

Міністерство науки і освіти 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 групи ПМ-42
Сватюк Д.Р.
Перевірила:
Візнович О.В.

Львів 2017

Мета: навчитися аналітично і чисельно розв'язувати задачі Коші.

Постановка задачі

Використовуючи мову програмування Go, написати та відлагодити програму чисельного розв'язування задачі Коші для ЗДР вкладеним методом Рунге-Кутта 3-го порядку з автоматичним вибором кроку при заданій точності $\varepsilon = 10^{-5}$ та $\varepsilon_M = 10^{-6}$. Знайти аналітичний розв'язок цієї ж задачі з використанням середовища Maple.

$$u' = \varphi(t)\psi(t) - g(t)u, \quad t \in [t_0, T], \quad u(t_0) = u_0$$

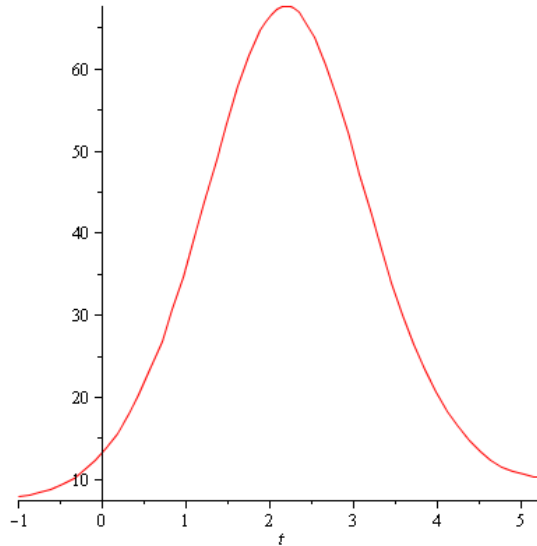
$$\tau_0 = 0,5, t_0 = -1, T = 2\pi - 1, u_0 = 8, g(t) = -\sin(t+1)$$

№	$\varphi(t)$	$\psi(t)$
14	e^{t_0}	$e^{1-\cos(t+1)}$

Аналітичний розв'язок задачі

> restart		
> t0 := -1; T := 2·Pi - 1;	$t0 := -1$ $T := 2\pi - 1$	(1)
> f1 := exp(t0);	$f1 := e^{-1}$	(2)
> f2 := exp(1 - cos(t + 1));	$f2 := e^{1 - \cos(t+1)}$	(3)
> g := -sin(t + 1);	$g := -\sin(t+1)$	(4)
> zdr := {diff(y(t), t) - f1·f2 + g·y(t) = 0};	$zdr := \left\{ \frac{d}{dt} y(t) - e^{-1} e^{1 - \cos(t+1)} - \sin(t+1) y(t) = 0 \right\}$	(5)
> pu := {y(t0) = 8};	$pu := \{y(-1) = 8\}$	(6)
> s := dsolve(zdr union pu, {y(t)});	$s := y(t) = e^{-\cos(t+1)} t - \frac{e^{-\cos(t+1)} (-8 - e^{-1})}{e^{-1}}$	(7)
> eval(y(t), s);	$e^{-\cos(t+1)} t - \frac{e^{-\cos(t+1)} (-8 - e^{-1})}{e^{-1}}$	(8)
> plot(%, t=t0..T);		

```
> plot(%, t=t0..T);
```



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

$$yI = \left(t \mapsto e^{-\cos(t+1)} t - \frac{e^{-\cos(t+1)} (-8 - e^{-1})}{e^{-1}} \right)$$

(9)

$$y(t_0) = 1 ; y(T) = 10.311454$$

Алгоритм чисельного розв'язування задачі Коші вкладеними методами Рунге-Кутта

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 10.
4. If $t + \tau > T$ then $\tau = T - t$.
5. $v := y; t_1 := t$.
6. Обчислити $k_1 := f(t, y)$.
7. Обчислити

$$k_2 = f(t + c_2 \tau, y + \tau a_{21} k_1);$$

.....

$$k_s = f(t + c_s \tau, y + \tau a_{s1} k_1 + \dots + \tau a_{s,s-1} k_{s-1});$$

$$w = y + \tau(b_1 k_1 + b_2 k_2 + \dots + b_s k_s);$$

$$y = y + \tau(\hat{b}_1 k_1 + \hat{b}_2 k_2 + \dots + \hat{b}_s k_s)$$

$$8. \text{ Обчислити } E := \frac{|y - w|}{\max(1, |y|)}, \tau_H := \tau \min \left(5, \max \left(0, 1, 0.9 \left(\frac{\varepsilon}{E} \right)^{\frac{1}{p+1}} \right) \right).$$

9. If $E \leq \varepsilon$ then

begin $t := t + \tau$; 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 7 end.

10. Вивести норму похибки e_{\max} .

11. End.

Програмний код

```
package main
import (
    "math"
    "strconv"
)
const t0 = -1.0
const T = 2*math.Pi - 1
const y0 = 8.0
const eps = 0.01
const tau0 = 0.2
const epsM = 0.0001
func main() {
    method := RungeKuttaMethod{}
    method.Start(t0, T, y0, eps, tau0, epsM)
    println("FINISHED!")
}
// u'(t) = f(t,u) = (e^t0)*(e^(1-cos(t+1))) + sin(t+1)*u
func f(t float64, u float64) float64 {
    return (1.0/math.E)*math.Pow(math.E, (1-math.Cos(t+1))) + math.Sin(t+1)*u
}
// Analytic solution of this differential equation found in Maple
func u(t float64) float64 {
    return math.Pow(math.E, -math.Cos(t+1))*t - math.E*(math.Pow(math.E, -
math.Cos(t+1))*(-8.0-1.0/math.E))
}
type RungeKuttaMethod struct {
    t          float64
    T          float64
    y          float64
    tau        float64
    eMax       float64
    v          float64
    t1         float64
    w          float64
    k1, k2, k3 float64
    E          float64
    tauH       float64
    epsM       float64
}
func (m RungeKuttaMethod) Start(t0 float64, T float64, y0 float64, eps float64, tau0
float64, epsM float64) {
    m.Step1(t0, T, y0, eps, tau0, epsM)
    return
}
func (m RungeKuttaMethod) Step1(t0 float64, T float64, y0 float64, eps float64, tau0
float64, epsM float64) {
    m.t = t0
    m.T = T
    m.y = y0
```

```

        m.tau = tau0
        m.eMax = 0
        m.epsM = epsM
        m.Step2()
        return
    }
    func (m RungeKuttaMethod) Step2() {
        println("t = " + strconv.FormatFloat(m.t, 'f', 8, 64))
        println("y = " + strconv.FormatFloat(m.y, 'f', 8, 64))
        println("u(t) = " + strconv.FormatFloat(u(m.t), 'f', 8, 64))
        println("|y - u(t)| = " + strconv.FormatFloat(math.Abs(m.y-u(m.t)), 'f', 8,
64))
        m.Step3()
        return
    }
    func (m RungeKuttaMethod) Step3() {
        if math.Abs(m.T-m.t) < m.epsM || m.t > m.T {
            m.Step10()
            return
        } else {
            m.Step4()
        }
        return
    }
    func (m RungeKuttaMethod) Step4() {
        if m.t+m.tau > m.T {
            m.tau = m.T - m.t
        }
        m.Step5()
        return
    }
    func (m RungeKuttaMethod) Step5() {
        m.v = m.y
        m.t1 = m.t
        m.Step6()
        return
    }
    func (m RungeKuttaMethod) Step6() {
        m.k1 = f(m.t, m.y)
        m.Step7()
        return
    }
    func (m RungeKuttaMethod) Step7() {
        m.k2 = f(m.t+1*m.tau, m.y+m.tau*1*m.k1)
        m.k3 = f(m.t+0.5*m.tau, m.y+m.tau*0.25*m.k1+m.tau*0.25*m.k2)
        m.w = m.y + m.tau*(0.5*m.k1+0.5*m.k2+0*m.k3)
        m.y = m.y + m.tau*(1.0/6.0*m.k1+1.0/6.0*m.k2+4.0/6.0*m.k3)
        m.Step8()
        return
    }
    func (m RungeKuttaMethod) Step8() {
        m.E = math.Abs(m.y-m.w) / math.Max(1.0, math.Abs(m.y))
        m.tauH = m.tau * math.Min(5.0, math.Max(0.1, 0.9*math.Pow(eps/m.E,
1.0/(1.0+8.0))))
        m.Step9()
        return
    }

```

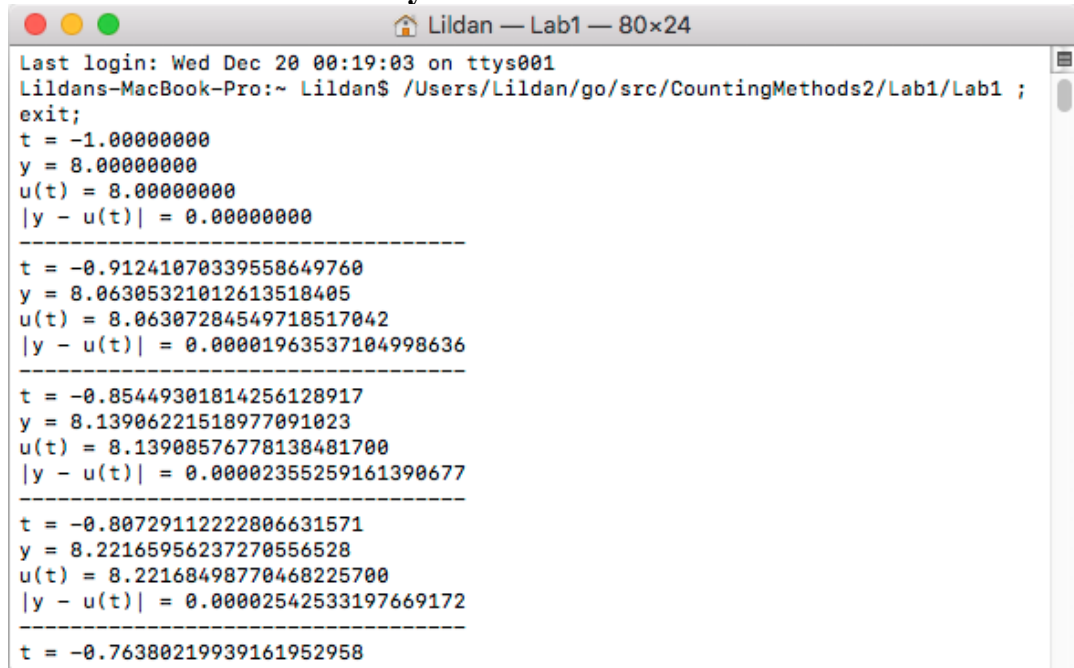
```

}
func (m RungeKuttaMethod) Step9() {
    if m.E <= eps {
        if m.tau > 0 {
            m.t = m.t + m.tau
        } else {
            m.Step10()
            return
        }
        un := u(m.t)
        m.tau = m.tauH
        println("-----")
        println("t = " + strconv.FormatFloat(m.t, 'f', 20, 64))
        println("y = " + strconv.FormatFloat(m.y, 'f', 20, 64))
        println("u(t) = " + strconv.FormatFloat(un, 'f', 20, 64))
        println("|y - u(t)| = " + strconv.FormatFloat(math.Abs(m.y-un), 'f', 20,
64))
        if m.eMax < math.Abs(m.y-un) {
            m.eMax = math.Abs(m.y - un)
        }
        m.Step3()
        return
    }

    m.y = m.v
    m.t = m.t1
    m.tau = m.tauH
    m.Step7()
    return
}
func (m RungeKuttaMethod) Step10() {
    println("Maximal error = " + strconv.FormatFloat(m.eMax, 'f', 8, 64))
    return
}

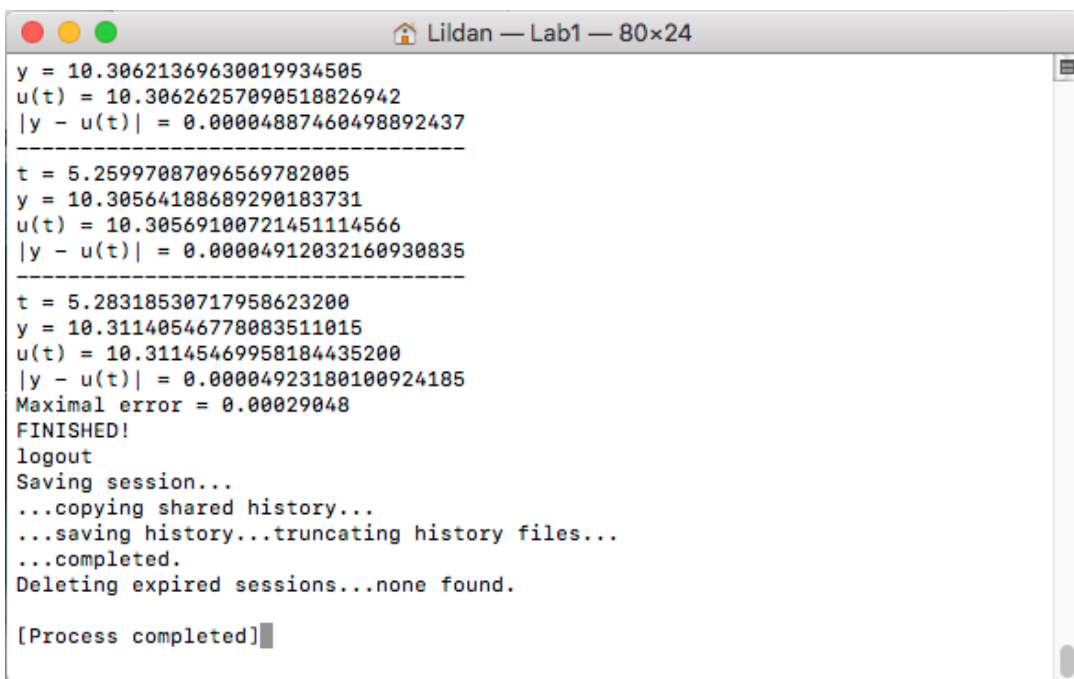
```

Результати виконання



```
Lildan — Lab1 — 80x24
Last login: Wed Dec 20 00:19:03 on ttys001
Lildans-MacBook-Pro:~ Lildan$ /Users/Lildan/go/src/CountingMethods2/Lab1/Lab1 ;
exit;
t = -1.00000000
y = 8.00000000
u(t) = 8.00000000
|y - u(t)| = 0.00000000
-----
t = -0.91241070339558649760
y = 8.06305321012613518405
u(t) = 8.06307284549718517042
|y - u(t)| = 0.00001963537104998636
-----
t = -0.85449301814256128917
y = 8.13906221518977091023
u(t) = 8.13908576778138481700
|y - u(t)| = 0.00002355259161390677
-----
t = -0.80729112222806631571
y = 8.22165956237270556528
u(t) = 8.22168498770468225700
|y - u(t)| = 0.00002542533197669172
-----
t = -0.76380219939161952958
```

Рис 1 Початок виводу



```
Lildan — Lab1 — 80x24
y = 10.30621369630019934505
u(t) = 10.30626257090518826942
|y - u(t)| = 0.00004887460498892437
-----
t = 5.25997087096569782005
y = 10.30564188689290183731
u(t) = 10.30569100721451114566
|y - u(t)| = 0.00004912032160930835
-----
t = 5.28318530717958623200
y = 10.31140546778083511015
u(t) = 10.31145469958184435200
|y - u(t)| = 0.00004923180100924185
Maximal error = 0.00029048
FINISHED!
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
Deleting expired sessions...none found.

[Process completed]
```

Рис 2 Кінець виводу

Висновок

Під час виконання лабораторної роботи було набуто навички аналітичного розв'язування задачі Коші у середовищі Maple.

Також було написано програму для чисельного розв'язування задачі Коші вкладеним методом Рунге-Кутта 3-го порядку. Для реалізації програми використано мову програмування Go.