Университет ИТМО, факультет ПИиКТ

Лабораторная работа №3

Дисциплина: Вычислительная математика

Вариант 19

Выполнил: Щелыкалов Виктор

Группа: Р3214

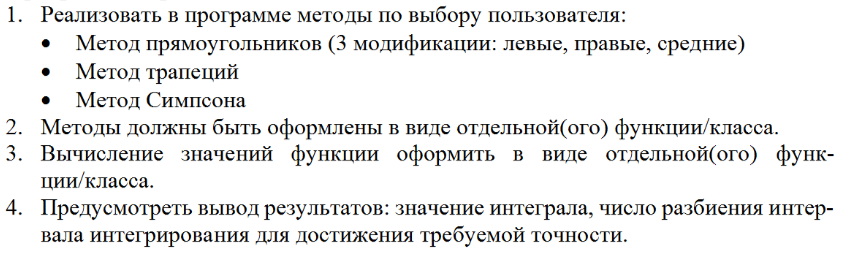
Преподаватель:

Малышева Татьяна Алексеевна

Санкт-Петербург, 2020 год

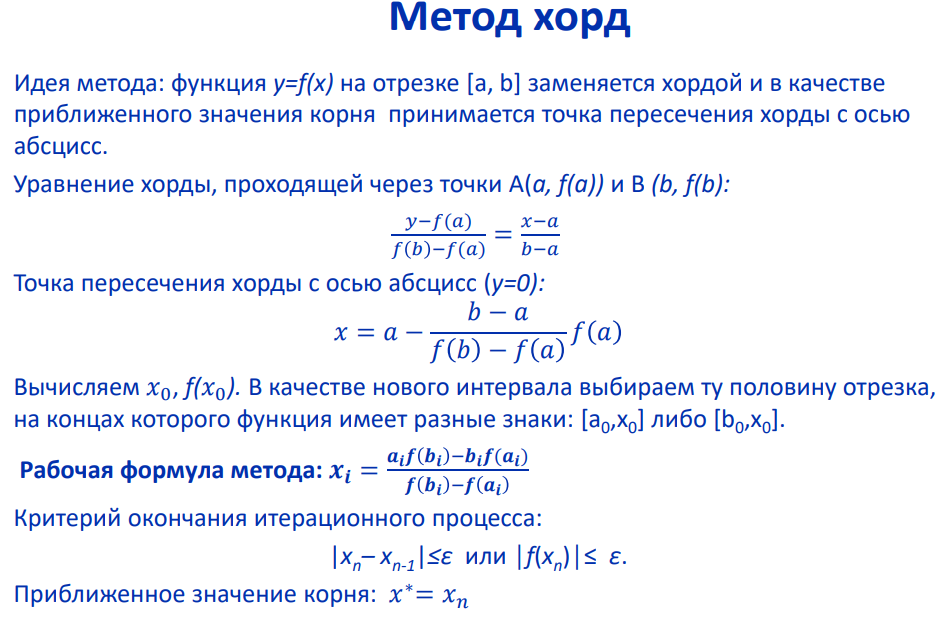
**Цель работы:** решить нелинейное уравнение

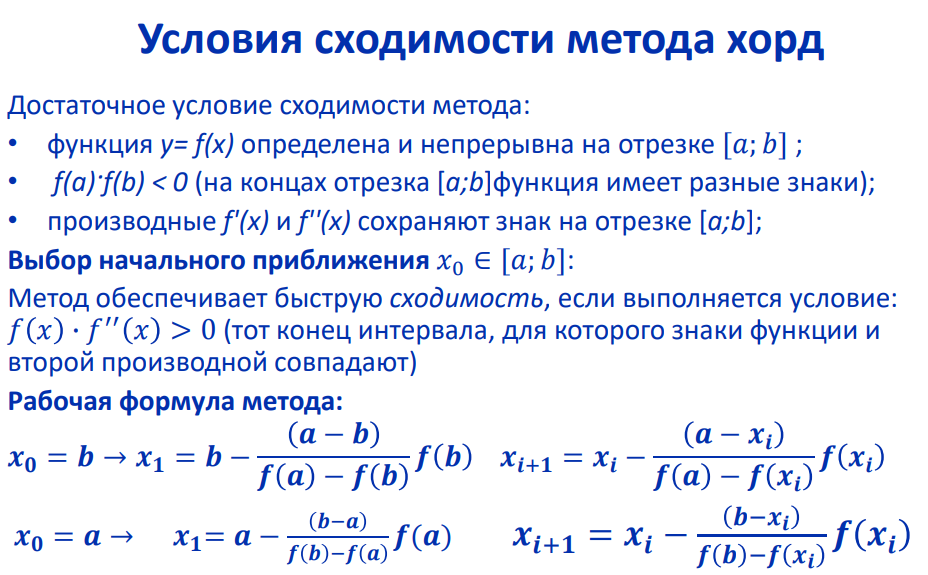
**Текст задания:**

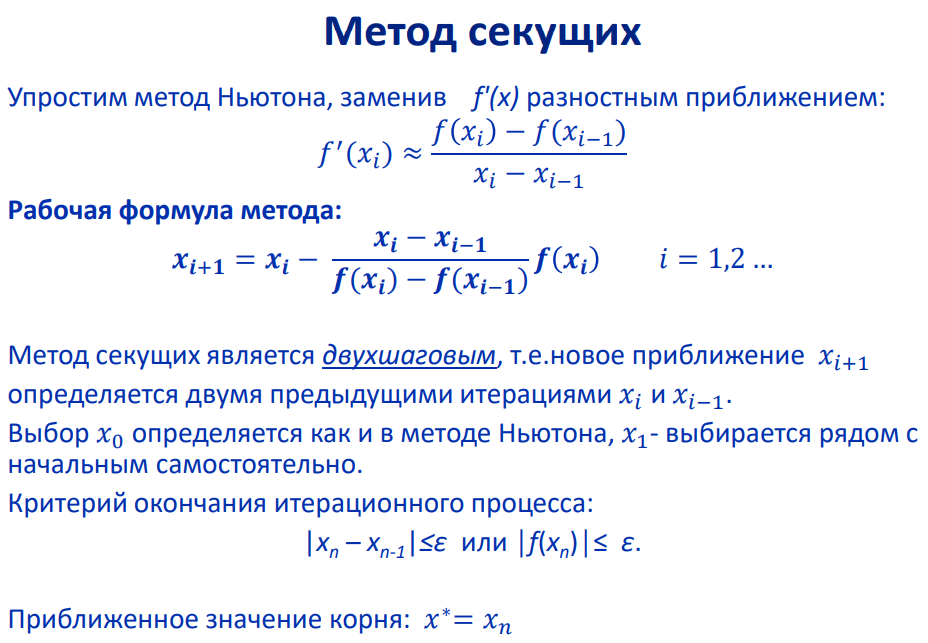


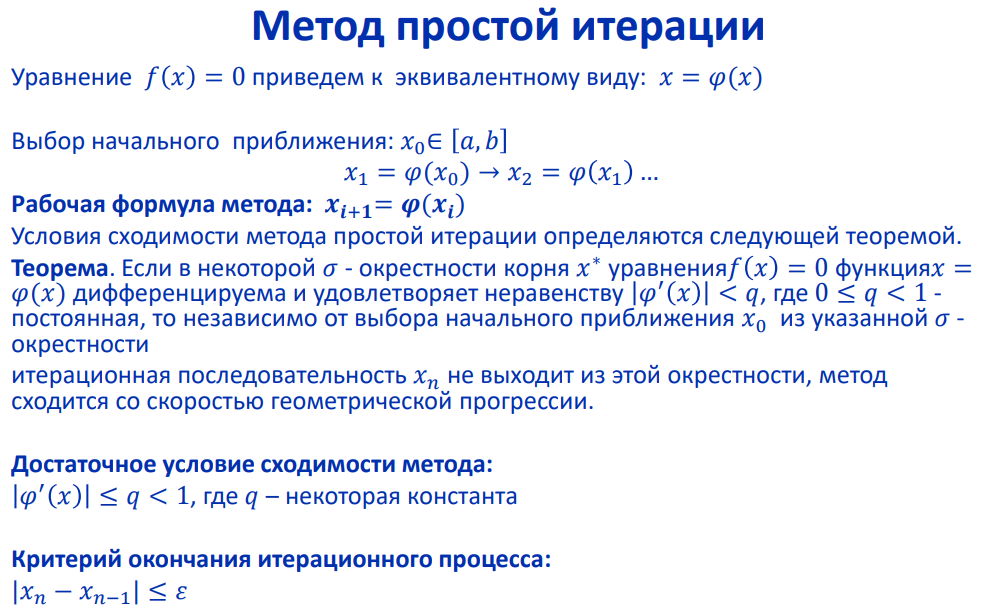
**Ход работы:**

В соответствии с вариантом мне необходимо реализовать три метода: метод секущих, метод хорд, метод простой итерации. Остановимся подробнее на каждом из них.



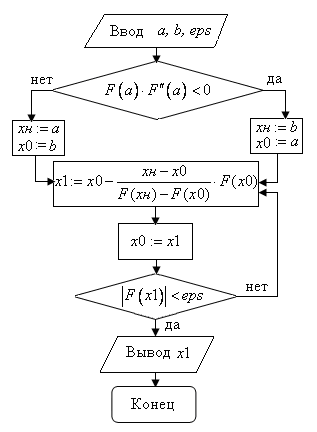




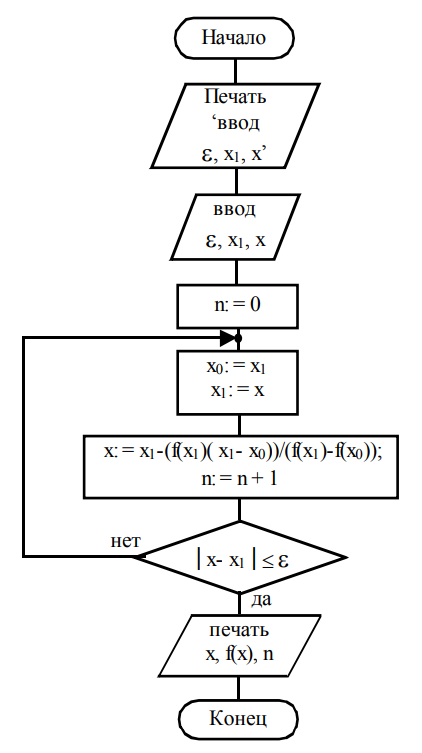


*Соответствующие блок-схемы:*

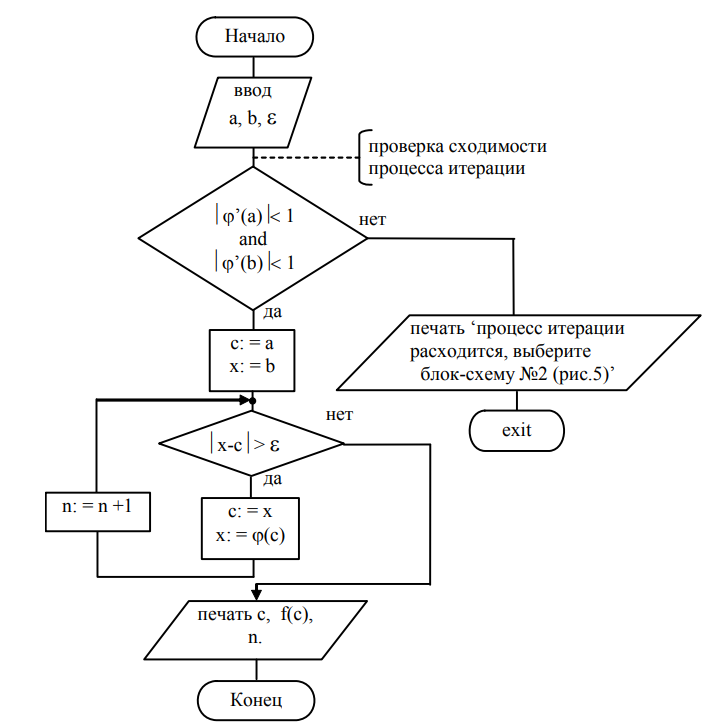
Метод хорд



Метод секущих:



Метод простой итерации:



**Листинг:**

Main.java

import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.chart.LineChart;

import javafx.scene.chart.NumberAxis;

import javafx.scene.chart.XYChart;

import javafx.stage.Stage;

import java.io.BufferedReader;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.util.Scanner;

public class Main extends Application {

public static void main(String[] args) {

launch(args);

}

@Override

public void start(Stage stage) throws Exception {

Scanner scanner = new Scanner(System.in);

System.out.println("Входные данные из:");

System.out.println("1 - файла");

System.out.println("2 - консоли");

int inpWay = scanner.nextInt();

double a = 0;

double b = 0;

double e = 0;

if (inpWay == 2) { // input from console

System.out.println("Введите начало интервала, конец интервала и погрешность через пробел:");

a = scanner.nextDouble();

b = scanner.nextDouble();

e = scanner.nextDouble();

} else if (inpWay == 1) { //input from file

System.out.println("Введите путь к файлу:");

//input-format of file: <start of the interval> <end of the interval> <epsilon>

//use dot in fractional number in file

String path = scanner.next().trim(); // file's path

String line = null;

try (BufferedReader reader = new BufferedReader(new FileReader(path))) {

line = reader.readLine();

} catch (FileNotFoundException e1) {

System.out.println("Файл не найден!");

} catch (IOException e2) {

System.out.println("Не удалось открыть файл!");

}

a = Double.parseDouble(line.split(" ")[0]);

b = Double.parseDouble(line.split(" ")[1]);

e = Double.parseDouble(line.split(" ")[2]);

} else {

System.out.println("Некорректный выбор!");

}

System.out.println("Записать ответ в:");

System.out.println("1 - файл");

System.out.println("2 - консоль");

int outWay = scanner.nextInt();

if (outWay == 2) { //consoles output

System.out.println("Meтод Хорд: " + Methods.horda(a, b, e));

System.out.println("Meтод Секущих: " + Methods.sekushie(a, b, e));

System.out.println("Meтод Простой итерации: " + Methods.iteraz(a, b, e));

} else if (outWay == 1) { // files output

System.out.println("Введите путь к файлу:");

String path = scanner.next().trim(); // file's path

try (FileWriter writer = new FileWriter(path, true)) {

writer.write("\nMeтод Хорд: " + Methods.horda(a, b, e) + "\n");

writer.write("Meтод Секущих: " + Methods.sekushie(a, b, e) + "\n");

writer.write("Meтод Простой итерации: " + Methods.iteraz(a, b, e) + "\n");

writer.flush();

} catch (IOException ex) {

System.out.println("Файл не найден!");

}

} else {

System.out.println("Некорректный выбор!");

}

//вывод графика

stage.setTitle("График");

final NumberAxis xAxis = new NumberAxis();

final NumberAxis yAxis = new NumberAxis();

xAxis.setLabel("X");

yAxis.setLabel("Y");

final LineChart<Number, Number> lineChart =

new LineChart<Number, Number>(xAxis, yAxis);

XYChart.Series series = new XYChart.Series();

double x = a - 15;

while(x < b + 10) {

series.getData().add(new XYChart.Data(x, Methods.f(x)));

x = x + 1;

Scene scene = new Scene(lineChart, 1000, 800);

lineChart.getData().add(series);

stage.setScene(scene);

stage.show();

}

}

Methods.java

public class Methods {

//Метод хорд

public static String horda(double a, double b, double e) {

if(f(a) \* f(b) > 0) return "в данном интервале нет корней.";

double x\_prev, x\_curr = 0;

do {

x\_prev = x\_curr;

x\_curr = (a \* f(b) - b \* f(a)) / (f(b) - f(a));

if ((f(x\_curr) > 0) && (f(a) < 0)) {

b = x\_curr;

} else if ((f(x\_curr) > 0) && (f(b) < 0)) {

a = b;

b = x\_curr;

} else {

return "значения функции на концах отрезка имеют одинаковые знаки";

}

} while ((Math.abs(x\_curr - x\_prev) > e) && (f(x\_curr) > e));

return Double.toString(x\_curr);

}

// Метод секущих

public static String sekushie(double x\_prev, double x\_curr, double e) {

if(f(x\_prev) \* f(x\_curr) > 0) return "невозможно выбрать начальное приближение";

double x\_next = 0;

double tmp;

do {

tmp = x\_next;

x\_next = x\_curr - (x\_curr - x\_prev) \* f(x\_curr) / (f(x\_curr) - f(x\_prev));

x\_prev = x\_curr;

x\_curr = tmp;

} while ((Math.abs(x\_next - x\_curr) > e) && (f(x\_next) > e));

return Double.toString(x\_next);

}

// Метод простых итераций

public static String iteraz(double a, double b, double e) {

double alf[] = proizvod(a, b);

if(f(a) \* f(b) > 0) return "в данном интервале нет корней.";

double x\_curr = alf[1], x\_next ;

x\_next = x\_curr + alf[0] \* f(x\_curr);

while ((Math.abs(x\_next - x\_curr) > e)) {

x\_curr = x\_next;

x\_next = x\_curr + alf[0] \* f(x\_curr);

}

return Double.toString(x\_next);

}

protected static double f(double x) {

return Math.pow(x, 3) - 2.92 \* Math.pow(x, 2) + 1.435 \* x + 0.791;

}

private static double[] proizvod(double a, double b) {

double a1, a2, ans[] = new double[3];

a1 = 3 \* Math.pow(a, 2) - 2.92 \* 2 \* a + 1.435;

a2 = 3 \* Math.pow(b, 2) - 2.92 \* 2 \* b + 1.435;

ans[2] = 0;

if (a1 > a2) {

if(Math.abs(a1) < 1) ans[2] = 1;

ans[0] = -1/a1;

ans[1] = a;

}

else{

if(Math.abs(a2) < 1) ans[2] = 1;

ans[0] = -1/a2;

ans[1] = b;

}

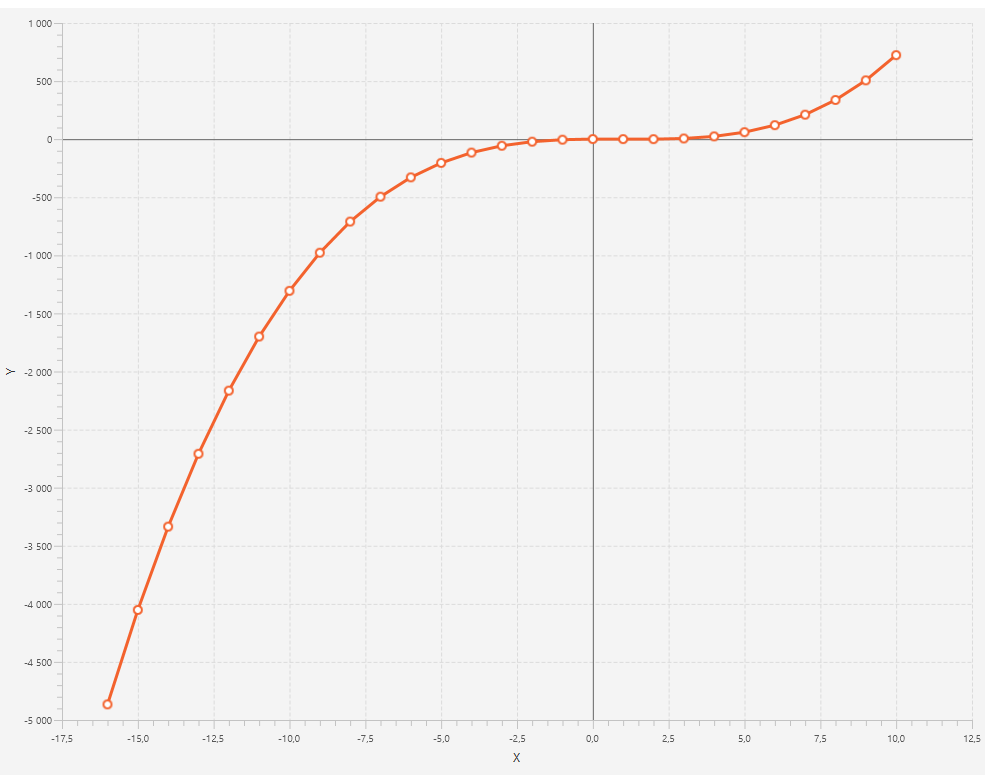
return ans;

}

}

**Примеры работы программы:**

Пример вывода графика:



|  |  |
| --- | --- |
| Входные данные из:  1 - файла  2 - консоли  2  Введите начало интервала, конец интервала и погрешность через пробел:  -100  100  0,01  Записать ответ в:  1 - файл  2 - консоль  2  Meтод Хорд: 2.919004401255333  Meтод Секущих: -0.5517786172232197  Meтод Простой итерации: -5.817424461674752 | Входные данные из:  1 - файла  2 - консоли  2  Введите начало интервала, конец интервала и погрешность через пробел:  -1  1  0,01  Записать ответ в:  1 - файл  2 - консоль  2  Meтод Хорд: -0.3157052789323239  Meтод Секущих: 1.6309278350515468  Meтод Простой итерации: -0.3324342071767712 |
| Входные данные из:  1 - файла  2 - консоли  1  Введите путь к файлу:  C:\All\1.txt  Записать ответ в:  1 - файл  2 - консоль  2  Meтод Хорд: 0.9999399459163655  Meтод Секущих: 1.6309278350515468  Meтод Простой итерации: -5.817424461674752 | Входные данные из:  1 - файла  2 - консоли  2  Введите начало интервала, конец интервала и погрешность через пробел:  1  10  0,01  Записать ответ в:  1 - файл  2 - консоль  2  Meтод Хорд: в данном интервале нет корней.  Meтод Секущих: невозможно выбрать начальное приближение  Meтод Простой итерации: в данном интервале нет корней. |

**Таблицы:**

Уточнение крайнего правого корня уравнения методом хорд с точностью ε=10-2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № | a | b | x | f(x) | f(a) | f(b) | |xi-xi+1| |
| 0 | 2 | 3 | 3 | -0.019 | -0.019 | 5.816 | - |
| 1 | 2 | 3 | 2.003 | -0.013 | -0.019 | 5.816 | 1.003 |
| 2 | 2.003 | 3 | 2.005 | -0.010 | -0.013 | 5.816 | 0.002 |

*Уточнение крайнего правого корня составляет 2.005*

Уточнение крайнего левого корня уравнения методом секущих с точностью ε=10-2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № | xi-1 | f(xi-1) | xi | f(xi) | xi+1 | f(xi+1) | |xi-xi+1| |
| 0 | -1 | -4.564 | 0 | 0.791 | -0.148 | 0.523 | 0.148 |
| 1 | 0 | 0.791 | -0.148 | 0.523 | -0.419 | -0.396 | 0.919 |
| 2 | -0.148 | 0.523 | -0.419 | -0.396 | -0.301 | 0.069 | 0.466 |
| 3 | -0.419 | -0.396 | -0.301 | 0.07 | -0.321 | -0.004 | 0.02 |
| 4 | -0.301 | 0.07 | -0.321 | -0.004 | -0.320 | 0.000024 | 0.001 |

*Уточнение крайнего левого корня составляет -0.320*

Уточнение центрального корня уравнения методом простой итерации с точностью ε=10-2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | xi | f(xi) | xi+1 | u(xi+1) | |xi-xi+1| |
| 0 | 1.3 | -0.081 | 1.225 | 1.388 | 0.075 |
| 1 | 1.225 | -0.002 | 1.230 | 1.227 | 0.005 |

*Уточнение крайнего левого корня составляет 1.230*

**Вывод:** в результате выполнения работы я познакомился с необычными(для меня) методами решения нелинейных уравнений. Разобравшись в большинстве из них, я успешно реализовал методы на машинном языке(Java). Основной сложностью стало определение сходимости метода в данном случае и верификация. Также вспомнил отрисовку графиков посредством JavaFX.