Федеральное государственное автономное образовательное учреждение высшего образования **«Национальный исследовательский университет ИТМО»**

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа по вычислительной математике №6

Численное дифференцирование

Преподаватель: Малышева Татьяна Алексеевна

Выполнила: Голованова Дарья Владимировна

Группа: Р3222

Санкт-Петербург,
2022г

Цель лабораторной работы:

Цель лабораторной работы: решить задачу Коши численными методами.

- Для исследования использовать:
- Одношаговые методы;
- Многошаговые методы.

Методы:

- **Одношаговые методы:**

1. Метод Эйлера;
2. **Усовершенствованный метод Эйлера;**
3. Метод Рунге-Кутта 4-го порядка.

- **Многошаговые методы:**

4. **Адамса;**
5. Милна.

Описание использованного метода:

1. **Метод Адамса** – многошаговый метод 4го порядка точности. Используемая в данном методе формула прогноза получена интегрированием обратной интерполяционной формулы Ньютона и имеет вид:

$$y_{i+1} = y_i + \frac{1}{24} h * (55 y_i{}' - 59 y'_{i-1} + 37 y'_{i-2} - 9 y'_{i-3})$$

На этапе коррекции используется формула:

$$y_{i+1} = y_i + \frac{1}{24} h * (9 y'_{i+1} + 19 y'_i - 5 y'_{i-1} + y'_{i-2})$$

Данный метод имеет четвёртый порядок точности, использует в качестве интерполяционного многочлена полином Лагранжа. А также ошибка, полученная на очередном шаге, не имеет тенденцию к экспоненциальному росту.

2. **Метод Эйлера** основан получении каждого следующего значение y из предыдущего.

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

Данный метод имеет большую погрешность, которая, к тому же, накапливается на каждом шаге. Порядок точности данного метода - первый.

Усовершенствованный метод Эйлера отличается от обычного тем, что значение правой части уравнения берется равным среднему арифметическому между $f(x_i; y_i)$ и $f(x_{i+1}; y_{i+1})$, т.е.

$$y_{i+1} = y_i + \frac{h}{2}\big(f(x_i; y_i) + f(x_{i+1} y_{i+1})\big)$$

затем вычисляется первое приближение $\tilde{y}_{i+1} = y_i + hf(x_i; y_i)$, затем подставляем значение в формулу выше и находим уточнённое значение

$$y_{i+1} = y_i + \frac{h}{2}(f(x_i; y_i) + f(x_{i+1}; \tilde{y}_{i+1}))$$

Данный метод точнее метода Эйлера и имеет второй порядок точности.

Листинг численного метода:

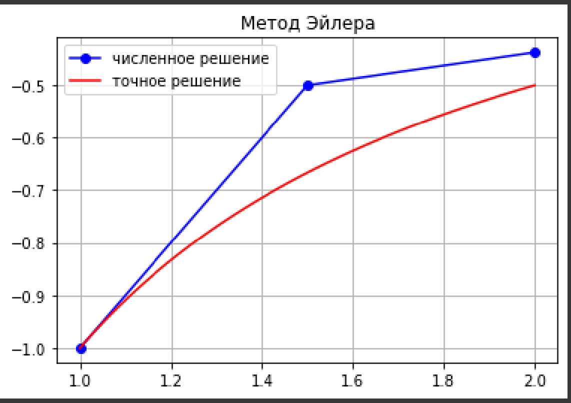
```
class EulerMethod(Method):
    def __init__(self):
        super().__init__()
        self.p = 2

    def get_name(self):
        return 'Метод Эйлера'

    def solve(self, fun, reference_func, const_c, a, b, y0, h, e):
        dots = self.process(fun, a, b, y0, h)
        dots_h2 = self.process(fun, a, b, y0, h / 2)
        if self._print_table(dots, dots_h2, reference_func, const_c):
            self.draw_graph(dots, reference_func, const_c, a, b)

    def process(self, fun, a, b, y0, h):
        dots = [(a, y0, fun(a, y0))]
        n = int((b - a) / h) + 1

        for i in range(1, n):
            try:
                x_prev = dots[i - 1][0]
                y_prev = dots[i - 1][1]
                y_cur = y_prev + h * (fun(x_prev, y_prev))
                x_cur = x_prev + h
                dots.append((x_cur, y_cur, fun(x_prev, y_prev)))
            except Exception:
                break
        return dots
```

```
class AdamsMethod(Method):
    def __init__(self):
        super().__init__()
        self.p = 4

    def get_name(self):
        return 'Метод Адамса'

    def solve(self, fun, reference_func, const_c, a, b, y0, h, e):
        dots = self.process(fun, a, b, y0, h, e)
        dots_h2 = self.process(fun, a, b, y0, h / 2, e)
        if self._print_table(dots, dots_h2, reference_func, const_c):
            self.draw_graph(dots, reference_func, const_c, a, b)

    def process(self, fun, a, b, y0, h, e):
        dots = [(a, y0)]
        fun_t = [fun(a, y0)]
        n = int((b - a) / h) + 1
```

```
        # Рунге-Кутта 4 порядка для нахождения 4 первых значений, чтобы запустить метод Адамса
        for i in range(1, 4):
            x_prev = dots[i - 1][0]
            y_prev = dots[i - 1][1]
            r1 = h * fun(x_prev, y_prev)
            r2 = h * fun(x_prev + h / 2, y_prev + r1 / 2)
            r3 = h * fun(x_prev + h / 2, y_prev + r2 / 2)
            r4 = h * fun(x_prev + h, y_prev + r3)

            x_cur = x_prev + h
            y_cur = y_prev + (r1 + 2 * r2 + 2 * r3 + r4) / 6

            dots.append((x_cur, y_cur))
            fun_t.append(fun(x_cur, y_cur))

        # Метод Адамса
        for i in range(4, n):
            try:
                x_cur = dots[i - 1][0] + h

                y_pred = dots[i - 1][1] + h / 24 * (
                        55 * fun_t[i - 1] - 59 * fun_t[i - 2] + 37 * fun_t[i - 3] - 9 * fun_t[i - 4])

                fun_t.append(fun(x_cur, y_pred))

                y_cor = dots[i - 1][1] + h / 24 * (9 * fun_t[i] + 19 * fun_t[i - 1] - 5 * fun_t[i - 2] + fun_t[i - 3])
                while e < abs(y_cor - y_pred):
                        y_pred = y_cor
                        fun_t[i] = fun(x_cur, y_pred)
                        y_cor = dots[i - 1][1] + h / 24 * (
                                9 * fun_t[i] + 19 * fun_t[i - 1] - 5 * fun_t[i - 2] + fun_t[i - 3])
            except Exception:
                break
            dots.append((x_cur, y_cor))

        return dots
```

Пример работы программы:

```
Выберите уравнение для задачи Коши:
1. y` = y + (1 + x) * y^2
2. y` = (x + 1)^3 - y
3. y` = xy
1
Интервал дифференцирования [a; b] (1 2): 1 2
Начальные условия y0 (-1): -1
Шаг дифференцирования (0.05): 0.5
Точность e (0.01): 0.1
Решение задачи Коши методом: Метод Эйлера
+-----+--------------+--------------+--------------+-----------------------+-------------------+
|   i |            x |         y(x) |        Рунге |   Погрешность с точным |   Точное значение |
+=====+==============+==============+==============+=======================+===================+
|   0 | 1.0000000000 | -1.0000000000 | 0.0000000000 |          0.0000000000 |     -1.0000000000 |
+-----+--------------+--------------+--------------+-----------------------+-------------------+
|   1 | 1.5000000000 | -0.5000000000 | 0.0403645833 |          0.1666666667 |     -0.6666666667 |
+-----+--------------+--------------+--------------+-----------------------+-------------------+
|   2 | 2.0000000000 | -0.4375000000 | 0.0115361728 |          0.0625000000 |     -0.5000000000 |
+-----+--------------+--------------+--------------+-----------------------+-------------------+
```




```
Решение задачи Коши методом: Метод Адамса
- - - - - - -   - - - -
Р е ш е н и я   н е т !
- - - - - - -   - - - -
```

Вывод:

Сравнение многошаговых методов:

Метод Адамса относится к многошаговым методам и представляет один из методов прогноза и коррекции, как и **метод Милна**. Разница между методами Адамса и Милна заключается в использовании разных формул прогноза и коррекции: в методе Адамса в качестве интерполяционного полинома используется полином Лагранжа, в методе Милна - полином Ньютона. Оба этих метода имеют четвертый порядок точности. Поскольку методы являются многошаговыми, для вычисления значения нам необходимо знать результаты нескольких предыдущих шагов, поэтому невозможно, если так можно выразиться, запустить метод: для этого необходимо предварительно получить одношаговыми методами первые три точки. Кроме того, методы прогноза и коррекции требуют дополнительного расхода памяти – поскольку для них требуются данные о предыдущих точках.

Сравнение одношаговых методов:

- **Метод Эйлера** основан получении каждого следующего значения y из предыдущего.

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

Данный метод имеет большую погрешность, которая, к тому же, накапливается на каждом шаге. Порядок точности данного метода - первый.

- **Усовершенствованный метод Эйлера** отличается от обычного тем, что значение правой части уравнения берется равным среднему арифметическому между $f(x_i; y_i)$ и $f(x_{i+1}; y_{i+1})$, т.е.

$$y_{i+1} = y_i + \frac{h}{2}\left(f(x_i; y_i) + f(x_{i+1} y_{i+1})\right)$$

затем вычисляется первое приближение $\tilde{y}_{i+1} = y_i + hf(x_i; y_i)$, затем подставляем значение в формулу выше и находим уточнённое значение

$$y_{i+1} = y_i + \frac{h}{2}(f(x_i; y_i) + f(x_{i+1}; \tilde{y}_{i+1}))$$

Данный метод точнее метода Эйлера и имеет второй порядок точности.

- **Метод Рунге-Кутта** имеет несколько разновидностей, различающихся порядком точности. В этих методах допускается вычисление правых частей не только в точках сетки, но и в некоторых промежуточных точках.

Рассмотрим **метод Рунге-Кутта 4-го порядка точности**. В данном методе вводятся 4 вспомогательные величины (k_0, k_1, k_2, k_3) и вычисление координат очередной точки сетки происходит исходя из известных координат предыдущей точки:

$$y_{i+1} = y_i + \frac{1}{6}(k_0 + 2 * k_1 + 2 * k_2 + k_4), i = 0, 1, \ldots$$

(формулы вычисления вспомогательных величин k_0, k_1, k_2, k_3 опущены). Таким образом, данный метод требует на каждом шаге четырехкратного вычисления правой части уравнения. Метод Рунге-Кутта требует большого объема вычислений, но имеет повышенную точность, что позволяет проводить вычисления с большим шагом.

При одинаковом шаге метод Эйлера и усовершенствованный метод Эйлера менее точные, в отличие от метода Рунге-Кутта четвёртого порядка.