Федеральное государственное автономное образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»

Лабораторная работа №2

Системы нелинейных уравнений

Вариант: 2ав

Выполнил:
Курносова Ирина
P3231

Преподаватель:
Перл Ольга Вячеславовна

Санкт-Петербург, 2022

Цель работы

Реализовать три численных метода:

1. Решение нелинейных уравнений **методом деления пополам**
2. Решение нелинейных уравнений **методом касательных**
3. Решение систем нелинейных уравнений **методом простой итерации**

Описание методов

Метод деления пополам:

Метод применим для численного решения уравнения f(x) = 0 для вещественной переменной x, где f-непрерывная функция, определенная на интервале [a, b] и где f(a) и f(b) имеют противоположные знаки. В этом случае говорят, что a и b заключают в скобки корень, так как по теореме о промежуточном значении непрерывная функция f должна иметь хотя бы один корень в интервале (a, b).

На каждом шаге метод делит интервал на две половины, вычисляя среднюю точку c = (a+b) / 2 интервала и значение функции f(c) в этой точке. Если c сам по себе не является корнем (что очень маловероятно, но возможно), теперь есть только две возможности: либо f(a) и f(c) имеют противоположные знаки и скобку корня, либо f(c) и f(b) имеют противоположные знаки и скобку корня. Метод выбирает подинтервал, который гарантированно будет скобкой, в качестве нового интервала, который будет использоваться на следующем шаге. Таким образом, интервал, содержащий ноль f уменьшается в ширину на 50% на каждом шаге. Процесс продолжается до тех пор, пока интервал не станет меньше заданной точности.

Метод касательных:

Геометрическая суть метода заключается в следующем: сначала выбирается один из концов заданного отрезка в качестве начального приближения, удовлетворяющий условию $f(x_0) * f''(x_0) > 0$. Далее проводится касательная к графику в выбранной точке x_0. Точка, полученная в результате пересечения касательной с осью Ox является первым приближением корня. Далее рассчитывается x_{i+1} по формуле: $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$

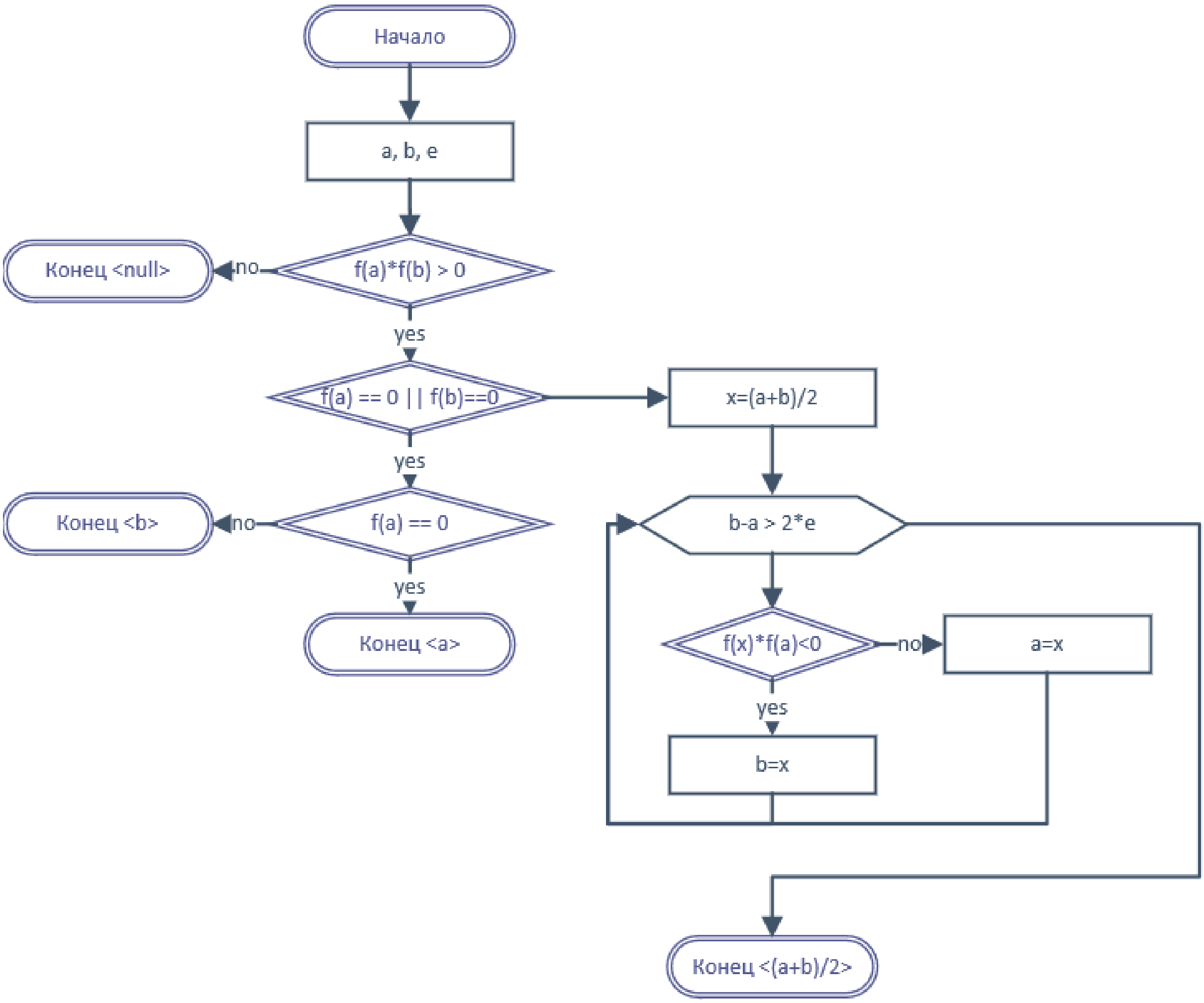
Данный процесс повторяется до тех пор, пока не будет достигнута необходимая точность.

Метод простой итерации:

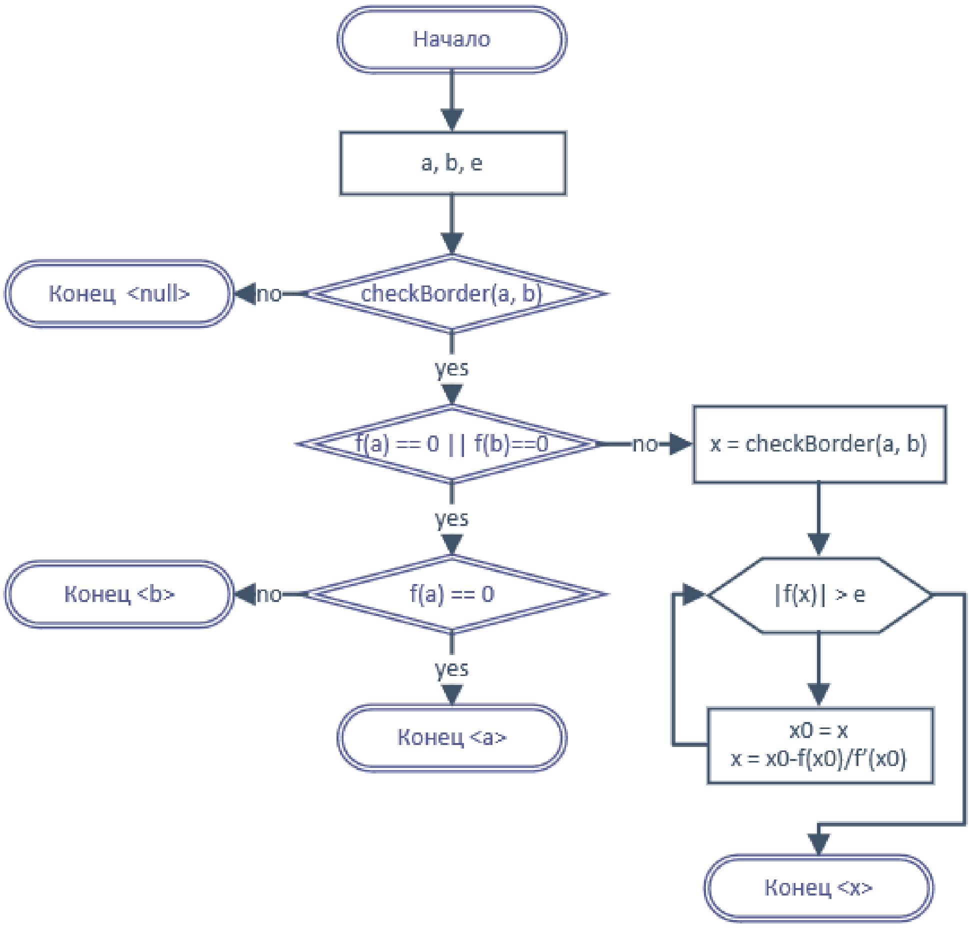
Используется для решения СНАУ вида $x_i = \varphi(x_1, x_2, \dots, x_n)$. Суть метода заключается в нахождении новых значений неизвестных через старые значения. Изначально задаются некоторые значения – начальные корни уравнений с большой погрешностью, а затем высчитываются новые корни через старые. Таким образом на каждой итерации уменьшается значение погрешности.

Блок-схемы методов

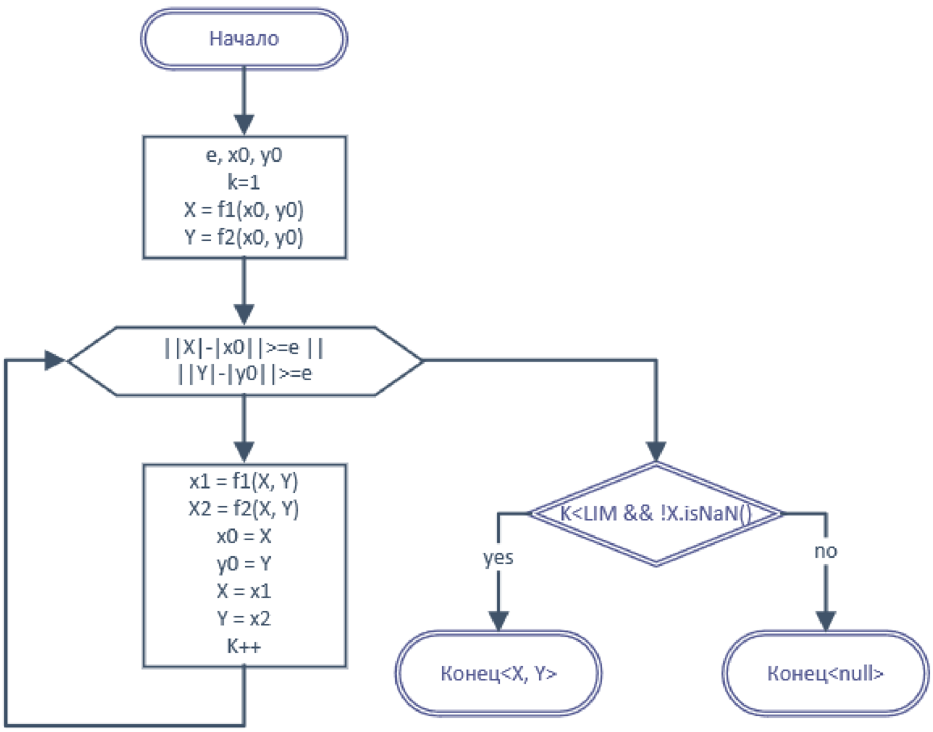
<u>Метод половинного деления</u>



Метод касательных



Метод простой итерации



Листинги вычислительных методов

1. Метод деления пополам

```
private static Double halfDivision(double e, double left, double right, int num) throws BorderExeption {
    Equation eq = Equations.getEquation(num);
    if (checkBorder(left, right, eq)) {

        if (eq.getEquation(left)==0 || eq.getEquation(right)==0) {
            if (eq.getEquation(left)==0)
                return left;
            else
                return  right;
        }

        double x = (left+right)/2;
        while (right-left>=2*e) {
            if (eq.getEquation(x)*eq.getEquation(left)<0)
                right=x;
            else
                left=x;
            x=(left+right)/2;
        }
        return (left+right)/2;
    }
    return null;
}
```

2. Метод касательных

```
private static Double tangents(double e, double left, double right, int num) throws BorderExeption {
    Equation eq = Equations.getEquation(num);
    if (checkBorder(left, right, eq)!=null) {
        if (eq.getEquation(left)==0 || eq.getEquation(right)==0) {
            if (eq.getEquation(left)==0)
                return left;
            else
                return  right;
        }
        Double x = checkBorder(left, right, eq);

        while (abs(eq.getEquation(x))>e) {
            double x0 = x;
            x = x0-eq.getEquation(x0)/eq.firstDerivative(x0);
        }
        return x;
    }
    return null;
}
```

3. Метод простой итерации

```
private static Double[] simpleIteartion(EquationSystem eq1, EquationSystem eq2) {
    double e = UserIO.getE();
    double x0 = UserIO.getX();
    double y0 = UserIO.getY();

    int k=1;
    Double[] res1 = {eq1.getEquation(x0, y0), eq2.getEquation(x0, y0), (double) 0};

    while ((abs(abs(res1[0])-abs(x0))>=e || abs(abs(res1[1])-abs(y0))>=e) && k<LIMIT) {
        double x1 = eq1.getEquation(res1[0], res1[1]);
        double x2 = eq2.getEquation(res1[0], res1[1]);

        x0 = res1[0];
        y0 = res1[1];
        res1[0] = x1;
        res1[1] = x2;
        k++;
    }
    res1[2] = (double)k;
    if (k<LIMIT && !res1[0].isNaN())
        return res1;

    return null;
}
```

Результаты работы программы

```
Доступные уравнения:
1)  0.5x^4 - x*e^x + 10 = 0
2)  sin(2x) * 5cos(x) + 1 = 0
3)  1/x - 2*sin(x) + 2^(-x) = 0
Введите номер уравнения:
1
Введите e:
0.0001
Введите левую границу:
2
Введите правую границу:
3

Результат метода половинного деления:
x = 2.41339111328125


Результат метода касательных:
x = 2.4134183797372453

Разница методов: 2.7266455995267336E-5
```

```
Доступные уравнения:
1)  0.5x^4 - x*e^x + 10 = 0
2)  sin(2x) * 5cos(x) + 1 = 0
3)  1/x - 2*sin(x) + 2^(-x) = 0
Введите номер уравнения:
3
Введите e:
0.01
Введите левую границу:
0.1
Введите правую границу:
1

Результат метода половинного деления:
x = 0.92265625


Результат метода касательных:
x = 0.9274988205118161

Разница методов: 0.004842570511816135 5
```

```
Система для решения:
1)  2*x^2 - xy -5x + 1 = 0
2)  x + 3*lg(x) - y^2 = 0
Введите e:
0.001
Введите начальное приближение X:
2
Введите начальное приближение Y:
4
Решение системы найдено за 15.0 итераций:
x = 3.48606248970948
y = 2.2609729086921306
```

Вывод

В ходе данной лабораторной работы было реализовано 2 численных метода для решения нелинейных уравнений и один метод для решения системы нелинейных уравнений. Исходя из проделанных тестов было выявлено, что метод касательных даёт необходимую точность за меньшее количество рабочих циклов, а также данный метод в принципе даёт большую точность при решении нелинейных уравнений, то есть он точнее и эффективнее, чем метод половинного деления. Но проблема данного метода заключается в том, что необходимо правильно подобрать отрезок изоляции корня, иначе не будет найдено начальное «хорошее» приближение и метод попросту не сработает.

Что касается метода простой итерации для решения СНАУ: эффективность данного метода сильно зависит от выбранных начальных приближений. Если начальное приближение такого, что в окрестностях данной точки существует решение системы, то данных метод буквально за несколько итераций найдёт решение заданной точности. Иначе, если начальное приближение далеко от истинного значение, то для высокой точности решения понадобится большое количество итераций и данных алгоритм не будет эффективным.