

Міністерство науки і освіти України
Національний університет “Львівська політехніка”
Інститут прикладної математики та фундаментальних наук

Кафедра прикладної математики

Звіт
про виконання лабораторної роботи №2
з курсу “Чисельні методи, частина 2”
на тему
“Однокрокові методи чисельного розв’язування задачі Коші для систем
звичайних диференціальних рівнянь”

Виконав:
студент групи ПМ-42
Сватюк Д.Р.
Перевірила:
Візнович О.В.

Львів 2017

Мета: навчитися чисельно знаходити розв’язок задачі Коші для системи рівнянь вкладеним методом Рунге-Кутта 3-го порядку.

Постановка задачі

Використовуючи мову програмування Go, написати та відлагодити програму чисельного розв’язування задачі Коші для систем ЗДР вкладеним методом Рунге-Кутта 3-го порядку при заданій точності ε .

| | | |
|------------|---|---|
| Варіант 14 | $y_1' = t + y_1 y_2,$ $y_2' = t - y_1^2,$ | $y_1(0) = 1, y_2(0) = 0.5,$ $t_k = 1, \varepsilon = 10^{-4}.$ |
|------------|---|---|

Алгоритм чисельного розв’язування задачі Коші для систем ЗДР вкладеними методами Рунге-Кутта

1. Ввести значення $t_0, T, y_{0i}, i = \overline{1, N}$ (N – розмірність системи рівнянь), $\varepsilon, \tau_0, \varepsilon_M$.
2. Ініціалізувати змінні $t := t_0; \tau := \tau_0; \text{for } i := 1 \text{ to } N \ y_i := y_{0i}$.

Вивести значення $t; y_i, i = \overline{1, N}$.

3. If $|T - t| < \varepsilon_M$ then go to 10.
4. If $t + \tau > T$ then $\tau = T - t$.
5. for $i := 1$ to $N \ v_i := y_i; t_1 := t$.
6. Обчислити $\vec{k}_1 := f(t, \vec{y})$.
7. Обчислити

$$\vec{k}_2 = f(t + c_2 \tau, \vec{y} + \tau a_{21} \vec{k}_1)$$

.....

$$\vec{k}_s = f(t + c_s \tau, \vec{y} + \tau a_{s1} \vec{k}_1 + \dots + \tau a_{s,s-1} \vec{k}_{s-1})$$

for $i := 1$ to N

begin

$$w_i = y_i + \tau(b_1 k_{1,i} + b_2 k_{2,i} + \dots + b_s k_{s,i})$$

$$y_i = y_i + \tau(\hat{b}_1 k_{1,i} + \hat{b}_2 k_{2,i} + \dots + \hat{b}_s k_{s,i})$$

end.

$$8. \text{ Обчислити } E := \frac{\max_{1 \leq i \leq N} |y_i - w_i|}{\max\left(1, \max_{1 \leq i \leq N} |y_i|\right)}, \tau_H := \tau \min\left(5; \max\left(0, 1; 0.9 \left(\frac{\varepsilon}{E}\right)^{\frac{1}{p+1}}\right)\right).$$

9. If $E \leq \varepsilon$ then

begin $t := t + \tau$; вивести значення $t, y_i, i = \overline{1, N}$;

$\tau := \tau_H$; go to 3

end;

else

begin for $i := 1$ to $N \ y_i := v_i; t := t_1; \tau := \tau_H$; go to 7 end.

10. End.

Программный код

```
package main
import ("math"
"strconv")
const t0 = 0.0
const T = 1
const eps = 0.0001
const epsM = 0.0001
const tau0 = 0.5
const y10 = 1.0
const y20 = 0.5
func y1d(t float64, y1 float64, y2 float64) float64 {
    return t + y1*y2
}
func y2d(t float64, y1 float64, y2 float64) float64 {
    return t - y1*y1
}
func main() {
    method := SystemRungeKuttaMethod{}
    method.Start(t0, T, eps, epsM, tau0, y10, y20)
    println("FINISHED!")
}
type SystemRungeKuttaMethod struct {
    t    float64
    T    float64
    tau  float64
    eps  float64
    epsM float64
    y1   float64
    y2   float64
    v1   float64
    v2   float64
    t1   float64
    k11, k12, k13 float64
    k21, k22, k23 float64
    w1, w2 float64
    E     float64
    tauH  float64
}
func (m SystemRungeKuttaMethod) Start(t0 float64, T float64, eps float64, epsM
float64, tau float64, y10 float64, y20 float64) {
    m.t = t0
    m.T = T
    m.tau = tau
    m.eps = eps
    m.epsM = epsM
    m.y1 = y10
    m.y2 = y20
    m.Step2()
    return
}
func (m SystemRungeKuttaMethod) Step2() {
    println("t = " + strconv.FormatFloat(m.t, 'f', 8, 64))
```

```

println("y1 = " + strconv.FormatFloat(m.y1, 'f', 8, 64))
println("y2 = " + strconv.FormatFloat(m.y2, 'f', 8, 64))
m.Step3()
return
}
func (m SystemRungeKuttaMethod) Step3() {
    if math.Abs(m.T-m.t) < m.epsM {
        m.Step10()
        return
    }
    m.Step4()
    return
}
func (m SystemRungeKuttaMethod) Step4() {
    if m.t+m.tau > m.T {
        m.tau = m.T - m.t
    }
    m.Step5()
    return
}
func (m SystemRungeKuttaMethod) Step5() {
    m.v1 = m.y1
    m.v2 = m.y2
    m.t1 = m.t
    m.Step6()
    return
}
func (m SystemRungeKuttaMethod) Step6() {
    m.k11 = y1d(m.t, m.y1, m.y2)
    m.k21 = y2d(m.t, m.y1, m.y2)
    m.Step7()
    return
}
func (m SystemRungeKuttaMethod) Step7() {
    m.k12 = y1d(m.t+1*m.tau, m.y1+1*m.tau*m.k11, m.y2+1*m.tau*m.k11)
    m.k22 = y2d(m.t+1*m.tau, m.y1+1*m.tau*m.k21, m.y2+1*m.tau*m.k21)
    m.k13 = y1d(m.t+0.5*m.tau, m.y1+0.25*m.tau*m.k11+0.25*m.tau*m.k12,
m.y2+0.25*m.tau*m.k11+0.25*m.tau*m.k12)
    m.k23 = y2d(m.t+0.5*m.tau, m.y1+0.25*m.tau*m.k21+0.25*m.tau*m.k22,
m.y2+0.25*m.tau*m.k21+0.25*m.tau*m.k22)
    m.w1 = m.y1 + m.tau*(0.5*m.k11+0.5*m.k12+0*m.k13)
    m.w2 = m.y2 + m.tau*(0.5*m.k21+0.5*m.k22+0*m.k23)
    m.y1 = m.y1 + m.tau*(1.0/6.0*m.k11+1.0/6.0*m.k12+4.0/6.0*m.k13)
    m.y2 = m.y2 + m.tau*(1.0/6.0*m.k21+1.0/6.0*m.k22+4.0/6.0*m.k23)
    m.Step8()
    return
}
func (m SystemRungeKuttaMethod) Step8() {
    m.E = math.Max(math.Abs(m.y1-m.w1), math.Abs(m.y2-m.w2)) /
        (math.Max(1, math.Max(m.y1, m.y2)))
    m.tauH = m.tau * math.Min(5, math.Max(0.1, 0.9*math.Pow(m.eps/m.E, 1.0/(8+1))))
    m.Step9()
    return
}
func (m SystemRungeKuttaMethod) Step9() {
    if m.E <= m.eps {

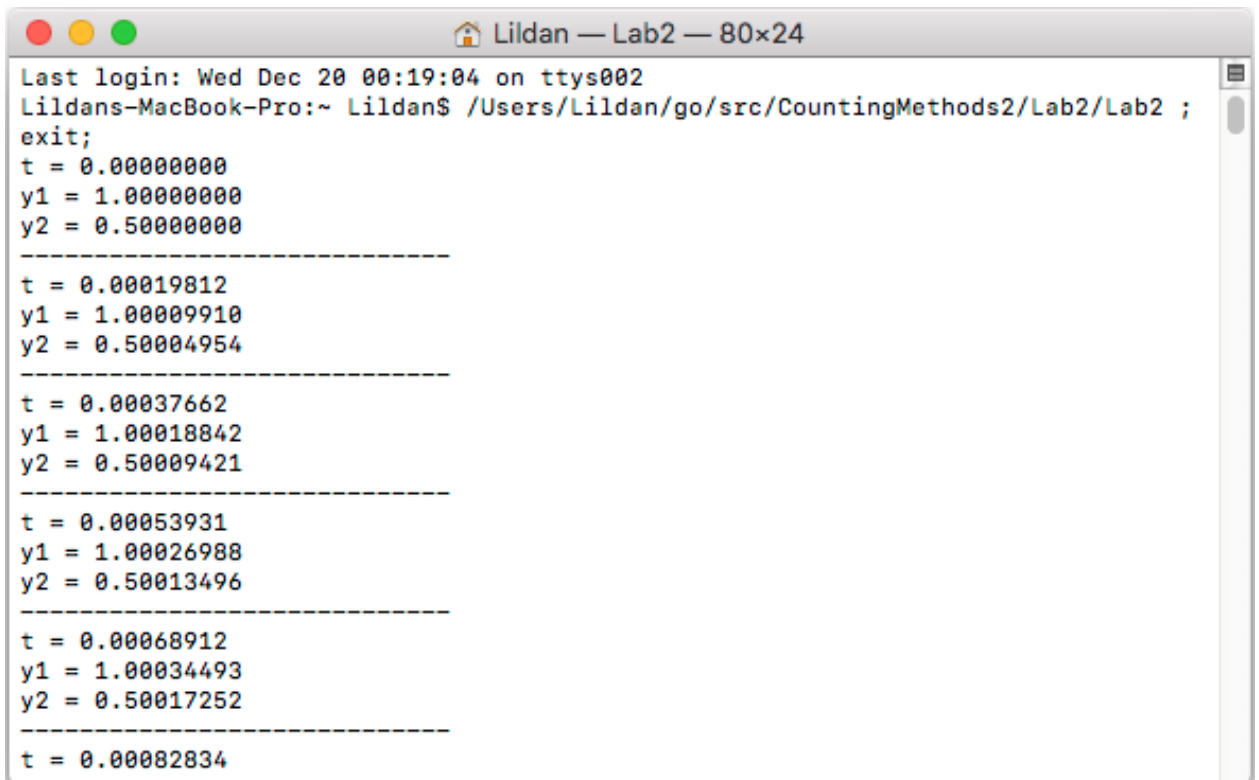
```

```

        m.t = m.t + m.tau
        m.tau = m.tauH
        println("-----")
        println("t = " + strconv.FormatFloat(m.t, 'f', 8, 64))
        println("y1 = " + strconv.FormatFloat(m.y1, 'f', 8, 64))
        println("y2 = " + strconv.FormatFloat(m.y2, 'f', 8, 64))
        m.Step3()
        return
    } else {
        m.y1 = m.v1
        m.y2 = m.v2
        m.t = m.t1
        m.tau = m.tauH
        m.Step7()
        return
    }
}
func (m SystemRungeKuttaMethod) Step10() {
    return
}

```

Результати виконання

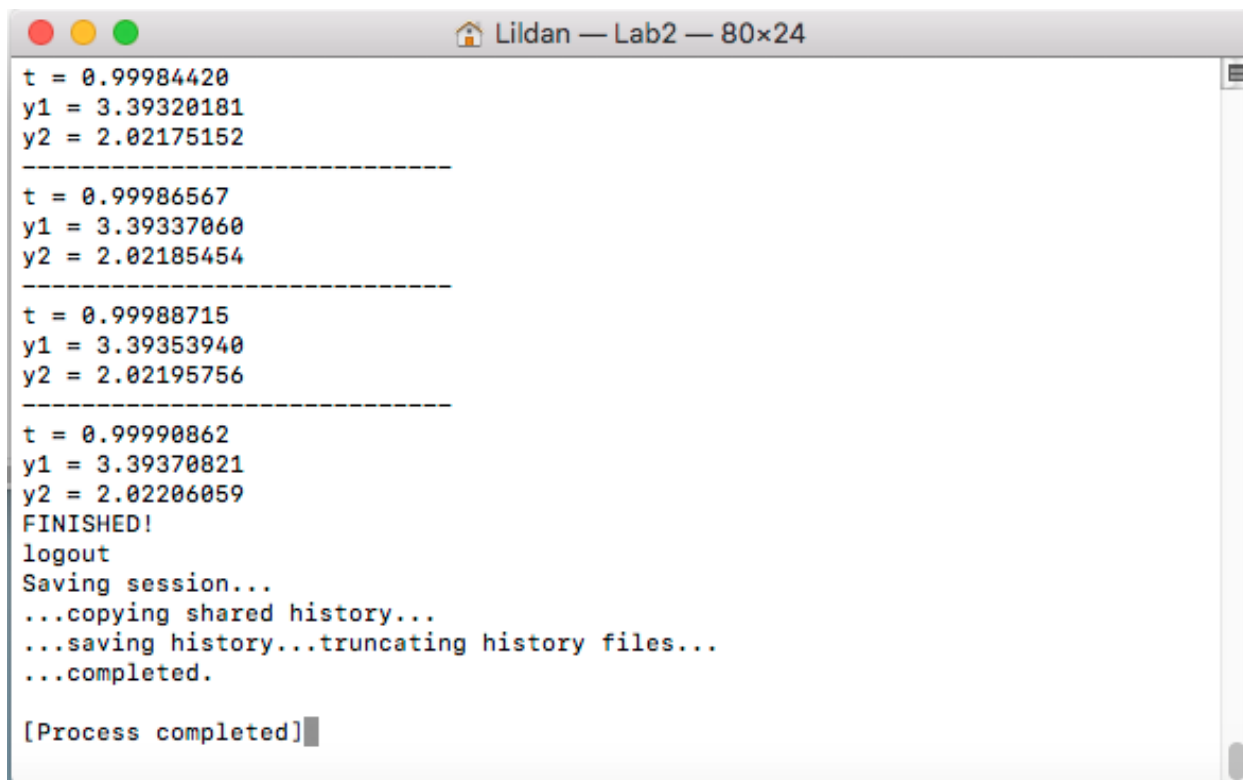


```

Last login: Wed Dec 20 00:19:04 on ttys002
Lildans-MacBook-Pro:~ Lildan$ /Users/Lildan/go/src/CountingMethods2/Lab2/Lab2 ;
exit;
t = 0.00000000
y1 = 1.00000000
y2 = 0.50000000
-----
t = 0.00019812
y1 = 1.00009910
y2 = 0.50004954
-----
t = 0.00037662
y1 = 1.00018842
y2 = 0.50009421
-----
t = 0.00053931
y1 = 1.00026988
y2 = 0.50013496
-----
t = 0.00068912
y1 = 1.00034493
y2 = 0.50017252
-----
t = 0.00082834

```

Рис 1 Початок виводу



```
t = 0.99984420
y1 = 3.39320181
y2 = 2.02175152
-----
t = 0.99986567
y1 = 3.39337060
y2 = 2.02185454
-----
t = 0.99988715
y1 = 3.39353940
y2 = 2.02195756
-----
t = 0.99990862
y1 = 3.39370821
y2 = 2.02206059
FINISHED!
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[Process completed]
```

Рис 2 кінець виводу

Висновок

Під час виконання лабораторної роботи було написано програму для чисельного розв'язування задачі Коші систем ЗДР вкладеним методом Рунге-Кутта 3-го порядку. Для реалізації програми використано мову програмування Go.