НИУ ИТМО
Факультет программной инженерии и компьютерных технологий

Отчет по лабораторной работе №6

по дисциплине Вычислительная математика

Студент группы № P32151 Шипулин Павел Андреевич
Преподаватель Машина Екатерина Алексеевна

Санкт-Петербург

2023

Цель работы

Решить задачу Коши для обыкновенных дифференциальных уравнений численными методами.

Алгоритм решения

Метод Эйлера.

$$y(x_i + h) = y(x_i) + y'(x_i) \cdot h \Rightarrow y'(x_i) = \frac{y(x_i + h) - y(x_i)}{h},$$

полагаем $y'(x_i) = f(x_i, y_i)$. Тогда

$$y(x_i + h) = y(x_i) + f(x_i) \cdot h$$

$$y_{i+1} = y_i + hf(x_i, y_i).$$

Функция $f(x, y)$ выражается аналитически.

Метод Рунге-Кутта 4-го порядка.

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = hf(x_i, y_i)$$

$$k_2 = hf\left(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right)$$

$$k_3 = hf\left(x_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right)$$

$$k_4 = hf(x_i + h, y_i + k_3)$$

Метод Милна.

1) этап прогноза:

$$y_i^{\text{прогн}} = y_{i-4} + \frac{4h}{3}(2f_{i-3} - f_{i-2} + 2f_{i-1})$$

2) этап коррекции:

$$y_i^{\text{корр}} = y_{i-2} + \frac{h}{3}\left(f_{i-2} + 4f_{i-1} + f_i^{\text{прогн}}\right)$$

$$f_i^{\text{прогн}} = f(x_i, y_i^{\text{прогн}})$$

Код численных методов

```
import math

from Labs.Lab6.data.table import Table


def runge_rule(y_h, y_half_h, p, epsilon):
    return abs(y_h - y_half_h) < epsilon * (2 ** p - 1)


def euler(f=lambda x, y: 0, x_0=0, y_0=0, x_n=0, h=0, epsilon=0):
    result = Table(head=["Номер шага", "h", "half_h", "y_h", "y_half_h",
"R"])

    if x_0 > x_n:
        raise Exception("x_0 > x_n???")

    y_h = 0
    y_half_h = math.inf
    iterations = 0
    y_values = []

    while not runge_rule(y_h, y_half_h, 1, epsilon):
        iterations += 1

        x = x_0
        y = y_0

        y_values = [y]

        while x < x_n:
            y = y + h * f(x, y)
            x = x + h

            y_values.append(y)

        y_h = y_half_h
        y_half_h = y

        result.add_row([iterations, h * 2, h, y_h, y_half_h, abs(y_h -
y_half_h)])

        h /= 2

    return result, y_values


def runge_kutta(f=lambda x, y: 0, x_0=0, y_0=0, x_n=0, h=0, epsilon=0):
    result = Table(head=["Номер шага", "h", "half_h", "y_h", "y_half_h",
"R"])
```

```
    if x_0 > x_n:
        raise Exception("x_0 > x_n???")

    y_h = 0
    y_half_h = math.inf
    iterations = 0
    y_values = []

    while not runge_rule(y_h, y_half_h, 4, epsilon):
        iterations += 1

        x = x_0
        y = y_0
        y_values = [y]

        while x < x_n:
            k1 = h * f(x, y)
            k2 = h * f(x + h / 2, y + k1 / 2)
            k3 = h * f(x + h / 2, y + k2 / 2)
            k4 = h * f(x + h, y + k3)
            y = y + 1 / 6 * (k1 + 2 * k2 + 2 * k3 + k4)
            x = x + h

            y_values.append(y)

        y_h = y_half_h
        y_half_h = y

        result.add_row([iterations, h * 2, h, y_h, y_half_h, abs(y_h -
y_half_h) / 15])

        h /= 2

    return result, y_values


def milne(f=lambda x, y: 0, x_0=0, y_0=0, x_n=0, h=0, epsilon=0):
    if x_0 > x_n:
        raise Exception("x_0 > x_n???")

    runge_kutta_table, y_values = runge_kutta(f, x_0, y_0, x_n, h, epsilon)
    h = runge_kutta_table.get_row(-1)[2]
    x_values = [(x_0 + i * h) for i in range(3)]
    y_values = y_values[:3]
    x = x_values[-1] + h

    while x < x_n:
        y_suggestion = y_values[-3] + 4 * h / 3 * (
                    2 * f(x_values[-3], y_values[-3]) - f(x_values[-2],
y_values[-2]) + 2 * f(x_values[-1],

y_values[-1]))
        y_corrected = y_values[-2] + h / 3 * (
                    f(x_values[-2], y_values[-2]) + 4 * f(x_values[-1],
y_values[-1]) + f(x, y_suggestion))

        while abs(y_suggestion - y_corrected) > epsilon:
            y_suggestion = y_corrected
            y_corrected = y_values[-2] + h / 3 * (
                    f(x_values[-2], y_values[-2]) + 4 * f(x_values[-1],
y_values[-1]) + f(x, y_suggestion))

        x_values.append(x)
```

```
        y_values.append(y_corrected)
        x += h
    return x_values, y_values
```

Результат выполнения программы

Пример 1

```
[Info]: Введите комманду:

help

[Input]: help

[Info]:

exit: Завершить работу приложения

help: Вывести информацию о доступных коммандах

lab6: Лабораторная работа 6 (ДУ)


[Info]: Введите комманду:

lab5

[Input]: lab5

[Error]: Нет такой команды 'lab5'. Напишите 'help', чтобы увидеть доступные команды

[Info]: Введите комманду:

lab6

[Input]: lab6

[Info]:

+-------+------------------------+

| Номер |               Уравнение |

+-------+------------------------+

|     1 | y' = xy / (x ** 2 + 1) |

+-------+------------------------+
```

```
|     2 |             y' = x + y |
+-------+------------------------+
|     3 |            y' = y + xy |
+-------+------------------------+


[Info]: Введите номер уравнения
1
[Input]: 1
[Info]: Ввод y(x_0) = y_0
[Info]: Введите x_0
1
[Input]: 1
[Info]: Введите y_0
1.5
[Input]: 1.5
[Info]: Введите x_n
3
[Input]: 3
[Info]: Введите шаг h
0.1
[Input]: 0.1
[Info]: Введите точность epsilon
0.001
[Input]: 0.001
[Info]: Метод Эйлера (h = 0.000390625):
+------------+---------+---------+---------+---------+----------+
| Номер шага |       h |   half_h |      y_h | y_half_h |        R |
```

1	0.200000	0.100000	inf	3.332078	inf
2	0.100000	0.050000	3.332078	3.393214	0.061136
3	0.050000	0.025000	3.393214	3.373694	0.019520
4	0.025000	0.012500	3.373694	3.351340	0.022354
5	0.012500	0.006250	3.351340	3.352721	0.001381
6	0.006250	0.003125	3.352721	3.356555	0.003834
7	0.003125	0.001563	3.356555	3.355329	0.001226
8	0.001563	0.000781	3.355329	3.353929	0.001399
9	0.000781	0.000391	3.353929	3.354016	0.000086

[Info]: Метод Рунге-Кутта 4-го порядка (h = 0.00625):

Номер шага	h	half_h	y_h	y_half_h	R
1	0.200000	0.100000	inf	3.354102	inf

| 2 | 0.100000 | 0.050000 | 3.354102 | 3.404455 | 0.003357 |

| 3 | 0.050000 | 0.025000 | 3.404455 | 3.379268 | 0.001679 |

| 4 | 0.025000 | 0.012500 | 3.379268 | 3.354102 | 0.001678 |

| 5 | 0.012500 | 0.006250 | 3.354102 | 3.354102 | 0.000000 |

[Info]: Метод Милна (h = 0.006250000000000089):

Номер	x_i	y_i	y_точн
0	1.000000	1.500000	1.500000
1	1.006250	1.504695	1.504695
2	1.012500	1.509404	1.509404
3	1.018750	1.514128	1.514128
4	1.025000	1.518866	1.518866
5	1.031250	1.523618	1.523618
6	1.037500	1.528384	1.528384

7	1.043750	1.533164	1.533164
8	1.050000	1.537957	1.537957
9	1.056250	1.542764	1.542764
10	1.062500	1.547585	1.547585
11	1.068750	1.552419	1.552419
12	1.075000	1.557266	1.557266
13	1.081250	1.562127	1.562127
14	1.087500	1.567000	1.567000
15	1.093750	1.571886	1.571886
16	1.100000	1.576785	1.576785
17	1.106250	1.581696	1.581696
18	1.112500	1.586620	1.586620
19	1.118750	1.591556	1.591556

| 20 | 1.125000 | 1.596505 | 1.596505 |

| 21 | 1.131250 | 1.601466 | 1.601466 |

| 22 | 1.137500 | 1.606439 | 1.606438 |

| 23 | 1.143750 | 1.611423 | 1.611423 |

| 24 | 1.150000 | 1.616420 | 1.616420 |

| 25 | 1.156250 | 1.621428 | 1.621428 |

| 26 | 1.162500 | 1.626448 | 1.626448 |

| 27 | 1.168750 | 1.631479 | 1.631479 |

| 28 | 1.175000 | 1.636522 | 1.636522 |

| 29 | 1.181250 | 1.641576 | 1.641576 |

| 30 | 1.187500 | 1.646641 | 1.646641 |

| 31 | 1.193750 | 1.651717 | 1.651717 |

| 32 | 1.200000 | 1.656804 | 1.656804 |

| 33 | 1.206250 | 1.661902 | 1.661902 |

34	1.212500	1.667011	1.667011
35	1.218750	1.672131	1.672131
36	1.225000	1.677261	1.677261
37	1.231250	1.682401	1.682401
38	1.237500	1.687552	1.687552
39	1.243750	1.692713	1.692713
40	1.250000	1.697885	1.697885
41	1.256250	1.703066	1.703066
42	1.262500	1.708258	1.708258
43	1.268750	1.713459	1.713459
44	1.275000	1.718671	1.718670
45	1.281250	1.723892	1.723891
46	1.287500	1.729122	1.729122

47	1.293750	1.734362	1.734362
48	1.300000	1.739612	1.739612
49	1.306250	1.744871	1.744871
50	1.312500	1.750140	1.750140
51	1.318750	1.755417	1.755417
52	1.325000	1.760704	1.760704
53	1.331250	1.766000	1.766000
54	1.337500	1.771305	1.771304
55	1.343750	1.776618	1.776618
56	1.350000	1.781941	1.781941
57	1.356250	1.787272	1.787272
58	1.362500	1.792612	1.792612
59	1.368750	1.797960	1.797960
60	1.375000	1.803317	1.803317

61	1.381250	1.808683	1.808683
62	1.387500	1.814057	1.814056
63	1.393750	1.819439	1.819438
64	1.400000	1.824829	1.824829
65	1.406250	1.830227	1.830227
66	1.412500	1.835634	1.835634
67	1.418750	1.841048	1.841048
68	1.425000	1.846471	1.846470
69	1.431250	1.851901	1.851901
70	1.437500	1.857339	1.857339
71	1.443750	1.862785	1.862784
72	1.450000	1.868238	1.868238
73	1.456250	1.873699	1.873699

| 74 | 1.462500 | 1.879167 | 1.879167 |

| 75 | 1.468750 | 1.884643 | 1.884643 |

| 76 | 1.475000 | 1.890127 | 1.890126 |

| 77 | 1.481250 | 1.895617 | 1.895617 |

| 78 | 1.487500 | 1.901115 | 1.901115 |

| 79 | 1.493750 | 1.906620 | 1.906620 |

| 80 | 1.500000 | 1.912133 | 1.912132 |

| 81 | 1.506250 | 1.917652 | 1.917652 |

| 82 | 1.512500 | 1.923178 | 1.923178 |

| 83 | 1.518750 | 1.928711 | 1.928711 |

| 84 | 1.525000 | 1.934252 | 1.934251 |

| 85 | 1.531250 | 1.939799 | 1.939798 |

| 86 | 1.537500 | 1.945352 | 1.945352 |

| 87 | 1.543750 | 1.950913 | 1.950912 |

88	1.550000	1.956480	1.956480
89	1.556250	1.962054	1.962053
90	1.562500	1.967634	1.967634
91	1.568750	1.973221	1.973220
92	1.575000	1.978814	1.978814
93	1.581250	1.984413	1.984413
94	1.587500	1.990019	1.990019
95	1.593750	1.995631	1.995631
96	1.600000	2.001250	2.001250
97	1.606250	2.006874	2.006874
98	1.612500	2.012505	2.012505
99	1.618750	2.018142	2.018142
100	1.625000	2.023785	2.023784

| 101 | 1.631250 | 2.029433 | 2.029433 |

| 102 | 1.637500 | 2.035088 | 2.035088 |

| 103 | 1.643750 | 2.040749 | 2.040748 |

| 104 | 1.650000 | 2.046415 | 2.046415 |

| 105 | 1.656250 | 2.052087 | 2.052087 |

| 106 | 1.662500 | 2.057765 | 2.057764 |

| 107 | 1.668750 | 2.063448 | 2.063448 |

| 108 | 1.675000 | 2.069137 | 2.069137 |

| 109 | 1.681250 | 2.074832 | 2.074832 |

| 110 | 1.687500 | 2.080532 | 2.080532 |

| 111 | 1.693750 | 2.086238 | 2.086238 |

| 112 | 1.700000 | 2.091949 | 2.091949 |

| 113 | 1.706250 | 2.097666 | 2.097665 |

| 114 | 1.712500 | 2.103388 | 2.103387 |

115	1.718750	2.109115	2.109115
116	1.725000	2.114847	2.114847
117	1.731250	2.120585	2.120585
118	1.737500	2.126328	2.126328
119	1.743750	2.132076	2.132076
120	1.750000	2.137829	2.137829
121	1.756250	2.143588	2.143587
122	1.762500	2.149351	2.149350
123	1.768750	2.155119	2.155119
124	1.775000	2.160892	2.160892
125	1.781250	2.166670	2.166670
126	1.787500	2.172453	2.172453
127	1.793750	2.178241	2.178240

| 128 | 1.800000 | 2.184033 | 2.184033 |

| 129 | 1.806250 | 2.189831 | 2.189830 |

| 130 | 1.812500 | 2.195633 | 2.195632 |

| 131 | 1.818750 | 2.201439 | 2.201439 |

| 132 | 1.825000 | 2.207251 | 2.207250 |

| 133 | 1.831250 | 2.213066 | 2.213066 |

| 134 | 1.837500 | 2.218887 | 2.218886 |

| 135 | 1.843750 | 2.224712 | 2.224711 |

| 136 | 1.850000 | 2.230541 | 2.230541 |

| 137 | 1.856250 | 2.236375 | 2.236375 |

| 138 | 1.862500 | 2.242214 | 2.242213 |

| 139 | 1.868750 | 2.248056 | 2.248056 |

| 140 | 1.875000 | 2.253903 | 2.253903 |

| 141 | 1.881250 | 2.259755 | 2.259754 |

142	1.887500	2.265610	2.265610
143	1.893750	2.271470	2.271470
144	1.900000	2.277334	2.277334
145	1.906250	2.283203	2.283202
146	1.912500	2.289075	2.289075
147	1.918750	2.294952	2.294951
148	1.925000	2.300833	2.300832
149	1.931250	2.306717	2.306717
150	1.937500	2.312606	2.312606
151	1.943750	2.318499	2.318498
152	1.950000	2.324396	2.324395
153	1.956250	2.330296	2.330296
154	1.962500	2.336201	2.336200

| 155 | 1.968750 | 2.342109 | 2.342109 |

| 156 | 1.975000 | 2.348022 | 2.348021 |

| 157 | 1.981250 | 2.353938 | 2.353937 |

| 158 | 1.987500 | 2.359858 | 2.359857 |

| 159 | 1.993750 | 2.365781 | 2.365781 |

| 160 | 2.000000 | 2.371709 | 2.371708 |

| 161 | 2.006250 | 2.377640 | 2.377639 |

| 162 | 2.012500 | 2.383575 | 2.383574 |

| 163 | 2.018750 | 2.389513 | 2.389513 |

| 164 | 2.025000 | 2.395455 | 2.395455 |

| 165 | 2.031250 | 2.401401 | 2.401400 |

| 166 | 2.037500 | 2.407350 | 2.407350 |

| 167 | 2.043750 | 2.413303 | 2.413302 |

| 168 | 2.050000 | 2.419259 | 2.419259 |

169	2.056250	2.425219	2.425218
170	2.062500	2.431182	2.431182
171	2.068750	2.437149	2.437148
172	2.075000	2.443119	2.443119
173	2.081250	2.449093	2.449092
174	2.087500	2.455069	2.455069
175	2.093750	2.461050	2.461049
176	2.100000	2.467033	2.467033
177	2.106250	2.473020	2.473019
178	2.112500	2.479010	2.479010
179	2.118750	2.485003	2.485003
180	2.125000	2.491000	2.490999
181	2.131250	2.497000	2.496999

| 182 | 2.137500 | 2.503003 | 2.503002 |

| 183 | 2.143750 | 2.509009 | 2.509008 |

| 184 | 2.150000 | 2.515018 | 2.515017 |

| 185 | 2.156250 | 2.521030 | 2.521030 |

| 186 | 2.162500 | 2.527046 | 2.527045 |

| 187 | 2.168750 | 2.533064 | 2.533064 |

| 188 | 2.175000 | 2.539086 | 2.539085 |

| 189 | 2.181250 | 2.545110 | 2.545110 |

| 190 | 2.187500 | 2.551138 | 2.551137 |

| 191 | 2.193750 | 2.557168 | 2.557168 |

| 192 | 2.200000 | 2.563202 | 2.563201 |

| 193 | 2.206250 | 2.569238 | 2.569238 |

| 194 | 2.212500 | 2.575277 | 2.575277 |

| 195 | 2.218750 | 2.581320 | 2.581319 |

196	2.225000	2.587365	2.587364
197	2.231250	2.593413	2.593412
198	2.237500	2.599463	2.599463
199	2.243750	2.605517	2.605516
200	2.250000	2.611573	2.611573
201	2.256250	2.617633	2.617632
202	2.262500	2.623694	2.623694
203	2.268750	2.629759	2.629759
204	2.275000	2.635827	2.635826
205	2.281250	2.641897	2.641896
206	2.287500	2.647969	2.647969
207	2.293750	2.654045	2.654044
208	2.300000	2.660123	2.660122

| 209 | 2.306250 | 2.666204 | 2.666203 |

| 210 | 2.312500 | 2.672287 | 2.672286 |

| 211 | 2.318750 | 2.678373 | 2.678372 |

| 212 | 2.325000 | 2.684461 | 2.684460 |

| 213 | 2.331250 | 2.690552 | 2.690552 |

| 214 | 2.337500 | 2.696646 | 2.696645 |

| 215 | 2.343750 | 2.702742 | 2.702741 |

| 216 | 2.350000 | 2.708840 | 2.708840 |

| 217 | 2.356250 | 2.714941 | 2.714941 |

| 218 | 2.362500 | 2.721045 | 2.721044 |

| 219 | 2.368750 | 2.727151 | 2.727150 |

| 220 | 2.375000 | 2.733259 | 2.733259 |

| 221 | 2.381250 | 2.739370 | 2.739370 |

| 222 | 2.387500 | 2.745483 | 2.745483 |

223	2.393750	2.751599	2.751598
224	2.400000	2.757717	2.757716
225	2.406250	2.763838	2.763837
226	2.412500	2.769960	2.769960
227	2.418750	2.776085	2.776085
228	2.425000	2.782213	2.782212
229	2.431250	2.788342	2.788342
230	2.437500	2.794474	2.794473
231	2.443750	2.800608	2.800608
232	2.450000	2.806745	2.806744
233	2.456250	2.812884	2.812883
234	2.462500	2.819024	2.819024
235	2.468750	2.825168	2.825167

| 236 | 2.475000 | 2.831313 | 2.831312 |

| 237 | 2.481250 | 2.837460 | 2.837460 |

| 238 | 2.487500 | 2.843610 | 2.843609 |

| 239 | 2.493750 | 2.849762 | 2.849761 |

| 240 | 2.500000 | 2.855916 | 2.855915 |

| 241 | 2.506250 | 2.862072 | 2.862071 |

| 242 | 2.512500 | 2.868230 | 2.868229 |

| 243 | 2.518750 | 2.874390 | 2.874389 |

| 244 | 2.525000 | 2.880552 | 2.880552 |

| 245 | 2.531250 | 2.886717 | 2.886716 |

| 246 | 2.537500 | 2.892883 | 2.892883 |

| 247 | 2.543750 | 2.899052 | 2.899051 |

| 248 | 2.550000 | 2.905222 | 2.905222 |

| 249 | 2.556250 | 2.911395 | 2.911394 |

250	2.562500	2.917569	2.917569
251	2.568750	2.923746	2.923745
252	2.575000	2.929925	2.929924
253	2.581250	2.936105	2.936104
254	2.587500	2.942287	2.942287
255	2.593750	2.948472	2.948471
256	2.600000	2.954658	2.954657
257	2.606250	2.960846	2.960846
258	2.612500	2.967036	2.967036
259	2.618750	2.973229	2.973228
260	2.625000	2.979422	2.979422
261	2.631250	2.985618	2.985617
262	2.637500	2.991816	2.991815

| 263 | 2.643750 | 2.998015 | 2.998014 |

| 264 | 2.650000 | 3.004217 | 3.004216 |

| 265 | 2.656250 | 3.010420 | 3.010419 |

| 266 | 2.662500 | 3.016625 | 3.016624 |

| 267 | 2.668750 | 3.022831 | 3.022831 |

| 268 | 2.675000 | 3.029040 | 3.029039 |

| 269 | 2.681250 | 3.035250 | 3.035249 |

| 270 | 2.687500 | 3.041462 | 3.041462 |

| 271 | 2.693750 | 3.047676 | 3.047675 |

| 272 | 2.700000 | 3.053892 | 3.053891 |

| 273 | 2.706250 | 3.060109 | 3.060108 |

| 274 | 2.712500 | 3.066328 | 3.066327 |

| 275 | 2.718750 | 3.072549 | 3.072548 |

| 276 | 2.725000 | 3.078771 | 3.078771 |

277	2.731250	3.084996	3.084995
278	2.737500	3.091221	3.091221
279	2.743750	3.097449	3.097448
280	2.750000	3.103678	3.103677
281	2.756250	3.109909	3.109908
282	2.762500	3.116141	3.116141
283	2.768750	3.122376	3.122375
284	2.775000	3.128611	3.128610
285	2.781250	3.134849	3.134848
286	2.787500	3.141088	3.141087
287	2.793750	3.147328	3.147327
288	2.800000	3.153570	3.153569
289	2.806250	3.159814	3.159813

| 290 | 2.812500 | 3.166059 | 3.166058 |

| 291 | 2.818750 | 3.172306 | 3.172305 |

| 292 | 2.825000 | 3.178555 | 3.178554 |

| 293 | 2.831250 | 3.184804 | 3.184804 |

| 294 | 2.837500 | 3.191056 | 3.191055 |

| 295 | 2.843750 | 3.197309 | 3.197308 |

| 296 | 2.850000 | 3.203563 | 3.203562 |

| 297 | 2.856250 | 3.209819 | 3.209818 |

| 298 | 2.862500 | 3.216077 | 3.216076 |

| 299 | 2.868750 | 3.222336 | 3.222335 |

| 300 | 2.875000 | 3.228596 | 3.228595 |

| 301 | 2.881250 | 3.234858 | 3.234857 |

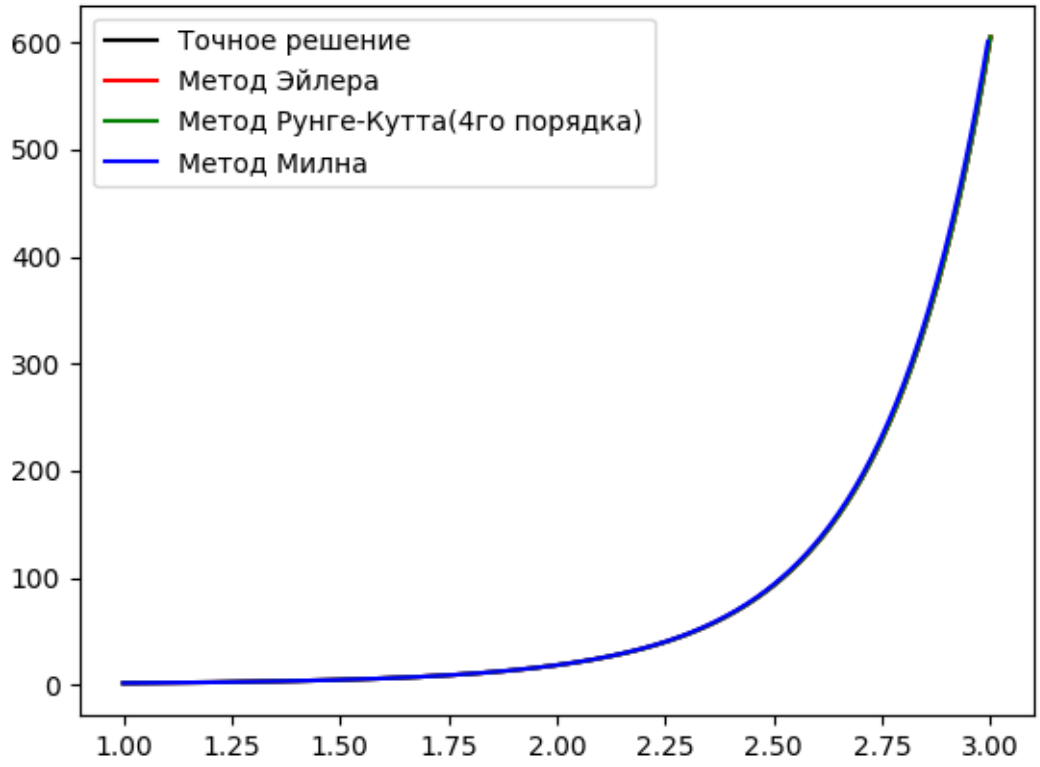
| 302 | 2.887500 | 3.241122 | 3.241121 |

| 303 | 2.893750 | 3.247386 | 3.247386 |

304	2.900000	3.253653	3.253652
305	2.906250	3.259920	3.259920
306	2.912500	3.266190	3.266189
307	2.918750	3.272460	3.272459
308	2.925000	3.278732	3.278731
309	2.931250	3.285006	3.285005
310	2.937500	3.291280	3.291279
311	2.943750	3.297556	3.297555
312	2.950000	3.303834	3.303833
313	2.956250	3.310113	3.310112
314	2.962500	3.316393	3.316392
315	2.968750	3.322675	3.322674
316	2.975000	3.328958	3.328957

```
|   317 | 2.981250 | 3.335242 | 3.335241 |
+-------+----------+----------+----------+
|   318 | 2.987500 | 3.341528 | 3.341527 |
+-------+----------+----------+----------+
|   319 | 2.993750 | 3.347815 | 3.347814 |
+-------+----------+----------+----------+
```

```
[Info]: Оценка погрешности метода Милна max(|y_i_точн - y_i|) = 1.0013834046418424e-06
```




```
[Info]: Лабораторная работа 6 (ДУ) завершилась
```

Выводы

Изучил численные методы решения задачи Коши.