Университет ИТМО
Факультет программной инженерии и компьютерной техники

Вычислительная математика

Лабораторная работа №3
Вариант – вг2

Выполнила: Екатерина Машина

Группа Р3210

Преподаватель: Ольга Вячеславовна Перл

Санкт-Петербург
2020 г.

Цель работы

Реализовать метод касательных (метод Ньютона) и метод простой итераций для решения нелинейных уравнений и реализовать решение систем линейных уравнений методом простой итерации.

Описание использованного метода

Метод касательных:

Суть метода заключается в том, что функция $y = f(x)$ на отрезке $[a, b]$ заменяется касательной, а в качестве приближенного значения корня $x^* = x_n$ принимается точка пересечения касательной с осью абсцисс (как показано на Рисунок *1*).

$$x_1 \; = \; x_0 \; - \; h_0$$

$$h_0 = \frac{f(x_0)}{\tan\alpha} = \frac{f(x_0)}{f'(x_0)x_1} = \; x_0 - \frac{f(x_0)}{f'(x_0)}$$

Рабочая формула метода:

$$x_i \; = \; x_i - 1 \; - \frac{f(\; x_i - 1)}{f'(x_i - 1)}$$

Критерий окончания итерационного процесса: $|x_n - x_{n-1}\,| \le \; \varepsilon$ или$\left|\frac{f(x_n)}{f'(x_n)}\right| \le \; \varepsilon$ или $|f(x_n)| \; \le \; \varepsilon$

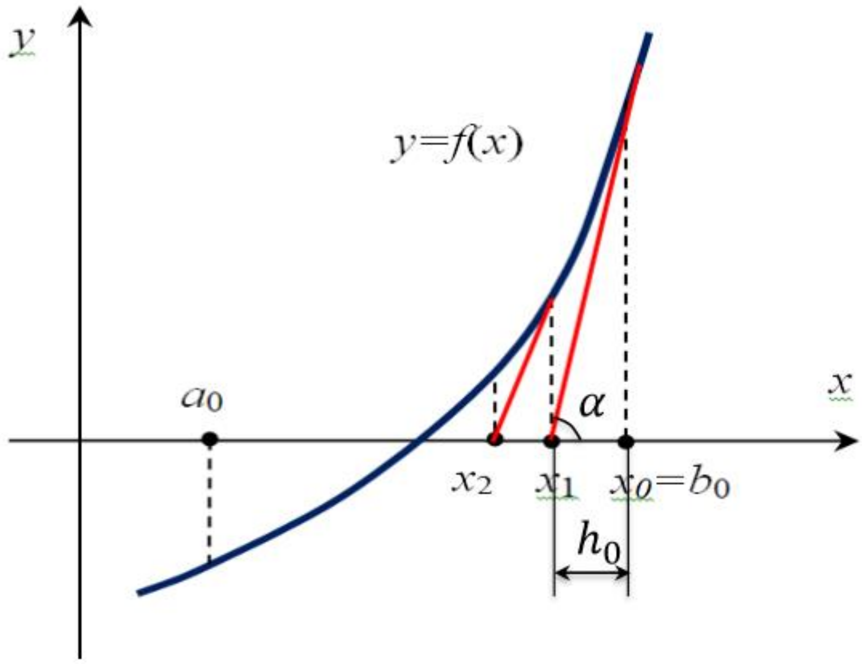


Рисунок 1

Метод имеет следующее достаточное условие сходимости:

Метод Ньютона применяется в том случае, если выполняются условия:

- функция $y = \; f(x)$ определена и непрерывна на отрезке $[a; b]$;
- $f(a)\cdot f(b) \; < \; 0$ (на концах отрезка $[a; b]$функция имеет разные знаки);
- производные $f'(x)$ и $f''(x)$ сохраняют знак на отрезке $[a; b]$;
- производная $f'(x) \neq 0$

Метод простой итерации:

Суть метода заключается в том, что уравнение $f(x) = 0$ с помощью некоторых преобразований необходимо переписать в виде $x = \varphi(x)$ (как показано на *Рисунок 2*).

Уравнение $f(x) = 0$ эквивалентно уравнению $x = x + \lambda(x)f(x)$ для любой функции $\lambda(x) \neq 0$. Возьмем $\varphi(x) = x - \lambda(x)f(x)$ и выберем функцию (или переменную) $\lambda(x) \neq 0$ так, чтобы функция $\varphi(x)$ удовлетворяла необходимым условиям.

Для нахождения корня уравнения $x = \varphi(x)$ выберем некоторое начальное значение x_0, которое должно находиться как можно ближе к корню уравнения. Дальше с помощью итерационной формулы $x_n + 1 = \varphi(x_n)$ будем находить каждое следующее приближение корня уравнения.

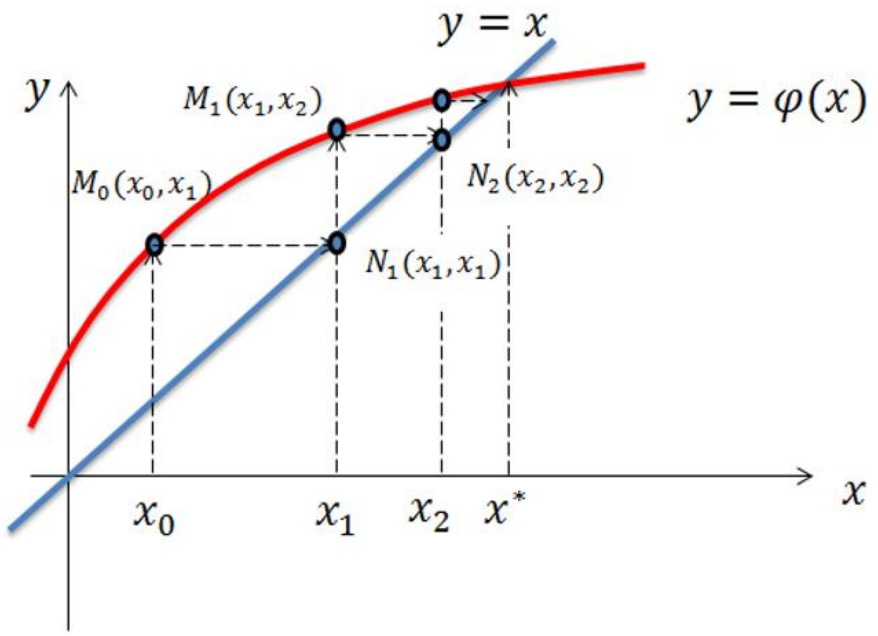


Рисунок 2

Рабочая формула метода:

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

Условия сходимости метода простой итерации определяются теоремой:

Если в некоторой σ - окрестности корня x* уравнения $f(x) = 0$ функция $x = \varphi(x)$ дифференцируема и удовлетворяет неравенству $|\varphi'(x)| < q$, где $0 \le q < 1$ постоянная, то независимо от выбора начального приближения x_0 из указанной σ окрестности итерационная последовательность x_n не выходит из этой окрестности, метод сходится со скоростью геометрической прогрессии.

Достаточное условие сходимости метода:

$|\varphi'(x)| \le q < 1$, где q – некоторая константа

Критерий окончания итерационного процесса:

$|x_n - x_{n-1}| \le \varepsilon$

Метод простой итерации для решения систем нелинейных уравнений:

Суть метода заключается в том, чтобы привести первоначальную систему уравнений к следующему виду:

$$\begin{cases} x_1 = \varphi_1(x_1, \dots, x_n), \\ x_2 = \varphi_1(x_2, \dots, x_n), \\ \quad \vdots \\ x_n = \varphi_n(x_1, \dots, x_n), \end{cases}$$

Для этого нам необходимо задать начальное приближение $x^{(0)} = (x_{10},\ x_{20}, \dots, x_{n0})^T$ и малое положительное число ε (точность).

Затем вычислить $x^{(k+1)}$ по формуле $x^{(k+1)} = \ \varphi_1\left(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}\right)$, так продолжать увеличивая k на единицу пока не будет достигнут критерий окончания итерационного процесса.

Критерий завершения итерационного процесса:

$\left|x_i^{(k+1)} - x_i^{(k)}\right| \le \varepsilon$, значит процесс завершен, $x^* = \mathrm{x}^{(k+1)}$.

Выводы

В результате выполнения я изучила 5 методов решения нелинейных уравнений и пришла к следующим выводам относительно их преимуществ и недостатков:

1. Метод касательных:

Достоинства: Метод обладает квадратичной сходимостью.

Недостатки: Необходимость вычисления производной на каждой итерации.

2. Метод простой итерации:

Достоинства: Простота реализации

Недостатки: Сходимость метода в малой окрестности корня и вытекающая отсюда необходимость выбора начального приближения к корню из этой малой окрестности. В противном случае итерационный процесс расходится или сходится к другому корню этого уравнения. Также при $|\varphi'(x)| \approx 1$, то сходимость может быть очень медленной.

3. Метод секущих:

Достоинства: Меньший объем вычислений по сравнению с методом Ньютона, т.к. не требуется вычислять производную.

Недостатки: Порядок сходимости метода секущих ниже, чем у метода касательных и равен золотому сечению ≈1,618 (сверхлинейная).

4. Метод половинного деления:

Достоинства: Обладает абсолютной сходимостью (близость получаемого численного решения задачи к истинному решению.) Устойчив к ошибкам округления.

Недостатки: если интервал содержит несколько корней, то неизвестно к какому относится вычислительный процесс. Медленный метод: имеет линейную сходимость.

Имеет смысл применять в случаях когда требуется высокая надежность счета, а скорость несущественна.

5. Метод хорд:

Достоинства: Простота реализации

Недостатки: Скорость сходимости – линейная. Порядок сходимости метода хорд выше, чем у метода половинного деления.

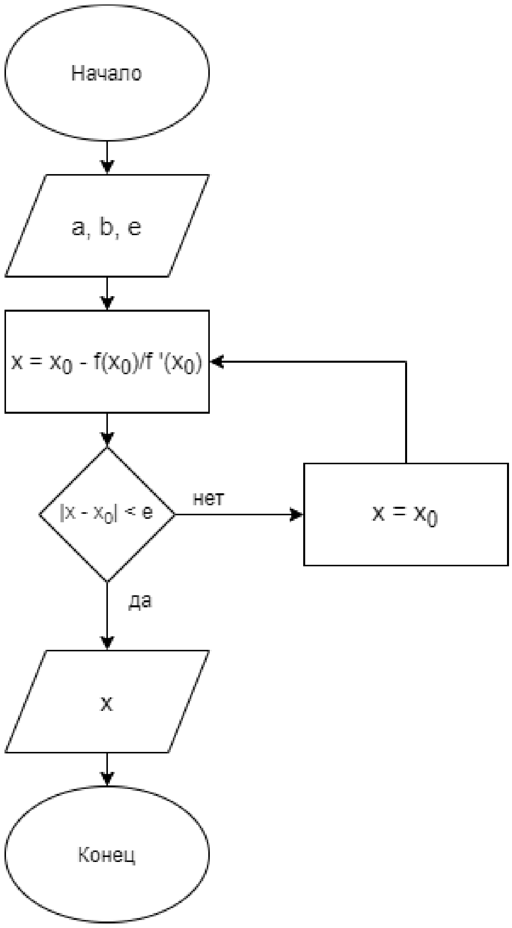
Что касается Методов решения систем нелинейных уравнений, то мною были изучены два метода решения и сделаны следующие выводы:

Метод Ньютона для решения СНАУ представляет собой обобщение метода Ньютона для решения НУ его сутью является попытка свести решение системы нелинейных уравнений к решению системы линейных уравнений. Основная сложность метода Ньютона заключается в обращении матрицы Якоби. Вводя обозначение $\Delta x^{(k)} = x^{(k+1)} - x^{(k)}\Delta x^{(k)} = x^{(k+1)} - x^{(k)}$ получаем СЛАУ для вычисления $\Delta x^{(k)}$. Решение этого СЛАУ создает основную вычислительную нагрузку алгоритма.

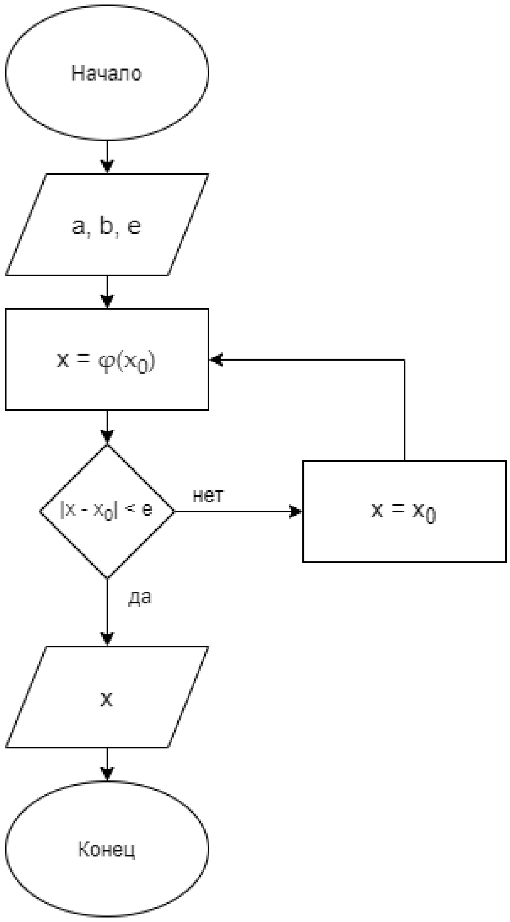
Метод простой итерации же в свою очередь позволяет грубо говоря подобрать вектор решений системы уравнений путем выражения значения одной неизвестной через все остальные и постепенной подстановки значений неизвестных, вычисляемых на каждом шаге.

Блок-схема

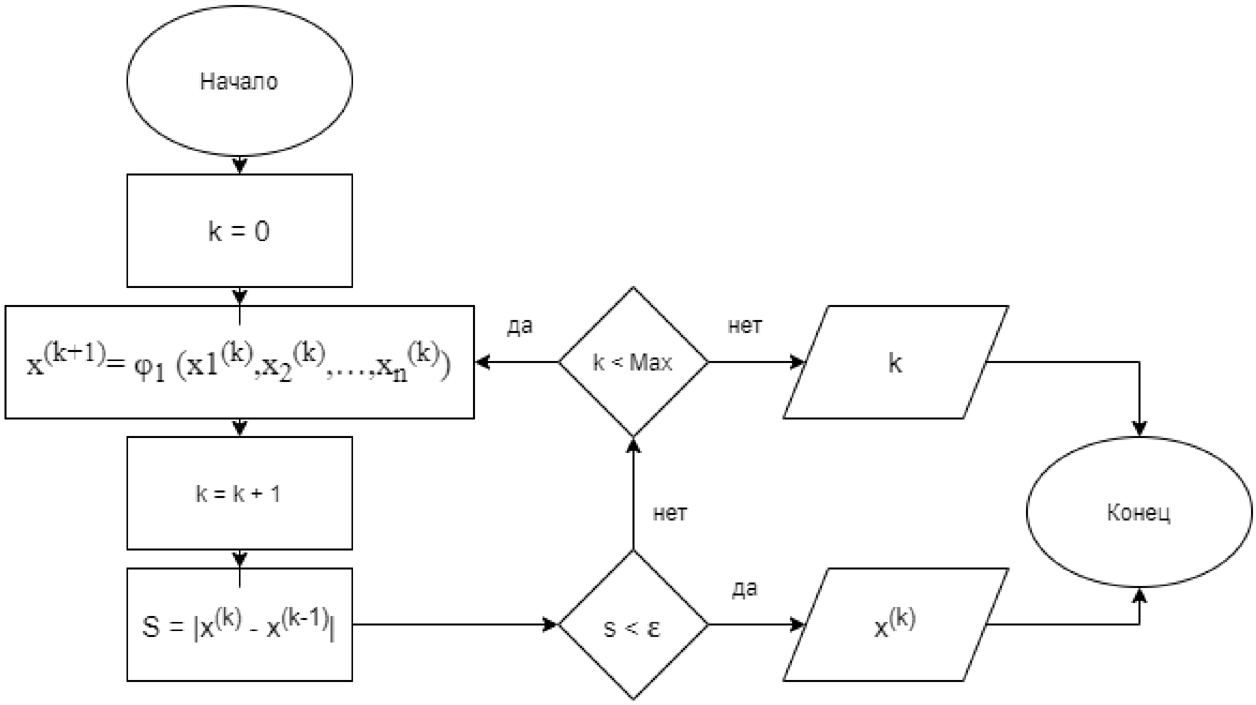
Метод касательных



Метод простой итерации (НУ)



Метод простой итерации (СНАУ)



Листинг численного метода

Метод касательных

```
public static double countXByNewton() {
    int n = 0;
    x = countStartX(MIN_RANGE, MAX_RANGE, x);
    double counter = 0;
    counter = Math.abs(df(x));
    while (counter > EPS) {
        x = x - (f(x) / df(x));
        n ++;
        counter--;
    }
    return x;
}
```

```
public static double countStartX(double minRange, double maxRange, double x) {
    if (f(minRange)*df(maxRange) < 0) {
        return minRange;
    }
    else
    {
        return maxRange;
    }
}
```

Метод простой итерации (НУ)

```
public static double countXByIterations() {
    x = countStartX(MIN_RANGE, MAX_RANGE, x);
    lambda = getLambda(x);
    double x0;
    double fx;
    int count = 0;
    do {
        x0 = x;
        x = x - lambda * (f(x));
        fx = f(x0);
        count++;
    } while (Math.abs(x - x0) >= EPS || count <= MAX_NUMBER_OF_ITERATIONS);
    return x0;
}
```

```
public static double countStartX(double minRange, double maxRange, double x) {
        double a = f(minRange);
        double b = f(maxRange);
        double c = df(x);
        if (a >= b && a >= c) return minRange;
        if (b >= a && b >= c) return maxRange;
        return x;
    }
```

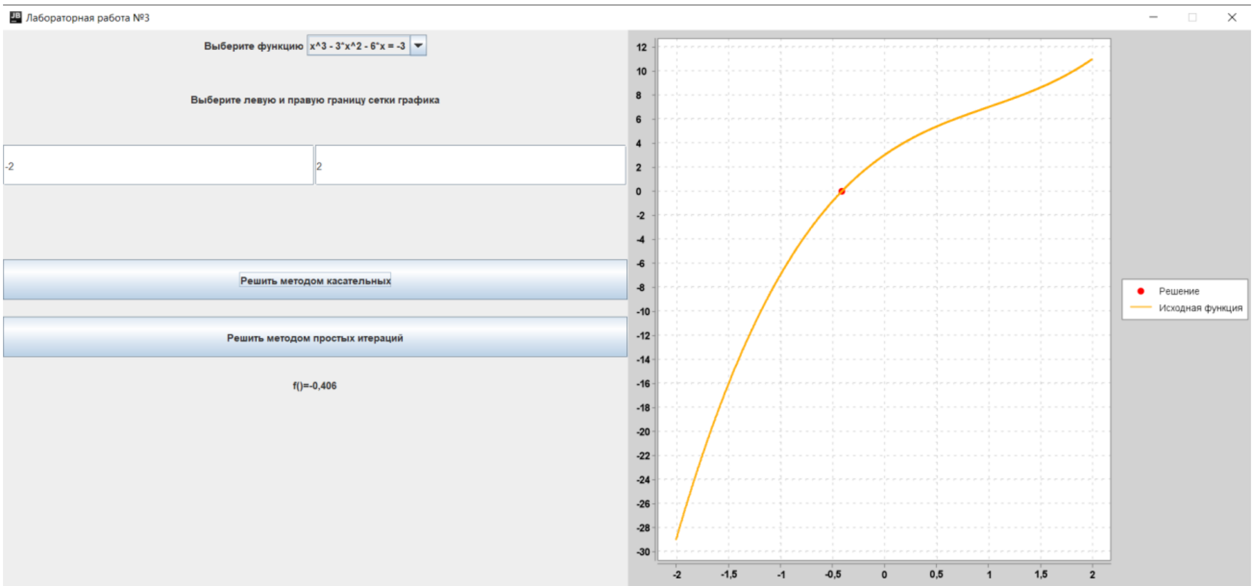
```
public static double getLambda(double x) {
    return 1.0 / df(x);
}
```

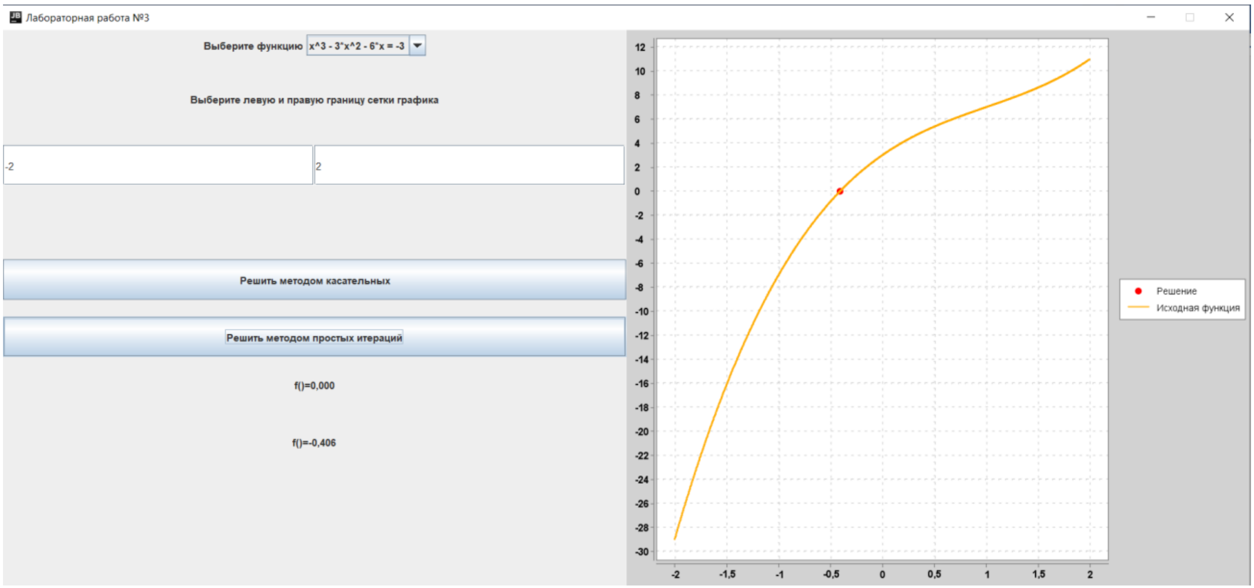
Метод простой итерации (СНАУ)

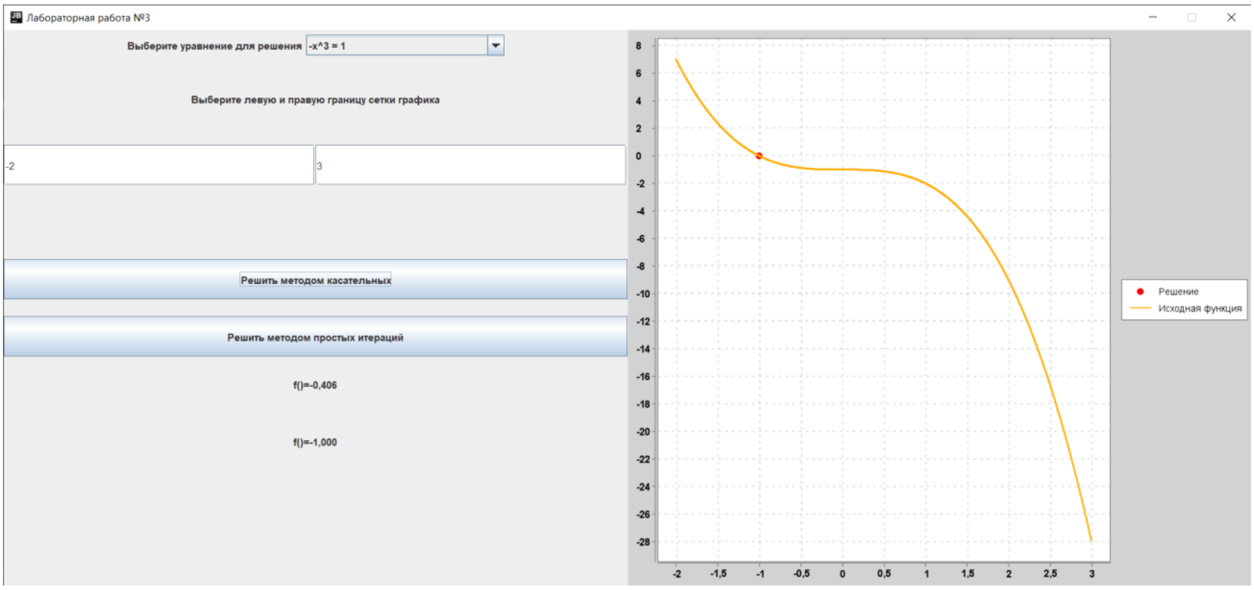
```java
public static double countX(Function f1, Function f2) {
        double d1, d2, x, y;
        int counter;
        do
        {
            x = f1.signifyX(y0);
            y = f2.getValue(x0);
            d1 = f1.equateToZero(x, y);
            d2 = f2.equateToZero(x, y);
            x0 = x;
            y0 = y;
            counter ++;
        } while (Math.abs(d1) > EPS && Math.abs(d2)> EPS || counter <=
MAX_NUMBER_OF_ITERATIONS);
        x0 = 0;
        y0 = 0;
        return x;
    }
```

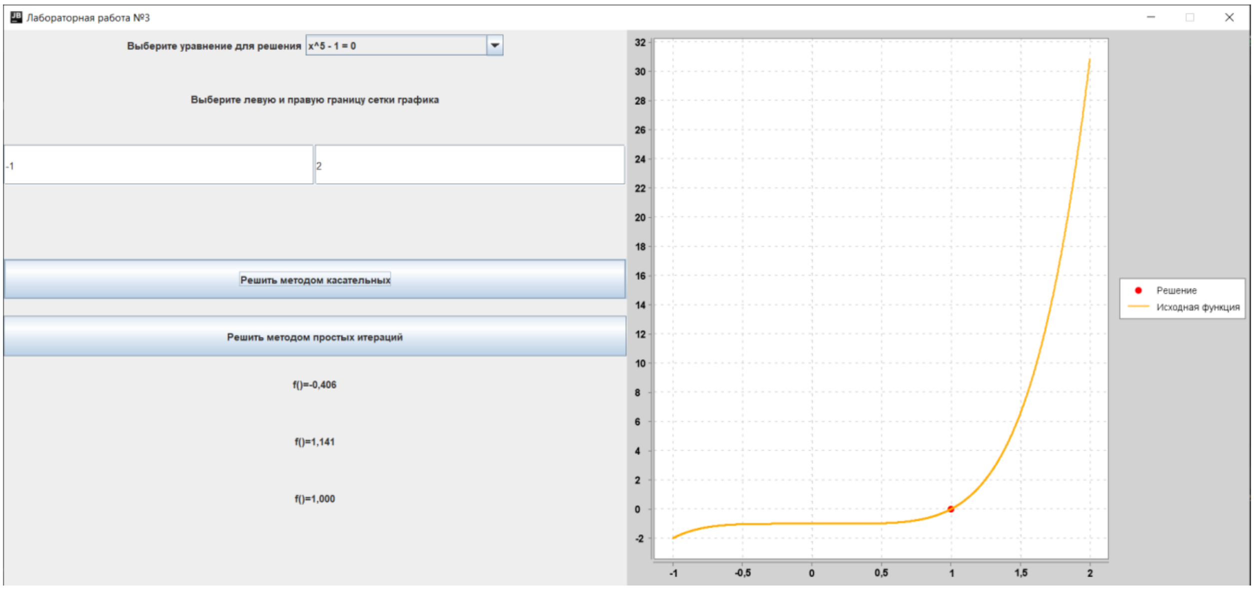
Примеры

Решение нелинейных уравнений:

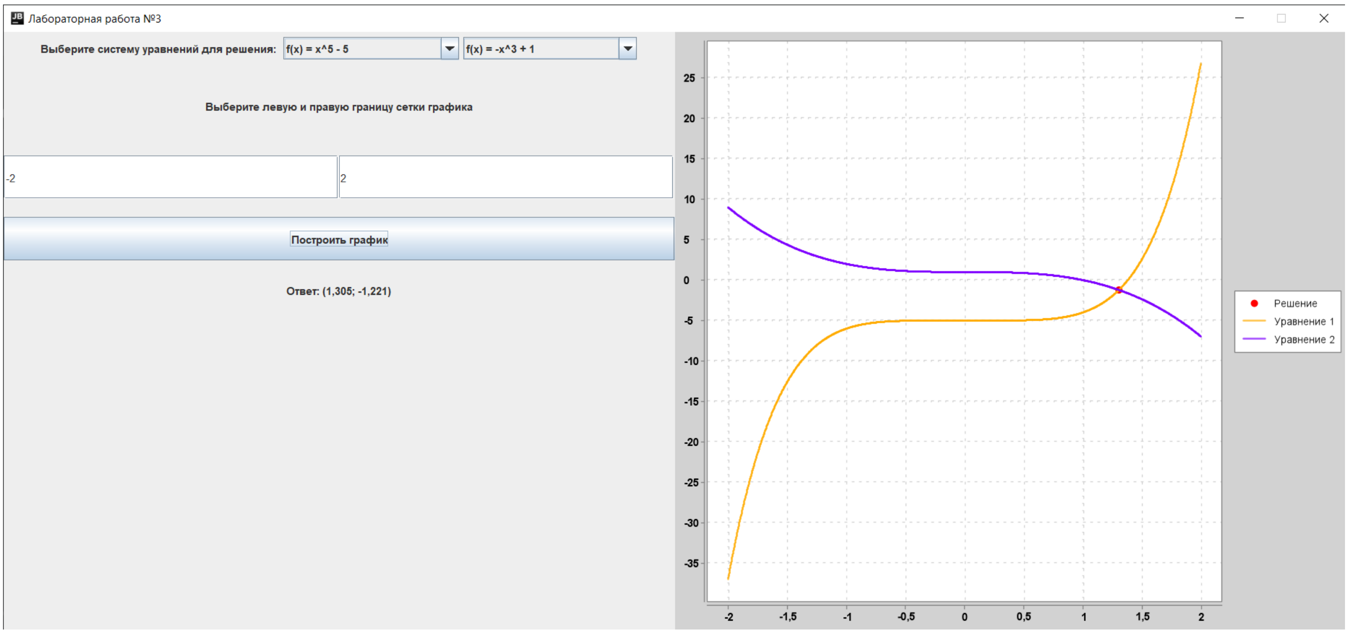


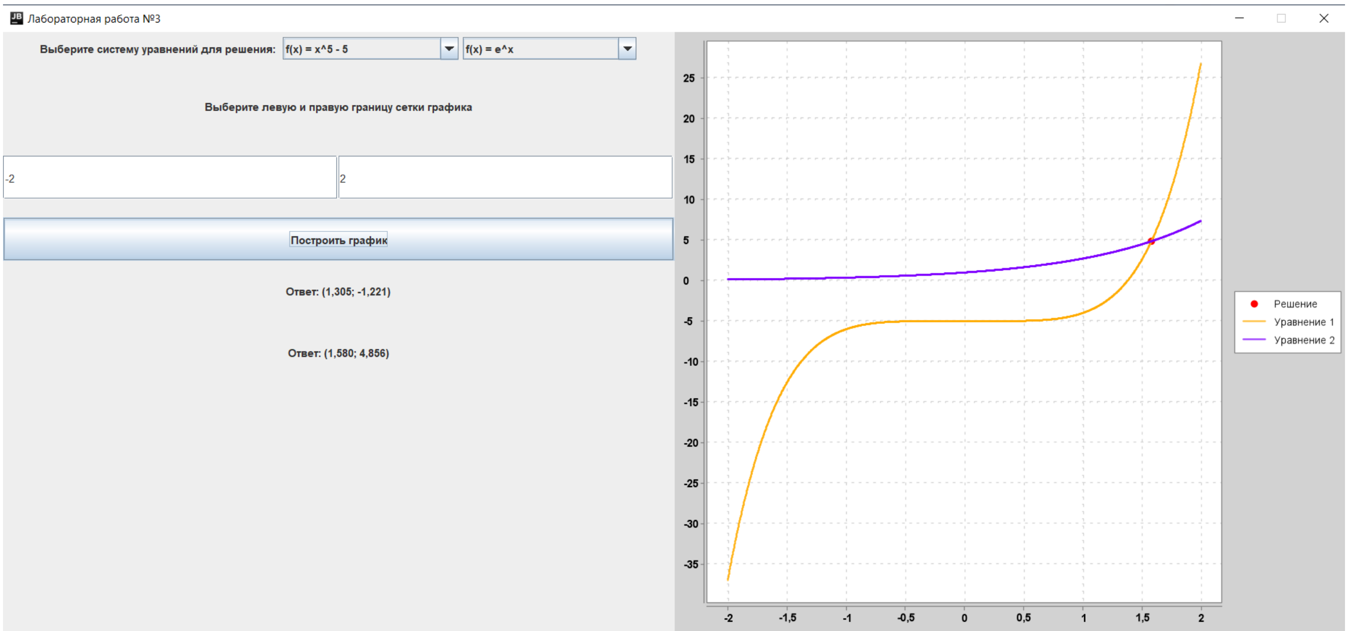


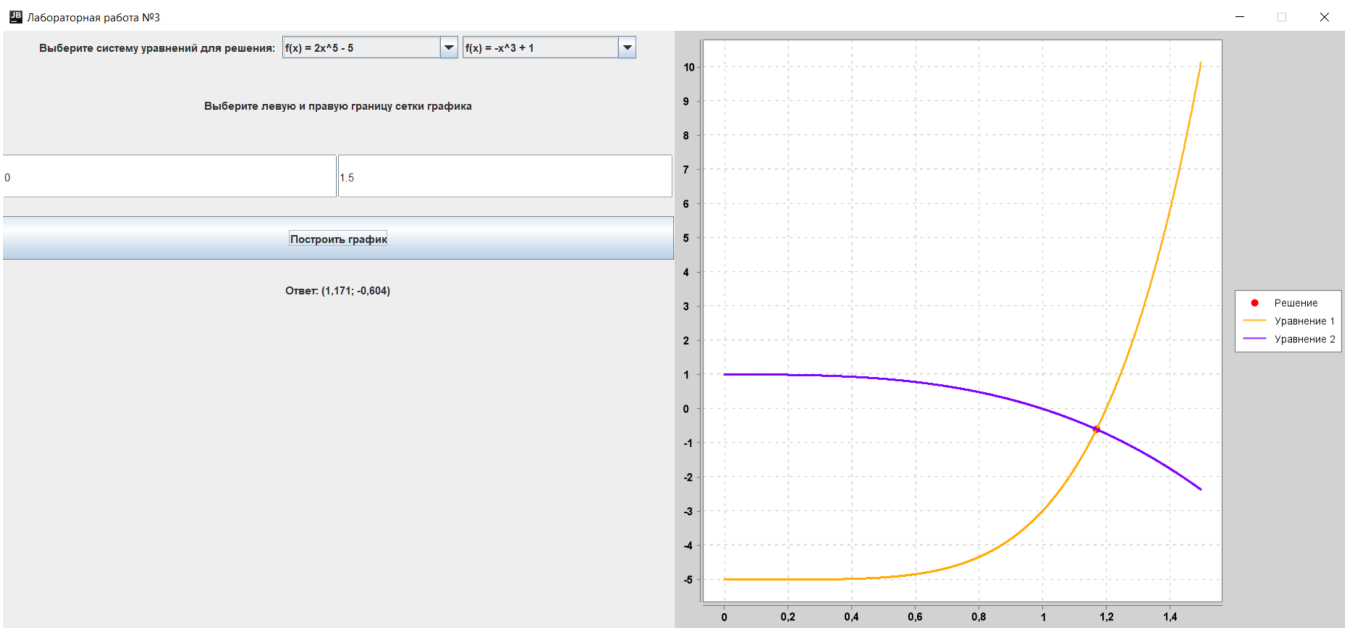


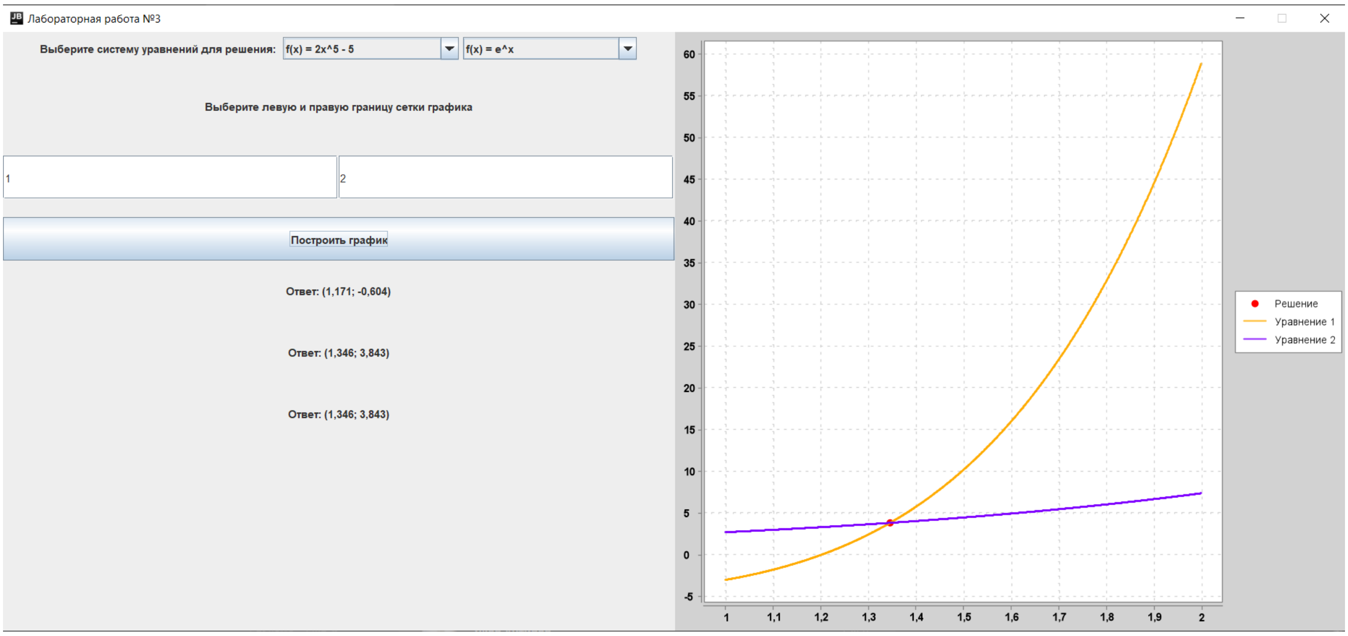


Решение систем нелинейных уравнений:


Лабораторная работа №3
Выберите систему уравнений для решения:
f(x) = x^5 - 5
f(x) = -x^3 + 1
Выберите левую и правую границу сетки графика
-2
2
Построить график
Ответ: (1,305; -1,221)
Решение
Уравнение 1
Уравнение 2


Лабораторная работа №3
Выберите систему уравнений для решения:
f(x) = x^5 - 5
f(x) = e^x
Выберите левую и правую границу сетки графика
-2
2
Построить график
Ответ: (1,305; -1,221)
Ответ: (1,580; 4,856)
Решение
Уравнение 1
Уравнение 2


Лабораторная работа №3
Выберите систему уравнений для решения:
f(x) = 2x^5 - 5
f(x) = -x^3 + 1
Выберите левую и правую границу сетки графика
0
1.5
Построить график
Ответ: (1,171; -0,604)
Решение
Уравнение 1
Уравнение 2


Лабораторная работа №3
Выберите систему уравнений для решения:
f(x) = 2x^5 - 5
f(x) = e^x
Выберите левую и правую границу сетки графика
1
2
Построить график
Ответ: (1,171; -0,604)
Ответ: (1,346; 3,843)
Ответ: (1,346; 3,843)
Решение
Уравнение 1
Уравнение 2
60
55
50
45
40
35
30
25
20
15
10
5
0
-5
1
1,1
1,2
1,3
1,4
1,5
1,6
1,7
1,8
1,9
2