

Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №6
по «Вычислительной математике»

Вариант 4

Выполнил:

Студент группы Р3208

Дашкевич Егор Вячеславович

Преподаватели:

Машина Екатерина Алексеевна

Санкт-Петербург

2024

Оглавление

Цель работы.....	3
Задание:	3
Вычислительная часть	Ошибка! Закладка не определена.
Листинг программы:.....	4
Вывод	7

Цель работы

решить задачу Коши для обыкновенных дифференциальных уравнений численными методами.

Задание:

1. В программе численные методы решения обыкновенных дифференциальных уравнений (ОДУ) должен быть реализован в виде отдельного класса /метода/функции;
2. Пользователь выбирает ОДУ вида $y' = f(x, y)$ (не менее трех уравнений), из тех, которые предлагает программа;
3. Предусмотреть ввод исходных данных с клавиатуры: начальные условия $y_0 = y(x_0)$, интервал дифференцирования $[x_0, x_n]$, шаг h , точность ε ;
4. Для исследования использовать одношаговые методы и многошаговые методы (см. табл.1);
5. Составить таблицу приближенных значений интеграла дифференциального уравнения, удовлетворяющего начальным условиям, для всех методов, реализуемых в программе;
6. Для оценки точности одношаговых методов использовать правило Рунге;
7. Для оценки точности многошаговых методов использовать точное решение задачи;
8. Построить графики точного решения и полученного приближенного решения (разными цветами);
9. Программа должна быть протестирована при различных наборах данных, в том числе и некорректных.
10. Проанализировать результаты работы программы.

Методы которые необходимо реализовать:

- Метод Эйлера
- Усовершенствованный метод Эйлера
- Метод Адамса

Рабочие формулы методов:

Эйлер:

Тогда получаем **формулу Эйлера**:

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

Усовершенствованный Эйлер:

$$y_{i+1} = y_i + \frac{h}{2} [f(x_i, y_i) + f(x_{i+1}, y_i + hf(x_i, y_i))], \quad i = 0, 1, \dots$$

Адамс:

$$\Delta f_i = f_i - f_{i-1}$$

$$\Delta^2 f_i = f_i - 2f_{i-1} + f_{i-2}$$

$$\Delta^3 f_i = f_i - 3f_{i-1} + 3f_{i-2} - f_{i-3}$$

$$y_{i+1} = y_i + hf_i + \frac{h^2}{2} \Delta f_i + \frac{5h^3}{12} \Delta^2 f_i + \frac{3h^4}{8} \Delta^3 f_i$$

Листинг программы:

Эйлер:

```
def solve(f, x0, y0, h, n):  
    result = [y0]  
    prev_y = y0  
  
    for i in range(n-1):  
        prev_y += h * f(x0 + i*h, prev_y)  
        result.append(prev_y)  
  
    return result
```

Усовершенствованный Эйлер:

```
def solve(f, x0, y0, h, n):
    result = [y0]
    prev_x, prev_y = x0, y0
    x = prev_x + h

    for i in range(n-1):
        prev_y += (h/2) * (f(prev_x, prev_y) + f(x, prev_y + h*f(prev_x, prev_y)))
        prev_x, x = x, x + h
        result.append(prev_y)

    return result
```

Адамс:

```
from methods import euler

def solve(f, x0, y0, h, n):
    result = euler.solve(f, x0, y0, h, n: 4)
    pre_f = []

    for i in range(4):
        pre_f.append(f(x0 + h * i, result[i]))

    prev_y = result[-1]

    for i in range(4, n):
        df = pre_f[-1] - pre_f[-2]
        d2f = pre_f[-1] - 2 * pre_f[-2] + pre_f[-3]
        d3f = pre_f[-1] - 2 * pre_f[-2] + 3 * pre_f[-3] - pre_f[-4]

        prev_y += h * pre_f[-1] + (h ** 2) / 2 * df + 5 * (h ** 3) / 12 * d2f + 3 * (h ** 4) / 8 * d3f
        result.append(prev_y)
        pre_f.append(f(x0 + h * i, result[-1]))

    return result
```

Вывод программы:

```
Enter filename or press Enter: test1
File read successfully
Function: y' = y + (1+x) * y^2
Method: Advanced Euler
x_0 = 1.0
y_0 = -1.0
Step h = 2.0
Intervals amount n = 10
Precision = 0.01

i      | 0      | 1      | 2      | 3      | 4      | 5      | 6      | 7      | 8      | 9      |
x      | 1.0    | 3.0    | 5.0    | 7.0    | 9.0    | 11.0   | 13.0   | 15.0   | 17.0   | 19.0   |
-----|-----|
y      | -1.0   | 5.0    | 277675.0 | 6.848577635124539e+24 | 5.631725759119254e+102 | inf    | inf    | inf    | inf    | inf    |
Desired precision haven't been achieved yet. Changing parameters
h = 1.0
n = 20
```

```
i      | 0      | 1      | 2      | 3      | 4      | 5      | 6      | 7      | 8      | 9      | 10     | 11     | 12     | 13     | 14     |
x      | 1.0    | 2.0    | 3.0    | 4.0    | 5.0    | 6.0    | 7.0    | 8.0    | 9.0    | 10.0   | 11.0   | 12.0   | 13.0   | 14.0   | 15.0   |
-----|-----|
y      | -1.0   | -0.5   | -0.375 | -0.2871 | -0.2268 | -0.1848 | -0.1548 | -0.1327 | -0.1161 | -0.1033 | -0.0931 | -0.0848 | -0.078 | -0.0722 | -0.0672 |
Desired precision haven't been achieved yet. Changing parameters
h = 0.5
n = 40

i      | 0      | 1      | 2      | 3      | 4      | 5      | 6      | 7      | 8      | 9      | 10     | 11     | 12     | 13     | 14     |
x      | 1.0    | 1.5    | 2.0    | 2.5    | 3.0    | 3.5    | 4.0    | 4.5    | 5.0    | 5.5    | 6.0    | 6.5    | 7.0    | 7.5    | 8.0    |
-----|-----|
y      | -1.0   | -0.7188 | -0.5434 | -0.4297 | -0.3526 | -0.2981 | -0.2581 | -0.2276 | -0.2036 | -0.1843 | -0.1685 | -0.1551 | -0.1438 | -0.1341 | -0.1256 |
Desired precision haven't been achieved yet. Changing parameters
h = 0.25
n = 80

i      | 0      | 1      | 2      | 3      | 4      | 5      | 6      | 7      | 8      | 9      | 10     | 11     | 12     | 13     | 14     |
x      | 1.0    | 1.25   | 1.5    | 1.75   | 2.0    | 2.25   | 2.5    | 2.75   | 3.0    | 3.25   | 3.5    | 3.75   | 4.0    | 4.25   | 4.5    |
-----|-----|
y      | -1.0   | -0.8105 | -0.6781 | -0.5813 | -0.508 | -0.4507 | -0.4049 | -0.3675 | -0.3363 | -0.3101 | -0.2876 | -0.2682 | -0.2512 | -0.2363 | -0.223 |
```

