

Университет ИТМО

Факультет ПИиКТ

Вычислительная математика

Лабораторная работа №4

Решение задачи Коши методом Милна

Выполнила: Наумова Надежда

Группа Р3201

Преподаватель: Перл Ольга Вячеславовна

Санкт-Петербург

2020 г.

Метод Милна относится к многошаговым методам и представляет один из методов прогноза и коррекции.

Решение в следующей точке находится в два этапа. На первом этапе осуществляется по специальной формуле прогноз значения функции, а затем на втором этапе - коррекция полученного значения. Если полученное значение  $y$  после коррекции существенно отличается от спрогнозированного, то проводят еще один этап коррекции. Если опять имеет место существенное отличие от предыдущего значения (т.е. от предыдущей коррекции), то проводят еще одну коррекцию и т.д. Однако очень часто ограничиваются одним этапом коррекции.

Пусть для уравнения  $y' = f(x, y)$  кроме начального условия  $y(x_0) = y_0$  известен "начальный отрезок" то есть значения искомой функции  $y(x_i) = y_i$  в точках  $x_i = x_0 + i \cdot h$ , ( $i = 1, 2, 3$ ), данные значения можно найти каким-либо одношаговым методом (в дальнейшем используется Метод Рунге-Кутты 4-го порядка).

Метод Милна имеет следующие вычислительные формулы:

а) этап предположения (прогноза):

$$(y_{i+1})^{prog} = y_{i-3} + 4 \cdot \frac{h}{3} \cdot (2f_{i-2} - f_{i-1} + 2f_i),$$

где для компактности записи использовано следующее обозначение  $f_i = f(x_i, y_i)$ ;

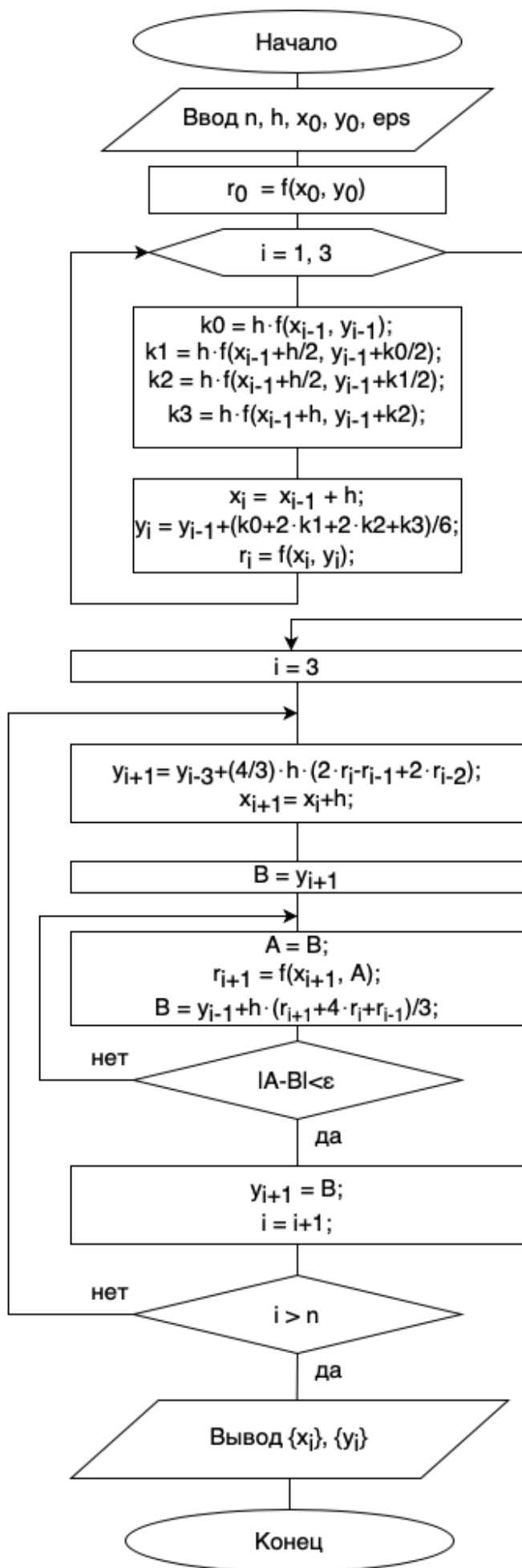
б) этап коррекции:

$$(y_{i+1})^{corr} = y_{i-1} + \frac{h}{3} \cdot (f_{i+1} + 4f_i + (y_{i+1})^{prog}),$$

Абсолютная погрешность определяется по формуле

$$\epsilon \approx \frac{(y_{i+1})^{corr} - (y_{i+1})^{prog}}{29}$$

# Блок-схема



Листинг численного метода

```
// RungeKuttaMethod.java
public RungeKuttaMethod(double x0, double y0, double xn,
    double accuracy, Function function) {
//    this.step = Math.pow(accuracy, 0.25);
    this.step = accuracy;
    this.function = function;
    values = new ArrayList<>();
    values.add(new Pair<>(x0, y0));
    double y_next = y0;
    for (double i = x0; i < xn; i += step) {
        y_next = getNextYValue(i, y_next);
        values.add(new Pair<>(i+step, y_next));
    }
}

private double getNextYValue(double x, double y) {
    double k0, k1, k2, k3, delta;
    k0 = function.getValue(x, y);
    k1 = function.getValue(x + step/2, y + k0 * step/2);
    k2 = function.getValue(x + step/2, y + k1 * step/2);
    k3 = function.getValue(x + step, y + k2 * step);
    delta = step * (k0 + 2 * k1 + 2 * k2 + k3)/6;
    return y + delta;
}

// MilneMethod.java
public MilneMethod(double x0, double y0, double xn, double accuracy,
    Function function) {
    values = new RungeKuttaMethod(x0, y0, xn, accuracy, function).
        getValues(4);
    this.function = function;
//    this.step = Math.pow(accuracy, 0.25);
    this.step = accuracy;
    this.accuracy = accuracy;
    this.xn = xn;
    this.currentX = x0 + 3 * this.step;
    calculate();
}

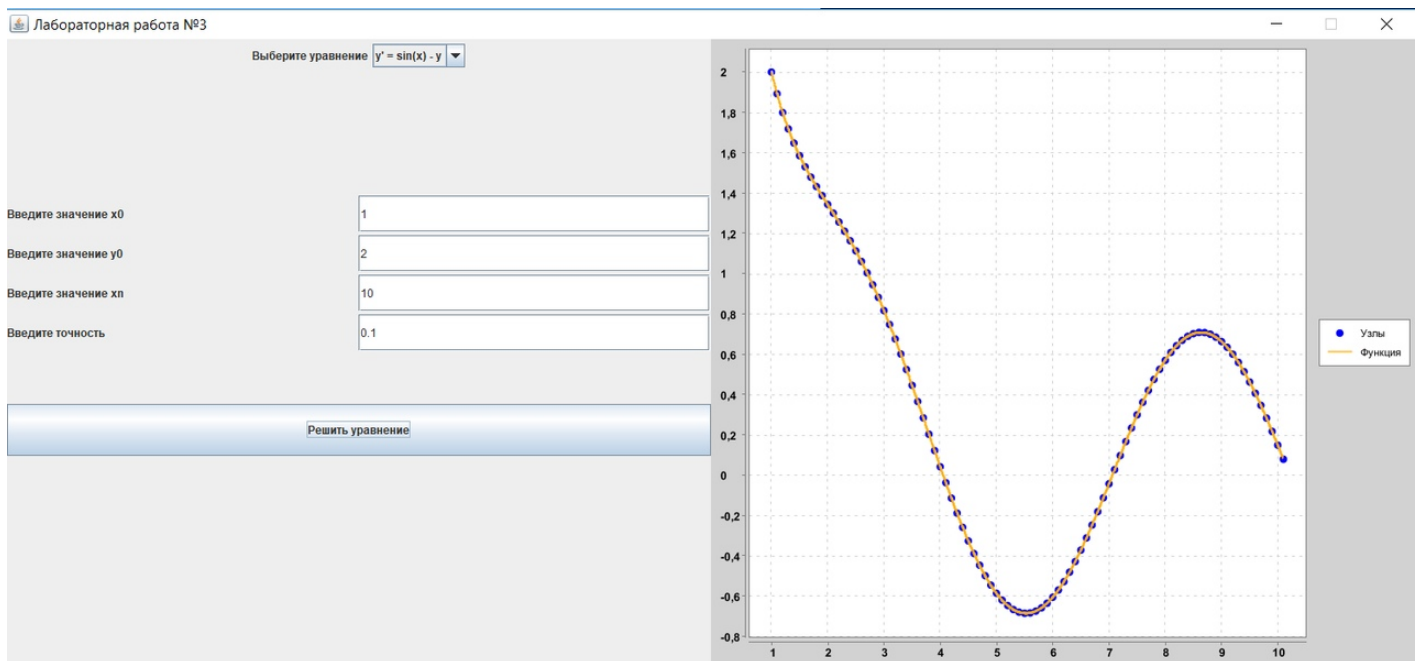
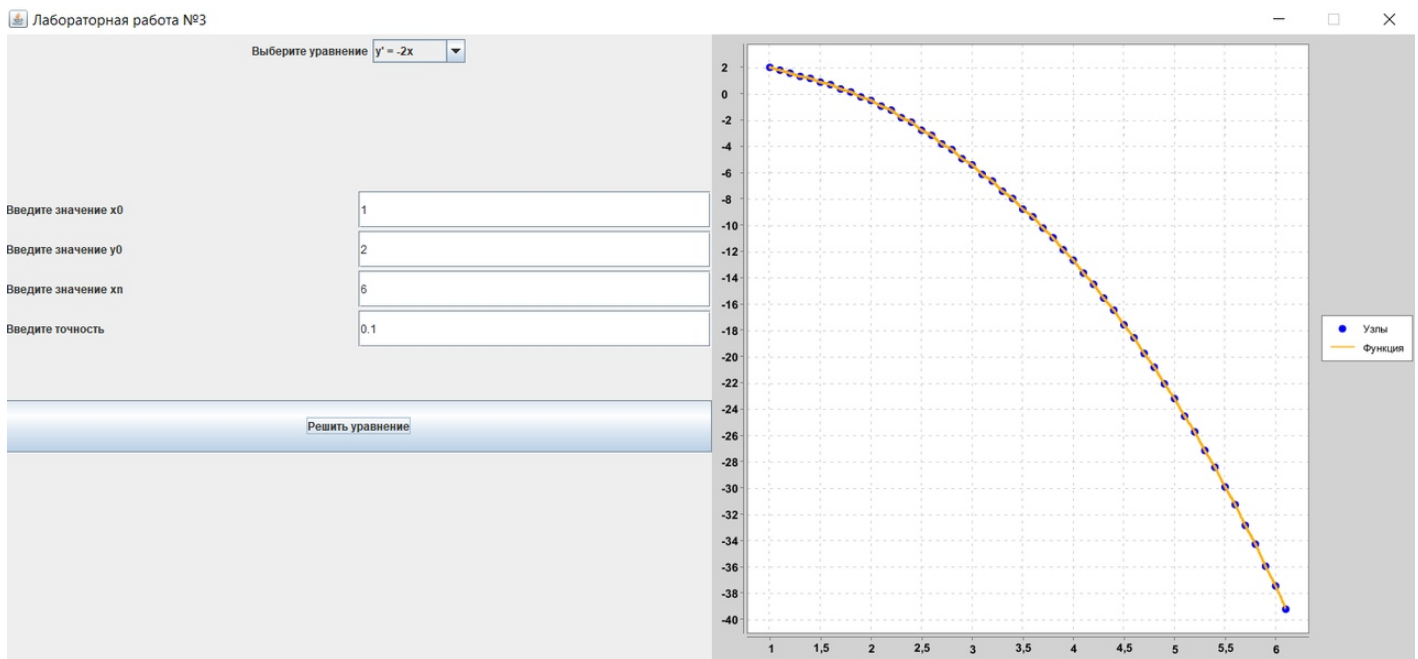
private void calculate() {
    int i = 3;
    while (currentX < xn) {
        i ++;
        currentX += step;
    }
}
```

```

double f3, f2, f1, f;
f3 = function.getValue(values.get(i - 3).getKey(),
    values.get(i - 3).getValue());
f2 = function.getValue(values.get(i - 2).getKey(),
    values.get(i - 2).getValue());
f1 = function.getValue(values.get(i - 1).getKey(),
    values.get(i - 1).getValue());
f = function.getValue(values.get(i - 1).getKey() + step,
    values.get(i - 4).getValue() +
        4 * step * (2 * f3 - f2 + 2 * f1)/3);
double y = f;
do {
    f = y;
    y = values.get(i - 2).getValue() + step*(f + 4 * f1 + f2)/3;
} while (Math.abs(f - y)/29 > accuracy);
values.add(new Pair<>(values.get(i - 1).getKey()+step, y));
}
}

```

# Примеры



Выберите уравнение  $y' = -2x$ Введите значение  $x_0$ 

-3

Введите значение  $y_0$ 

4

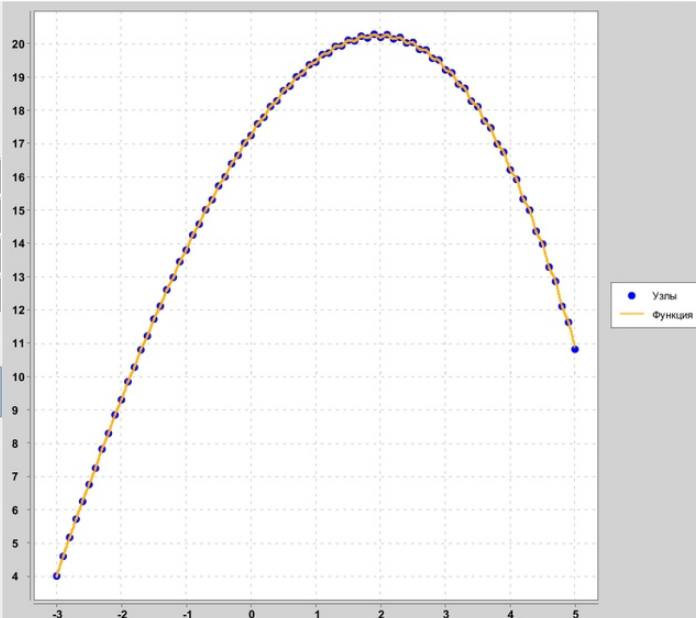
Введите значение  $x_1$ 

5

Введите точность

0.1

Решить уравнение



Вывод: **метод Милна** относится к многошаговым методам и представляет один из методов прогноза и коррекции, как и **метод Адамса**. Разница между методами Адамса и Милна заключается в использовании разных формул прогноза и коррекции: в методе Милна в качестве интерполяционного полинома используется полином Ньютона, в методе Адамса - полином Лагранжа. 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.

Далее перейдем к сравнению одношаговых методов.

**Метод Эйлера** основан на получении каждого следующего значения  $y$  из предыдущего:

$$y_{i+1} = y_i + h \cdot f(x_i; y_i).$$

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  $f(x_{i+1}; y_{i+1})$ , то есть

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

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

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

Данный метод точнее метода Эйлера и имеет второй порядок точности.

**Метод Рунге-Кутты** 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-Кутты 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:

$$y_{i+1} = y_i + \frac{1}{6} \cdot (k_0 + 2 \cdot k_1 + 2 \cdot k_2 + k_3), i = 0, 1, \dots$$

(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-Кутты 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-Кутты четвертого порядка.