

Университет ИТМО
Факультет ФПИ и КТ

Лабораторная работа №2
“Численное решение нелинейных уравнений и систем”
По вычислительной математике
Вариант 8

Выполнил: Рогачев М. С.
Группа: Р32082
Преподаватель: Машина Е. А.

1) Задание

Задание:

1. Отделить корни заданного нелинейного уравнения графически (вид уравнения представлен в табл. 6)
2. Определить интервалы изоляции корней.
3. Уточнить корни нелинейного уравнения (см. табл. 6) с точностью $\varepsilon=10^{-2}$.
4. Используемые методы для уточнения каждого из 3-х корней многочлена представлены в таблице 7.
5. Вычисления оформить в виде таблиц (1-5), в зависимости от заданного метода. Для всех значений в таблице удерживать 3 знака после запятой.
 - 5.1 Для метода половинного деления заполнить таблицу 1.
 - 5.2 Для метода хорд заполнить таблицу 2.
 - 5.3 Для метода Ньютона заполнить таблицу 3.
 - 5.4 Для метода секущих заполнить таблицу 4.
 - 5.5 Для метода простой итерации заполнить таблицу 5.
6. Заполненные таблицы отобразить в отчете.

8	$3x^3 + 1,7x^2 - 15,42x + 6,89$		
8	5	2	3

2) Рабочие формулы

$$x_0 = \frac{a_0 + b_0}{2} \quad x_i = \frac{a_i + b_i}{2} \quad |f(x_n)| \leq \varepsilon$$

$$x_{i+1} = x_i - \frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})} f(x_i) \quad i = 1, 2, \dots$$

$$x_{i+1} = \varphi(x_i)$$

$$\begin{cases} F_1(x_1, x_2, \dots, x_n) = 0 \\ F_2(x_1, x_2, \dots, x_n) = 0 \\ \dots \dots \dots \dots \dots \dots \dots \\ F_n(x_1, x_2, \dots, x_n) = 0 \end{cases}$$

3) Вычислительная реализация

Функция для исследования: $3x^3 + 1,7x^2 - 15,42x + 6,89$

Вычислительная часть лабораторной работы №2

Функция: $3x^3 + 1,7x^2 - 15,42x + 6,89$

Крайний правый корень: метод хорд и итераций

n итерации	x_k	x_{k+1}	$f(x_{k+1})$	$ x_{k+1} - x_k $
1	2	1,7498	1,186	0,25
2	1,7498	1,7065	0,435	0,043
3	1,7065	1,69	0,175	0,01589
4	1,69	1,684	0,0734	0,00642

$x = 1,684$

Крайний левый корень: метод хорд

n шага	a	b	x	f(a)	f(b)	f(x)	a-b
1	-3	-2	-2,62	-12,55	20,53	4,98	1
2	-3	-2,62	-2,73	-12,55	5,008	0,687	0,38
3	-3	-2,73	-2,742	-12,55	0,617	0,077	0,22
4	-2,742	-2,73	-2,7444	0,105	0,617	-0,0008	0,012
5	-2,7444	-2,742	-2,74445	0,0023	0,105	-3,07	0,0024

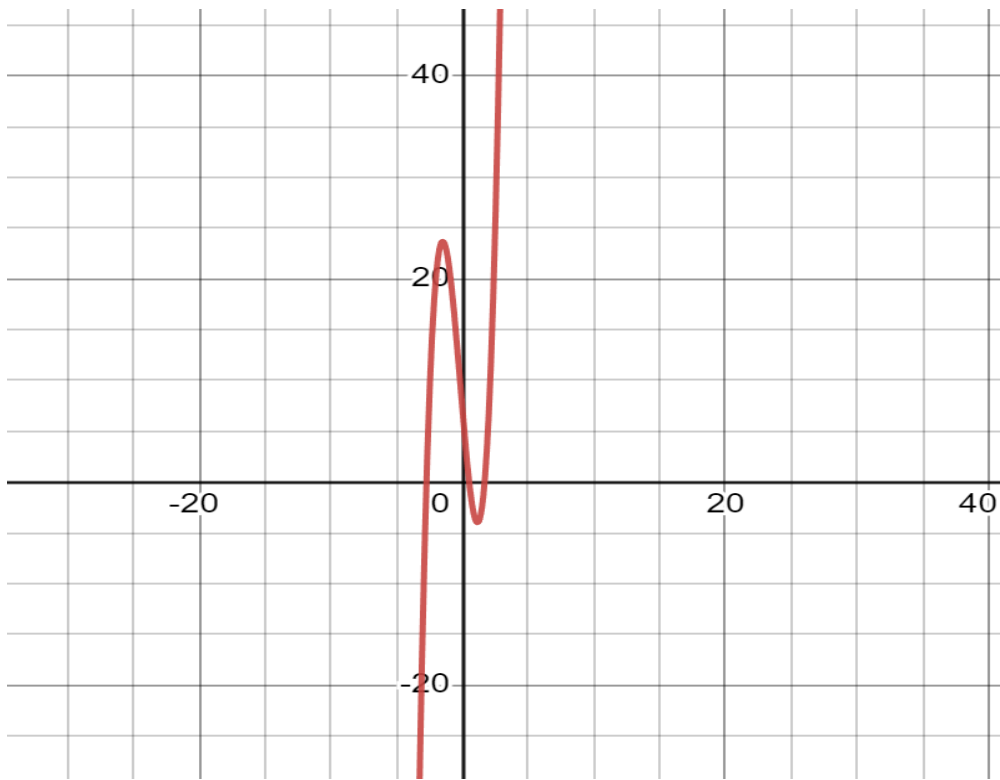
$x = -2,74445$

Центральный корень: метод Ньютона

n итерации	x_k	$f(x_k)$	$f'(x_k)$	x_{k+1}	$ x_{k+1} - x_k $
1	0	6,89	-15,42	0,447	0,447
2	0,447	0,607	-12,104	0,497	0,05
3	0,497	0,0148	-11,507	0,4983	0,00128

$x = 0,4983$

График функции:



4) Листинг программы

Метод половинного деления

```
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;

public class Method1 implements Method{
    ArrayList<Double> a = new ArrayList<Double>();
    ArrayList<Double> b = new ArrayList<Double>();
    ArrayList<Double> x = new ArrayList<Double>();
    ArrayList<Double> fa = new ArrayList<Double>();
    ArrayList<Double> fb = new ArrayList<Double>();
    ArrayList<Double> fx = new ArrayList<Double>();
    ArrayList<Double> ab = new ArrayList<Double>();

    public void solve(double sa, double sb, double eps, int n, int save)
    {
        int check = 200;

        a.add(sa);
        b.add(sb);
        ab.add(Math.abs(a.get(a.size() - 1) - b.get(b.size() - 1)));
        x.add((a.get(a.size() - 1) + b.get(b.size() - 1)) / 2);
        fa.add(Function.getFunction(a.get(a.size() - 1), n));
        fb.add(Function.getFunction(b.get(b.size() - 1), n));
        fx.add(Function.getFunction(x.get(x.size() - 1), n));

        while ((ab.get(ab.size() - 1) > eps || fx.get(fx.size() - 1) >
eps) && check-- > 0) {

            if (fa.get(fa.size() - 1) * fx.get(fx.size() - 1) > 0) {
                a.add(x.get(x.size() - 1));
                b.add(b.get(b.size() - 1));
            }
        }
    }
}
```

```

        } else {
            b.add(x.get(x.size() - 1));
            a.add(a.get(a.size() - 1));
        }
        ab.add(Math.abs(a.get(a.size() - 1) - b.get(b.size() - 1)));
        x.add((a.get(a.size() - 1) + b.get(b.size() - 1)) / 2);
        fa.add(Function.getFunction(a.get(a.size() - 1), n));
        fb.add(Function.getFunction(b.get(b.size() - 1), n));
        fx.add(Function.getFunction(x.get(x.size() - 1), n));
    }

    if (check <= 0) {
        System.out.println("Метод не сходится");
        return;
    }

    if (save == 1) {
        printConsole();
    } else {
        printFile(n);
    }
}

public void printConsole() {
    System.out.println("+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+");
    System.out.println("-----" +
        "-----+");
    System.out.printf("|%-15s|%-15s|%-15s|%-15s|%-15s|%-15s|%-15s|%-15s|%-15s|%-15s|%-15s|%-15s|%-15s|%-15s|%-15s|%-15s|%-15s|%-15s|%-15s|%-15s|\n",
        "№", "a", "b", "x", "F(a)", "F(b)", "F(x)", " |a - b|");
    System.out.println("+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+");
    System.out.println("-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+");
    for (int i = 0; i < a.size(); i++) {
        System.out.printf("|%-15d|%-15.5f|%-15.5f|%-15.5f|%-15.5f|%-15.5f|%-15.5f|%-15.5f|%-15.5f|%-15.5f|%-15.5f|%-15.5f|%-15.5f|%-15.5f|%-15.5f|%-15.5f|%-15.5f|%-15.5f|%-15.5f|%-15.5f|\n",
            i, a.get(i), b.get(i), x.get(i), fa.get(i), fb.get(i), fx.get(i), ab.get(i));
        System.out.println("|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+");
    }
    System.out.println("x = " + x.get(x.size() - 1) + " f(x) = " + fx.get(fx.size() - 1) + " n = " + a.size());
}

public void printFile(int n) {
    try {
        FileWriter fileWriter = new
FileWriter("src/main/resources/method1-" + n + ".txt", false);
        fileWriter.write("+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+");
    }
}

```

```

-----" +
                "-----
-----+\\n");
        fileWriter.write(String.format("|%-15s|%-15s|%-15s|%-15s|%-
15s|%-15s|%-15s|%-15s|\\n",
                "N", "a", "b", "x", "F(a)", "F(b)", "F(x)", "|a -
b|"));
        fileWriter.write("+-----+-----+-----
-----+" +
                "-----+-----+-----+-----
-----+-----+\\n");
        for (int i = 0; i < fa.size(); i++) {
            fileWriter.write(String.format("|%-15d|%-15.5f|%-
15.5f|%-15.5f|%-15.5f|%-15.5f|%-15.5f|\\n",
                i, a.get(i), b.get(i), x.get(i), fa.get(i),
fb.get(i), fx.get(i), ab.get(i)));
            fileWriter.write("|-----+-----+-----
-----+" +
                "-----+-----+-----
+-----+-----+-----|\\n");
        }
        fileWriter.flush();
        fileWriter.close();
    } catch (IOException e) {
        System.out.println("Не удалось сохранить");
    }
}
}

```

Метод секущих

```

import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;

public class SecantMethod{

    ArrayList<Double> a = new ArrayList<Double>();
    ArrayList<Double> b = new ArrayList<Double>();
    ArrayList<Double> fa = new ArrayList<Double>();
    ArrayList<Double> x = new ArrayList<Double>();
    ArrayList<Double> ab = new ArrayList<Double>();

    Scanner scanner = new Scanner(System.in);

    public void solve(double sa, double sb, double eps, int n, int
save) {
        System.out.println("f(a) = "+ Function.getFunction(sa, n));
        System.out.println("f` `(a) = "+
Function.getDerivativeFunction2(sa, n));
        System.out.println("f(b) = "+ Function.getFunction(sb, n));
        System.out.println("f` `(b) = "+
Function.getDerivativeFunction2(sb, n));
        if (Function.getFunction(sa, n) *
Function.getDerivativeFunction2(sa, n) > 0) {

```



```

        System.out.println("x0 = "+ sa);
        a.add(sa);
        System.out.println("Введите x1:");
        double x1 = scanner.nextDouble();
        b.add(x1);
    } else {
        System.out.println("x0 = "+ sb);
        a.add(sb);
        System.out.println("Введите x1:");
        double x1 = scanner.nextDouble();
        b.add(x1);
    }

    int check = 200;

    do {
        fa.add(b.get(b.size() - 1) - (b.get(b.size() - 1) -
a.get(a.size() - 1)) /
            (Function.getFunction(b.get(b.size() - 1), n) -
Function.getFunction(a.get(a.size() - 1), n)) *
            Function.getFunction(b.get(b.size() - 1), n));
        x.add(Function.getFunction(fa.get(fa.size() - 1), n));
        ab.add(Math.abs(fa.get(fa.size() - 1) - b.get(b.size() -
1)));
        a.add(b.get(b.size() - 1));
        b.add(fa.get(fa.size() - 1));
    } while (ab.get(ab.size() - 1) > eps && check-- > 0);

    fa.add(b.get(b.size() - 1) - (b.get(b.size() - 1) -
a.get(a.size() - 1)) /
        (Function.getFunction(b.get(b.size() - 1), n) -
Function.getFunction(a.get(a.size() - 1), n)) *
        Function.getFunction(b.get(b.size() - 1), n));
    x.add(Function.getFunction(fa.get(fa.size() - 1), n));
    ab.add(Math.abs(fa.get(fa.size() - 1) - b.get(b.size() - 1)));

    if (check <= 0) {
        System.out.println("Метод не сходится");
        return;
    }
    if (save == 1) {
        printConsole(n);
    } else {
        printFile(n);
    }
}

public void printConsole(int n) {
    System.out.println("+-----+
-----" +
        "-----+\\n");
    System.out.printf("|%-15s|%-15s|%-15s|%-15s|%-15s|%-15s|\\n",
        "№", " xk-1", " xk", " xk+1", " f(x", " |xk+1 - xk| ");
    System.out.println("+-----+-----+-----+-----+
-----");

```

```

-----+" +
                "-----+-----+-----+\n");

        for (int i = 0; i < a.size(); i++) {
            System.out.printf("|%-15d|%-15.5f|%-15.5f|%-15.5f|%-15.5f|%-15.5f|\n",
                                i, a.get(i), fa.get(i), x.get(i), b.get(i),
ab.get(i));
            System.out.println("+-----+-----+-----+-----+
-----+" +
                "-----+-----+-----+
+\n");
        }
        System.out.println("x = " + b.get(b.size() - 1) + " f(x) = " +
Function.getFunction(b.get(b.size() - 1), n) + " n = " + a.size());
    }

    public void printFile(int n) {
        try {
            FileWriter fileWriter = new
FileWriter("src/main/resources/method3-" + n + ".txt");
            fileWriter.write("+-----+-----+-----+
-----" +
                "-----+");
            fileWriter.write(String.format("|%-15s|%-15s|%-15s|%-15s|%-15s|%-15s|\n",
                                "№", " xk", " f(xk)", " f` (xk)", " xk+1", " |xk+1 -
xk| "));
            fileWriter.write("+-----+-----+-----+
-----+" +
                "-----+");
            for (int i = 0; i < fa.size(); i++) {
                fileWriter.write(String.format("|%-15d|%-15.5f|%-15.5f|%-15.5f|%-15.5f|\n",
                                i, a.get(i), fa.get(i), x.get(i), b.get(i),
ab.get(i)));
                fileWriter.write("+-----+-----+-----+
-----+" +
                    "-----+
+");
            }
            System.out.println("x = " + b.get(b.size() - 1) + " f(x) = "
+ Function.getFunction(b.get(b.size() - 1), n) + " n = " + a.size());
        } catch (IOException e) {
            System.out.println("Не удалось сохранить");
        }
    }
}

```

Метод простой итерации

```

import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;

public class SimpleIterationMethod implements Method{

```



```

ArrayList<Double> a = new ArrayList<Double>();
ArrayList<Double> b = new ArrayList<Double>();
ArrayList<Double> fb = new ArrayList<Double>();
ArrayList<Double> ab = new ArrayList<Double>();

public void solve(double sa, double sb, double eps, int n, int save)
{
    double z = Math.abs(Function.getDerivativeMaxValue(sa, sb, n));

    if (z > 1.0) {
        System.out.println("Достаточное условие сходимости не
выполнено q = " + z);
    } else {
        System.out.println("Достаточное условие сходимости выполнено
q = " + z);
    }

    int check = 200;

    if (Function.getFunction(sa, n) *
Function.getDerivativeFunction2(sa, n) > 0) {
        a.add(sa);
    } else {
        a.add(sb);
    }

    do {
        b.add(Function.getFiFunction(a.get(a.size() - 1), sa, sb,
n));

        fb.add(Function.getFunction(b.get(b.size() - 1), n));
        ab.add(Math.abs(a.get(a.size() - 1) - b.get(b.size() - 1)));
        a.add(b.get(b.size() - 1));
    } while ((ab.get(ab.size() - 1) > eps || fb.get(fb.size() - 1) >
eps) && check-- > 0);

    b.add(Function.getFiFunction(a.get(a.size() - 1), sa, sb, n));
    fb.add(Function.getFunction(b.get(b.size() - 1), n));
    ab.add(Math.abs(a.get(a.size() - 1) - b.get(b.size() - 1)));

    if (check <= 0) {
        System.out.println("Метод не сходится");
        return;
    }

    if (save == 1) {
        printConsole();
    } else {
        printFile(n);
    }
}

public void printConsole() {
    System.out.println("+-----+
-----" +
        "-----+");
    System.out.printf("|%-15s|%-15s|%-15s|%-15s|%-15s|\n",

```

```

        "№", " xk", " xk+1", " f(xk+1)", " |xk+1 - xk| ");
        System.out.println("+-----+-----+-----+-----+");
        +
            "-----+-----+");

        for (int i = 0; i < a.size(); i++) {
            System.out.printf("|%-15d|%-15.5f|%-15.5f|%-15.5f|%-15.5f|\n",
                i, a.get(i), b.get(i), fb.get(i), ab.get(i));
            System.out.println("+-----+-----+-----+-----+");
            +
                "-----+-----+");
        }

        System.out.println("x = " + b.get(b.size() - 1) + " f(x) = " +
fb.get(fb.size() - 1) + " n = " + a.size());
    }

    public void printFile(int n) {
        try {
            FileWriter fileWriter = new
FileWriter("src/main/resources/method5-" + n + ".txt");
            fileWriter.write("+-----+-----+-----+-----+");
            +
                "-----+-----+\n");
            fileWriter.write(String.format("|%-15s|%-15s|%-15s|%-15s|%-15s|\n",
                "№", " xk", " xk+1", " f(xk+1)", " |xk+1 - xk| "));
            fileWriter.write("+-----+-----+-----+-----+");
            +
                "-----+-----+-----+\n");
            for (int i = 0; i < a.size(); i++) {
                fileWriter.write(String.format("|%-15d|%-15.5f|%-15.5f|%-15.5f|%-15.5f|\n",
                    i, a.get(i), b.get(i), fb.get(i), ab.get(i)));
                fileWriter.write("+-----+-----+-----+-----+");
                +
                    "-----+-----+-----+\n");
            }
        } catch (IOException e) {
            System.out.println("Не удалось сохранить");
        }
    }
}

```

Метод Ньютона для системы нелинейных уравнений

```
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;

public class NewtonMethod{

    ArrayList<Double> a = new ArrayList<Double>();
    ArrayList<Double> b = new ArrayList<Double>();
```

```

ArrayList<Double> da = new ArrayList<Double>();
ArrayList<Double> db = new ArrayList<Double>();
ArrayList<Double> fa = new ArrayList<Double>();
ArrayList<Double> fb = new ArrayList<Double>();

public void solve(double sa, double sb, double eps, int n, int save)
{
    int check = 200;
    Pair<Double, Double> x;

    a.add(sa);
    b.add(sb);

    if (n == 1) {
        do {
            x = solve(2 * a.get(a.size() - 1), 2 * b.get(b.size() -
1),
                    4 - Math.pow(a.get(a.size() - 1), 2) -
Math.pow(b.get(b.size() - 1), 2),
                    -6 * a.get(a.size() - 1), 1,
                    3 * Math.pow(a.get(a.size() - 1), 2) -
b.get(b.size() - 1));
            db.add(x.value());
            da.add(x.key());
            a.add(a.get(a.size() - 1) + da.get(da.size() - 1));
            b.add(b.get(b.size() - 1) + db.get(db.size() - 1));
            fa.add(Math.pow(a.get(a.size() - 1), 2) +
Math.pow(b.get(b.size() - 1), 2) - 4);
            fb.add(3 * Math.pow(a.get(a.size() - 1), 2) -
b.get(b.size() - 1));
        } while (Math.max(Math.abs(da.get(da.size() - 1)),
Math.abs(db.get(db.size() - 1))) > eps && check-- > 0);
    } else {
        do {
            x = solve(2 * a.get(a.size() - 1), 2 * b.get(b.size() -
1),
                    1 - Math.pow(a.get(a.size() - 1), 2) -
Math.pow(b.get(b.size() - 1), 2),
                    -1, 1,
                    a.get(a.size() - 1) - b.get(b.size() - 1));
            da.add(x.key());
            db.add(x.value());
            a.add(a.get(a.size() - 1) + da.get(da.size() - 1));
            b.add(b.get(b.size() - 1) + db.get(db.size() - 1));
            fa.add(Math.pow(a.get(a.size() - 1), 2) +
Math.pow(b.get(b.size() - 1), 2) - 1);
            fb.add(a.get(a.size() - 1) - b.get(b.size() - 1));
        } while (Math.max(Math.abs(da.get(da.size() - 1)),
Math.abs(db.get(db.size() - 1))) > eps && check-- > 0);
    }

    if (check <= 0) {
        System.out.println("Алгоритм не сходится");
        return;
    }
}

```

```

        if (save == 1) {
            printConsole();
        } else {
            printFile(n);
        }
    }

    public void printConsole() {
        System.out.println("+-----+-----+-----+-----+-----+-----+");
        System.out.printf("%-15s|%-15s|%-15s|%-15s|%-15s|%-15s|%-15s|\n",
            "N", "x", "y", "dx", "dy", "f(x+dx,y+dy)",
            "g(x+dx,y+dy)");
        System.out.println("+-----+-----+-----+-----+-----+-----+");
        for (int i = 0; i < fa.size(); i++) {
            System.out.printf("%-15d|%-15.5f|%-15.5f|%-15.5f|%-15.5f|%-15.5f|%-15.5f|\n",
                i, a.get(i), b.get(i), da.get(i), db.get(i),
                fa.get(i), fb.get(i));
            System.out.println("|-----+-----+-----+-----+-----+");
        }
        System.out.println("x = " + a.get(a.size() - 1) + " y = " +
            b.get(b.size() - 1) +
            "\nf(x) = " + fa.get(fa.size() - 1) + " g(x) = " +
            fb.get(fb.size() - 1) + "\nn = " + a.size());
    }

    public void printFile(int n) {
        try {
            FileWriter fileWriter = new
            FileWriter("src/main/resources/method6-" + n + ".txt");
            fileWriter.write("+-----+-----+-----+-----+-----+-----+");
            fileWriter.write(String.format("%-15s|%-15s|%-15s|%-15s|%-15s|%-15s|%-15s|\n",
                "N", "x", "y", "dx", "dy", "f(x+dx,y+dy)",
                "g(x+dx,y+dy)"));
            fileWriter.write("+-----+-----+-----+-----+-----+-----+");
            for (int i = 0; i < fa.size(); i++) {
                fileWriter.write(String.format("%-15d|%-15.5f|%-15.5f|%-15.5f|%-15.5f|%-15.5f|%-15.5f|\n",
                    i, a.get(i), b.get(i), da.get(i), db.get(i),
                    fa.get(i), fb.get(i)));
            }
        } catch (IOException e) {}
    }
}

```

```

        i, a.get(i), b.get(i), da.get(i), db.get(i),
fa.get(i), fb.get(i)));
        fileWriter.write("|-----+-----+-----
-----+" +
        "-----+-----+-----
+-----|\n");
    }
} catch (IOException e) {
    System.out.println("Не удалось сохранить");
}
}

public Pair<Double, Double> solve(double x1, double y1, double a1,
double x2, double y2, double a2) {
    double d = x1 * y2 - x2 * y1;

    return new Pair<>((a1 * y2 - a2 * y1) / d, (x1 * a2 - x2 * a1) /
d);
}
}

```

5) Примеры работы программы

```

Введите:
0 - для выхода
1 - для ввода данных из консоли
2 - для ввода данных из файла
1
Введите номер метода:
1 - Метод половинного деления
4 - Метод секущих
5 - Метод простой итерации
6 - Метод Ньютона для решение систем нелинейных уравнений
1
Введите номер функции:
1 - x^3 + 2.28 * x^2 - 1.934 * x - 3.907
2 - x^3 + 3 * x^2 + 3 * x + 1
3 - cos(x)
4 - x^3 - x + 4
1
Введите a:
-3
Введите b:
-2
Введите погрешность [0,0001; 1]:
0,01
0 - сохранить в файл
1 - вывести ответ в консоль

```

```

0 - сохранить в файл
1 - вывести ответ в консоль

```

№	a	b	x	F(a)	F(b)	F(x)	a - b
0	-3,00000	-2,00000	-2,50000	-4,58500	1,08100	-0,44700	1,00000
1	-2,50000	-2,00000	-2,25000	-0,44700	1,08100	0,59637	0,50000
2	-2,50000	-2,25000	-2,37500	-0,44700	0,59637	0,15039	0,25000
3	-2,50000	-2,37500	-2,43750	-0,44700	0,15039	-0,12865	0,12500
4	-2,43750	-2,37500	-2,40625	-0,12865	0,15039	0,01570	0,06250
5	-2,43750	-2,40625	-2,42188	-0,12865	0,01570	-0,05526	0,03125
6	-2,42188	-2,40625	-2,41406	-0,05526	0,01570	-0,01948	0,01563
7	-2,41406	-2,40625	-2,41016	-0,01948	0,01570	-0,00182	0,00781

x = -2.41015625 f(x) = -0.0018163552284251594 n = 8

6) Вывод

В ходе работы я познакомился с методами численных решений систем нелинейных уравнений, а также написал программную часть для решения нелинейных уравнений с заданной точностью.