ИТМО

ПИИКТ

Лабораторная работа №4

“Вычислительная математика”

Группа P3201

Метод Адамса

Выполнил: Братчиков Иван Станиславович

Приняла: Перл Ольга Вячеславовна

Санкт-Петербург

2020

Описание метода:

Метод Адамса имеет четвёртый порядок точности. Используемая в нём формула прогноза получена интегрированием обратной интерполяционной формулы Ньютона и имеет вид:

$$y_{i+1} = y_i + \frac{1}{24}h \cdot (55y_i' - 59y_{i-1}' + 37y_{i-2}' - 9y_{i-3}') + O(h^5),$$

$$где\ O(h^5) = \frac{251}{720}h^5 y^{(5)}.$$

На этапе коррекции используется формула:

$$y_{i+1} = y_i + \frac{1}{24}h \cdot (9y_{i+1}' + 19y_i' - 5y_{i-1}' + y_{i-2}') + O(h^5),$$

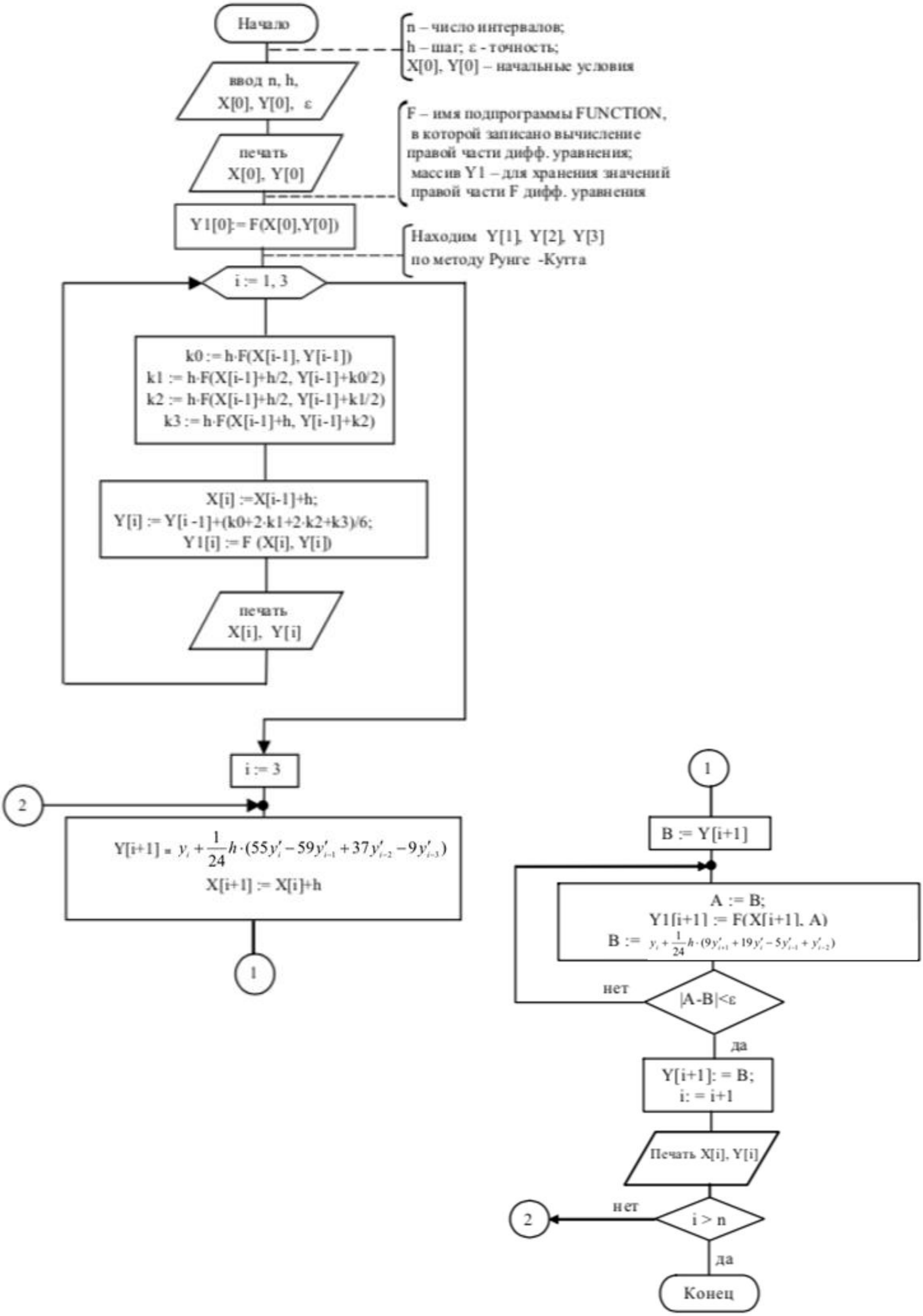
$$где\ O(h^5) = -\frac{19}{720}h^5 y^{(5)}.$$

Код вычислительного метода:

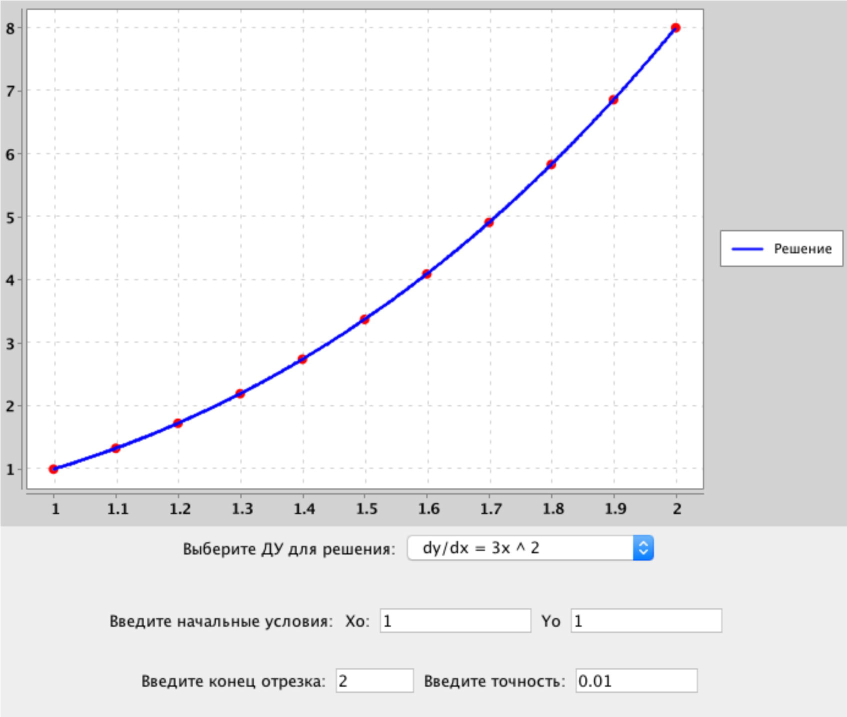
```
private void setInitialData(double[] xData, double[] yData, double h){
    for(int i = 1; i < 4; i++){
        double x = xData[i - 1];
        double y = yData[i - 1];

        double k0 = function.getValue(x, y);
        double k1 = function.getValue(x + h / 2, y + h / 2 * k0);
        double k2 = function.getValue(x + h / 2, y + h / 2 * k1);
        double k3 = function.getValue(x + h, y + h * k2);

        yData[i] = y + h / 6 * (k0 + 2 * k1 + 2 * k2 + k3);
    }
}

public double[][] solve(double x0, double y0, double endPoint, double accuracy){
    int intervals = 5;
    double[] xData;
    double[] yData;
    double averageError;
    double segmentLength = endPoint - x0;
    do {
        intervals *= 2;
        if(intervals > 10000000)
            throw new SolutionException("Достигнут максимум разбиений");
        double h = segmentLength / intervals;

        xData = new double[intervals + 1];
        xData[0] = x0;
        yData = new double[intervals + 1];
        yData[0] = y0;

        for(int i = 1; i < xData.length; i++)
            xData[i] = xData[i - 1] + h;
        setInitialData(xData, yData, h);
```

```
        double[] error = new double[yData.length - 4];
        System.out.println(Arrays.toString(yData));
        System.out.println(yData.length);
        for(int i = 4; i < yData.length; i++){
            double prediction = yData[i - 1] + (1 * h / 24) * (55 * function.getValue(xData[i - 1], yData[i - 1])
                    - 59 * function.getValue(xData[i - 2], yData[i - 2])
                    + 37 * function.getValue(xData[i - 3], yData[i - 3]) - 9 * function.getValue(xData[i-4], yData[i-4]));

            yData[i] = yData[i-1] + h / 24 * (- 5 * function.getValue(xData[i - 2], yData[i - 2])
                    + 19 * function.getValue(xData[i - 1], yData[i - 1])
                    + 9 * function.getValue(xData[i], prediction) + function.getValue(xData[i - 3], yData[i - 3]));
            if(!Double.isFinite(yData[i]))
                throw new SolutionException(String.format("При аргументе, лежащем внутри отрезка от %f до %f, " +
                        "значения функции выходят за пределы допустимых значений",
                        Math.min(x0, endPoint), Math.max(x0, endPoint)));
            error[i - 4] = Math.abs(yData[i] - prediction);
        }
        averageError = Arrays.stream(error).average().orElse(0);
    } while (averageError >= accuracy);
    return new double[][]{xData, yData};
}
```

Блок схема:


Начало
n – число интервалов;
h – шаг; ε - точность;
X[0], Y[0] – начальные условия
ввод n, h,
X[0], Y[0], ε
F – имя подпрограммы FUNCTION,
в которой записано вычисление
правой части дифф. уравнения;
массив Y1 – для хранения значений
правой части F дифф. уравнения
печать
X[0], Y[0]
Y1[0]:= F(X[0],Y[0])
Находим Y[1], Y[2], Y[3]
по методу Рунге -Кутта
i := 1, 3
k0 := h·F(X[i-1], Y[i-1])
k1 := h·F(X[i-1]+h/2, Y[i-1]+k0/2)
k2 := h·F(X[i-1]+h/2, Y[i-1]+k1/2)
k3 := h·F(X[i-1]+h, Y[i-1]+k2)
X[i] :=X[i-1]+h;
Y[i] := Y[i -1]+(k0+2·k1+2·k2+k3)/6;
Y1[i] := F (X[i], Y[i])
печать
X[i], Y[i]
i := 3
2
Y[i+1] = $y_i + \frac{1}{24}h\cdot(55y'_i - 59y'_{i-1} + 37y'_{i-2} - 9y'_{i-3})$
X[i+1] := X[i]+h
1
1
B := Y[i+1]
A := B;
Y1[i+1] := F(X[i+1], A)
B := $y_i + \frac{1}{24}h\cdot(9y'_{i+1} + 19y'_i - 5y'_{i-1} + y'_{i-2})$
нет
|A-B|<ε
да
Y[i+1]: = B;
i: = i+1
Печать X[i], Y[i]
нет
2
i > n
да
Конец

Пример:



Вывод:

Методы Адамса k-го порядка требуют предварительного вычисления решения в k начальных точках. Для вычисления начальных значений обычно используют одношаговые методы, например, 4-стадийный метод Рунге — Кутты 4-го порядка точности. Расчёты выполняются так же, как и по методу Милна, однако в отличие от последнего ошибка, внесённая на каком-либо шаге, не имеет тенденции к экспоненциальному росту.