МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПИИКТ

ЛАБОРАТОРНАЯ РАБОТА № 5

по дисциплине

‘Вычислительная математика’

Вариант:

Метод Адамса-Башфорта

Выполнил:
Студент группы P3233
Хасаншин Марат Айратович

УНИВЕРСИТЕТ ИТМО

Санкт-Петербург, 2024

Описание численного метода:

Метод Адамса-Башфорта является многошаговым методом для решения задачи Коши.

Для данного метода необходимы несколько дополнительных точек, поэтому для этого используются другие метода (часто одношаговые). В моем случае – метод Рунге-Кутты.

В методе Рунге-Кутты для нахождения следующего значения y мы используем следующие формулы:

$$\begin{cases} k_1 = hf(x_i, y_i) \\ k_2 = hf(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}) \\ k_3 = hf\left(x_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right) \\ k_4 = hf(x_i + h, y_i + k_3) \end{cases}$$

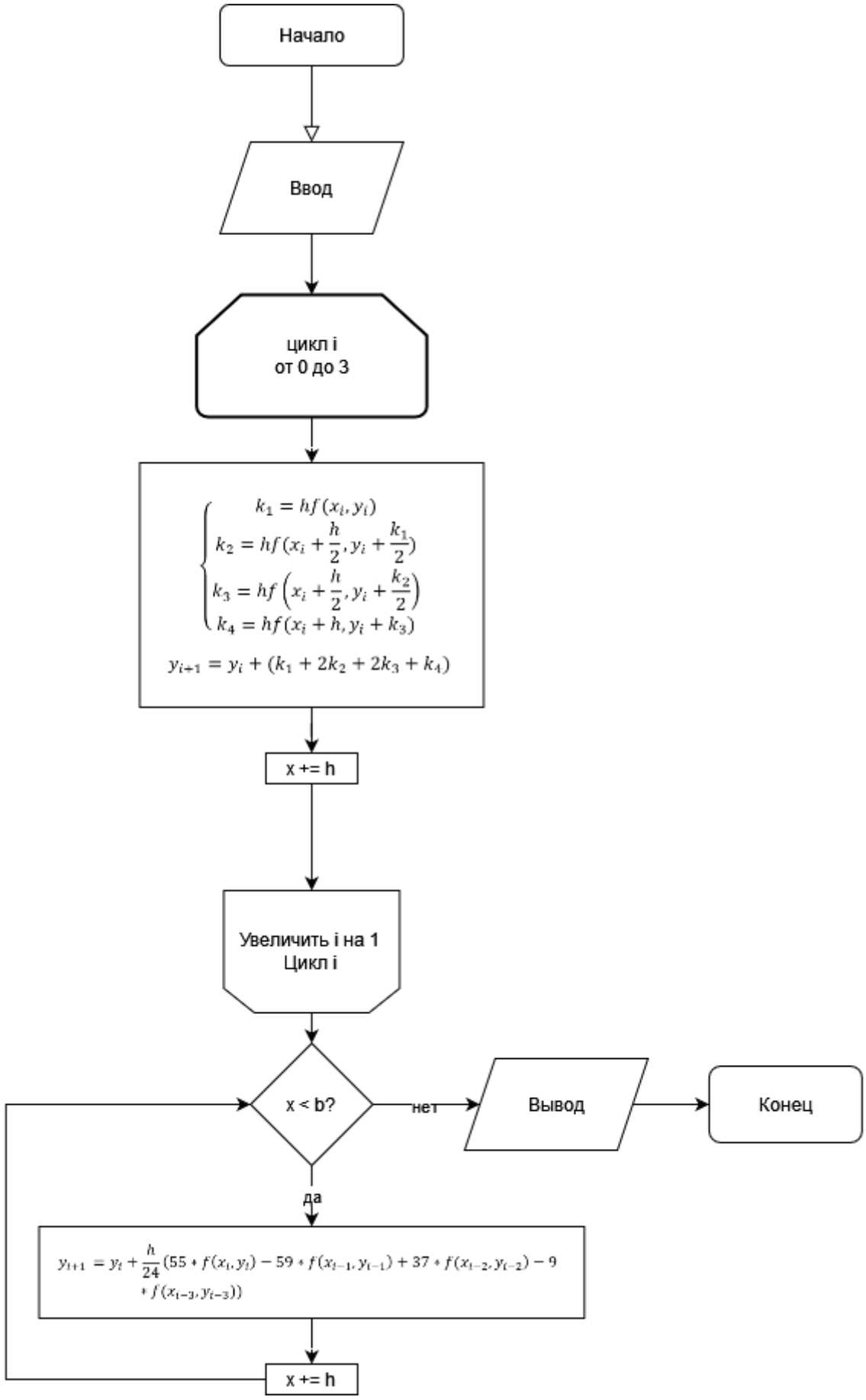
$$y_{i+1} = y_i + (k_1 + 2k_2 + 2k_3 + k_4)$$

Используя данные формулы, находим 3 дополнительные разгоночные точки для метода Адамса-Башфорта.

Двигаясь с некоторым шагом (в моем случае 0.01) до крайнего значения, высчитывая следующее значение по формуле:

$$y_{i+1} = y_i + \frac{h}{24}(55 * f(x_i, y_i) - 59 * f(x_{i-1}, y_{i-1}) + 37 * f(x_{i-2}, y_{i-2}) - 9 * f(x_{i-3}, y_{i-3}))$$

Блок-схема:


Начало
Ввод
цикл i
от 0 до 3
$k_1 = hf(x_i, y_i)$
$k_2 = hf(x_i + \frac{h}{2}, y_i + \frac{k_1}{2})$
$k_3 = hf\left(x_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right)$
$k_4 = hf(x_i + h, y_i + k_3)$
$y_{i+1} = y_i + (k_1 + 2k_2 + 2k_3 + k_4)$
x += h
Увеличить i на 1
Цикл i
x < b?
нет
Вывод
Конец
да
$y_{i+1} = y_i + \frac{h}{24}(55 * f(x_i, y_i) - 59 * f(x_{i-1}, y_{i-1}) + 37 * f(x_{i-2}, y_{i-2}) - 9 * f(x_{i-3}, y_{i-3}))$
x += h

Код:

Рунге-Кутта:

```
def runge_kutta(x, y, h, f):
    k1 = h * f(x, y)
    k2 = h * f(x + 0.5 * h, y + 0.5 * k1)
    k3 = h * f(x + 0.5 * h, y + 0.5 * k2)
    k4 = h * f(x + h, y + k3)
    return y + (k1 + 2 * k2 + 2 * k3 + k4) / 6
```

Адамса-Башфорта:

```
def solveByAdams(f, epsilon, a, y_a, b):

    h = 0.01
    func = Result.get_function(f)

    x_values = []
    x_values.append(a)
    y_values = []
    y_values.append(y_a)

    for i in range(3):
        y = Result.runge_kutta(x, y, h, f)
        x += h
        y_values.append(y)
        x_values.append(x)

    while (x < b):
        if (x + h > b):
            h = b - x
        y = y_values[-1] + h/24 * (55 * func(x_values[-1], y_values[-1]) - 59
* func(x_values[-2], y_values[-2]) + 37 * func(x_values[-3],y_values[-3]) - 9
* func(x_values[-4],y_values[-4]))
        y_values.append(y)
        x += h
        x_values.append(x)

    return y
```

Примеры работы программы:

№1:

```
1
0.01
1
3
5
3.2566401195008763
```

№2:

```
2
0.01
1
5
3
36.945277549240174
```

№3:

```
3
0.01
5
10
2
10.274407654055725
```

№4:

```
4
0.01
5
-9
-3
-9.121363601852886
```

№5:

```
5
0.01
1
3
5
3.0
```

Вывод:

В тестах 1 - 4 проверена работа программы с различными функциями. Так же по ответам видно, что погрешность небольшая.
В тесте 5 продемонстрирована работа с дефолтной функцией.

По сравнению с другими методами можно выделить следующие плюсы и минусы.

Минусы:

- Требуется реализация другого метода для расчета разгоночных точек
- Необходимо хранить предыдущие значения

Плюсы:

- Высокая точность метода
- Для вычисления используются только предыдущие точки и значения

Алгоритм может быть применен, если у системы есть решение и нет 0 элементов на главной диагонали.

Алгоритмическая сложность зависит от выбранного шага – $O(\frac{b-a}{h})$, где b – конец промежутка, a – начало, h – выбранный шаг.

Численная погрешность зависит от выбранного шага и точности. Порядок точности – $O(h^4)$.