



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»

# Отчёт

## по лабораторной работе № 1

**Название:** Программная реализация приближенного  
аналитического метода и численных алгоритмов первого  
и второго порядков точности при решении задачи Коши для ОДУ.

**Дисциплина:** Моделирование

Студент

ИУ7-65Б

(Группа)

Д.В. Сусликов

(Подпись, дата)

(И.О. Фамилия)

Преподаватель

В.М. Градов

(Подпись, дата)

(И.О. Фамилия)

Москва, 2021

## Введение

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

**Исходные данные:**

ОДУ, не имеющее аналитического решения

$$u'(x) = x^2 + u^2,$$

$$u(0) = 0$$

## Результат работы программы

Таблица, содержащая значения аргумента с заданным шагом в интервале  $[0, x_{\max}]$  и результаты расчета функции  $u(x)$  в приближениях Пикара (от 1-го до 4-го), а также численными методами. Границу интервала  $x_{\max}$  выбирать максимально возможной из условия, чтобы численные методы обеспечивали точность вычисления решения уравнения  $u(x)$  до второго знака после запятой.

### 1. Метод Пикара

$$y^{(1)} = \frac{x^3}{3}$$

$$y^{(2)} = \frac{x^3}{3} + \frac{x^7}{63}$$

$$y^{(3)} = \frac{x^3}{3} + \frac{x^7}{63} + \frac{2x^{11}}{2079} + \frac{x^{15}}{59535}$$

$$y^{(4)} = \frac{x^3}{3} + \frac{x^7}{63} + \frac{2x^{11}}{2079} + \frac{13x^{15}}{218295} + \frac{82x^{19}}{37328445} + \frac{662x^{23}}{10438212015} + \frac{4x^{27}}{3341878155} + \frac{x^{31}}{109876902975}$$

### 2. Метод Эйлера

$$y_{n+1} = y_n + h \cdot f(x_n, y_n), \text{ где } f(x, y) = x^2 + y^2$$

### 3. Метод Рунге-Кутта

$$y_{n+1} = y_n + h \cdot ((1 - \alpha)k_1 + \alpha k_2),$$

$$k_1 = f(x_n, y_n)$$

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

$$\alpha = \frac{1}{2} \text{ or } 1$$

#### Результаты

Ниже приведены результаты расчёта функции  $u(x)$  с шагом  $10^{-5}$  и значением  $\alpha = 1$  в методе Рунге-Кутта.

x	Пикара 1	Пикара 2	Пикара 3	Пикара 4	Эйлера	Рунге
0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.05	0.00	0.00	0.00	0.00	0.00	0.00
0.10	0.00	0.00	0.00	0.00	0.00	0.00
0.15	0.00	0.00	0.00	0.00	0.00	0.00
0.20	0.00	0.00	0.00	0.00	0.00	0.00
0.25	0.01	0.01	0.01	0.01	0.01	0.01
0.30	0.01	0.01	0.01	0.01	0.01	0.01
0.35	0.01	0.01	0.01	0.01	0.01	0.01
0.40	0.02	0.02	0.02	0.02	0.02	0.02
0.45	0.03	0.03	0.03	0.03	0.03	0.03
0.50	0.04	0.04	0.04	0.04	0.04	0.04
0.55	0.06	0.06	0.06	0.06	0.06	0.06
0.60	0.07	0.07	0.07	0.07	0.07	0.07
0.65	0.09	0.09	0.09	0.09	0.09	0.09
0.70	0.11	0.12	0.12	0.12	0.12	0.12
0.75	0.14	0.14	0.14	0.14	0.14	0.14
0.80	0.17	0.17	0.17	0.17	0.17	0.17
0.85	0.20	0.21	0.21	0.21	0.21	0.21
0.90	0.24	0.25	0.25	0.25	0.25	0.25
0.95	0.29	0.30	0.30	0.30	0.30	0.30
1.00	0.33	0.35	0.35	0.35	0.35	0.35
1.05	0.39	0.41	0.41	0.41	0.41	0.41
1.10	0.44	0.47	0.48	0.48	0.48	0.48
1.15	0.51	0.55	0.55	0.55	0.55	0.55
1.20	0.58	0.63	0.64	0.64	0.64	0.64
1.25	0.65	0.73	0.74	0.74	0.74	0.74
1.30	0.73	0.83	0.85	0.85	0.85	0.85
1.35	0.82	0.95	0.98	0.98	0.98	0.98
1.40	0.91	1.08	1.12	1.13	1.13	1.13
1.45	1.02	1.23	1.29	1.30	1.31	1.31
1.50	1.13	1.40	1.49	1.50	1.52	1.52
1.55	1.24	1.58	1.71	1.74	1.77	1.77
1.60	1.37	1.79	1.98	2.02	2.08	2.08
1.65	1.50	2.03	2.29	2.36	2.47	2.47
1.70	1.64	2.29	2.67	2.77	2.97	2.97
1.75	1.79	2.58	3.11	3.28	3.67	3.67
1.80	1.94	2.92	3.65	3.92	4.69	4.69
1.85	2.11	3.29	4.29	4.74	6.35	6.35
1.90	2.29	3.71	5.08	5.79	9.57	9.57
1.95	2.47	4.17	6.04	7.18	18.74	18.75
2.00	2.67	4.70	7.22	9.03	313.04	318.73
2.05	2.87	5.29	8.67	11.54	null	null
2.10	3.09	5.95	10.46	15.01	null	null
2.15	3.31	6.68	12.68	19.86	null	null
2.20	3.55	7.51	15.43	26.75	null	null
2.25	3.80	8.43	18.85	36.67	null	null

## Ответы на вопросы

- 1) **Вопрос:** 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.

**Ответ:** При оценке значений до 2-й цифры после запятой общим интервалом, где совпадают значения у всех приближений Пикара, будет  $[0, 65]$ . Если оценивать по отдельности, то для 1-го -  $[0, 65]$ , для 2-го -  $[0, 1.05]$ , для 3-го -  $[0, 1.4]$ , для 4-го -  $[0, 1.4]$ .

Интервалы для 3-его и 4-ого приближения одинаковые из-за того, что нельзя гарантировать, что 4-ое приближение вычисляется правильно, так как для этого нужно приближение более высокого порядка.

Интервалы разнятся, так как чем больше приближение, тем точнее результат.

- 2) **Вопрос:** Пояснить, каким образом можно доказать правильность полученного результата при фиксированном значении аргумента в численных методах.

**Ответ:** Численные методы работают пошагово, поэтому точность полученного результата зависит от величины шага. То есть, чтобы доказать правильность результата нужно вычислять значения сперва с большим шагом, постепенно уменьшая его, до тех пор, пока изменение результата будет пренебрежительно мало. Итоговый результат можно считать верным. Приведу пример вычисления значения  $u(2)$ :

шаг	Метод Эйлера	Метод Рунге
$10^{-2}$	36.49	575.95
$10^{-3}$	142.63	420.29
$10^{-4}$	277/36	327.89
$10^{-5}$	313.04	318.73
$10^{-6}$	317.25	317.82

3) **Вопрос:** Каково значение функции при  $x=2$ , т.е. привести значение  $u(2)$ .

**Ответ:**

х	Метод Эйлера	Метод Рунге
2.00	317.25	317.82

**Код программы**

#### Листинг 1 – Метод Эйлера

```

1  def euler(h, x):
2      y = 0
3      x0 = h
4      while (x0 < x + h / 2):
5          try:
6              y += h * func(x0, y)
7              x0 += h
8          except:
9              return 'null'
10
11     return y

```

#### Листинг 2 – Метод Рунге-Кутта

```

1  def runge(h, x):
2      alpha = 1
3      y = 0

```

```

4      x0 = h
5      while (x0 < x + h / 2):
6          try:
7              k1 = func(x0, y)
8              k2 = func(x0 + h / 2 / alpha, y + h / 2 / alpha * k1)
9              y += h * ((1 - alpha) * k1 + alpha * k2)
10             x0 += h
11         except:
12             return 'null'
13
14     return y

```

### Листинг 3 – Метод Пикара

```

1  def approximation1(arg):
2      return arg ** 3 / 3
3
4  def approximation2(arg):
5      return approximation1(arg) + arg ** 7 / 63
6
7  def approximation3(arg):
8      return approximation2(arg) + (arg ** 11) * (2 / 2079) + (arg
9      ** 15) / 59535
10
11 def approximation4(arg):
12     return approximation3(arg) + (arg ** 15)*(2 / 93555) + (arg **
13     19)*(2 / 3393495) + \
14     (arg ** 19)*(2 / 2488563) + (arg ** 23)*(2 / 86266215) + (arg
15     ** 23)*(1 / 99411543) + \
16     (arg ** 27)*(2 / 3341878155) + (arg ** 31)*(1 / 109876902975)
17
18 def picar(x):
19     y_approx1 = approximation1(x)
20     y_approx2 = approximation2(x)
21     y_approx3 = approximation3(x)
22     y_approx4 = approximation4(x)

```

20

21

```
return [y_approx1 , y_approx2 , y_approx3 , y_approx4]
```