Университет ИТМО, факультет ПИиКТ

Лабораторная работа №3 по

Вычислительной математике

“Решение нелинейных уравнений”

Вариант 7

Выполнил: Клюев Андрей Викторович  
Группа: №3214

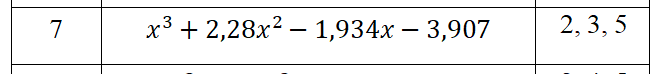
Преподаватель: Малышева Татьяна Алексеевна

Санкт-Петербург, 2020 год

Текст задания:

Цель работы: найти значение нелинейного уравнения с помощью трёх методов у выданного уравнения.

Описание программы:  
Пользователь выбирает уравнение и метод, с помощью которого он будет решать уравнение. После чего пользователю даётся выбор ввести данные для решения в терминал или из файла. После валидации поданных данных, пользователь выбирает, куда он хочет вывести решение: в файл или в терминал. Также, пользователю выводится отдельное окно, в котором расположен график выбранной функции. Затем происходит вывод решения.



В программе реализованы методы: хорд, Ньютона, а также метод простых итераций.

Решение:

Логическая часть программы:

private void chordMethod(double a, double b, Function<Double, Double> func, Function<Double, Double> dFunc, double E){

//Метод Хорд

// a, b - интервал, func - функция, E - необходимая погрешность

int count = 0;

double x, x0, tmpX = 12345;

if(func.apply(a)\*func.apply(b) >= 0){

System.out.println("Корней нет! Функция имеет разные знаки на концах интервала!");

}else {

System.out.println("Как вы хотите вывести решение?");

System.out.println("1. В файл");

System.out.println("2. В консоль");

boolean fileOut = (scn.nextInt() == 1);

if(fileOut){

System.out.println("Поместите файл в папку src и введите название файла:");

String path = "E://CompMath\_Lab3\_NonlinearEquations-master/src/" + scn.next();

try {

fos = new FileOutputStream(path);

filePrintStream = new PrintStream(fos);

}catch (IOException e){

e.printStackTrace();

}

}

if(fileOut){

filePrintStream.printf("%15s %15s %15s %15s %15s %15s %15s %15s %n",

"Шаг", "a", "b", "x", "func(a)", "func(b)", "func(x)", "|b - a|");

}else {

System.out.printf("%15s %15s %15s %15s %15s %15s %15s %15s %n",

"Шаг", "a", "b", "x", "func(a)", "func(b)", "func(x)", "|b - a|");

}

x = (a \* func.apply(b) - b \* func.apply(a)) / (func.apply(b) - func.apply(a));

do{

count++;

if(count != 1) tmpX = x;

x = (a \* func.apply(b) - b \* func.apply(a)) / (func.apply(b) - func.apply(a));

x0 = tmpX;

if(func.apply(a) \* func.apply(x) > 0){

a = x;

} else{

b =x;

}

if(fileOut){

filePrintStream.printf("%15d %15f %15f %15f %15f %15f %15f %15f %n",

count, a, b, x, func.apply(a), func.apply(b), func.apply(x), Math.abs(b - a));

}else {

System.out.printf("%15d %15f %15f %15f %15f %15f %15f %15f %n",

count, a, b, x, func.apply(a), func.apply(b), func.apply(x), Math.abs(b - a));

}

if (count > 350){

break;

}

}while (Math.abs(func.apply(x)) > E & Math.abs(x-x0) > E);

if(fileOut){

filePrintStream.close();

}

}

}

private void newtonMethod(double a, double b, Function<Double, Double> func, Function<Double, Double> dFunc, double E){

//Метод Ньютона

//a, b - начальные границы, func - функция, dFunc- производная функции, E - необходимая погрешность

int count = 0;

Function<Double, Double> ddfunc = x -> 6\*x + 4.56;

double x0, x, div;

if(func.apply(a)\*func.apply(b) >=0){

System.out.println("Корней нет! Знаки функции на краях промежутка совпадают!");

}else {

System.out.println("Как вы хотите вывести решение?");

System.out.println("1. В файл");

System.out.println("2. В консоль");

boolean fileOut = (scn.nextInt() == 1);

if (fileOut) {

System.out.println("Поместите файл в папку src и введите название файла:");

String path = "E:\\CompMath\_Lab3\_NonlinearEquations-master\\src\\" + scn.next();

try {

fos = new FileOutputStream(path);

filePrintStream = new PrintStream(fos);

} catch (IOException e) {

System.err.println("Ошибка чтения файла!");

}

}

if (fileOut) {

filePrintStream.printf("%15s %15s %15s %15s %15s %15s %n",

"Шаг", "Xn", "func(Xn)", "dfunc(Xn)", "Xn+1", "|Xn+1 - Xn|");

} else {

System.out.printf("%15s %15s %15s %15s %15s %15s %n",

"Шаг", "Xn", "func(Xn)", "dfunc(Xn)", "Xn+1", "|Xn+1 - Xn|");

}

//Если значения функции и второй производной функции имеют одинаковые знаки, то берём Xn за левую границу

if(func.apply(a) \* ddfunc.apply(a) > 0){

x = a;

}else {

x = b;

}

div = func.apply(x) / dFunc.apply(x);

do{

count++;

x0 = x;

x = x0 - div;

div = func.apply(x) / dFunc.apply(x);

if (fileOut) {

filePrintStream.printf("%15d %15f %15f %15f %15f %15f %n",

count, x0, func.apply(x), dFunc.apply(x), x, Math.abs(x - x0));

} else {

System.out.printf("%15d %15f %15f %15f %15f %15f %n",

count, x0, func.apply(x), dFunc.apply(x), x, Math.abs(x - x0));

}

if (count > 350){

break;

}

}while(Math.abs(x-x0) > E);

if (fileOut) {

filePrintStream.close();

}

}

}

private void simpleIterationsMethod(double a, double b, Function<Double, Double> func, Function<Double, Double> dFunc, double E){

//a, b- границы, func - функция, dFunc - произовдная функции, E - необходимая погрешность

// Function<Double, Double> dfi2 = x -> ((1.934-4.56\*x) \* Math.pow((1.934\*x - 2.28\*Math.pow(x, 2) + 3.907), (-2/3)))/3;

int count = 0;

double x0, x, lambda;

if(func.apply(a)\*func.apply(b) > 0){

System.out.println("Решений нет! На краях интервала функция имеет одинаковые знаки!");

}else {

System.out.println("Как вы хотите вывести решение?");

System.out.println("1. В файл");

System.out.println("2. В консоль");

boolean fileOut = (scn.nextInt() == 1);

if (fileOut) {

System.out.println("Поместите файл в папку src и введите название файла:");

String path = "E:\\CompMath\_Lab3\_NonlinearEquations-master\\src\\" + scn.next();

try {

fos = new FileOutputStream(path);

filePrintStream = new PrintStream(fos);

} catch (IOException e) {

System.err.println("Ошибка чтения файла!");

}

}

if (fileOut) {

filePrintStream.printf("%15s %15s %15s %15s %15s %n",

"Шаг", "Xn", "func(Xn)", "Xn+1", "|Xn+1 - Xn|");

} else {

System.out.printf("%15s %15s %15s %15s %15s %n",

"Шаг", "Xn", "func(Xn)", "Xn+1", "|Xn+1 - Xn|");

}

if (dFunc.apply(a) > dFunc.apply(b)) {

x = a;

} else {

x = b;

}

lambda = -1 / dFunc.apply(x);

double dfia = (3 \* Math.pow(a, 2) + 4.56 \* a - 1.934) \* lambda + 1;

double dfib = (3 \* Math.pow(b, 2) + 4.56 \* b - 1.934) \* lambda + 1;

//Проверка сходимости метода простой итерации

if (Math.abs(dfia) >= 1 & Math.abs(dfib) >= 1) {

System.err.println("Начальное приближение выбрано неверно, не выполняется достаточное условие сходимости метода.");

} else {

x = x + lambda \* func.apply(x);

do {

count++;

x0 = x;

x = x0 + lambda \* func.apply(x0);

if (fileOut) {

filePrintStream.printf("%15d %15f %15f %15f %15f %n",

count, x0, func.apply(x0), x, Math.abs(x - x0));

} else {

System.out.printf("%15d %15f %15f %15f %15f %n",

count, x0, func.apply(x0), x, Math.abs(x - x0));

}

if (count > 350) {

break;

}

} while (Math.abs(x - x0) > E | Math.abs(func.apply(x0)) > E);

if (fileOut) {

filePrintStream.close();

}

}

}

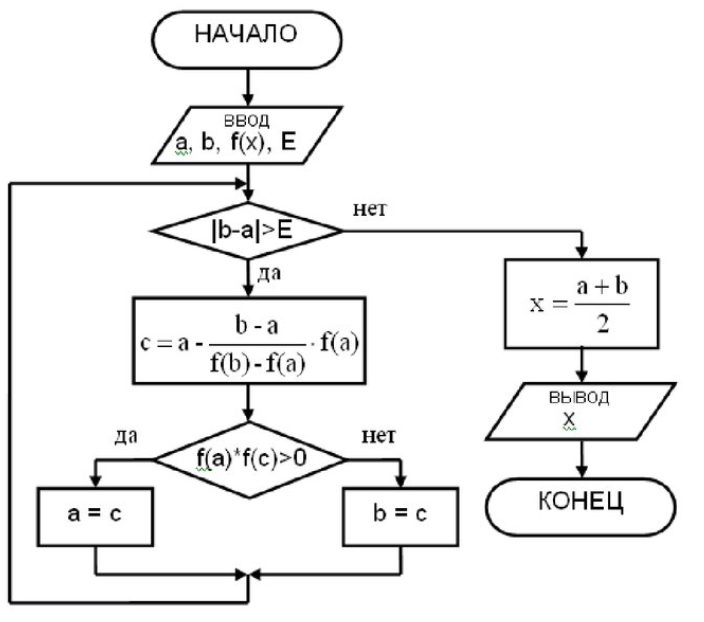
}

**ВЕСЬ КОД ПРОГРАММЫ:**

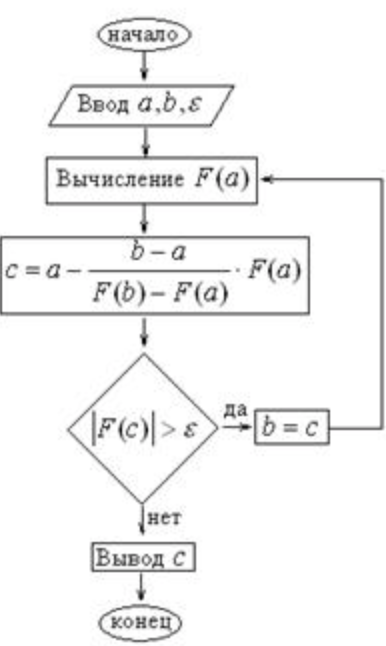
https://github.com/Amur27RUS/CompMath\_Lab3\_NonlinearEquations

Блок-схемы:

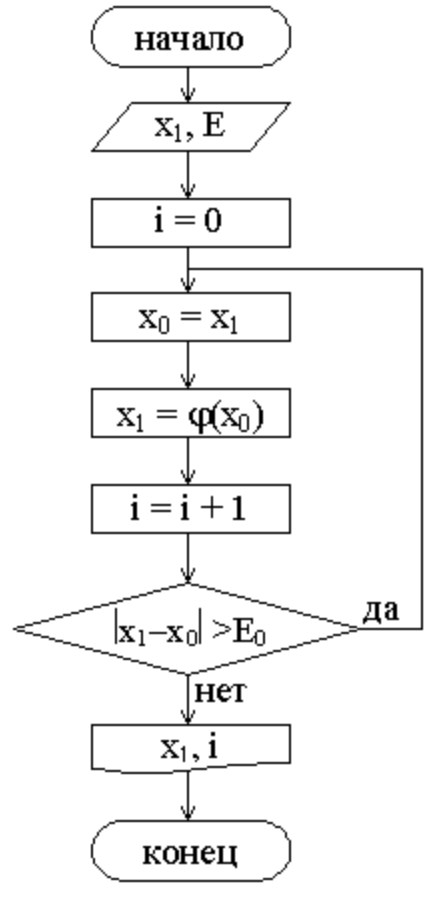
Метод хорд:



Метод Ньютона:



Метод простых итераций:



Заполненные таблицы:

Вводимые данные: a = -3, b = -2, E =0,01.

Получаемый результат:

Метод Хорд:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Шаг | a | b | x | func(a) | func(b) | func(x) | |b-a| |
| 1 | -3,000000 | -2,190787 | -2,190787 | -4,585000 | 0,758164 | 0,758164 | 0,809213 |
| 2 | -3,000000 | -2,305610 | -2,305610 | -4,585000 | 0,415912 | 0,415912 | 0,694390 |
| 3 | -3,000000 | -2,363360 | -2,363360 | -4,585000 | 0,198132 | 0,198132 | 0,636640 |
| 4 | -3,000000 | -2,389732 | -2,389732 | -4,585000 | 0,088083 | 0,088083 | 0,610268 |
| 5 | -3,000000 | -2,401235 | -2,401235 | -4,585000 | 0,037957 | 0,037957 | 0,598765 |
| 6 | -3,000000 | -2,406151 | -2,406151 | -4,585000 | 0,016137 | 0,016137 | 0,593849 |

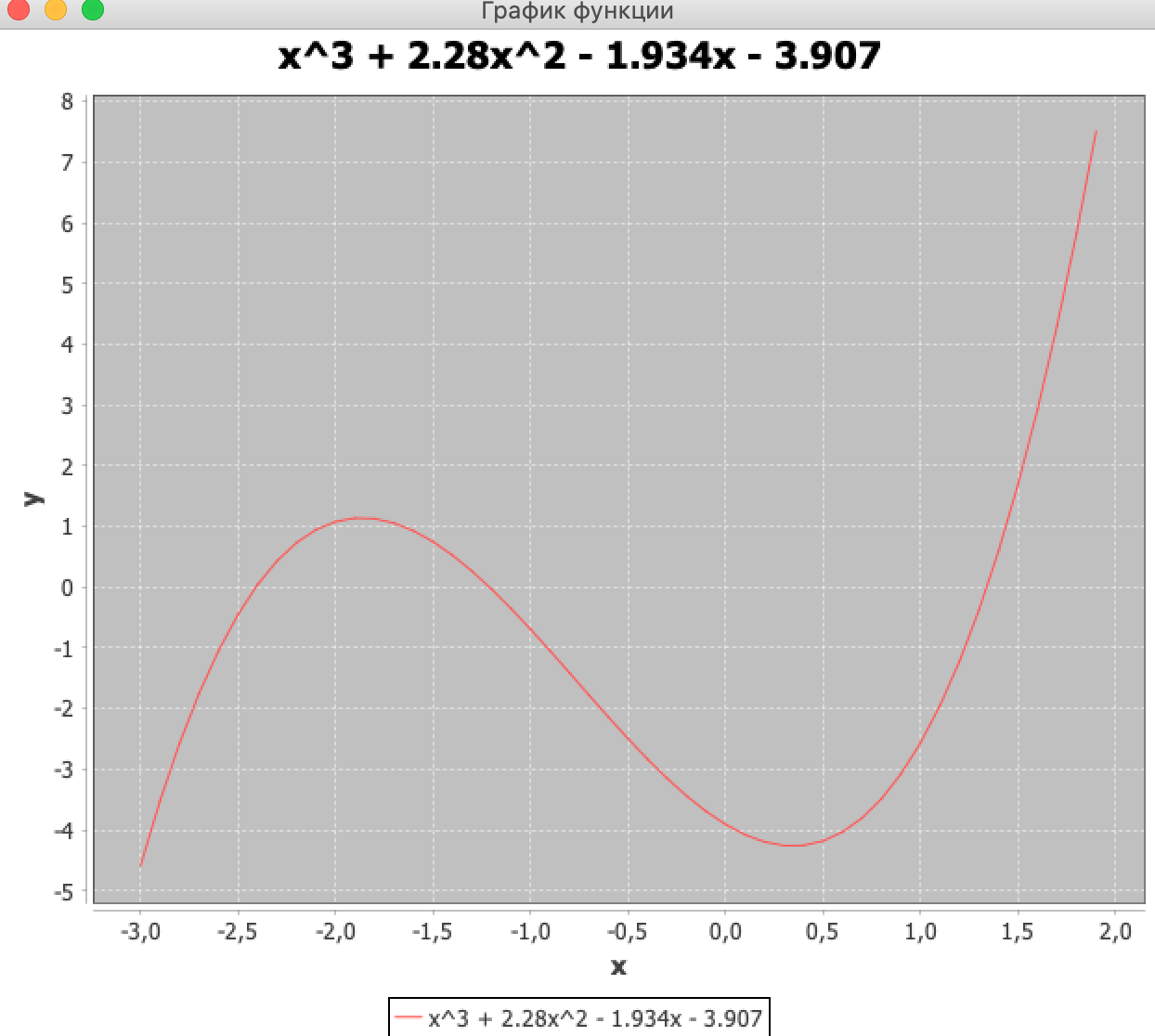
Метод Ньютона:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Шаг | Xn | func(Xn) | dfunc(Xn) | Xn+1 | |Xn+1 – Xn| |
| 1 | -3,000000 | -1,024398 | 6,460352 | -2,597312 | 0,402688 |
| 2 | -2,597312 | -0,134602 | 4,787760 | -2,438746 | 0,158567 |
| 3 | -2,438746 | -0,003958 | 4,506956 | -2,410632 | 0,028114 |
| 4 | -2,410632 | -0,000004 | 4,498260 | -2,409753 | 0,000878 |

Метод простых итераций:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Шаг | Xn | func(Xn) | Xn+1 | |Xn+1 – Xn| |
| 1 | -2,597312 | -1,024398 | -2,507343 | 0,089970 |
| 2 | -2,507343 | -0,487049 | -2,464566 | 0,042776 |
| 3 | -2,464566 | -0,261601 | -2,441591 | 0,022976 |
| 4 | -2,441591 | -0,148265 | -2,428569 | 0,013022 |
| 5 | -2,428569 | -0,086400 | -2,420981 | 0,007588 |
| 6 | -2,420981 | -0,051132 | -2,416490 | 0,004491 |
| 7 | -2,416490 | -0,030531 | -2,413809 | 0,002681 |
| 8 | -2,413809 | -0,018326 | -2,412199 | 0,001610 |
| 9 | -2,412199 | -0,011034 | -2,411230 | 0,000969 |
| 10 | -2,411230 | -0,006656 | -2,410645 | 0,000585 |

Пример выводимого графика:



Вывод:  
В ходе выполнения работы я разобрался с тем, как можно находить решения нелинейных уравнений, а также сделал их реализацию на языке программирования Java.