

Министерство образования и науки РФ  
Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологий  
Высшая школа программной инженерии

Отчёт по лабораторной работе №3  
по дисциплине «Вычислительная математика»

**Вариант №15**

Выполнил

студент гр. 3530904/00005  
Рябикин В.М.

Руководитель

Воскобойников С.П.

## **Оглавление**

Оглавление .....	2
Описание работы .....	3
1. Постановка задачи: .....	3
2. Текст программы .....	3
3. Результаты работы программы .....	3
4. Выводы по результатам .....	7

# Описание работы

## 1. Постановка задачи:

Решить систему дифференциальных уравнений:

$$\frac{dx_1}{dt} = -44x_1 - 160x_2 + \cos(t + 1);$$

$$\frac{dx_2}{dt} = x_1 + \arctg(1 + t^2);$$

$$x_1(0) = 2, x_2(0) = 0.5; t \in [0, 1.6]$$

следующими способами с одним и тем же шагом печати  $h_{print} = 0.08$ :

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

II) методом Адамса 4-й степени точности

$$Z_{n+1} = Z_n + h(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) / 24;$$

с двумя постоянными шагами интегрирования:

а)  $h_{int} = 0.008$ :

б) любой другой, позволяющий получить качественно верное решение.

Сравнить результаты. Дополнительные начальные условия для метода

Адамса получить с помощью RKF45.

## 2. Текст программы

```
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include "cmath.h"

# define n 2
double abserr = 0.0001;
double relerr = 0.0001;

int diffs(int ndf, double t, double *x, double *dx)
{
    dx[0] = -44 * x[0] - 160 * x[1] + cos(t + 1);
    dx[1] = x[0] + atan(1 + t * t);
    return (0);
}

double adamsdx0(double t, double x1, double x2)
{
    return -44 * x1 - 160 * x2 + cos(t + 1);
}

double adamsdx1(double t, double x1, double x2)
{
    return x1 + atan(1 + t * t);
}

void adams(double h_adams, double hprint)
{
    double x[n], xp[n], arr1[4], arr2[4], t, tout, h;
    double z1[2000], z2[2000];
```

```

int fail, nfe, flag = 1, maxnfe = 5000;

rkfinit(2, &fail);
if (fail)
{
    printf("RKF initialization failed.\n");
    exit(-1);
}

t = 0;
tout = h_adams;
x[0] = 2.0;
x[1] = 0.5;
z1[0] = x[0];
z2[0] = x[1];

for (int i = 1; i <= 3; i++)
{
    rkf45(diffs, n, x, xp, &t, tout, &relerr, abserr, &h, &nfe, maxnfe, &flag);
    z1[i] = x[0];
    z2[i] = x[1];
    t = tout;
    tout += h_adams;
}
rkfend();

printf("-----\n");
printf("|                                Adams, h(int) = %.3f\n", h_adams);
printf("-----\n");
printf("|          t          |          z1          |          z2\n");
printf("-----\n");
for (int step = 4; step <= (1.6 / h_adams); step++)
{
    tout = h_adams * step;
    for (int k = 0; k < 4; k++)
    {
        arr1[k] = adamsdx0(tout - h_adams * k, z1[step - (k + 1)], z2[step - (k + 1)]);
        arr2[k] = adamsdx1(tout - h_adams * k, z1[step - (k + 1)], z2[step - (k + 1)]);
    }
    z1[step] = z1[step - 1] + h_adams / 24 * (55 * arr1[0] - 59 * arr1[1] + 37 * arr1[2]
- 9 * arr1[3]);
    z2[step] = z2[step - 1] + h_adams / 24 * (55 * arr2[0] - 59 * arr2[1] + 37 * arr2[2]
- 9 * arr2[3]);
}

int m = hprint / h_adams;
for (int i = 1; i <= (1.6 / hprint); i++)
{
    t = hprint * i;
    printf("|%14.2f          |%18.6f          |%18.6f          |\n", t, z1[i * m], z2[i *
m]);
}
printf("-----\n\n");
}

void rkfer(double hprint, int maxnfe)
{
    int fail, nfe, flag = 1;
    double x[n], xp[n], h, t, tout;

```

```

x[0] = 2.0;
x[1] = 0.5;
rkfinit(2, &fail);
if (fail)
{
    printf("RKF initialization failed.\n");
    exit(-1);
}
printf("-----\n");
printf("|                                RKF45, MAXNFE = %d\n", maxnfe);
printf("-----\n");
printf("|          t          |          x1          |          x2\n");
printf("|          flag          |\n");
printf("-----\n");
for (int step = 1; step <= 1.6 / hprint; step++)
{
    tout = hprint * step;
    t = tout - hprint;
    rkf45(diffs, n, x, xp, &t, tout, &relerr, abserr, &h, &nfe, maxnfe, &flag);
    printf("|%14.2f          |%17.6f          |%17.6f          |%14d          |\n", t,
x[0], x[1], flag);
}
rkfend();
printf("-----\n\n");
}

int main(void)
{
    int maxnfe = 5000;
    double h_adams = 0.008, h2_adams = 0.001, hprint = 0.08;

    // RKF45 part
    rkfer(hprint, maxnfe);

    // Adams part
    adams(h_adams, hprint);
    adams(h2_adams, hprint);
    adams(hprint, hprint);

    return (0);
}

```

### 3. Результаты работы программы

RK45, MAXNFE = 1000			
t	x1	x2	flag
0.08	-1.740085	0.497596	2
0.16	-1.602897	0.424849	2
0.24	-1.389920	0.369927	2
0.32	-1.235086	0.331158	2
0.40	-1.128464	0.304604	2
0.48	-1.058152	0.287235	2
0.56	-1.015235	0.276781	2
0.64	-0.992878	0.271522	2
0.72	-0.985912	0.270142	2
0.80	-0.990328	0.271624	2
0.88	-1.003014	0.275174	2
0.96	-1.021537	0.280174	2
1.04	-1.043995	0.286141	2
1.12	-1.068906	0.292698	2
1.20	-1.095130	0.299560	2
1.28	-1.121801	0.306509	2
1.36	-1.148274	0.313387	2
1.44	-1.174083	0.320081	2
1.52	-1.198904	0.326512	2
1.60	-1.222523	0.332633	2

  

Adams, h(int) = 0.008		
t	z1	z2
0.08	-1.737102	0.497536
0.16	-1.601903	0.424890
0.24	-1.388838	0.370043
0.32	-1.233451	0.331352
0.40	-1.125751	0.304869
0.48	-1.053633	0.287550
0.56	-1.007653	0.277107
0.64	-0.980344	0.271800
0.72	-0.965496	0.270285
0.80	-0.957546	0.271504
0.88	-0.950977	0.274603
0.96	-0.939653	0.278872
1.04	-0.915960	0.283686
1.12	-0.869605	0.288453
1.20	-0.785849	0.292546
1.28	-0.642837	0.295228
1.36	-0.407536	0.295531
1.44	-0.029503	0.292094
1.52	0.568701	0.282914
1.60	1.506278	0.264968

  

Adams, h(int) = 0.001		
t	z1	z2
0.08	-1.740084	0.497598
0.16	-1.602926	0.424858
0.24	-1.389991	0.369946
0.32	-1.235208	0.331190
0.40	-1.128605	0.304649
0.48	-1.058366	0.287294
0.56	-1.015490	0.276851
0.64	-0.993180	0.271603
0.72	-0.986249	0.270232
0.80	-0.990694	0.271720
0.88	-1.003399	0.275275
0.96	-1.021934	0.280278
1.04	-1.044394	0.286244
1.12	-1.069302	0.292800
1.20	-1.095517	0.299659
1.28	-1.122175	0.306605
1.36	-1.148632	0.313479
1.44	-1.174423	0.320168
1.52	-1.199223	0.326595
1.60	-1.222822	0.332710

  

Adams, h(int) = 0.08		
t	z1	z2
0.08	-1.740085	0.497596
0.16	-1.602897	0.424849
0.24	-1.389920	0.369927
0.32	3.149460	0.223448
0.40	-28.781038	0.998600
0.48	208.548463	-4.948329
0.56	-1567.782597	39.450904
0.64	11712.402952	-292.556899
0.72	-87578.424133	2189.712425
0.80	654782.348956	-16369.304319
0.88	-4895571.005033	122389.532190
0.96	36602342.630303	-915058.303196
1.04	-273662019.058663	6841550.743503
1.12	2046068435.039660	-51151710.602219
1.20	-15297687550.986359	382442189.053434
1.28	114375081613.444946	-2859377040.050326
1.36	-855139657659.100952	21378491441.768143
1.44	6393559014569.871094	-159838975363.949341
1.52	-47802246693552.234375	1195056167339.107910
1.60	357399499049497.562500	-8934987476237.130859

#### 4. Выводы по результатам

По результатам, полученным в ходе работы программы, мы видим, что RKF45 с шагом 0.08 выдаёт приблизительно те же точные результаты, что и метод Адамса 4-ой степени точности с  $h_{\text{int}} = 0.001$ . Однако при шаге интегрирования равном 0.08 Адамс выдаёт до  $t = 0.24$  относительно точные значения, далее идут сильные расхождения в сравнении с остальными полученными данными. Исходя из этого, можно сказать, что метод Адамса 4-го порядка неустойчив при  $h_{\text{int}} = 0.08$ , нужно использовать меньший шаг интегрирования (например, тот же 0.001).

Оценка шага метода Адамса:

$$\begin{pmatrix} -44 & -160 \\ 1 & 0 \end{pmatrix}$$

$$\lambda^2 + 44\lambda + 160 = 0$$

$$\lambda_1 = -40, \lambda_2 = -4$$

$$h < \frac{0.3}{|\lambda_k|_{\max}} = \frac{0.3}{40} = 0.0075$$

Шаг интегрирования метода Адамса 4-ой степени точности не должен превышать 0.0075, иначе способ неустойчив и выдаёт некорректные результаты.