



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»

Отчет по лабораторной работе №1 по дисциплине «Моделирование»

Тема Программная реализация приближенного аналитического метода и
численных алгоритмов первого и второго порядков точности при
решении задачи Коши для ОДУ

Студент Зайцева А. А.

Группа ИУ7-62Б

Оценка (баллы)

Преподаватель Градов В. М.

1 | Задание

Цель работы.

Получение навыков решения задачи Коши для ОДУ методами Пикара и явными методами первого порядка точности (Эйлера) и второго порядка точности (Рунге-Кутты).

Исходные данные.

ОДУ 1.1, не имеющее аналитического решения:

$$\begin{cases} u'(x) = u^2 + x^2 \\ u(0) = 0. \end{cases} \quad (1.1)$$

Результат работы программы.

1. 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.

2. График функции в диапазоне $[-x_{\max}, x_{\max}]$.

2 | Теоретическая часть

Обыкновенное дифференциальное уравнение (ОДУ) n -ого порядка имеет вид 2.1:

$$F(x, u', u'', \dots, u^{(n)}) = 0. \quad (2.1)$$

Задача Коши состоит в нахождении решения дифференциального уравнения, удовлетворяющего начальным условиям 2.2:

$$\begin{cases} u'(x) = f(x, u) \\ u(\xi) = \eta \end{cases} \quad (2.2)$$

Рассмотрим методы решения этой задачи.

2.1 Метод Пикара

Метод Пикара является приближенно-аналитическим. Идея состоит в том, чтобы заменить дифференциальное уравнение интегральным 2.3.:

$$y^s(x) = \eta + \int_{\xi}^x f(t, y^{s-1}(t)) dt \quad (2.3)$$

$$y^{(0)} = \eta \quad (2.4)$$

Для данного в задании ОДУ 1.1:

$$y^{(1)} = 0 + \int_0^x t^2 dt = \frac{x^3}{3} \quad (2.5)$$

$$y^{(2)} = 0 + \int_0^x \left[\left(t^2 + \frac{t^3}{3} \right)^2 \right] dt = \frac{x^3}{3} + \frac{x^7}{63} \quad (2.6)$$

$$y^{(3)} = 0 + \int_0^x \left[t^2 + \left(\frac{t^7}{63} + \frac{t^3}{3} \right)^2 \right] dt = \frac{x^3}{3} + \frac{x^7}{63} + \frac{2x^{11}}{2079} + \frac{x^{15}}{59535} \quad (2.7)$$

$$\begin{aligned}
y^{(4)} &= 0 + \int_0^x \left[t^2 + \left(\frac{t^3}{3} + \frac{t^7}{63} + \frac{2t^{11}}{2079} + \frac{t^{15}}{59535} \right)^2 \right] dt = \\
&= \frac{x^3}{3} + \frac{x^7}{63} + \frac{2x^{11}}{2079} + \frac{13x^{15}}{218295} + \frac{82x^{19}}{37328445} + \\
&\quad + \frac{662x^{23}}{10438212015} + \frac{4x^{27}}{3341878155} + \frac{x^{31}}{109876902975}
\end{aligned} \tag{2.8}$$

2.2 Метод Эйлера

Метод Эйлера – это явный (численный) метод первого порядка точности, использующий формулу 2.9:

$$y_{n+1} = y_n + h * f(x_n, y_n). \tag{2.9}$$

2.3 Метод Рунге-Кутты

Метод Рунге-Кутты – это явный (численный) метод второго порядка точности, использующий формулу 2.10:

$$y_{n+1} = y_n + h * [(1 - \alpha)k_1 + \alpha * k_2], \tag{2.10}$$

где $k_1 = f(x_n, y_n)$, $k_2 = f(x_n + \frac{h}{2\alpha}, y_n + \frac{h}{2\alpha}k_1)$, $\alpha = 1$ или $\frac{1}{2}$

3 | Исходный код алгоритмов

```
1 class UDESolver:
2     def __init__(self, x_start, y_start, x_max, step, f, f_derivatives):
3         :
4         if (x_start < x_max) != (step > 0):
5             raise ValueError('Ошибка вшаге ')
6
7     self.x_start = x_start
8     self.y_start = y_start
9     self.x_max = x_max
10    self.step = step
11    self.f = f
12    self.f_derivativesf_derivatives = f_derivativesf_derivatives
13    self.cmp_func = lambda x1, x2: x1 < x2 + EPS
14
15    def reverse_move(self):
16        self.step *= -1
17        self.x_max *= -1
18        self.cmp_func = lambda x1, x2: x1 > x2 - EPS
19
20    def x_range(self):
21        result = []
22        x = self.x_start
23        while self.cmp_func(x, self.x_max):
24            result.append(x)
25            x += self.step
26        return result
27
28    def solve_euler(self):
29        result = []
30        x, y = self.x_start, self.y_start
31
32        while self.cmp_func(x, self.x_max):
33            result.append(y)
34
35            y = y + self.step * self.f(x, y)
36            x += self.step
37
38        return result
39
40    def solve_runge_kutta(self):
41        a = 0.5
42        result = []
43        x, y = self.x_start, self.y_start
44
45        while self.cmp_func(x, self.x_max):
```

```

45         result.append(y)
46
47         k1 = self.f(x, y)
48         k2 = self.f(x + self.step / (2 * a), y + self.step * k1 / (2 *
49             a))
50         y += self.step * ((1 - a) * k1 + a * k2)
51         x += self.step
52
53     return result
54
55 def solve_picar(self, approx):
56     func = self.f_derivatives[approx - 1]
57     result = []
58     x, y = self.x_start, self.y_start
59
60     while self.cmp_func(x, self.x_max):
61         result.append(y)
62         x += self.step
63         y = func(x)
64
65     return result
66
67 def function(x, u):
68     return x * x + u * u
69
70 def fd1(x):
71     return pow(x, 3) / 3
72
73
74 def fd2(x):
75     return fd1(x) + pow(x, 7) / 63
76
77
78 def fd3(x):
79     return fd2(x) + 2 * pow(x, 11) / 2079 + pow(x, 15) / 59535
80
81
82 def fd4(x):
83     return (fd2(x) + 2 * pow(x, 11) / 2079 + 13 * pow(x, 15) / 218295 +
84         82 * pow(x, 19) / 37328445 + 662 * pow(x, 23) / 10438212015 +
85         4 * pow(x, 27) / 3341878155 + pow(x, 31) / 109876902975)

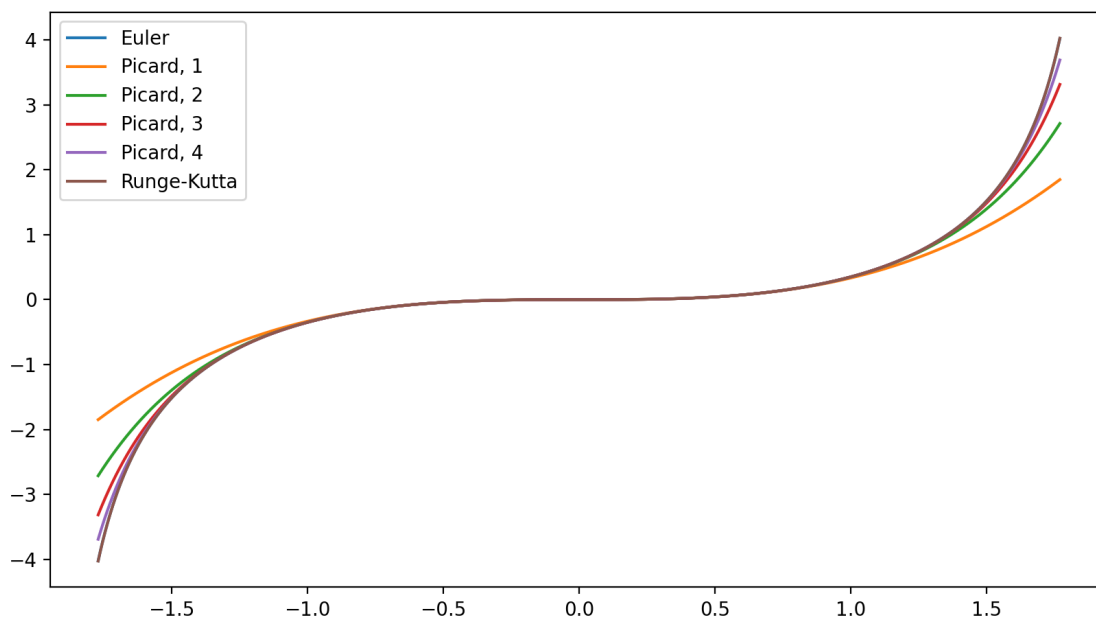
```

4 | Результат работы программы

МЫ

Исходные данные: $h=10^{-4}$, $x_{\max} = 1.77$, $x_0=0$, $y_0 = 0$, округление при выводе - до 2 знака после запятой, шаг вывода=0.01.

На рисунке 4 приведен график функции в диапазоне $[-x_{\max}; x_{\max}]$.



Ниже приведена таблица с полученными данными.

1	x	Euler	Runge—Kutta	Picard , 1	Picard , 2	Picard , 3	Picard , 4
2							
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.01	0.00	0.00	0.00	0.00	0.00	0.00
5	0.02	0.00	0.00	0.00	0.00	0.00	0.00
6	0.03	0.00	0.00	0.00	0.00	0.00	0.00
7	0.04	0.00	0.00	0.00	0.00	0.00	0.00
8	0.05	0.00	0.00	0.00	0.00	0.00	0.00
9	0.06	0.00	0.00	0.00	0.00	0.00	0.00
10	0.07	0.00	0.00	0.00	0.00	0.00	0.00
11	0.08	0.00	0.00	0.00	0.00	0.00	0.00
12	0.09	0.00	0.00	0.00	0.00	0.00	0.00
13	0.10	0.00	0.00	0.00	0.00	0.00	0.00
14	0.11	0.00	0.00	0.00	0.00	0.00	0.00
15	0.12	0.00	0.00	0.00	0.00	0.00	0.00
16	0.13	0.00	0.00	0.00	0.00	0.00	0.00
17	0.14	0.00	0.00	0.00	0.00	0.00	0.00
18	0.15	0.00	0.00	0.00	0.00	0.00	0.00
19	0.16	0.00	0.00	0.00	0.00	0.00	0.00

20	0.17	0.00	0.00	0.00	0.00	0.00	0.00
21	0.18	0.00	0.00	0.00	0.00	0.00	0.00
22	0.19	0.00	0.00	0.00	0.00	0.00	0.00
23	0.20	0.00	0.00	0.00	0.00	0.00	0.00
24	0.21	0.00	0.00	0.00	0.00	0.00	0.00
25	0.22	0.00	0.00	0.00	0.00	0.00	0.00
26	0.23	0.00	0.00	0.00	0.00	0.00	0.00
27	0.24	0.00	0.00	0.00	0.00	0.00	0.00
28	0.25	0.01	0.01	0.01	0.01	0.01	0.01
29	0.26	0.01	0.01	0.01	0.01	0.01	0.01
30	0.27	0.01	0.01	0.01	0.01	0.01	0.01
31	0.28	0.01	0.01	0.01	0.01	0.01	0.01
32	0.29	0.01	0.01	0.01	0.01	0.01	0.01
33	0.30	0.01	0.01	0.01	0.01	0.01	0.01
34	0.31	0.01	0.01	0.01	0.01	0.01	0.01
35	0.32	0.01	0.01	0.01	0.01	0.01	0.01
36	0.33	0.01	0.01	0.01	0.01	0.01	0.01
37	0.34	0.01	0.01	0.01	0.01	0.01	0.01
38	0.35	0.01	0.01	0.01	0.01	0.01	0.01
39	0.36	0.02	0.02	0.02	0.02	0.02	0.02
40	0.37	0.02	0.02	0.02	0.02	0.02	0.02
41	0.38	0.02	0.02	0.02	0.02	0.02	0.02
42	0.39	0.02	0.02	0.02	0.02	0.02	0.02
43	0.40	0.02	0.02	0.02	0.02	0.02	0.02
44	0.41	0.02	0.02	0.02	0.02	0.02	0.02
45	0.42	0.02	0.02	0.02	0.02	0.02	0.02
46	0.43	0.03	0.03	0.03	0.03	0.03	0.03
47	0.44	0.03	0.03	0.03	0.03	0.03	0.03
48	0.45	0.03	0.03	0.03	0.03	0.03	0.03
49	0.46	0.03	0.03	0.03	0.03	0.03	0.03
50	0.47	0.03	0.03	0.03	0.03	0.03	0.03
51	0.48	0.04	0.04	0.04	0.04	0.04	0.04
52	0.49	0.04	0.04	0.04	0.04	0.04	0.04
53	0.50	0.04	0.04	0.04	0.04	0.04	0.04
54	0.51	0.04	0.04	0.04	0.04	0.04	0.04
55	0.52	0.05	0.05	0.05	0.05	0.05	0.05
56	0.53	0.05	0.05	0.05	0.05	0.05	0.05
57	0.54	0.05	0.05	0.05	0.05	0.05	0.05
58	0.55	0.06	0.06	0.06	0.06	0.06	0.06
59	0.56	0.06	0.06	0.06	0.06	0.06	0.06
60	0.57	0.06	0.06	0.06	0.06	0.06	0.06
61	0.58	0.07	0.07	0.07	0.07	0.07	0.07
62	0.59	0.07	0.07	0.07	0.07	0.07	0.07
63	0.60	0.07	0.07	0.07	0.07	0.07	0.07
64	0.61	0.08	0.08	0.08	0.08	0.08	0.08
65	0.62	0.08	0.08	0.08	0.08	0.08	0.08
66	0.63	0.08	0.08	0.08	0.08	0.08	0.08
67	0.64	0.09	0.09	0.09	0.09	0.09	0.09

68	0.65	0.09	0.09	0.09	0.09	0.09	0.09
69	0.66	0.10	0.10	0.10	0.10	0.10	0.10
70	0.67	0.10	0.10	0.10	0.10	0.10	0.10
71	0.68	0.11	0.11	0.10	0.11	0.11	0.11
72	0.69	0.11	0.11	0.11	0.11	0.11	0.11
73	0.70	0.12	0.12	0.11	0.12	0.12	0.12
74	0.71	0.12	0.12	0.12	0.12	0.12	0.12
75	0.72	0.13	0.13	0.12	0.13	0.13	0.13
76	0.73	0.13	0.13	0.13	0.13	0.13	0.13
77	0.74	0.14	0.14	0.14	0.14	0.14	0.14
78	0.75	0.14	0.14	0.14	0.14	0.14	0.14
79	0.76	0.15	0.15	0.15	0.15	0.15	0.15
80	0.77	0.15	0.15	0.15	0.15	0.15	0.15
81	0.78	0.16	0.16	0.16	0.16	0.16	0.16
82	0.79	0.17	0.17	0.16	0.17	0.17	0.17
83	0.80	0.17	0.17	0.17	0.17	0.17	0.17
84	0.81	0.18	0.18	0.18	0.18	0.18	0.18
85	0.82	0.19	0.19	0.18	0.19	0.19	0.19
86	0.83	0.19	0.20	0.19	0.19	0.20	0.20
87	0.84	0.20	0.20	0.20	0.20	0.20	0.20
88	0.85	0.21	0.21	0.20	0.21	0.21	0.21
89	0.86	0.22	0.22	0.21	0.22	0.22	0.22
90	0.87	0.23	0.23	0.22	0.23	0.23	0.23
91	0.88	0.23	0.23	0.23	0.23	0.23	0.23
92	0.89	0.24	0.24	0.23	0.24	0.24	0.24
93	0.90	0.25	0.25	0.24	0.25	0.25	0.25
94	0.91	0.26	0.26	0.25	0.26	0.26	0.26
95	0.92	0.27	0.27	0.26	0.27	0.27	0.27
96	0.93	0.28	0.28	0.27	0.28	0.28	0.28
97	0.94	0.29	0.29	0.28	0.29	0.29	0.29
98	0.95	0.30	0.30	0.29	0.30	0.30	0.30
99	0.96	0.31	0.31	0.29	0.31	0.31	0.31
100	0.97	0.32	0.32	0.30	0.32	0.32	0.32
101	0.98	0.33	0.33	0.31	0.33	0.33	0.33
102	0.99	0.34	0.34	0.32	0.34	0.34	0.34
103	1.00	0.35	0.35	0.33	0.35	0.35	0.35
104	1.01	0.36	0.36	0.34	0.36	0.36	0.36
105	1.02	0.37	0.37	0.35	0.37	0.37	0.37
106	1.03	0.39	0.39	0.36	0.38	0.39	0.39
107	1.04	0.40	0.40	0.37	0.40	0.40	0.40
108	1.05	0.41	0.41	0.39	0.41	0.41	0.41
109	1.06	0.42	0.42	0.40	0.42	0.42	0.42
110	1.07	0.44	0.44	0.41	0.43	0.44	0.44
111	1.08	0.45	0.45	0.42	0.45	0.45	0.45
112	1.09	0.46	0.46	0.43	0.46	0.46	0.46
113	1.10	0.48	0.48	0.44	0.47	0.48	0.48
114	1.11	0.49	0.49	0.46	0.49	0.49	0.49
115	1.12	0.51	0.51	0.47	0.50	0.51	0.51

116	1.13	0.52	0.52	0.48	0.52	0.52	0.52
117	1.14	0.54	0.54	0.49	0.53	0.54	0.54
118	1.15	0.55	0.55	0.51	0.55	0.55	0.55
119	1.16	0.57	0.57	0.52	0.57	0.57	0.57
120	1.17	0.59	0.59	0.53	0.58	0.59	0.59
121	1.18	0.60	0.60	0.55	0.60	0.60	0.60
122	1.19	0.62	0.62	0.56	0.62	0.62	0.62
123	1.20	0.64	0.64	0.58	0.63	0.64	0.64
124	1.21	0.66	0.66	0.59	0.65	0.66	0.66
125	1.22	0.68	0.68	0.61	0.67	0.68	0.68
126	1.23	0.70	0.70	0.62	0.69	0.70	0.70
127	1.24	0.72	0.72	0.64	0.71	0.72	0.72
128	1.25	0.74	0.74	0.65	0.73	0.74	0.74
129	1.26	0.76	0.76	0.67	0.75	0.76	0.76
130	1.27	0.78	0.78	0.68	0.77	0.78	0.78
131	1.28	0.81	0.81	0.70	0.79	0.80	0.81
132	1.29	0.83	0.83	0.72	0.81	0.83	0.83
133	1.30	0.85	0.85	0.73	0.83	0.85	0.85
134	1.31	0.88	0.88	0.75	0.85	0.87	0.88
135	1.32	0.90	0.90	0.77	0.88	0.90	0.90
136	1.33	0.93	0.93	0.78	0.90	0.92	0.93
137	1.34	0.96	0.96	0.80	0.93	0.95	0.95
138	1.35	0.98	0.98	0.82	0.95	0.98	0.98
139	1.36	1.01	1.01	0.84	0.98	1.01	1.01
140	1.37	1.04	1.04	0.86	1.00	1.03	1.04
141	1.38	1.07	1.07	0.88	1.03	1.06	1.07
142	1.39	1.10	1.10	0.90	1.05	1.09	1.10
143	1.40	1.13	1.13	0.91	1.08	1.12	1.13
144	1.41	1.17	1.17	0.93	1.11	1.16	1.16
145	1.42	1.20	1.20	0.95	1.14	1.19	1.20
146	1.43	1.23	1.24	0.97	1.17	1.22	1.23
147	1.44	1.27	1.27	1.00	1.20	1.26	1.27
148	1.45	1.31	1.31	1.02	1.23	1.29	1.31
149	1.46	1.35	1.35	1.04	1.26	1.33	1.34
150	1.47	1.39	1.39	1.06	1.29	1.37	1.38
151	1.48	1.43	1.43	1.08	1.33	1.41	1.42
152	1.49	1.47	1.47	1.10	1.36	1.45	1.47
153	1.50	1.52	1.52	1.12	1.40	1.49	1.51
154	1.51	1.56	1.56	1.15	1.43	1.53	1.56
155	1.52	1.61	1.61	1.17	1.47	1.57	1.60
156	1.53	1.66	1.66	1.19	1.51	1.62	1.65
157	1.54	1.71	1.71	1.22	1.54	1.67	1.70
158	1.55	1.77	1.77	1.24	1.58	1.71	1.76
159	1.56	1.82	1.82	1.27	1.62	1.76	1.81
160	1.57	1.88	1.88	1.29	1.66	1.82	1.87
161	1.58	1.94	1.95	1.31	1.70	1.87	1.92
162	1.59	2.01	2.01	1.34	1.75	1.92	1.99
163	1.60	2.08	2.08	1.37	1.79	1.98	2.05

164	1.61	2.15	2.15	1.39	1.84	2.04	2.12
165	1.62	2.22	2.22	1.42	1.88	2.10	2.18
166	1.63	2.30	2.30	1.44	1.93	2.16	2.26
167	1.64	2.38	2.38	1.47	1.98	2.23	2.33
168	1.65	2.46	2.47	1.50	2.03	2.29	2.41
169	1.66	2.55	2.56	1.52	2.08	2.36	2.49
170	1.67	2.65	2.65	1.55	2.13	2.44	2.58
171	1.68	2.75	2.75	1.58	2.18	2.51	2.67
172	1.69	2.86	2.86	1.61	2.23	2.59	2.76
173	1.70	2.97	2.97	1.64	2.29	2.67	2.86
174	1.71	3.09	3.09	1.67	2.35	2.75	2.96
175	1.72	3.22	3.22	1.70	2.40	2.84	3.07
176	1.73	3.36	3.36	1.73	2.46	2.92	3.18
177	1.74	3.51	3.51	1.76	2.52	3.02	3.30
178	1.75	3.67	3.67	1.79	2.58	3.11	3.42
179	1.76	3.84	3.84	1.82	2.65	3.21	3.55
180	1.77	4.02	4.03	1.85	2.71	3.31	3.69

На рисунке 4 приведен вывод программы при шагах $1e-4$ (слева) и $1e-5$ (справа). Порядок колонок сохранен. Нас интересуют первые 3 (справа и слева) – значение аргумента, результат, вычисленный методом Эйлера и результат, вычисленный методом Рунге-Кутты.

1.66	2.55	2.56	1.52	2.08	2.36	2.49	170	170	1.66	2.56	2.56	1.52	2.08	2.36	2.49
1.67	2.65	2.65	1.55	2.13	2.44	2.58	171	171	1.67	2.65	2.65	1.55	2.13	2.44	2.58
1.68	2.75	2.75	1.58	2.18	2.51	2.67	172	172	1.68	2.75	2.75	1.58	2.18	2.51	2.67
1.69	2.86	2.86	1.61	2.23	2.59	2.76	173	173	1.69	2.86	2.86	1.61	2.23	2.59	2.76
1.70	2.97	2.97	1.64	2.29	2.67	2.86	174	174	1.70	2.97	2.97	1.64	2.29	2.67	2.86
1.71	3.09	3.09	1.67	2.35	2.75	2.96	175	175	1.71	3.09	3.09	1.67	2.35	2.75	2.96
1.72	3.22	3.22	1.70	2.40	2.84	3.07	176	176	1.72	3.22	3.22	1.70	2.40	2.84	3.07
1.73	3.36	3.36	1.73	2.46	2.92	3.18	177	177	1.73	3.36	3.36	1.73	2.46	2.92	3.18
1.74	3.51	3.51	1.76	2.52	3.02	3.30	178	178	1.74	3.51	3.51	1.76	2.52	3.02	3.30
1.75	3.67	3.67	1.79	2.58	3.11	3.42	179	179	1.75	3.67	3.67	1.79	2.58	3.11	3.42
1.76	3.84	3.84	1.82	2.65	3.21	3.55	180	180	1.76	3.84	3.84	1.82	2.65	3.21	3.55
1.77	4.02	4.03	1.85	2.71	3.31	3.69	181	181	1.77	4.03	4.03	1.85	2.71	3.31	3.69
1.78	4.22	4.23	1.88	2.78	3.42	3.83	182	182	1.78	4.23	4.23	1.88	2.78	3.42	3.83
1.79	4.44	4.45	1.91	2.85	3.53	3.99	183	183	1.79	4.45	4.45	1.91	2.85	3.53	3.99
1.80	4.68	4.69	1.94	2.92	3.65	4.15	184	184	1.80	4.69	4.69	1.94	2.92	3.65	4.15
1.81	4.95	4.95	1.98	2.99	3.77	4.32	185	185	1.81	4.95	4.95	1.98	2.99	3.77	4.32
1.82	5.24	5.25	2.01	3.06	3.89	4.50	186	186	1.82	5.25	5.25	2.01	3.06	3.89	4.50
1.83	5.57	5.57	2.04	3.13	4.02	4.69	187	187	1.83	5.57	5.57	2.04	3.13	4.02	4.69
1.84	5.93	5.94	2.08	3.21	4.15	4.89	188	188	1.84	5.94	5.94	2.08	3.21	4.15	4.89
1.85	6.34	6.35	2.11	3.29	4.29	5.10	189	189	1.85	6.35	6.35	2.11	3.29	4.29	5.10
1.86	6.80	6.81	2.14	3.37	4.44	5.33	190	190	1.86	6.81	6.81	2.14	3.37	4.44	5.33
1.87	7.34	7.35	2.18	3.45	4.59	5.56	191	191	1.87	7.35	7.35	2.18	3.45	4.59	5.56
1.88	7.96	7.97	2.21	3.53	4.75	5.82	192	192	1.88	7.97	7.97	2.21	3.53	4.75	5.82

На рисунке видно, что результаты, полученные методом Эйлера на одних и тех же аргументах, но с разным шагом, начинают сильно отличаться при $x=1.77$, поэтому именно это значение и выбрано в качестве максимального.

5 | 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.

i -е приближение Пикара можно считать решением уравнения до тех пор, пока совпадают результаты для i -го и $(i+1)$ -го приближений до второго знака после запятой.

Основываясь на таблице результатов:

- 1-ое приближение можно считать решением на отрезке $[0, 0.84]$;
- 2-ое приближение можно считать решением на отрезке $[0, 1.06]$;
- 3-ое приближение можно считать решением на отрезке $[0, 1.37]$;
- для определения промежутка, на котором 4-ое приближение можно считать решением, необходимо использовать 5-ое приближение. При его расчете происходит ошибка `OverflowError: int too large to convert to float: second_mult += (chisl / znam)`

2. Пояснить, каким образом можно доказать правильность полученного результата при фиксированном значении аргумента в численных методах.

Путем уменьшения шага для того же метода. Если при меньшем шаге этим же методом получен примерно такой же результат, значит и при большем шаге был получен корректный ответ.

Рассмотрим, например, значение функции при $x=1$.

По Эйлера:

- при шаге= 10^{-1} $y=0.29$,
- при шаге= 10^{-2} $y=0.34$,
- при шаге= 10^{-3} $y=0.35$,
- при шаге= 10^{-4} $y=0.35$.

По Рунге-Кутта:

- при шаге= 10^{-1} $y=0.35$,

- при шаге $=10^{-2}$ $y=0.35$.

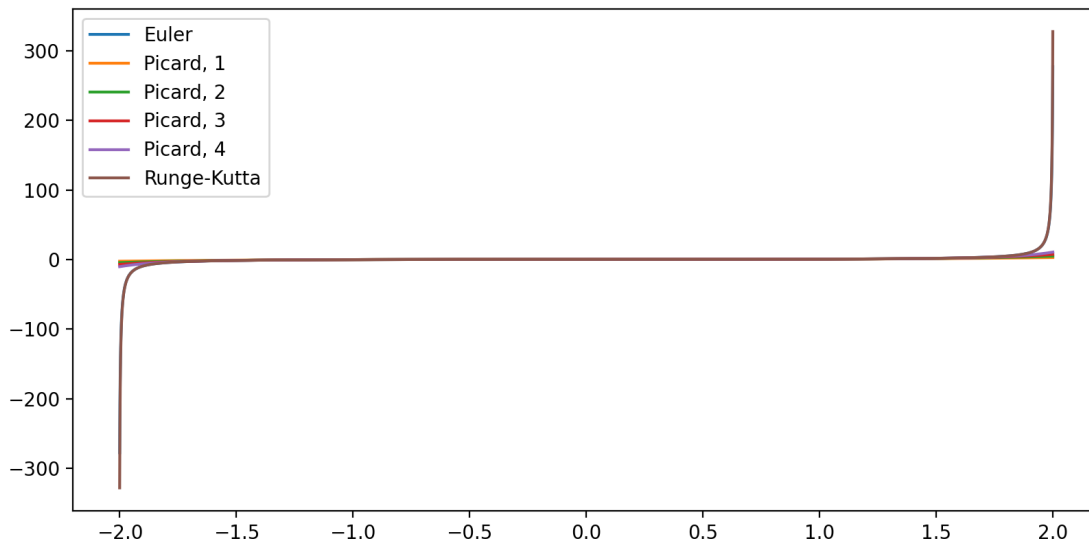
Таким образом, правильным ответом является $y=0.35$, причем при вычислении по Рунге-Кутта можно остановиться на шаге 10^{-1} , при вычислении по Эйлера – на шаге 10^{-3}

3. Каково значение решения уравнения в точке $x=2$, т.е. привести значение $u(2)$

$$u(2) \approx 317.72 \quad (5.1)$$

4. Дайте оценку точки разрыва решения уравнения.

На рисунке 5 приведен график функции в диапазоне $[-2.0001; 2.0001]$.



По графику видно, что $x=-2$ и $x=2$ являются точками разрыва (2 рода) решения уравнения, так как численные методы Эйлера и Рунге-Кутта асимптотически растут при приближении аргумента к этим значениям справа и слева, соответственно.

5. Покажите, что метод Пикара сходится к точному аналитическому решению уравнения:

$$5) \begin{cases} u' = x^2 + u \\ u(0) = 0 \end{cases}$$

Аналитическое решение:

$u' + a(x)u = b(x)$ - линейное уравнение 1-го порядка,

где $a(x) = -1$, $b(x) = x^2$

1) пусть $u = fg$; $u' = fg' + gf'$

$$f'g + gf' - fg = x^2$$

$$g'f + g(f' - f) = x^2$$

$$1.1) f' - f = 0$$

$$\frac{df}{f} = dx \Rightarrow f = e^x$$

$$1.2) g'f + g(f' - f) = x^2 \text{ при } f = e^x \text{ и } (f' - f) = 0$$

$$g'e^x = x^2$$

$$g = \int \frac{x^2}{e^x} dx = C - \frac{x^2 + 2x + 2}{e^x}$$

2) обратная замена

$$u = fg = e^x \left(C - \frac{x^2 + 2x + 2}{e^x} \right) = Ce^x - x^2 - 2x - 2$$

3) подставим нач. значения.

$$0 = Ce^0 - 0^2 - 2 \cdot 0 - 2 = C - 2 \Rightarrow C = 2$$

$$\text{Ответ: } u = 2e^x - x^2 - 2x - 2$$

Метод Тейлора

$$y^{(1)} = 0 + \int_0^x (t^2 + 0) dt = \frac{x^3}{3}$$

$$y^{(2)} = 0 + \int_0^x (t^2 + \frac{t^3}{3}) dt = \frac{t^3}{3} + \frac{t^4}{12}$$

$$y^{(3)} = 0 + \int_0^x (t^2 + \frac{t^3}{3} + \frac{t^4}{12}) dt = \frac{t^3}{3} + \frac{t^4}{12} + \frac{t^5}{60}$$

$$y^{(n)} = \frac{t^n}{n} + \frac{t^{n+1}}{n(n+1)} + \frac{t^{n+2}}{n(n+1)(n+2)} + \dots$$

[пог Тейлора $e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$]

$$y^{(n)} = 2(\frac{t^3}{3!} + \frac{t^4}{4!} + \frac{t^5}{5!} + \dots + \frac{t^{(n+2)}}{(n+2)!}) = 2(e^t - \frac{t^2}{2!} - \frac{t}{1!} - 1) = 2e^t - t^2 - 2t - 2$$