

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Інститут прикладної математики та фундаментальних наук  
Кафедра прикладної математики



Звіт  
про виконання лабораторної роботи №1  
з курсу “Чисельні методи частина 2”  
на тему:  
«ОДНОКРОКОВІ МЕТОДИ ЧИСЕЛЬНОГО РОЗВ’ЯЗУВАННЯ ЗАДАЧІ КОШІ  
ДЛЯ ЗВИЧАЙНИХ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ»

*Виконав:*  
*студент гр. ПМ-41*  
*Дудяк М.С.*  
*Прийняв:*  
*доцент*  
*Пізюр Я. В.*

Львів-2016

## Варіант 4

### Постановка задачі

Задано задачу Коші для звичайного диференціального рівняння:

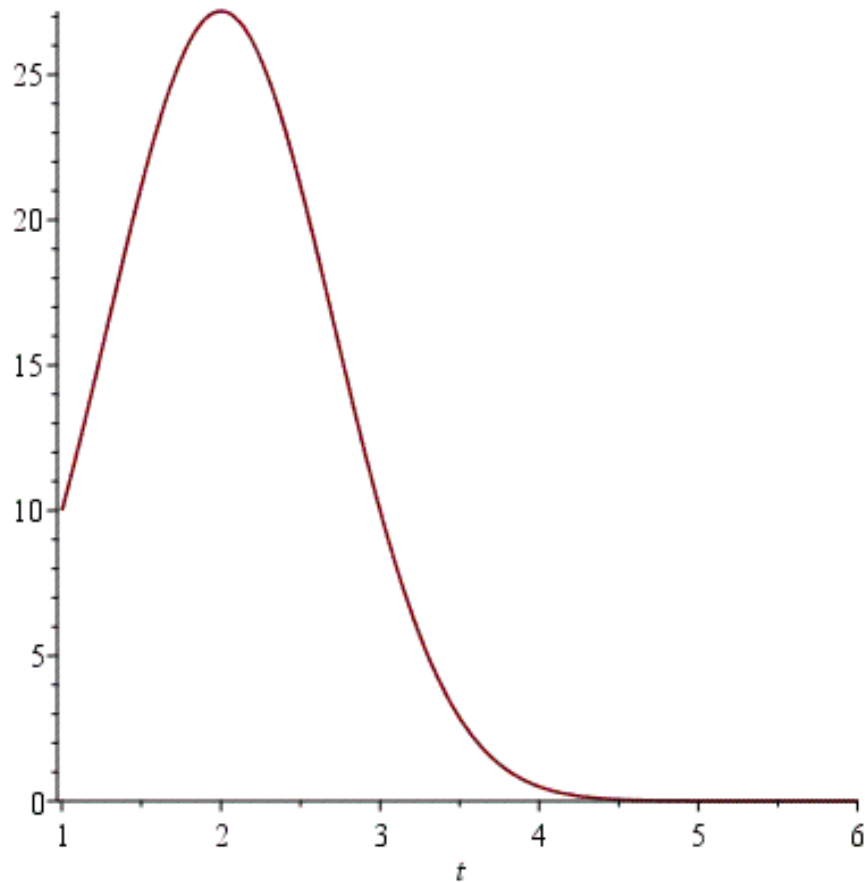
$$\begin{cases} u' = 0.01 e^{-t^2} + (2t-4)u \\ u(1) = 10 \end{cases}.$$

1. Знайти аналітичний розв'язок з допомогою пакету Maple
2. Використовуючи мову програмування C++, написати та відлагодити програму чисельного розв'язування задачі Коші для ЗДР методом Рунге-Кутта з автоматичним вибором кроку при заданій точності  $\epsilon = 10^{-5}$  та  $\epsilon_M = 10^{-6}$ .

### Аналітичний розв'язок задачі

```
> restart;
> t0 := 1; T := 6;
>
> f1 := exp(-t*t);
>
> f2 := 0.01;
>
> g := 2*(t-2)
>
> zdr := {diff(y(t), t) - f1*f2 + g*y(t) = 0};
>
> pu := {y(t0) = 10};
>
> s := dsolve(zdr union pu, {y(t)});
>
> evalf(y(t), s);
>
> plot(%, t = t0..T);
```

$$t0 := 1$$
$$T := 6$$
$$f1 := e^{-t^2}$$
$$f2 := 0.01$$
$$g := 2t - 4$$
$$zdr := \left\{ \frac{d}{dt} y(t) - 0.01 e^{-t^2} + (2t - 4) y(t) = 0 \right\}$$
$$pu := \{y(1) = 10\}$$
$$s := y(t) = \left( -\frac{1}{400} e^{-4t} + \frac{1}{400} \frac{e^3 e^{-4} + 4000}{e^3} \right) e^{-t(t-4)}$$
$$\left( -\frac{1}{400} e^{-4t} + \frac{1}{400} \frac{e^3 e^{-4} + 4000}{e^3} \right) e^{-t(t-4)}$$



> `yI := unapply(%%, t);`

$$yI := t \rightarrow \left( -\frac{1}{400} e^{-4t} + \frac{1}{400} \frac{e^3 e^{-4} + 4000}{e^3} \right) e^{-t(t-4)}$$

> `evalf(yI(t0));`

9.999999999

> `evalf(yI(T));`

0.000003059304543

### Алгоритм для конкретного чисельного методу

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

1. Ввести значення  $t_0$ ,  $T$ ,  $y_0$ ,  $\varepsilon$ ,  $\tau_0$ ,  $\varepsilon_M$ .
2. Ініціалізувати змінні  $t := t_0$ ;  $y := y_0$ ;  $\tau := \tau_0$ ;  $e_{\max} := 0$ .

Вивести значення  $t$ ;  $y$ ;  $u(t)$ ;  $|y - u(t)|$ .

3. If  $|T - t| < \varepsilon_M$  then go to 13.
4. If  $t + \tau > T$  then  $\tau = T - t$ .
5.  $v := y$ ;  $t_1 := t$ .
6.  $kf := 0$ .

7. **Обчислити**  $k_1 := f(t, y)$ .

8. **Обчислити**

$$\begin{cases} k_2 = f\left(t + \frac{\tau}{2}, y + \frac{\tau k_1}{2}\right) \\ k_3 = f\left(t + \frac{\tau}{2}, y + \frac{\tau k_2}{2}\right) \\ k_4 = f(t + \tau, y + \tau k_3) \\ y_{n+1} = y_n + \frac{\tau}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{cases}$$

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

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

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

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

11. **Обчислити**

$$E := \frac{|y-w|}{15 \max(1, |y|)} \quad , \quad \tau_H := 2 \tau \min\left(5; \max\left(0.1; 0.9 \left(\frac{\epsilon}{E}\right)^{\frac{1}{5}}\right)\right) .$$

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

**begin**  $t := t + \tau$ ;  $y := y + \frac{y-w}{15}$ ; **обчислити точний розв'язок**  $u(t)$ ;  $\tau := \tau_H$ ;

**вивести значення**  $t$ ;  $y$ ;  $u(t)$ ;  $|y - u(t)|$ ;

**if**  $e_{\max} < |y - u(t)|$  **then**  $e_{\max} := |y - u(t)|$ ;

**go to 3**

**end;**

**else**

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

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

14. **End.**

## Текст програми

```
// Lab1.cpp : Defines the entry point for the console application.
//

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

num u(num t)
{
    return (-exp(-4.0*t)+(exp(-1.0)+4000.0)/exp(3.0))*exp(-t*(t-4.0))/400.0;
}

num f(num t, num u)
{
    return exp(-t*t)*0.01 - 2.0 * (t - 2.0)*u;
}

bool debug = false;
bool intermed_out = false;

int main()
{
    if(debug)
    {
        cout << u(1) << endl;
    }
    // Part 1
    if (debug) cout << "Part 1" << endl;
    num t0, T, y0, eps, tau0, eM;
    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;
    cout << "Input eM: "; cin >> eM;
    // Part 2
    if (debug) cout << "Part 2" << endl;
    num t = t0;
    num y = y0;
    num tau = tau0;
    cout << "t=" << t << " y=" << y << " 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;
        num v = y;
        num t1 = t;
        num w;
        part6:
        // Part 6
        if (debug) cout << "Part 6" << endl;
        int kf = 0;
        while (kf < 3) {
            // Part 7
            if (debug) cout << "Part 7" << endl;
            num k1 = f(t, y);
            // Part 8
            if (debug) cout << "Part 8" << endl;
            num k2 = f(t + tau / 2.0, y + tau*k1 / 2.0);
            num k3 = f(t + tau / 2.0, y + tau*k2 / 2.0);
            num 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 = abs(y - w) / (15.0 * max(1.0, abs(y)));
num tauH = 2*tau*min(5.0, max(0.1, 0.9*pow(eps / E, 0.2)));
// Part 12
if (debug) cout << "Part 12" << endl;
if(E<=eps)
{
    t = t + tau;
    y = y + (y - w) / 15.0;
    num precise = u(t);
    tau = tauH;
    num diff = abs(y - u(t));
    cout << "t=" << t << "\ty=" << y << "\tu(t)=" << u(t) << "\t|y-u(t)|=" << diff
<< endl;

    if (eM < diff)
    //{
        eM = diff;
        continue;///TODO: Check this
    //}
} 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;
}

```

## Результат виконання програми

C:\Users\mnxoid\documents\visual studio 2015\Projects\NumericalMethods2016\Debug\Lab1.exe

```
Input t0: 1
Input T: 6
Input y0: 10
Input eps: 0.00001
Input tau0: 0.5
Input eM: 0.000001
t=1 y=10 u(t)=10 |y-u(t)|=1.77636e-15
t=1.24621 y=15.4012 u(t)=15.4012 |y-u(t)|=1.51903e-05
t=1.48727 y=20.9005 u(t)=20.9006 |y-u(t)|=2.05761e-05
t=1.79737 y=26.0917 u(t)=26.0917 |y-u(t)|=1.84418e-05
t=2.22394 y=25.8556 u(t)=25.8556 |y-u(t)|=2.81109e-05
t=2.38965 y=23.3558 u(t)=23.3558 |y-u(t)|=2.50831e-05
t=2.72515 y=16.0682 u(t)=16.0682 |y-u(t)|=5.44872e-05
t=2.93368 y=11.3693 u(t)=11.3692 |y-u(t)|=4.02788e-05
t=3.11605 y=7.8234 u(t)=7.82338 |y-u(t)|=2.50679e-05
t=3.27399 y=5.36361 u(t)=5.3636 |y-u(t)|=1.40431e-05
t=3.41492 y=3.67179 u(t)=3.67178 |y-u(t)|=7.02947e-06
t=3.54327 y=2.51176 u(t)=2.51176 |y-u(t)|=2.88055e-06
t=3.67324 y=1.65357 u(t)=1.65357 |y-u(t)|=5.13494e-07
t=3.79185 y=1.09633 u(t)=1.09633 |y-u(t)|=1.81808e-06
t=3.90194 y=0.730001 u(t)=0.730003 |y-u(t)|=2.14867e-06
t=4.01193 y=0.474643 u(t)=0.474645 |y-u(t)|=2.27839e-06
t=4.12456 y=0.297874 u(t)=0.297877 |y-u(t)|=2.34684e-06
t=4.24079 y=0.17934 u(t)=0.179343 |y-u(t)|=2.36551e-06
t=4.36179 y=0.102756 u(t)=0.102758 |y-u(t)|=2.34633e-06
t=4.48896 y=0.0554493 u(t)=0.0554516 |y-u(t)|=2.29755e-06
t=4.62404 y=0.0277926 u(t)=0.0277948 |y-u(t)|=2.22419e-06
t=4.76929 y=0.0126962 u(t)=0.0126984 |y-u(t)|=2.12827e-06
t=4.92776 y=0.00514617 u(t)=0.00514818 |y-u(t)|=2.00875e-06
t=5.10378 y=0.00177883 u(t)=0.00178069 |y-u(t)|=1.86126e-06
t=5.30376 y=0.000492735 u(t)=0.000494413 |y-u(t)|=1.67746e-06
t=5.53788 y=9.81998e-05 u(t)=9.96435e-05 |y-u(t)|=1.44372e-06
t=5.82363 y=1.10211e-05 u(t)=1.21585e-05 |y-u(t)|=1.13736e-06
t=6 y=2.7622e-06 u(t)=3.0593e-06 |y-u(t)|=2.97107e-07
eM=5.44872e-05
Press any key to continue . . .
```

## **Висновок**

В цій лабораторній роботі я навчився: використовувати програмний пакет Maple для пошуку аналітичного розв'язку ЗДР, з допомогою мови програмування C++ розв'язувати ЗДР методом Рунге-Кутта четвертого порядку точності з автоматичним вибором довжини кроку для заданої точності.