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»

Лабораторная работа № 1

Тема: Приближенный аналитический метод Пикара, сравнение с численными методами

Студент: Тимонин Антон

Группа ИУ7-62б

Оценка (баллы) _______________

Преподаватель Градов В. М.

Москва.
2020 г.

Задание

Решить задачу Коши тремя различными методами, явным методом Эйлера, неявным методом Эйлера, Рунге-Кутта.

$$\begin{cases} y' = f(x,\ y) \\ y(x_0) = y_0 \end{cases}$$

$$\frac{dy}{dx} = f(x, y)$$

$$y_{x=x_0} = y_0$$

Решение можно найти приближенным аналитическим **методом Пикара**

$$y^{(n+1)}\left(x\right) = \eta + \int_{\xi}^{x} f\left(t,\ y^{(n)}\left(t\right)\right)\mathrm{d}t y$$

$$y^{(0)}(x) = \eta$$

Рассмотрим на примере применение метода Пикара:

$$y'(x) = x^2 + y^2$$

$$y(0) = 0$$

$$y^{(1)}\left(x\right) = 0 + \int_0^x t^2 \mathrm{d}t y = \frac{t^3}{3}$$

$$y^{(2)}\left(x\right) = 0 + \int_0^x \left[t^2 + \left(\frac{t^3}{3}\right)^2\right] dt y = \frac{t^3}{3} + \frac{t^7}{7 \cdot 9} = \frac{t^3}{3}\left[1 + \frac{t^4}{21}\right]$$

$$y^{(3)}\left(x\right) = 0 + \int_0^x \left[t^2 + \left(\frac{t^3}{3}\left[1 + \frac{t^4}{21}\right]\right)^2\right] dt y = \frac{t^3}{3} + \frac{t^7}{7 \cdot 9} = \frac{t^3}{3}\left[1 + \frac{t^4}{21}\right] =$$

$$= \frac{x^3}{3}\left[1 + \frac{x^4}{21} + \frac{2x^8}{693} + \frac{x^{12}}{19845}\right]$$

Чем больше итераций *n*, тем точнее результат.

Явный метод Эйлера

Формула:

$$y_i = y_{i-1} + \left(x_i - x_{i-1}\right) f\left(x_{i-1}, y_{i-1}\right),\ i = 1,\ 2,\ 3,\ \ldots,\ n$$

Неявный метод Эйлера

Прогноз:

$$y_i' = y_{i-1} + \left(x_i - x_{i-1}\right) f\left(x_{i-1}, y_{i-1}\right)$$

Коррекция:

$$y_i = y_{i-1} + \left(x_i - x_{i-1}\right) \frac{f\left(x_{i-1},\ y_{i-1}\right) + f\left(x_i, y_i'\right)}{2}$$

Чем меньше шаг в Эйлере, тем точнее результат. Это связано с тем, что при маленьком шаге, каждое новое значение итерации будет меньше отдаляться от истинного значения, поэтому по итогу мы получим максимально приближенный ответ.

Листинг 1 – Метод Пикара

```
def f(x, y):
    return x**2 + y**2

def pikar_1(x):
    return 1/3*(x**3)

def pikar_2(x):
    return 1/3*(x**3) + \
           1/54*(x**7)

def pikar_3(x):
    return 1/3*(x**3) + \
           1/54*(x**7) + \
           1/810*(x**11) + \
           1/40824*(x**15)

def pikar_4(x):
    return (
              1/3*(x**3) +
              1/63*(x**7) +
              1/891*(x**11) +
              1/135*(x**15)*(1/324+1/135) +
              1/171*(x**19)*(1/6804+1/2430) +
              1/207*(x**23)*(1/122472+1/72900) +
              1/49601160*(x**27) +
              1/5740507584*(x**31)
              )



def pikar(x0, x1, h):
    result1 = list()
    result2 = list()
    result3 = list()
    result4 = list()

    while fabs(x0 - x1) > EPS:
        result1.append(round(pikar_1(x0), ROUNDED_NUM))
        result2.append(round(pikar_2(x0), ROUNDED_NUM))
        result3.append(round(pikar_3(x0), ROUNDED_NUM))
        result4.append(round(pikar_4(x0), ROUNDED_NUM))

        x0 += h

    return [result1, result2, result3, result4]
```

Листинг 2 – Явный Эйлер

```
def euler_simple(x0, x1, h):
    result = list()

    # x0 -> a
    # x1 -> b
    # h  -> step

    yk = 0
    while fabs(x0 - x1) > EPS:
        x0 += h
        yk += f(x0, yk) * h


        result.append(round(yk, ROUNDED_NUM))

    return result
```

Листинг 3 – Неявный Эйлер

```
def euler_imprv(x0, x1, h):
    result = list()

    # x0 -> a
    # x1 -> b
    # h  -> step

    yk = 0
    while fabs(x0 - x1) > EPS:

        tmp = yk + f(x0 + h , yk) * h
        yk += (f(x0, yk) + f(x0 + h, tmp)) * h * 0.5
        x0 += h

        result.append(round(yk, ROUNDED_NUM))

    return result
```