Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

О Т Ч Е Т

по лабораторной работе № 0 1

Дисциплина: ***Моделирование***

Студент	***ИУ7И-66Б***		**Нгуен Ф. С.**
	(Группа)	(Подпись, дата)	(И.О. Фамилия)
Преподаватель			**Градов В. М.**
		(Подпись, дата)	(И.О. Фамилия)

Москва, 2021

- **Цель работы:** Получение навыков решения задачи Коши для ОДУ методами Пикара и явными методами первого порядка точности (Эйлера) и второго порядка точности (РунгеКутта).
- **Исходные данные.**

$$\begin{cases} u'(x) = x^2 + u^2 \\ u(0) = 0 \end{cases} \quad (1)$$

1. **Метод Пикара**

$$y_s(x) = u_0 + \int_{x_0}^{x} f(t, y_{s-1}(t))dt \qquad (2)$$

$$\boldsymbol{y_1(x) = 0 + \int_0^x f(t,0)dt = \int_0^x t^2 dt = \frac{x^3}{3}}$$

$$\boldsymbol{y_2(x) = 0 + \int_0^x f\left(t, \frac{x^3}{3}\right)dt = \int_0^x (t^2 + \frac{t^6}{9})dt = \frac{x^3}{3} + \frac{x^7}{63}}$$

$$\boldsymbol{y_3(x) = 0 + \int_0^x f\left(t, \frac{x^3}{3} + \frac{x^7}{63}\right)dt}$$

$$\boldsymbol{= \int_0^x \left(t^2 + \frac{t^6}{9} + \frac{2t^{10}}{189} + \frac{t^{14}}{3969}\right)dt = \frac{x^3}{3} + \frac{x^7}{63} + \frac{2x^{11}}{2079} + \frac{x^{15}}{59535}}$$

$$\boldsymbol{y_4(x) = 0 + \int_0^x f\left(t, \frac{x^3}{3} + \frac{x^7}{63} + \frac{2x^{11}}{2079} + \frac{x^{15}}{59535}\right)dt}$$

$$\boldsymbol{= 0 + \int_0^x [t^2 + \left(\frac{x^3}{3} + \frac{x^7}{63} + \frac{2x^{11}}{2079} + \frac{x^{15}}{59535}\right)^2]dt}$$

$$\boldsymbol{= \frac{x^3}{3} + \frac{x^7}{63} + \frac{2x^{11}}{2079} + \frac{x^{15}}{59535} + \frac{2x^{15}}{93555} + \frac{2x^{19}}{3393495} + \frac{2x^{19}}{2488563}}$$

$$\boldsymbol{+ \frac{2x^{23}}{86266215} + \frac{x^{23}}{99411543} + \frac{2x^{27}}{3341878155} + \frac{x^{31}}{109876902975}}$$

2. **Метод Эйлера:**

$$\boldsymbol{y_{n+1} = y_n + h * f(x_n, y_n)} \qquad (3)$$

$$\boldsymbol{y_{n+1} = y_n + h * (x_n^2 + y_n^2)}$$

3. **Метод РунгеКутта:**

$$\boldsymbol{y_{n+1} = y_n + h[(1-\alpha)k_1 + \alpha k_2]} \qquad (4)$$

$$\boldsymbol{k_1 = f(x_n, y_n)}$$

$$\boldsymbol{k_2 = f(x_n + \frac{h}{2\alpha}, y_n + \frac{h}{2\alpha}k_1)}$$

$$\boldsymbol{\alpha = 1} \text{ или } \boldsymbol{\frac{1}{2}}$$

4. Программа

```
# Эйлера
def euler(n, h, x, y):
    y_out = []
    for i in range(n):
        try:
            y += h * f(x, y)
            y_out.append(y)
            x += h
        except OverflowError:
            y_out.append('overflow')
            for j in range(i, n-1):
                y_out.append('-----')
            break
    return y_out

# РунгеКутта
def rungeKutta(n, h, x, y):
    y_out = [y]
    alpha = 1 / 2
    h1 = h / (2 * alpha)
    for i in range(n):
        try:
            k1 = f(x, y)
            k2 = f(x + h1, y + h1 * k1)
            y += h * ((1 - alpha) * k1 + alpha * k2)
            y_out.append(y)
            x += h
        except:
            y_out.append('overflow')
            for j in range(i, n-1):
                y_out.append('-----')
            break
    return y_out


# Пикар
def picar(n, h, x, y0):
    def f1(a):
        return a ** 3 / 3
    def f2(a):
        return f1(a) + a ** 7 / 63
    def f3(a):
        return f2(a) +  (a ** 11) * (2 / 2079) + (a ** 15) / 59535
    def f4(a):
        return f3(a) + (a ** 15)*(2 / 93555) + (a ** 19)*(2 / 3393495) + (a ** 19)*(2 / 2488563) + \
    (a ** 23)*(2 / 86266215) + (a ** 23)*(1 / 99411543) + (a ** 27)*(2 / 3341878155) + (a ** 31)*(1 /
109876902975)

    y_out = [[y0, y0, y0, y0]]
    for i in range(n-1):
        x += h
        y_f1 = f1(x)
        y_f2 = f2(x)
        y_f3 = f3(x)
        y_f4 = f4(x)
        y_out.append([y_f1, y_f2, y_f3, y_f4])
    return y_out


def main():
    h = 10 ** -5
    x = 0
    y0 = 0
    end = 2.1

    n = ceil(abs(end - x)/h)+1

```

```
    x_arr = [x + h*i for i in range(n)]
    y_picar = picar(n, h, x, y0)
    y_euler = euler(n, h, x, y0)
    y_RungeKutta = rungeKutta(n, h, x, y0)

    print("|    x    |    Пикара 1   |    Пикара 2   |    Пикара 3   |    Пикара 4   |     Эйлер     |
RungeKutt |")
    print("-"*75)
    output_step = int(n/h) # выводим только 100 значений в таблице
    for i in range(0, n, 10000):
        print("|{:^9.5f}|{:^15.8f}|{:^15.8f}|{:^15.8f}|{:^15.8f}|{:^15s}|{:^15s}|".format(x_arr[i],
y_picar[i][0], y_picar[i][1], y_picar[i][1], y_picar[i][1], output(y_euler[i]), output(y_RungeKutta[i])))


main()
```

5. Результат

x	Пикара 1	Пикара 2	Пикара 3	Пикара 4	Эйлер	РунгеКутт
0.00000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0
0.00001	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
0.00002	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
0.00003	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
0.00004	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
0.00005	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000

...

1.99982	2.66594673	4.69641311	4.69641311	4.69641311	296.38485376	300.53283740
1.99983	2.66598672	4.69652418	4.69652418	4.69652418	297.26333357	301.43879587
1.99984	2.66602672	4.69663524	4.69663524	4.69663524	298.14702846	302.35023262
1.99985	2.66606671	4.69674632	4.69674632	4.69674632	299.03598496	303.26719750
1.99986	2.66610671	4.69685739	4.69685739	4.69685739	299.93025016	304.18974095
1.99987	2.66614670	4.69696847	4.69696847	4.69696847	300.82987170	305.11791405
1.99988	2.66618670	4.69707955	4.69707955	4.69707955	301.73489781	306.05176847
1.99989	2.66622669	4.69719063	4.69719063	4.69719063	302.64537730	306.99135654
1.99990	2.66626669	4.69730171	4.69730171	4.69730171	303.56135954	307.93673123
1.99991	2.66630668	4.69741280	4.69741280	4.69741280	304.48289452	308.88794616
1.99992	2.66634668	4.69752389	4.69752389	4.69752389	305.41003285	309.84505562
1.99993	2.66638668	4.69763498	4.69763498	4.69763498	306.34282573	310.80811457
1.99994	2.66642667	4.69774608	4.69774608	4.69774608	307.28132499	311.77717866
1.99995	2.66646667	4.69785717	4.69785717	4.69785717	308.22558312	312.75230422
1.99996	2.66650667	4.69796827	4.69796827	4.69796827	309.17565322	313.73354833
1.99997	2.66654667	4.69807938	4.69807938	4.69807938	310.13158906	314.72096873
1.99998	2.66658667	4.69819048	4.69819048	4.69819048	311.09344509	315.71462393
1.99999	2.66662667	4.69830159	4.69830159	4.69830159	312.06127640	316.71457317
2.00000	2.66666667	4.69841270	4.69841270	4.69841270	313.03513881	317.72087644
2.00001	2.66670667	4.69852381	4.69852381	4.69852381	314.01508879	318.73359450
2.00002	2.66674667	4.69863493	4.69863493	4.69863493	315.00118355	319.75278888
2.00003	2.66678667	4.69874604	4.69874604	4.69874604	315.99348101	320.77852191
2.00004	2.66682667	4.69885716	4.69885716	4.69885716	316.99203981	321.81085671
2.00005	2.66686667	4.69896829	4.69896829	4.69896829	317.99691934	322.84985722
2.00006	2.66690667	4.69907941	4.69907941	4.69907941	319.00817975	323.89558822
2.00007	2.66694668	4.69919054	4.69919054	4.69919054	320.02588194	324.94811531
2.00008	2.66698668	4.69930167	4.69930167	4.69930167	321.05008760	326.00750496
2.00009	2.66702668	4.69941280	4.69941280	4.69941280	322.08085919	327.07382451
2.00010	2.66706669	4.69952394	4.69952394	4.69952394	323.11825999	328.14714218
2.00011	2.66710669	4.69963507	4.69963507	4.69963507	324.16235409	329.22752709
2.00012	2.66714670	4.69974621	4.69974621	4.69974621	325.21320642	330.31504927
2.00013	2.66718670	4.69985736	4.69985736	4.69985736	326.27088272	331.40977968
2.00014	2.66722671	4.69996850	4.69996850	4.69996850	327.33544961	332.51179023
2.00015	2.66726671	4.70007965	4.70007965	4.70007965	328.40697458	333.62115377
2.00016	2.66730672	4.70019080	4.70019080	4.70019080	329.48552600	334.73794415
2.00017	2.66734672	4.70030195	4.70030195	4.70030195	330.57117313	335.86223621
2.00018	2.66738673	4.70041311	4.70041311	4.70041311	331.66398614	336.99410577
2.00019	2.66742674	4.70052427	4.70052427	4.70052427	332.76403614	338.13362971
2.00020	2.66746675	4.70063543	4.70063543	4.70063543	333.87139519	339.28088593
2.00021	2.66750675	4.70074659	4.70074659	4.70074659	334.98613628	340.43595341

...

6. **Ответы на вопросы:**
   1) ***Укажите интервалы значений аргумента, в которых можно считать решением заданного уравнения каждое из первых 4-х приближений Пикара. Точность результата оценивать до второй цифры после запятой. Объяснить свой ответ.***
      - Разница меньше 0.005 в интервал (0, 0.85):

$$\frac{x^7}{63} + \frac{2x^{11}}{2079} + \frac{x^{15}}{59535} + \cdots < 0.005$$

        ⇨ В интервал (0, 0.85) можно считать решением заданного уравнения каждое из первых 4-х приближений Пикара.
   2) ***Пояснить, каким образом можно доказать правильность полученного результата при фиксированном значении аргумента в численных методах.***
      - Численные методы тем точнее, чем меньше шаг
      - Мы сравниваем значения Эйлера при разном шаге (x = 2):
        - При шаге = 10^{-4} , получаем ≈ 270
        - При шаге = 10^{-5}, получаем ≈ 313
          ⇨ *Погрешность при шаге 10-4 была большая.*
        - Далее, при шаге = 10^{-6}, получаем ≈ 317.
        - При шаге = 10^{-7}, получаем ≈ 317.
          ⇨ *Можно считать, что результат около 317*
   3) ***Каково значение функции при x=2, т.е. Привести значение u(2).***
      - U(2) ≈ 317