



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 курсу: "Моделирование"

Студент _____ Сукочева Алис
Группа _____ ИУ7-63Б
Название предприятия _____ МГТУ им. Н. Э. Баумана, каф. ИУ7
Тема _____ Метод Пикара.

Студент:	_____	Сукочева А.
	подпись, дата	Фамилия, И.О.
Преподаватель:	_____	Градов В.М.
	подпись, дата	Фамилия, И. О.

Теоретические сведения

Моделирование - исследование объектов, в ходе которого он заменяется моделью и исследование объекта проводится на его модели. Результат переносится на исходный объект. Объектом может быть система, явление, процесс и т.д..

Модель - представление объекта в виде, отличном от облика или способа его реального существования или способа функционирования.

Корректно поставленная задача, если ее решение существует единственно и устойчиво по входным данным.

Устойчивая задача - малое изменение входных данных должно порождать малые изменения выходных данных.

ОДУ

Дано ОДУ (Обыкновенное Дифференциальное уравнение) n -ого порядка (0.1).

$$F(x, u', u'', \dots, u^{(n)} = 0) \quad (0.1)$$

ОДУ любого порядка может быть сведено к системе ОДУ 1-ого порядка.

Задача Коши

Задача Коши состоит в нахождении решения дифференциального уравнения, удовлетворяющего начальным условиям. Это одна из основных задач теории дифференциальных уравнений.

Имеется задача Коши (0.2).

$$\begin{cases} u'(x) = f(x, u) \\ u(\xi) = \eta \end{cases} \quad (0.2)$$

Методы решения ОДУ в задачи Коши:

- а) аналитические;
- б) приближенно аналитические;
- в) численные.

Методы решения задачи Коши

При отсутствии аналитического решения можно воспользоваться приближенно аналитическим методом Пикара. Заменяв дифференциальное уравнение интегральным получим (0.3).

$$y(x)^s = \eta + \int_{\xi}^x f(t, y^{s-1}(t)) dt \quad (0.3)$$

$$y^{(0)} = \eta \quad (0.4)$$

Метод сходится если:

- а) правая часть непрерывная;
- б) выполнено условие Липшица (0.5)

$$|f(x, u_1) - f(x, u_2)| \leq L|u_1 - u_2| \quad (0.5)$$

где L - константа Липшица.

Метод Эйлера (0.6).

$$y_{n+1} = y_n + h * f(x_n, y_n) \quad (0.6)$$

Метод Рунге-Кутты (0.7).

$$y_{n+1} = y_n + h * [(1 - \alpha)k_1 + \alpha * k_2] \quad (0.7)$$

Где k_1 и k_2 представлены как (0.8) и (0.9) соответственно. А $\alpha = 1$ или $\frac{1}{2}$

$$k_1 = f(x_n, y_n) \quad (0.8)$$

$$k_2 = f(x_n + \frac{h}{2\alpha}, y_n + \frac{h}{2\alpha}k_1) \quad (0.9)$$

Задание и вычисления приближений для метода Пикара

Дана задача 0.10

$$\begin{cases} u'(x) = u^2 + x^2 \\ u(0) = 0 \end{cases} \quad (0.10)$$

Используя описанные выше методы построить таблицу для:

а) метода Пикара:

- 1) Первое приближение;
- 2) Второе приближение;
- 3) третье приближение;
- 4) четверное приближение.

б) метод Эйлера;

в) метод Рунге-Кутты.

Приближения:

$$y^{(1)} = 0 + \int_0^x t^2 dt = \frac{x^3}{3} \quad (0.11)$$

$$y^{(2)} = 0 + \int_0^x \left[\left(\frac{t^3}{3} \right)^2 + t^2 \right] dt = \frac{x^3}{3} + \frac{x^7}{63} \quad (0.12)$$

$$\begin{aligned} y^{(3)} &= 0 + \int_0^x \left[\left(\frac{t^7}{63} + \frac{t^3}{3} \right)^2 + t^2 \right] dt = \\ &= \int_0^x \left[\frac{t^{14}}{63^2} + \frac{2}{63 \cdot 3} t^{10} + \frac{t^6}{9} + t^2 \right] dt = \\ &= \frac{x^3}{3} + \frac{x^7}{63} + \frac{2x^{11}}{2079} + \frac{x^{15}}{59535} \end{aligned} \quad (0.13)$$

$$\begin{aligned} y^{(4)} &= 0 + \int_0^x \left[\left(\frac{t^3}{3} + \frac{t^7}{63} + \frac{2t^{11}}{2079} + \frac{t^{15}}{59535} \right)^2 + t^2 \right] dt = \\ &= \frac{x^3}{3} + \frac{x^7}{63} + \frac{2x^{11}}{2079} + \frac{13x^{15}}{218295} + \frac{82x^{19}}{37328445} + \\ &+ \frac{662x^{23}}{10438212015} + \frac{4x^{27}}{3341878155} + \frac{x^{31}}{109876903905} \end{aligned} \quad (0.14)$$

Листинги реализованных методов

```
1 from prettytable import PrettyTable
2
3 # Подбираем шаг:
4
5 # Euler:
6 # При  $1e-1$   $y(1) = 0.2925421046$ 
7 # При  $1e-2$   $y(1) = 0.3331073593$ 
8 # При  $1e-3$   $y(1) = 0.3484859823$ 
9 # При  $1e-4$   $y(1) = 0.3501691515$ 
10 # При  $1e-5$   $y(1) = 0.3502255745$ 
11 # Шаг ничего не меняет (между  $1e-3$  и  $1e-4$ )
12 # Значит мы подобрали нулевой нам шаг.
13
14 # Runge:
15 # При  $1e-1$   $y(1) = 0.3933413435$ 
16 # При  $1e-2$   $y(1) = 0.4015176873$ 
17 # При  $1e-3$   $y(1) = 0.4184706977$ 
18 # При  $1e-4$   $y(1) = 0.420382457$ 
19 # При  $1e-5$   $y(1) = 0.4204145637$ 
20 # Аналогично.
21
22 ACCURACY = 10
23 MAX_X = 1
24 STEP = 0.01 #  $1e-3$ 
25
26
27 def f(x, y):
28     return pow(x, 2) + pow(y, 2)
29
30 def fp1(x):
31     return pow(x, 3) / 3
32
33 def fp2(x):
34     return pow(x, 7) / 63 + \
35         fp1(x)
36
37 def fp3(x):
38     return pow(x, 15) / 59535 + \
39         2 * pow(x, 11) / 2079 + \
40         fp2(x)
41
42 def fp4(x):
43     return pow(x, 31) / 109876903905 + \
44         4 * pow(x, 27) / 3341878155 + \
45         662 * pow(x, 23) / 10438212015 + \
```

```

46         82 * pow(x, 19) / 37328445 + \
47         fp3(x)
48
49
50 def Picar(x_max, h, func):
51     result = list()
52     x, y = 0, 0
53
54     while x < x_max:
55         result.append(round(y, ACCURACY))
56         x += h
57         y = func(x)
58
59     return result
60
61
62 def Euler(x_max, h):
63     result = list()
64     x, y = 0, 0      # Начальное условие.
65
66     while x < x_max:
67         result.append(round(y, ACCURACY))
68         y = y + h * f(x, y)
69         x += h
70
71     return result
72
73
74 def Runge(x_max, h):
75     result = list()
76     alpha = 0.5      # or 1
77     coeff = h / (2 * alpha)
78     x, y = 0, 0
79
80     while x < x_max:
81         result.append(round(y, ACCURACY))
82         k1 = f(x, y)
83         k2 = f(x + coeff, y + alpha * k1)
84         y = y + h * ((1 - alpha) * k1 + alpha * k2)
85         x += h
86
87     return result
88
89
90 def x_range(x_max, h):
91     result = list()
92     x = 0

```

```

93     while x < x_max:
94         result.append(round(x, 2))
95         x += h
96     return result
97
98
99 def main():
100     column_names = ["X", "Picard 1", "Picard 2", "Picard 3", "Picard 4",
101                     "Runge"]
102
103     tb = PrettyTable()
104     tb.add_column("X", x_range(MAX_X, STEP))
105     tb.add_column("Picard 1", Picar(MAX_X, STEP, fp1))
106     tb.add_column("Picard 2", Picar(MAX_X, STEP, fp2))
107     tb.add_column("Picard 3", Picar(MAX_X, STEP, fp3))
108     tb.add_column("Picard 4", Picar(MAX_X, STEP, fp4))
109     tb.add_column("Euler", Euler(MAX_X, STEP))
110     tb.add_column("Runge", Runge(MAX_X, STEP))
111
112     print(tb)
113
114 if __name__ == "__main__":
115     main()

```