

Отчёт по лабораторной работе №3

по дисциплине «Вычислительная математика»

Студент гр. 3530202/90002

Потапова А.М.

Преподаватель

Леонтьева Т.В.

2021г

Вариант №13

Постановка задачи:

Решить систему дифференциальных уравнений:

$$\frac{dx_1}{dt} = -40x_1 + 260x_2 + 1/(10t^2 + 1);$$

$$\frac{dx_2}{dt} = 30x_1 - 270x_2 + e^{-2t};$$

$$x_1(0) = 0, \quad x_2(0) = 1; \quad t \in [0, 0.4]$$

Следующими способами с одним и тем же шагом печати $h_{print} = 0.02$:

- 1) по программе **RKF45** с $EPS = 0.0001$;
- 2) методом Адамса 3-й степени точности

$$z_{n+1} = z_n + h(23f_n - 16f_{n-1} + 5f_{n-2})/12;$$

с двумя постоянными шагами интегрирования:

а) $h_{int} = 0.002$

б) любой другой, позволяющий получить качественно верное решение. Сравнить результаты. Дополнительные начальные условия для метода Адамса получить с помощью **RKF45**.

Ход работы

1. В первую очередь было получено решение по программе **RKF45**;
2. Далее методом Адамса 3-й степени точности были получены 2 набора решений. В первом случае постоянный шаг интегрирования — 0.002, как и требовалось в задании, а во втором — 0.00002 (рассчитанный шаг интегрирования).
3. После этого стало возможным сравнить полученные решения.

Результаты:

RKF45			ADAMS(0.002)			ADAMS(0.00002)		
x	y[0]	y[1]	x	y[0]	y[1]	x	y[0]	y[1]
0.020000	0.761431	0.093159	0.020000	0.793958	0.093728	0.020000	0.761897	0.093189
0.040000	0.656251	0.078445	0.040000	0.674514	0.079409	0.040000	0.656562	0.078471
0.060000	0.567499	0.068085	0.060000	0.576624	0.068205	0.060000	0.567701	0.068099
0.080000	0.493912	0.059478	0.080000	0.496565	0.059026	0.080000	0.494025	0.059492
0.100000	0.432642	0.052323	0.100000	0.430832	0.051474	0.100000	0.432710	0.052318
0.120000	0.381437	0.046314	0.120000	0.376612	0.045230	0.120000	0.381457	0.046310
0.140000	0.338422	0.041252	0.140000	0.331655	0.040040	0.140000	0.338403	0.041254
0.160000	0.302068	0.036988	0.160000	0.294160	0.035699	0.160000	0.302041	0.036975
0.180000	0.271199	0.033341	0.180000	0.262689	0.032045	0.180000	0.271153	0.033332
0.200000	0.244819	0.030213	0.200000	0.236099	0.028946	0.200000	0.244757	0.030211
0.220000	0.222113	0.027535	0.220000	0.213474	0.026301	0.220000	0.222059	0.027519
0.240000	0.202482	0.025194	0.240000	0.194086	0.024025	0.240000	0.202419	0.025184
0.260000	0.185391	0.023147	0.260000	0.177353	0.022052	0.260000	0.185321	0.023145
0.280000	0.170401	0.021367	0.280000	0.162809	0.020331	0.280000	0.170344	0.021352
0.300000	0.157210	0.019777	0.300000	0.150081	0.018818	0.300000	0.157150	0.019768
0.320000	0.145524	0.018362	0.320000	0.138868	0.017479	0.320000	0.145460	0.018360
0.340000	0.135096	0.017115	0.340000	0.128929	0.016287	0.340000	0.135048	0.017101
0.360000	0.125778	0.015978	0.360000	0.120065	0.015219	0.360000	0.125727	0.015970
0.380000	0.117400	0.014949	0.380000	0.112117	0.014257	0.380000	0.117344	0.014949
0.400000	0.109811	0.014035	0.400000	0.104952	0.013387	0.400000	0.109772	0.014023

Текст программы:

```
#include "cmath.h"

int f(n, x, y, dydx)
int n;
double x, y[], dydx[];
{
    dydx[0] = -40.0 * y[0] + 260.0 * y[1] + 1 / (10.0 * x * x + 1);
    dydx[1] = 30.0 * y[0] - 270.0 * y[1] + exp(-2 * x);
    return (0);
}

double adams1(double x, double y0, double y1)
{
    return -40.0 * y0 + 260.0 * y1 + 1 / (10 * x * x + 1);
}

double adams2(double x, double y0, double y1)
{
    return 30.0 * y0 - 270 * y1 + exp(-2 * x);
}

rkf()
{
    int f(), nfe, fail, step;
    int n = 2;
    int flag = 1;
    int maxfe = 5000;

    double h, x1, x2;
    double relerr = 1.0e-4;
    double abserr = 1.0e-4;
    double y[2], yp[2];

    rkfinit(n, &fail);

    if (fail == 0)
    {
        y[0] = 0.0;
        y[1] = 1.0;

        printf("\n          RKF45\n");
        printf("-----|-----|-----\n");
        printf("          x      y[0]      y[1]\n");
        printf("-----|-----|-----\n");

        for (step = 1; step <= 0.4 / 0.02; step++)
        {
            x2 = 0.02 * step;
            x1 = x2 - 0.02;

            rkf45(f, n, y, yp, &x1, x2, &relerr, abserr, &h, &nfe, maxfe, &flag);
            printf("%10.6f  |%10.6f  |%10.6f \n", x1, y[0], y[1]);

            if (flag != 2)
            {
                printf("%s\n", cmathmsg(RKF45_C, flag));
                break;
            }
        }

        rkfend();
    }

    return (0);
}

adams(double shag)
{
    int f(), nfe, fail, step;
    int n = 2;
    int flag = 1;
    int maxfe = 5000;

    double h, x1, x2;
    double relerr = 1.0e-4;
    double abserr = 1.0e-4;
    double y[2], yp[2];
```

```

rkfinit(n, &fail);

if (fail == 0)
{
    y[0] = 0.0;
    y[1] = 1.0;

    if (shag > 1.0e-3)
    {
        printf("\n          ADAMS(%.3f)\n", shag);
    }
    else
    {
        printf("\n          ADAMS(%.5f)\n", shag);
    }

    printf("-----|-----|-----\n");
    printf("          x          y[0]          y[1]\n");
    printf("-----|-----|-----\n");

    x2 = shag;
    x1 = x2 - shag;

    //дополнительные начальные условия для метода Адамса
    rkf45(f, n, y, yp, &x1, x2, &relerr, abserr,
        &h, &nfe, maxfe, &flag);

    rkf45(f, n, y, yp, &x1, x2, &relerr, abserr,
        &h, &nfe, maxfe, &flag);

    //метод Адамса
    x2 = shag * 4;
    for (step = 4; step <= (0.4 / shag + 1); ++step)
    {
        x2 = shag * step;
        x1 = x2 - shag;

        y[0] = y[0] + shag * (23 * adams1(x2, y[0], y[1]) - 16 * adams1(x2 - shag,
            y[0], y[1]) + 5 * adams1(x2 - 2 * shag, y[0], y[1])) / 12;
        y[1] = y[1] + shag*(23 * adams2(x2, y[0], y[1]) - 16 * adams2(x2 - shag, y[0],
            y[1]) + 5 * adams2(x2 - 2 * shag, y[0], y[1])) / 12;

        if (shag > 1.0e-3)
        {
            if (step % 10 - 1 == 0)
            {
                printf("%10.6f  |%10.6f  |%10.6f \n", x1, y[0], y[1]);
            }
        }
        else
        {
            if (step % 1000 - 1 == 0)
            {
                printf("%10.6f  |%10.6f  |%10.6f \n", x1, y[0], y[1]);
            }
        }
    }

    rkfend();
}

return (0);
}

main()
{
    rkf();
    adams(0.002);
    adams(0.00002);
    return (0);
}

```

Вывод:

Используя разные способы с разными шагами интегрирования получили следующие результаты: значения, полученные с помощью подпрограммы **RKF45** и **метода Адамса 3й степени** точности с шагом 0.002, менее точны, чем решения, полученные с помощью того же метода, но с шагом интегрирования 0.00002, что хорошо заметно на графиках. Таким образом, получаем, что для получения более точных результатов необходим достаточно малый шаг интегрирования.

