

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Інститут прикладної математики та фундаментальних наук
Кафедра прикладної математики

Звіт

про виконання лабораторної роботи №1а
з курсу “Чисельні методи частина 2”

на тему:

«ОДНОКРОКОВІ МЕТОДИ ЧИСЕЛЬНОГО РОЗВ’ЯЗУВАННЯ ЗАДАЧІ КОШІ
ДЛЯ СИСТЕМ ЗВИЧАЙНИХ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ»

Виконав:

студент гр. ПМ-41

Дудяк М.С.

Прийняв:

доцент

Пізюр Я. В.

Львів-2016

Варіант 6

Постановка задачі

Задано задачу Коші для системи звичайних диференціальних рівнянь:

$$\begin{cases} y_1' = -2y_1 + 5y_2 \\ y_2' = -(1 - \sin t)y_1 - y_2 + 3y_3 \\ y_3' = -y_1 + 2y_3 \end{cases},$$

$$\begin{aligned} y_1(0) &= 2 \\ y_2(0) &= 1 \\ y_3(0) &= 1 \\ t_k &= 1 \end{aligned}.$$

1. Використовуючи мову програмування C++, написати та відлагодити програму чисельного розв'язування задачі Коші для ЗДР методом Рунге-Кутта 4го порядку точності з автоматичним вибором кроку при заданій точності $\epsilon = 10^{-5}$ та $\epsilon = 10^{-7}$.

Алгоритм для конкретного чисельного методу

В своєму варіанті я використовував метод Рунге-Кутта четвертого порядку точності за наступним алгоритмом:

1. Ввести значення $t_0, T, \bar{y}_0, \varepsilon, \tau_0, \varepsilon_M$.
2. Ініціалізувати змінні $t := t_0; \bar{y} := \bar{y}_0; \tau := \tau_0; e_{\max} := 0$.
Вивести значення $t; \bar{y}$.
3. If $|T - t| < \varepsilon_M$ then go to 13.
4. If $t + \tau > T$ then $\tau = T - t$.
5. $\bar{v} := \bar{y}; t_1 := t$.
6. $k_f := 0$.
7. Обчислити $\bar{k}_1 := \bar{f}(t, \bar{y})$.
8. Обчислити

$$\left\{ \begin{array}{l} \overline{k}_2 = \overline{f}\left(t + \frac{\tau}{2}, \overline{y} + \frac{\tau \overline{k}_1}{2}\right) \\ \overline{k}_3 = \overline{f}\left(t + \frac{\tau}{2}, \overline{y} + \frac{\tau \overline{k}_2}{2}\right) \\ \overline{k}_4 = \overline{f}\left(t + \tau, \overline{y} + \tau \overline{k}_3\right) \\ \overline{y}_{n+1} = \overline{y}_n + \frac{\tau}{6}(\overline{k}_1 + 2 \overline{k}_2 + 2 \overline{k}_3 + \overline{k}_4) \end{array} \right.$$

9. **If** $kf = 0$ **then**

begin $\overline{w} := \overline{y}$; $\overline{y} := \overline{v}$; $\tau := \tau / 2$; $kf := 1$; **go to 8 end.**

10. **If** $kf = 1$ **then**

begin $t := t + \tau$; $kf := 2$; **go to 7 end.**

11. **Обчислити**

$$E := \frac{\max_{1 \leq i \leq N} |y_i - w_i|}{15 \max(1, \max_{1 \leq i \leq N} |y_i|)} \quad \tau_H := 2 \tau \min\left(5; \max\left(0.1; 0.9 \left(\frac{\epsilon}{E}\right)^{\frac{1}{5}}\right)\right) \text{ ,.}$$

12. **If** $E \leq \epsilon$ **then**

begin $t := t + \tau$; $\overline{y} := \overline{y} + \frac{\overline{y} - \overline{w}}{15}$; $\tau := \tau_H$

вивести значення t ; \overline{y} ;

go to 3

end;

else

begin $\overline{y} := \overline{v}$; $t := t_1$; $\tau := \tau_H$; **go to 6 end.**

13. **Вивести норму похибки** e_{\max} .

14. **End.**

Текст програми

```
// Lab2.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include "Vect.h"
#include <iostream>
#include <algorithm>
#include <iomanip>
using namespace std;
typedef double num;

//Vector u(num t)
//{
//    Vector v = { 1,1,1 };
//    return v*(-exp(-4.0*t) + (exp(-1.0) + 4000.0) / exp(3.0))*exp(-t*(t - 4.0)) /
//    400.0;
//}

Vector f(num t, Vector u)
{
    Vector res{
        -2 * u[0] + 5 * u[1],
        -(1 - sin(t))*u[0] - u[1] + 3 * u[2],
        -u[0]+2*u[2]
    };
    return res;
}

bool debug = false;
bool debug_input = true;
bool intermed_out = false;

int main()
{
    try {
        /*if (debug)
        {
            Vector v = u(1);
            cout << v << endl;
        }*/
        // Part 1
        if (debug) cout << "Part 1" << endl;
        num t0, T, eps, tau0, eM;
        Vector y0(3);
        if(debug_input)
        {
            t0 = 0.0;
            T = 1.0;
            y0 = { 2,1,1 };
            eps = 0.00001;
            tau0 = 1;
        } else
        {
            cout << "Input t0: "; cin >> t0;
            cout << "Input T: "; cin >> T;
            cout << "Input y0: "; cin >> y0;
            cout << "Input eps: "; cin >> eps;
            cout << "Input tau0: "; cin >> tau0;
        }
        eM = eps;
        // Part 2
        if (debug) cout << "Part 2" << endl;
        num t = t0;
        Vector y = y0;
        num tau = tau0;
        cout << "t=" << left << setw(6) << t << "\t" << "y=" << y << endl; //"
        u(t)=" << u(t) << " |y-u(t)|=" << abs(y - u(t)) << endl;
```

```

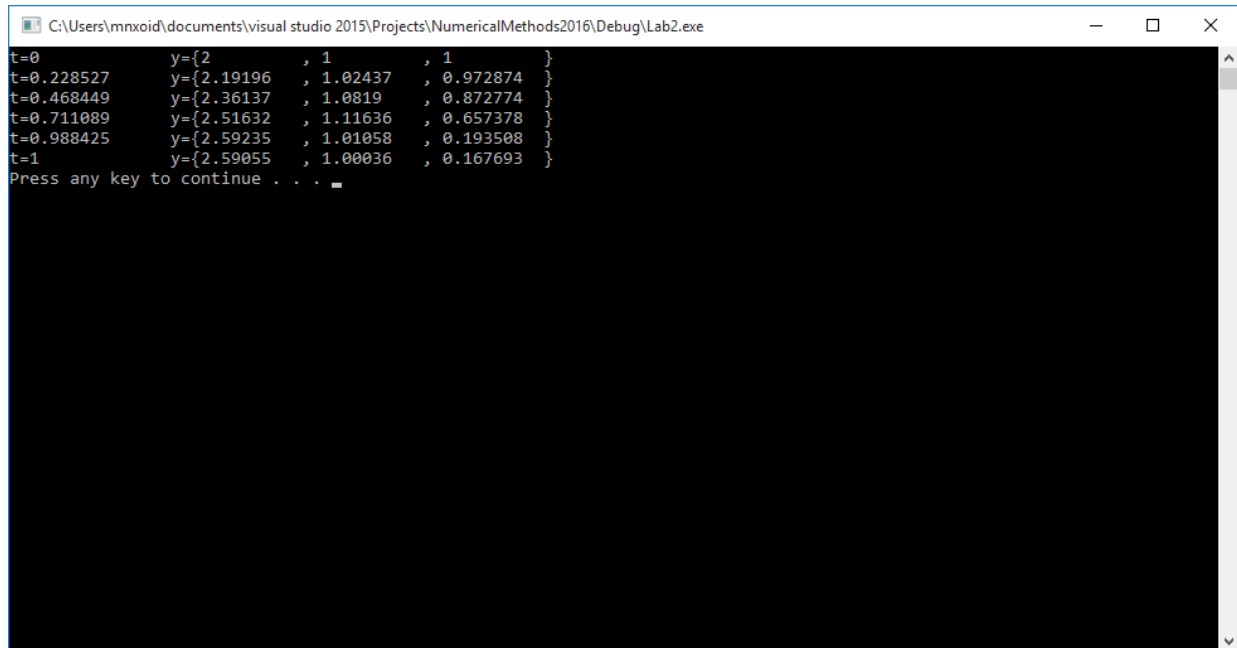
// Part 3
if (debug) cout << "Part 3" << endl;
while (abs(T - t) >= eM)
{
    // Part 4
    if (debug) cout << "Part 4" << endl;
    if (t + tau > T) tau = T - t;
    // Part 5
    if (debug) cout << "Part 5" << endl;
    Vector v = y;
    num t1 = t;
    Vector w;
part6:
    // Part 6
    if (debug) cout << "Part 6" << endl;
    int kf = 0;
    while (kf < 3) {
        // Part 7
        if (debug) cout << "Part 7" << endl;
        Vector k1 = f(t, y);
        // Part 8
        if (debug) cout << "Part 8" << endl;
        Vector k2 = f(t + tau / 2.0, y + tau*k1 / 2.0);
        Vector k3 = f(t + tau / 2.0, y + tau*k2 / 2.0);
        Vector k4 = f(t + tau, y + tau*k3);
        y = y + tau*(k1 + 2.0 * k2 + 2.0 * k3 + k4) / 6.0;
        /*if (intermed_out)
        {
            num diff = abs(y - u(t));
            cout << "t=" << t << "\ty=" << y << "\tu(t)=" << u(t)
<< "\t|y-u(t)|=" << diff << endl;
        }*/
        // Part 9
        if (debug) cout << "Part 9" << endl;
        if (kf == 0)
        {
            w = y;
            y = v;
            tau = tau / 2;
        }
        else if (kf == 1)
        {
            t = t + tau;
        }
        kf++;
    }
    // Part 11
    if (debug) cout << "Part 11" << endl;
    num E = (y - w).abs().max() / (15.0 * max(1.0, y.abs().max()));
    num tauH = 2 * tau*min(5.0, max(0.1, 0.9*pow(eM / E, 0.2)));
    // Part 12
    if (debug) cout << "Part 12" << endl;
    if (E <= eM)
    {
        t = t + tau;
        y = y + (y - w) / 15.0;
        //Vector precise = u(t);
        tau = tauH;
        //num diff = abs(y - u(t));
        cout << "t=" << left << setw(6) << t << "\t" << "y=" << y
<< endl; // "\tu(t)=" << u(t) << "\t|y-u(t)|=" << diff << endl;
        /*if (eM < diff)
            eM = diff;*/
        continue;
    }
    else
    {
        y = v;
        t = t1;
        tau = tauH;
    }
}

```

```
        goto part6;
    }
    break;
}
// Part 13
if (debug) cout << "Part 13" << endl;
//cout << "eM=" << eM << endl;
system("pause");
return 0;
}
catch (const exception &e) {
    cerr << "ERROR: " << e.what() << endl;
    system("pause");
    return 0;
}
}
```

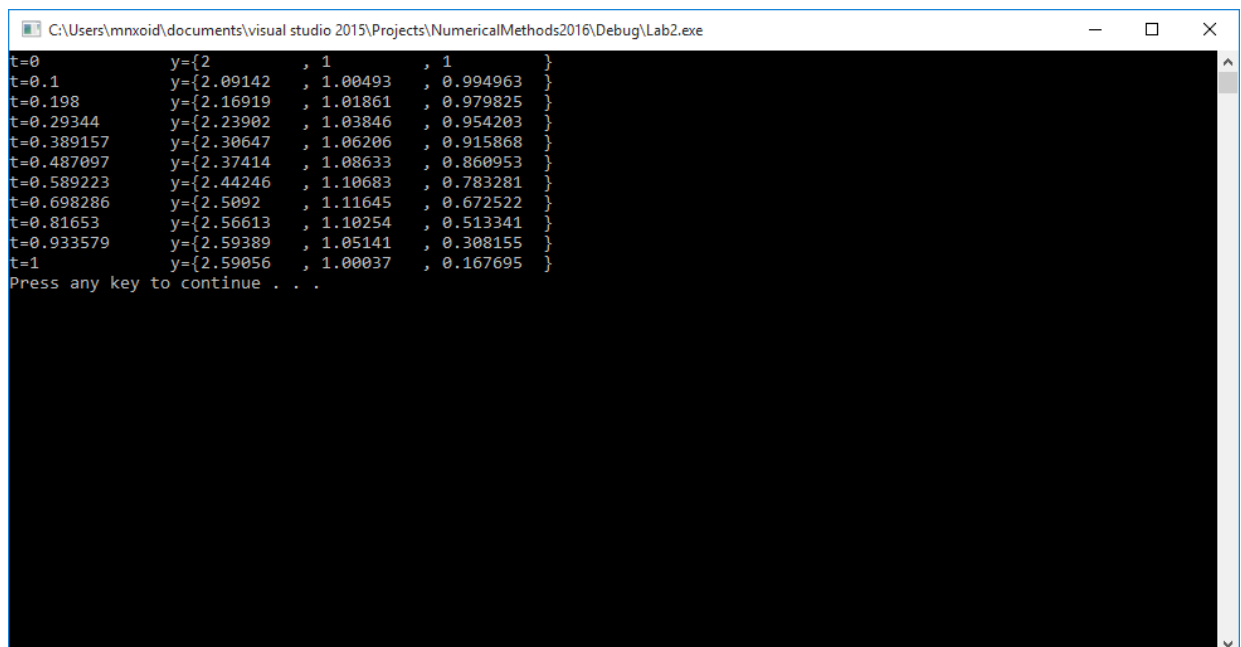
Результат виконання програми

$\epsilon = 10^{-5}$:



```
C:\Users\mnxoid\documents\visual studio 2015\Projects\NumericalMethods2016\Debug\Lab2.exe
t=0      y={2      , 1      , 1      }
t=0.228527 y={2.19196 , 1.02437 , 0.972874 }
t=0.468449 y={2.36137 , 1.0819 , 0.872774 }
t=0.711089 y={2.51632 , 1.11636 , 0.657378 }
t=0.988425 y={2.59235 , 1.01058 , 0.193508 }
t=1      y={2.59055 , 1.00036 , 0.167693 }
Press any key to continue . . .
```

$\epsilon = 10^{-7}$:



```
C:\Users\mnxoid\documents\visual studio 2015\Projects\NumericalMethods2016\Debug\Lab2.exe
t=0      y={2      , 1      , 1      }
t=0.1     y={2.09142 , 1.00493 , 0.994963 }
t=0.198   y={2.16919 , 1.01861 , 0.979825 }
t=0.29344 y={2.23902 , 1.03846 , 0.954203 }
t=0.389157 y={2.30647 , 1.06206 , 0.915868 }
t=0.487097 y={2.37414 , 1.08633 , 0.860953 }
t=0.589223 y={2.44246 , 1.10683 , 0.783281 }
t=0.698286 y={2.5092 , 1.11645 , 0.672522 }
t=0.81653 y={2.56613 , 1.10254 , 0.513341 }
t=0.933579 y={2.59389 , 1.05141 , 0.308155 }
t=1      y={2.59056 , 1.00037 , 0.167695 }
Press any key to continue . . .
```

Висновок

В цій лабораторній роботі я навчився: з допомогою мови програмування C++ розв'язувати системи ЗДР методом Рунге-Кутта четвертого порядку точності з автоматичним вибором довжини кроку для заданої точності.