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.

Москва — 2021 г.

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).

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

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).

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

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. Заменив дифференциальное уравнение интегральным получим (0.3).

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

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

Метод сходится если:

а) правая часть непрерывная;

б) выполнено условие Липшица (0.5)

$$|f(x, u_1) - f(x, u_2| \leqslant L|u_1 - u_2| \tag{0.5}$$

где L - константа Липшица.

Метод Эйлера (0.6).

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

Метод Рунге-Кутты (0.15).

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

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

$$k_1 = f(x_n, y_n) \tag{0.8}$$

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

Задание и вычисления приближений для метода Пикара

Дана задача 0.10

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

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:

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

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

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

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

Листинги реализованных методов

```
from prettytable import PrettyTable

# Подбираем шаг:

# Euler:
# При 1e-1 y(1) = 0.2925421046
# При 1e-2 y(1) = 0.3331073593
# При 1e-3 y(1) = 0.3484859823
# При 1e-4 y(1) = 0.3501691515
# При 1e-5 y(1) = 0.3502255745
# Изменение шага ничего не меняет (между 1e-4 и 1e-5)
# Значит мы подобрали нужный нам шаг.

# Runge:
# При 1e-1 y(1) = 0.3485453439
# При 1e-2 y(1) = 0.3391265967
# При 1e-3 y(1) = 0.3491103993
# При 1e-4 y(1) = 0.3502318426
# При 1e-5 y(1) = 0.3502318443
# Аналогично.

MAX_X = 1
STEP = 1e-4


def f(x, y):
    return pow(x, 2) + pow(y, 2)

def fp1(x):
    return pow(x, 3) / 3

def fp2(x):
    return pow(x, 7) / 63 + \
        fp1(x)

def fp3(x):
    return pow(x, 15) / 59535 + \
        2 * pow(x, 11) / 2079 + \
        fp2(x)

def fp4(x):
    return pow(x, 31) / 109876903905 + \
        4 * pow(x, 27) / 3341878155 + \
        662 * pow(x, 23) / 10438212015 + \
        82 * pow(x, 19) / 37328445 + \
```

```
        fp3(x)


def Picar(x_max, h, func):
    result = list()
    x, y = 0, 0

    while x < x_max:
        result.append(y)
        x += h
        y = func(x)

    return result


def Euler(x_max, h):
    result = list()
    x, y = 0, 0      # Начальное условие.

    while x < x_max:
        result.append(y)
        y = y + h * f(x, y)
        x += h

    return result


def Runge(x_max, h):
    result = list()
    coeff = h / 2
    x, y = 0, 0

    while x < x_max:
        result.append(y)
        y = y + h * f(x + coeff, y + coeff * f(x, y))
        x += h

    return result


def x_range(x_max, h):
    result = list()
    x = 0
    while x < x_max:
        result.append(round(x, 2))
        x += h
    return result
```

```python


def main():
    column_names = ["X","Picard 1", "Picard 2", "Picard 3", "Picard 4",
        "Runge"]

    tb = PrettyTable()
    tb.add_column("X", x_range(MAX_X, STEP))
    tb.add_column("Picard 1", Picar(MAX_X, STEP, fp1))
    tb.add_column("Picard 2", Picar(MAX_X, STEP, fp2))
    tb.add_column("Picard 3", Picar(MAX_X, STEP, fp3))
    tb.add_column("Picard 4", Picar(MAX_X, STEP, fp4))
    tb.add_column("Euler", Euler(MAX_X, STEP))
    tb.add_column("Runge", Runge(MAX_X, STEP))

    print(tb)


if __name__ == "__main__":
    main()
```

Ответы на вопросы

1. Для того, чтобы указать интервал значений аргумента, в которых можно считать решением заданного уравнения каждое из первых 4-х приближений проанализируем полученные значения. Так как нам дано начальное приближение, то левой границей будет 0. Для определения правой границы границы мы будем анализировать полученные решения методом Пикара для конкретного приближения и сравнивать со значениями более высоких порядков приближения и с результатами численных методов при определенном шаге. В листинге был подобран шаг 1e-4. Для первого приближения искомым интервалом будет [0, 0.89], для второго [0, 1.12], для третьего [0, 1.39], для четвертого [0, 1.4]

2. В численных методах правильность полученного результата, при фиксированном значении аргумента, доказывается путем уменьшения шага. Правильно полученный результат - это когда при уменьшение шага значение аргумента незначительно (или вообще) не меняется.

3. Примерное значение функции при x = 2.

$$u(2) \approx 316.713 \tag{0.15}$$