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 в сравнении с численными методами

Студент: Лаврова А. А.

Группа: ИУ7-65Б

Оценка (баллы) _______________

Преподаватель: Градов В.М.

Москва.
2020 г.

Цель работы: рассмотрение методов решения дифференциального уравнения

Метод Пикара:

Метод Пикара – это приближенный метод решения, являющийся обобщением метода последовательных приближений.

Рассмотрим с 1 по 4 приближения функции f = x^2+u^2

$$u(0) = 0$$

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

$$y^{(2)} = 0 + \int_0^x [t^2 + \left(\frac{x^3}{3}\right)^2 dt = \left.\frac{t^3}{3}\right|_0^x + \left.\frac{t^7}{7*9}\right|_0^x = \frac{x^3}{3}\left[1 + \frac{x^4}{21}\right]$$

$$y^{(3)} = 0 + \int_0^x [t^2 + \left\{\frac{x^3}{3}\left(1 + \frac{x^4}{21}\right)\right\}^2 dt = \left.\frac{t^3}{3}\right|_0^x + \int_0^x \frac{x^{14}}{3969} + \frac{2x^{10}}{189} + \frac{x^6}{9} dt =$$

$$= \left.\frac{t^3}{3}\right|_0^x + \int_0^x \frac{x^{14}}{3969} dt + \int_0^x \frac{2x^{10}}{189} dt + \int_0^x \frac{x^6}{9} dt = \left.\frac{t^3}{3}\right|_0^x$$

$$= + \int_0^x \frac{x^{14}}{3969} dt + \int_0^x \frac{2x^{10}}{189} dt + \int_0^x \frac{x^6}{9} dt =$$

$$= \left.\frac{t^3}{3}\right|_0^x + \left.\frac{t^{15}}{15*3969}\right|_0^x + \left.\frac{2t^{11}}{11*189}\right|_0^x + \left.\frac{t^7}{7*9}\right|_0^x = \frac{x^3}{3} + \frac{x^{15}}{59535} + \frac{2x^{11}}{2079} + \frac{x^7}{63}$$

$$y^{(4)} = 0 + \int_0^x t^2 + \left(\frac{t^3}{3} + \frac{t^{15}}{59535} + \frac{2t^{11}}{2079} + \frac{t^7}{63}\right)^2 dt$$

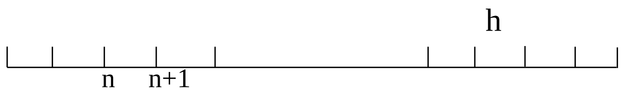
$$= \int_0^x t^2 + \frac{t^{30}}{3544416225} + \frac{4t^{26}}{123773265} + \frac{662t^{22}}{45383505} + \frac{82t^{18}}{1964655} + \frac{13t^{14}}{14553} + \frac{2t^{10}}{189} + \frac{t^6}{9} dt =$$

$$= \frac{x^{31}}{109876902975} + \frac{4*x^{27}}{3341878155} + \frac{662x^{23}}{10438212015} + \frac{82x^{19}}{37328445} + \frac{13x^{15}}{218295} + \frac{2x^{11}}{2079} + \frac{x^7}{63} + \frac{x^3}{3}$$

Метод Эйлера:

Метод Эйлера - простейший численный метод решения систем обыкновенных дифференциальных уравнений. Отличается низкой точностью.

В данной лабораторной работе рассмотрены две схемы: явная и неявная



Явная схема:

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

Неявная схема:

$y_{n+1} = y_n + h * (f(x_{n+1}, y_{n+1}))$

Листинг программы:

```
def func(x, u):
    return x ** 2 + u ** 2


def euler(n, h, x, y):
    y_out = []
    for i in range(n):
        y += h * func(x, y)
        x += h
        y_out.append(y)
    return y_out


def un_euler(n, h, x, y):
    y_out = []
    for i in range(n):
        y += h * (func(x, y) + func(x + h, y + h * func(x, y))) / 2
        x += h
        y_out.append(y)
    return y_out

def picar_1(n, h, x):
    def f(a):
        return (a ** 3) / 3

    y_out = [0]
    for i in range(n - 1):
        x += h
        y_out.append(f(x))
    return y_out


def picar_2(n, h, x):
    def f(a):
        return ((a ** 3)/3) * (1 + (x ** 4)/21)
```

```
    y_out = [0]
    for i in range(n - 1):
        x += h
        y_out.append(f(x))
    return y_out


def picar_3(n, h, x):
    def f(a):
        return (a ** 3)/3 + (a ** 15)/59535 + (2 * a ** 11)/2079 + (a ** 7)/63

    y_out = [0]
    for i in range(n - 1):
        x += h
        y_out.append(f(x))
    return y_out


def picar_4(n, h, x):
    def f(a):
        return ((a ** 31)/109876902975 + (4 * a ** 27)/3341878155 + (662 * a ** 23)/10438212015 +
                + (82 * a ** 19)/37328445  + (13 * a ** 15)/218295 + (2 * a ** 11)/2079 + (a ** 3)/3)

    y_out = [0]
    for i in range(n - 1):
        x += h
        y_out.append(f(x))
    return y_out



#n = 2*10 ** 6
#h = 10 ** -6
n = 21
h = 0.1
x = 0
y0 = 0
x_arr = [x + h * i for i in range(n)]
y1 = euler(n, h, x, y0)
y2 = un_euler(n, h, x, y0)
y_p1 = picar_1(n, h, x)
y_p2 = picar_2(n, h, x)
y_p3 = picar_3(n, h, x)
y_p4 = picar_4(n, h, x)

print("|  x  |    Picar #1  |    Picar #2   |    Picar #3   |    Picar #4   |     Eyler     |     non Eyler     |")
print("-" * 114)
for i in range(len(y1)):
    print("|{:.2f} |  {:.8f}   |  {:.8f}   |  {:.8f}   |  {:.8f}   |   {:.8f}   |    {:.8f}     |".format(x_arr[i], y_p1[i], y_p2[i],
y_p3[i], y_p4[i],
                                                                                 y1[i], y2[i]))
print("-" * 114)
```