + 3.75k_3);$$
$$k_5 = hf(t_n + 2h/3, z_n + (6k_1 + 90k_2 - 50k_3 + 8k_4)/81);$$
$$k_6 = hf(t_n + 4h/5, z_n + (6k_1 + 36k_2 + 10k_3 + 8k_4)/75);$$

с двумя постоянными шагами интегрирования:

а)  $h_{int} = 0.1$

б) любой другой, позволяющий получить качественно верное решение.

Сравнить результаты.

# Результаты

RKF45 with eps = 0.0001

0 0 1

0.2 -0.932477 0.980375

0.4 -0.885233 0.926926

0.6 -0.818626 0.85193

0.8 -0.731589 0.755573

1 -0.624716 0.639564

1.2 -0.500028 0.506894

1.4 -0.36125 0.361564

1.6 -0.213045 0.208296

1.8 -0.0607653 0.0522638

2 0.0899907 -0.101146

2.2 0.233651 -0.246562

2.4 0.364937 -0.378846

2.6 0.479056 -0.493291

2.8 0.571876 -0.585815

3 0.640065 -0.653132

3.2 0.681226 -0.692892

3.4 0.693987 -0.703784

3.6 0.678062 -0.6856

3.8 0.634201 -0.639247

4 0.564488 -0.566728

Runge Kutta 6 with step 0.1

	0	0	1
0.2	5959.65	-84.1707	
0.4	3.69023e+07	-527175	
0.6	2.28465e+11	-3.26378e+09	
0.8	1.41444e+15	-2.02063e+13	
1	8.7569e+18	-1.25099e+17	
1.2	5.42146e+22	-7.74494e+20	
1.4	3.35646e+26	-4.79494e+24	
1.6	2.07801e+30	-2.96858e+28	
1.8	1.28651e+34	-1.83787e+32	
2	7.96486e+37	-1.13784e+36	
2.2	4.9311e+41	-7.04442e+39	
2.4	3.05287e+45	-4.36125e+43	
2.6	1.89005e+49	-2.70008e+47	
2.8	1.17015e+53	-1.67164e+51	
3	7.24445e+56	-1.03492e+55	
3.2	4.48509e+60	-6.40727e+58	
3.4	2.77675e+64	-3.96678e+62	
3.6	1.7191e+68	-2.45586e+66	
3.8	1.06431e+72	-1.52044e+70	
4	6.5892e+75	-9.41315e+73	

Result is  $x = 6.5892e+75$ ,  $y = -9.41315e+73$

Runge Kutta 6 with step 1e-05

	0	0	1
0.2	-0.932477	0.980375	
0.4	-0.885234	0.926927	
0.6	-0.818626	0.85193	
0.8	-0.731589	0.755573	
1	-0.624659	0.639563	
1.2	-0.500008	0.506894	
1.4	-0.361243	0.361564	
1.6	-0.213048	0.208296	
1.8	-0.0607649	0.0522638	
2	0.0899913	-0.101146	
2.2	0.233652	-0.246562	
2.4	0.364937	-0.378846	
2.6	0.479057	-0.493291	
2.8	0.571876	-0.585815	
3	0.640065	-0.653132	
3.2	0.681226	-0.692892	
3.4	0.693987	-0.703784	
3.6	0.678062	-0.6856	
3.8	0.634271	-0.639248	
4	0.564511	-0.566728	

Result is  $x = 0.564511$ ,  $y = -0.566728$

## Выводы

Решения двух систем максимально схожи для rkf45 и Рунге-Кутты 6 порядка с шагом 1e-05. Можно говорить о трех знаках точности после запятой. Уменьшение шага в Рунге-Кутты ведет к увеличению времени вычисления, в то время, как 1e-05 вычисляется быстро и достаточно точно. Шаг 0.1 дает экспонентный рост результата из-за неустойчивости системы, накопления численных ошибок при вычислении и коэффициентов при вычислении параметров метода. Есть два варианта решения проблема: другой шаг или адаптивная формула Рунге-Кутты 6 порядка.

# Код программы

```
<DIR>/computational_mathematics/third_lab/Rkf45.cpp
#include "Rkf45.h"
#include "rkf.h"
#include <cmath>
#include <iostream>

void dimkashelk::Rkf45::calculate(int (*F)(int n, double t, double y[],
double yp[]),
                                double valueArray[],
                                double t, double tout) {
    int flag = 0;
    int NEQN = 2;
    rkfinit(NEQN, &flag);
    double dop[2]{0.0, 0.0};
    flag = 1;
    double ABS = 0.0001, REL = 0.0001;
    double H[NEQN];
    double STEP = 0.2;
    int NFE = 0;
    int MAXNFE = 100000;
    while (t <= tout) {
        std::cout << "\t" << t << " " << valueArray[0] << " " <<
valueArray[1] << "\n";
        rkf45(F, NEQN, valueArray, dop, &t, t + STEP, &REL, ABS, H, &NFE,
MAXNFE, &flag);
    }
    std::cout << "\t" << t << " " << valueArray[0] << " " << valueArray[1] <<
"\n";
    rkfend();
}
```

```
<DIR>/computational_mathematics/third_lab/Rkf45.h
#ifndef RKF45_H
#define RKF45_H

namespace dimkashelk {
    class Rkf45 {
    public:
        static void calculate(int (*F)(int n, double t, double y[], double
yp[]),
                                double Y[],
                                double T,
                                double TOUT);
    };
}
```

```
#endif
```

```
<DIR>/computational_mathematics/third_lab/main.cpp
```

```
#include <iostream>
```

```
#include <iomanip>
```

```
#include <cmath>
```

```
#include <functional>
```

```
#include "Rkf45.h"
```

```
int func(int n, double t, double *value, double *res) {  
    res[0] = -71 * value[0] - 70 * value[1] + std::exp(1 - t * t);  
    res[1] = value[0] + std::sin(1 - t);  
    return 0;  
}
```

```
bool areEqualRel(float a, float b, float epsilon) {  
    return (fabs(a - b) <= epsilon * std::max(fabs(a), fabs(b)));  
}
```

```
using ODEFunction = std::function<double(double, double, double)>;
```

```
void rungeKutta6(ODEFunction f1, ODEFunction f2, double &y1, double &y2,  
double t0, double tEnd, double h) {  
    double k1 = 0.0, k2 = 0.0, k3 = 0.0, k4 = 0.0, k5 = 0.0, k6 = 0.0;  
    double l1 = 0.0, l2 = 0.0, l3 = 0.0, l4 = 0.0, l5 = 0.0, l6 = 0.0;
```

```
    for (double t = t0, s = t0; t < tEnd; t += h) {  
        if (areEqualRel(s, t, 0.000001)) {  
            std::cout << "\t" << t << "\t" << y1 << "\t" << y2 << "\n";  
            s += 0.2;  
        }  
        k1 = h * f1(t, y1, y2);  
        l1 = h * f2(t, y1, y2);  
  
        k2 = h * f1(t + h / 3, y1 + k1 / 3, y2 + l1 / 3);  
        l2 = h * f2(t + h / 3, y1 + k1 / 3, y2 + l1 / 3);  
  
        k3 = h * f1(t + 0.4 * h, y1 + 0.16 * k1 + 0.24 * k2, y2 + 0.16 * l1 +  
0.24 * l2);  
        l3 = h * f2(t + 0.4 * h, y1 + 0.16 * k1 + 0.24 * k2, y2 + 0.16 * l1 +  
0.24 * l2);  
  
        k4 = h * f1(t + h, y1 + 0.25 * k1 - 3 * k2 + 3.75 * k3, y2 + 0.25 *  
l1 - 3 * l2 + 3.75 * l3);  
        l4 = h * f2(t + h, y1 + 0.25 * k1 - 3 * k2 + 3.75 * k3, y2 + 0.25 *  
l1 - 3 * l2 + 3.75 * l3);  
  
        k5 = h * f1(t + 2 * h / 3, y1 + (6 * k1 + 90 * k2 - 50 * k3 + 8 * k4)  
/ 81,  
y2 + (6 * l1 + 90 * l2 - 50 * l3 + 8 * l4) / 81);
```

```

        15 = h * f2(t + 2 * h / 3, y1 + (6 * k1 + 90 * k2 - 50 * k3 + 8 * k4)
/ 81,
        y2 + (6 * l1 + 90 * l2 - 50 * l3 + 8 * l4) / 81);

        k6 = h * f1(t + 4 * h / 5, y1 + (6 * k1 + 36 * k2 + 10 * k3 + 8 * k4)
/ 75,
        y2 + (6 * l1 + 36 * l2 + 10 * l3 + 8 * l4) / 75);
        16 = h * f2(t + 4 * h / 5, y1 + (6 * k1 + 36 * k2 + 10 * k3 + 8 * k4)
/ 75,
        y2 + (6 * l1 + 36 * l2 + 10 * l3 + 8 * l4) / 75);

        y1 += (23 * k1 + 125 * k3 - 81 * k5 + 125 * k6) / 192;
        y2 += (23 * l1 + 125 * l3 - 81 * l5 + 125 * l6) / 192;
    }
    std::cout << "\t" << tEnd << " " << y1 << " " << y2 << "\n";
}

int main() {
    double data[2]{0.0, 1.0};
    std::cout << "RKF45 with eps = 0.0001\n";
    dimkashelk::Rkf45::calculate(func, data, 0, 4);
    std::cout << "\n\n\n\n";

    double y1 = 0.0, y2 = 1.0;
    double t0 = 0.0, tEnd = 4.0, h = 0.1;
    ODEFunction f1 = [](double t, double x, double y) {
        return -71 * x - 70 * y + exp(1 - t * t);
    };
    ODEFunction f2 = [](double t, double x, double y) {
        return x + sin(1 - t);
    };
    std::cout << "Runge Kutta 6 with step " << h << "\n";
    rungeKutta6(f1, f2, y1, y2, t0, tEnd, h);
    std::cout << "Result is x = " << y1 << ", y = " << y2 << "\n\n\n";

    y1 = 0.0, y2 = 1.0;
    h = 0.00001;
    std::cout << "Runge Kutta 6 with step " << h << "\n";
    rungeKutta6(f1, f2, y1, y2, t0, tEnd, h);
    std::cout << "Result is x = " << y1 << ", y = " << y2 << "\n\n\n";
    return 0;
}

```