Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе №5

по дисциплине «**Вычислительная математика**»

Автор: Баянов Равиль Динарович

Факультет: ПИиКТ

Группа: Р3234

Преподаватель: Перл О. В.

Санкт-Петербург, 2024

Оглавление

Описание метода

Метод Адамса-Башфорта – это явный, многошаговый метод для решения задачи Коши. Данный метод не является самостоятельным, так как нужно с помощью одношагового метода рассчитать разгоночные точки. Чаще всего для этой подзадачи используется метод Рунге-Кутты в силу того, что порядок точности одношагового метода должен быть выше или совпадать с порядком точности многошагового метода. Метод Адамса-Башфорта использует интерполяционный полином Лагранжа для аппроксимации решения задачи Коши с точностью до 4 порядка. Не будем рассматривать вычисление точек методом Рунге-Кутты сразу перейдём к вычислению точек в методе Адамса-Башфорта.

Пусть у нас заданы условия: $y' = f(x, y)$ и $y(x_0) = y_0$, для которых надо найти решение.

Экстраполяционная формула Адамса выглядит так:

$$\delta y_i = h y_i' + \frac{1}{2}\delta\left(h y_{i-1}'\right) + \frac{5}{12}\delta^2\left(h y_{i-2}'\right) + \frac{3}{8}\delta^3\left(h y_{i-3}'\right)$$

Формула метода Адамса-Башфорта:

$$y_{i+i}^{(0)} = y_i + \frac{h}{24}\left(55 y_i' - 59 y_{i-1}' + 37 y_{i-2}' - 9 y_{i-3}'\right)(i = 4,5, \ldots)$$

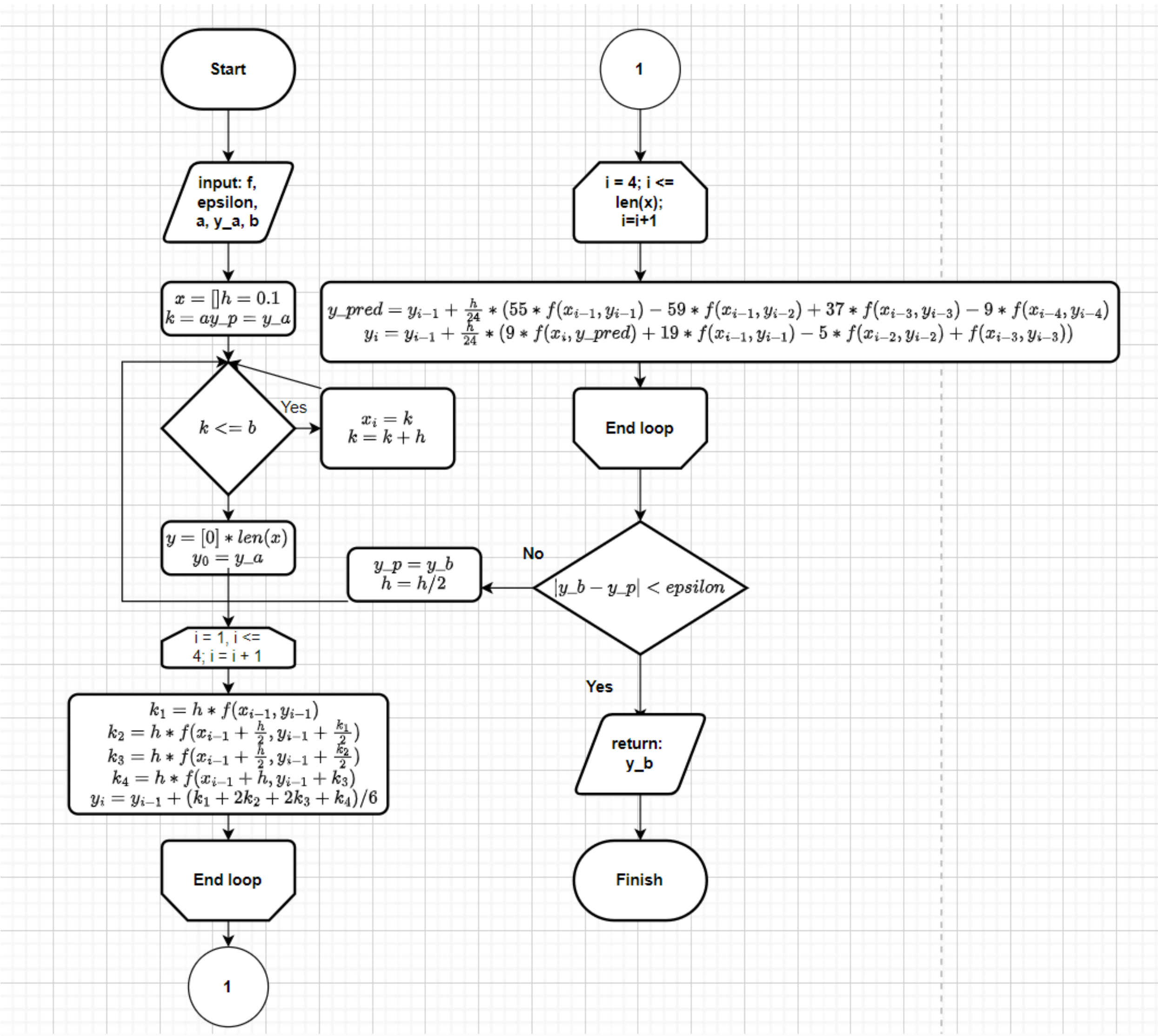
Таким образом, получив первые 4 точки методом Рунге-Кутты, мы можем перейти к методу Адамса-Башфорта.

На практике неявный метод Адамса-Мультона используется вместе с явным методом Адамса_Башфорта. Формула метода Адамса-Мультона выглядит так:

$$y_{i+1} = y_i + \frac{h}{24}\left(9 y_{i+1}' + 19 y_i' - 5 y_{i-1}' + y_{i-2}'\right)(i = 4,5, \ldots)$$

После вычисления начальных точек делаем прогноз для величин y_{i+1} с помощью явного метода Адамса-Башфорта. Затем корректируем величину y_{i+1} по формуле Адамса-Мультона. А теперь превратим этот метод в блок-схему и в код.

Блок-схема



Start
input: f, epsilon, a, y_a, b
x = []h = 0.1
k = ay_p = y_a
k <= b
Yes
x_i = k
k = k + h
y = [0] * len(x)
y_0 = y_a
i = 1, i <= 4; i = i + 1
k_1 = h * f(x_{i-1}, y_{i-1})
k_2 = h * f(x_{i-1} + h/2, y_{i-1} + k_1/2)
k_3 = h * f(x_{i-1} + h/2, y_{i-1} + k_2/2)
k_4 = h * f(x_{i-1} + h, y_{i-1} + k_3)
y_i = y_{i-1} + (k_1 + 2k_2 + 2k_3 + k_4)/6
End loop
1
1
i = 4; i <= len(x); i=i+1
y_pred = y_{i-1} + h/24 * (55 * f(x_{i-1}, y_{i-1}) - 59 * f(x_{i-1}, y_{i-2}) + 37 * f(x_{i-3}, y_{i-3}) - 9 * f(x_{i-4}, y_{i-4}))
y_i = y_{i-1} + h/24 * (9 * f(x_i, y_pred) + 19 * f(x_{i-1}, y_{i-1}) - 5 * f(x_{i-2}, y_{i-2}) + f(x_{i-3}, y_{i-3}))
End loop
|y_b - y_p| < epsilon
No
y_p = y_b
h = h/2
Yes
return: y_b
Finish

Исходный код метода на языке программирования Python

```python
class Result:

    def first_function(x: float, y: float):
        return math.sin(x)

    def second_function(x: float, y: float):
        return (x * y) / 2

    def third_function(x: float, y: float):
        return y - (2 * x) / y

    def fourth_function(x: float, y: float):
        return x + y

    def default_function(x: float, y: float):
        return 0.0

    # How to use this function:
    # func = Result.get_function(4)
    # func(0.01)
    def get_function(n: int):
        if n == 1:
            return Result.first_function
        elif n == 2:
            return Result.second_function
        elif n == 3:
            return Result.third_function
        elif n == 4:
            return Result.fourth_function
        else:
            return Result.default_function

    #
    # Complete the 'solveByAdams' function below.
    #
    # The function is expected to return a DOUBLE.
    # The function accepts following parameters:
    #  1. INTEGER f
    #  2. DOUBLE epsilon
    #  3. DOUBLE a
    #  4. DOUBLE y_a
    #  5. DOUBLE b
    #
    def runge_kutta(f, x0, y0, h):
        k1 = h * f(x0, y0)
```

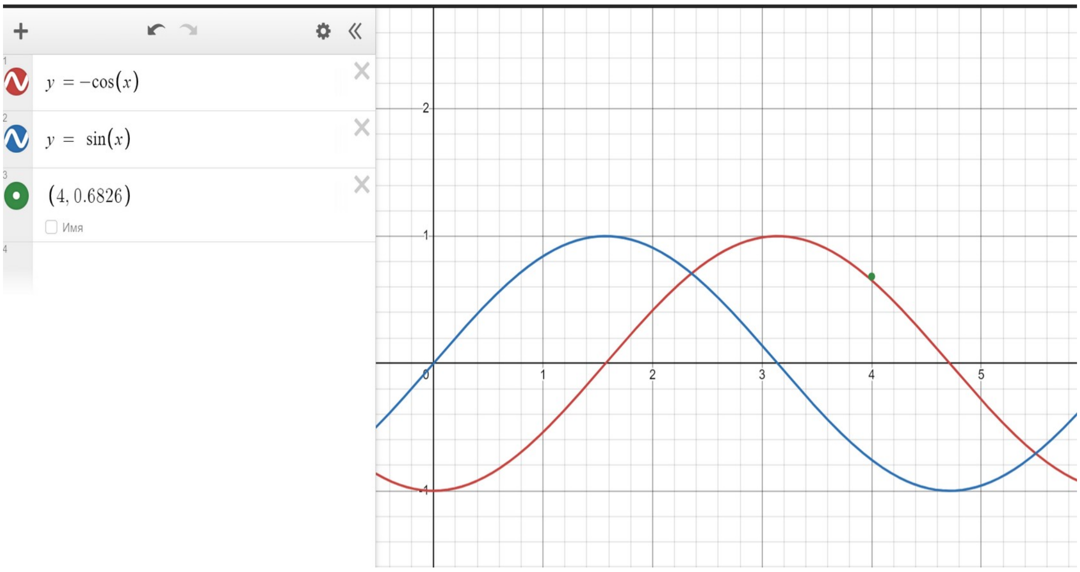
```
        k2 = h * f(x0 + h / 2, y0 + k1 / 2)
        k3 = h * f(x0 + h / 2, y0 + k2 / 2)
        k4 = h * f(x0 + h, y0 + k3)
        return y0 + (k1 + 2 * k2 + 2 * k3 + k4) / 6
    def solve(f, eps, a, y_a, b, h):
        x = []
        k = a
        while k <= b:
            x.append(k)
            k += h
        y = [0] * len(x)
        y[0] = y_a
        for i in range(1, 4):
            y[i] = Result.runge_kutta(f, x[0], y[0], h)
        for i in range(4, len(x)):
            y[i] = y[i - 1] + h / 24 * (
                        9 * f(x[i], y[i - 1] + h / 24 * (
                        55 * f(x[i - 1], y[i - 1]) - 59 * f(x[i - 2], y[i - 2]) + 37 * f(x[i - 3], y[i - 3]) - 9 * f(
                    x[i - 4], y[i - 4]))) + 19 * f(x[i - 1], y[i - 1]) - 5 * f(x[i - 2], y[i - 2]) + f(x[i - 3],
                    y[i - 3]))
        return y[-1]

    def solveByAdams(f, eps, a, y_a, b):
        f = Result.get_function(f)
        h = 0.1
        y_p = y_a
        while True:
            y_b = Result.solve(f, eps, a, y_a, b, h)
            if (abs(y_b - y_p) < eps):
                return y_b
            y_p = y_b
            h /= 2
            return y_b
```

Примеры работы программы

1) Первая функция:

```
1
0.001
0
-1
4
0.6862692530288825
```



2) Вторая функция:

```
2
0.01
3
4
5
158.30644506119552
```

3) Третья функция:

```
3
0.001
1
-20
30
-57966151392220.97
```

4) Четвёртая функция:

```
4
0.001
0.3
5
0.8
6.943688242155609
```

5) Четвёртая функция:

```
4
0.01
40
5
45
5028.076400243744
```

6) Default функция:

```
6
4
5
6
7
6.0
```

Вывод

Метод Адамса-Башфорта – это численный метод для решения обыкновенных дифференциальных уравнений. Он основан на идее использования интерполяции для вычисления приближённого значения функции в следующей точке. Этот метод является явным и многошаговым.

Сравним наш метод с другими:

- Точность: Метод Адамса-Башфорта имеет более высокий порядок точности, чем простые методы, такие как метод Эйлера. Например, метод Адамса-Башфорта четвертого порядка имеет порядок точности 4, в то время как метод Эйлера имеет порядок точности 1. Однако, метод Рунге-Кутты также имеет высокий порядок точности, и он может быть более точным, чем метод Адамса-Башфорта в некоторых случаях.

- Устойчивость: Метод Адамса-Башфорта может быть неустойчивым при решении жестких дифференциальных уравнений, в то время как метод Рунге-Кутты и некоторые другие методы более устойчивы. Это означает, что метод Адамса-Башфорта может давать неверные результаты при решении некоторых типов уравнений.

- Вычислительная сложность: Метод Адамса-Башфорта требует вычисления дополнительных значений функции, что может увеличить вычислительную сложность метода. Метод Эйлера проще и требует меньше вычислений, но он менее точен. Метод Рунге-Кутты требует большего количества вычислений, но он более точен и устойчив.

Метод Адамса-Башфорта может быть неустойчивым в некоторых случаях, но требует не очень много вычислений по сравнению с другими методами.

Данный метод не стоит применять для жёстких функций, так как метод крайне неустойчив. Также для метода Адамса-Башфорта требуется наличие начальных условий.

Для метода Адамса-Башфорта порядка k, количество арифметических операций, необходимых для вычисления одного шага, пропорционально k. Это означает, что вычислительная сложность одного шага метода Адамса-Башфорта порядка k равна O(k).

Общая вычислительная сложность метода Адамса-Башфорта зависит от количества шагов, необходимых для решения дифференциального уравнения. Если N - количество шагов, необходимых для решения уравнения, то общая вычислительная сложность метода Адамса-Башфорта порядка k будет равна O(Nk).

Ошибка метода сильно зависит от начальных условий, от гладкости решения и от поведения дифференциального уравнения. В целом, метод Адамса-Башфорта имеет относительно небольшую численную ошибку за счёт высокого порядка точности.

Таким образом, метод Адамса-Башфорта один из многих методов для решения обыкновенных дифференциальных уравнений. Он очень часто используется вместе с методом Адамса-Мультона для корректировки ответа. Данный метод очень эффективен и точен.