Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»

Факультет Программной инженерии и компьютерной техники

Лабораторная работа №2

По дисциплине “Вычислительная математика”

Вариант – 2бв

Группа: P32312

Выполнил: Обляшевский С.А.

Преподаватель:

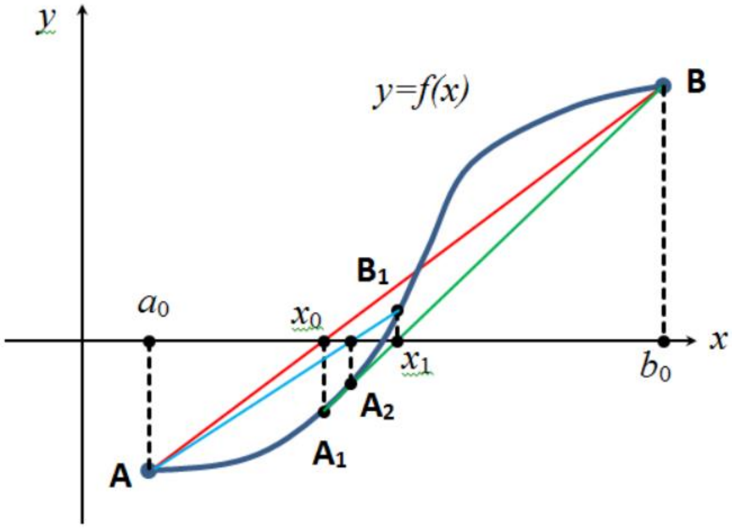
Перл О. В.

Санкт-Петербург, 2023

Описание метода, расчетные формулы:

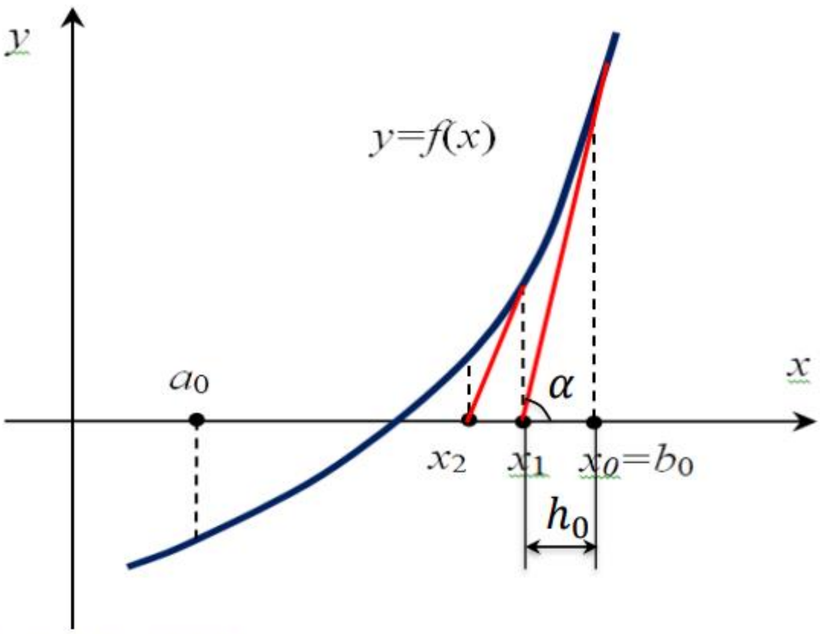
1. Метод хорд:

Это итерационный метод для нахождения корней нелинейного уравнения.
Функция $y = f(x)$ на отрезке [a, b] заменяется хордой и в качестве приближенного значения корня принимается точка пересечения хорды с осью абсцисс.
Выбираем новый интервал, на концах которого функция имеет разные знаки: [a, x] или [x, b].
Повторяем, пока не соблюдется критерий окончания сходимости метода: $|x_n - x_{n-1}|$ < eps.



2. Метод касательных:

Итерационный метод для нахождения корней нелинейного уравнения.
Функция $y = f(x)$ на отрезке [a, b] заменяется касательной и в качестве приближенного значения корня. В качестве новой точки берется пересечение касательной с осью абсцисс.
Метод выполняется до достижения условия $|x_n - x_{n-1}|$ < eps.



3. Метод простой итерации:

Итерационный метод для нахождения корней у СНАУ.

В каждом уравнении системы мы выражаем переменную.

$$x_n = \varphi_n(x_1, x_2, \dots, x_n)$$

Пользователь сам вводит начальное приближения (начальные значения переменных).

Затем производятся вычисления приближений по следующей формуле:

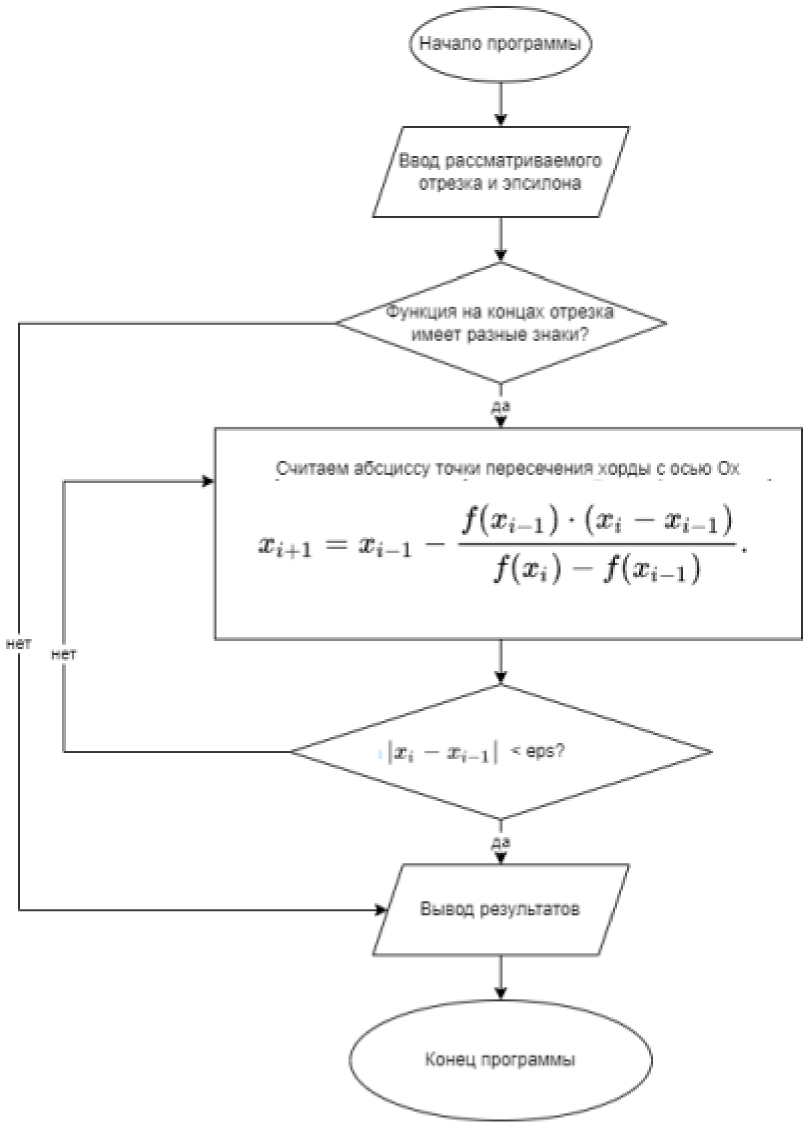
$$x_n^{(k+1)} = \varphi_n\left(x_1^k, x_2^k, \dots, x_n^k\right)$$

Критерий окончания метода:

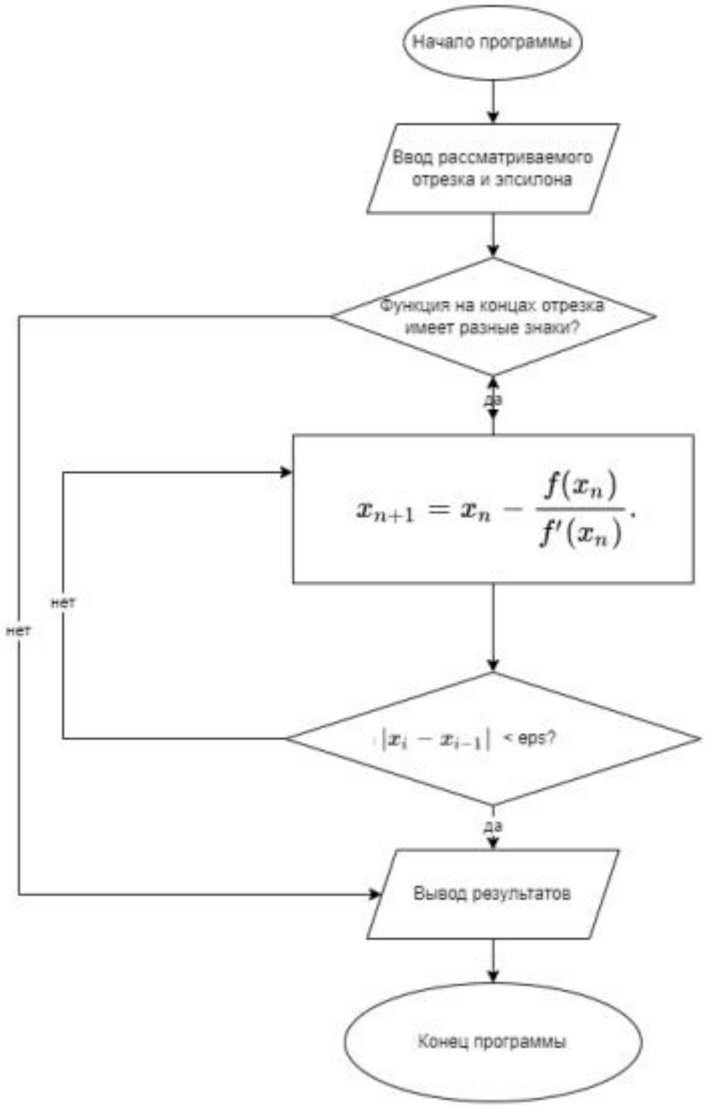
$$\max_{1<i<n} \left| x_i^{(k+1)} - x_i^k \right| \leq \varepsilon$$

Блок – схема:

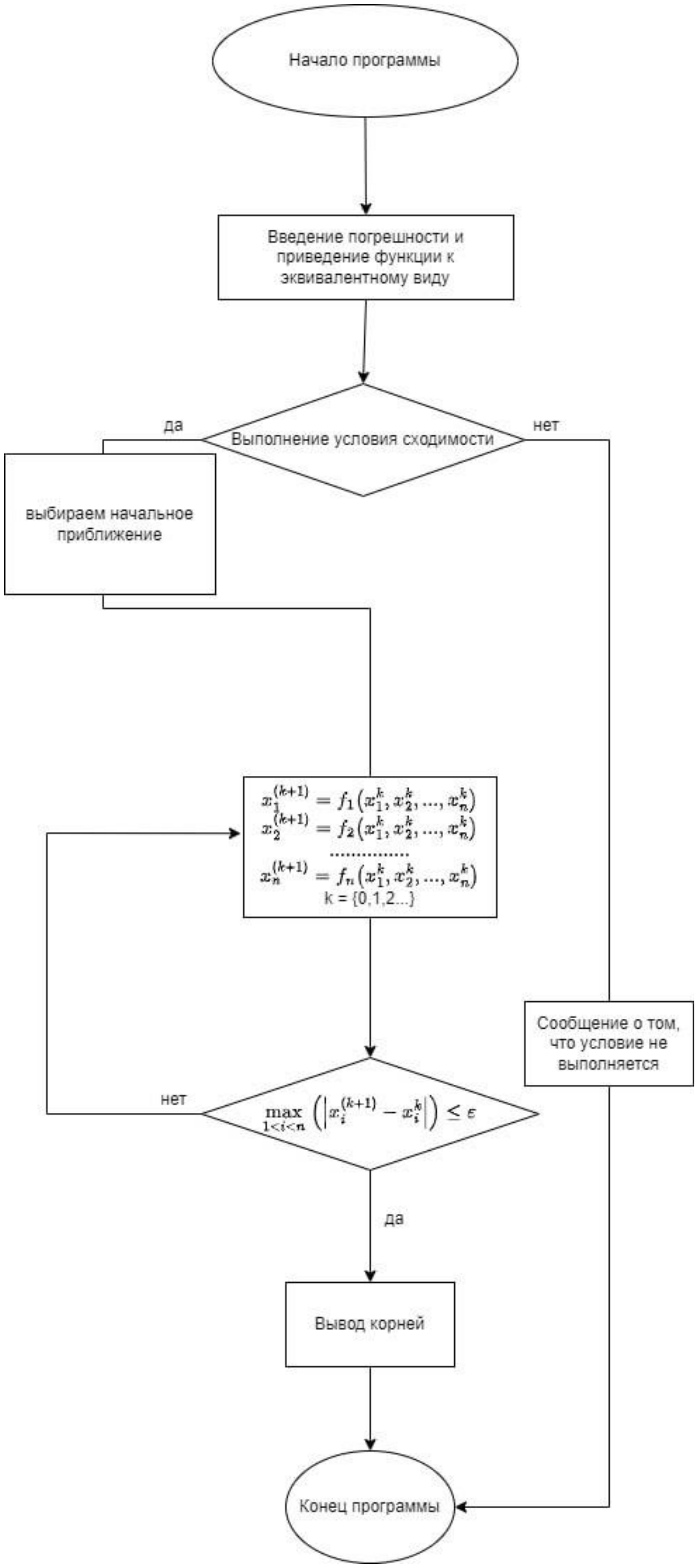
Метод хорд:



Метод касательных:



Метод простых итераций:



Листинг численного метода:

Метод хорд:

```
@Override
public double solve(Equation equation, double[] e, double eps) {
    double[] edges = e.clone();
    if (equation instanceof IrrationalEquation && (e[0] < 0 || e[1] < 0)){
        System.err.println("Метод хорд: в уравнении с корнем не может быть
отрицательных x.");
        return Double.MAX_VALUE;
    }
    System.out.print("Метод хорд, ответ - ");
    double x = 0;
    if (equation.getEquation(edges[0]) > 0 && equation.getEquation(edges[1])
> 0 ||
            equation.getEquation(edges[0]) < 0 &&
equation.getEquation(edges[1]) < 0){
        System.out.println("корней нет.");
        return Double.MAX_VALUE;
    }
    else {
        while (!isFinish(edges, eps)){
            x = edges[0] - ((edges[1] - edges[0]) *
equation.getEquation(edges[0]))/(equation.getEquation(edges[1]) -
equation.getEquation(edges[0]));
            edges[0] = edges[1];
            edges[1] = x;
        }
    }
    System.out.printf("%10s", new DecimalFormat("#.##########").format(x));
    System.out.println();
    return x;
}

private boolean isFinish(double[] edges, double eps){
    return Math.abs(edges[0] - edges[1]) < eps;
}
```

Метод касательных:

```
@Override
public double solve(Equation equation, double[] e, double eps) {
    double[] edges = e.clone();
    if (equation instanceof IrrationalEquation && (e[0] < 0 || e[1] < 0)){
        System.err.println("Метод касательных: в уравнении с корнем не может
быть отрицательных x.");
        return Double.MAX_VALUE;
    }
    System.out.print("Метод касательных, ответ - ");
    if (equation.getEquation(edges[0]) > 0 && equation.getEquation(edges[1])
> 0 ||
            equation.getEquation(edges[0]) < 0 &&
equation.getEquation(edges[1]) < 0){
        System.out.println("корней нет.");
        return Double.MAX_VALUE;
    }
    double x0 = edges[1];
    double x1 = x0 - equation.getEquation(x0)/equation.getDerivative(x0);
    while (!isFinish(x0, x1, eps)){
        x0 = x1;
```

```
        x1 = x0 - equation.getEquation(x0)/equation.getDerivative(x0);
    }
    System.out.printf("%10s", new DecimalFormat("#.##########").format(x1));
    System.out.println();
    return x1;
}

private boolean isFinish(double x0, double x1, double eps){
    return Math.abs(x0 - x1) < eps;
}
```

Метод простой итерации:

```
public void solve(EquationSystem system, double x, double y, double eps) {
    double curX = system.getX1(y);
    double curY = system.getY2(x);

    while (!(Math.abs(curX - x) < eps && Math.abs(curY - y) < eps)){
        x = curX;
        y = curY;
        curX = system.getX1(y);
        curY = system.getY2(x);
    }
    System.out.println("Решение системы:");
    System.out.printf("%10s", "x = " + new
DecimalFormat("#.########").format(curX));
    System.out.printf("%12s", ", y = " + new
DecimalFormat("#.########").format(curY));
}
```

Примеры и рез-ты работы:

Методы хорд и касательных:

```
Выберите, что нужно сделать:
1) Решить нелинейное уравнение.
2) Решить систему нелинейных уравнений.
1
Выберите, какое нелинейное уравнение решить:
1) Квадратное уравнение.
2) Кубическое уравнение.
3) Уравнение с корнем.
2
Вы выбрали уравнение вида: c1*x^3 + c2*x^2 + c3*x + c4 = 0
Введите константы:
Формат - c1 c2 c3 c4
1 0 0 -8
[1.0, 0.0, 0.0, -8.0]
Введите нижнюю границу рассматриваемого отрезка:
-3 5
Введите верхнюю границу рассматриваемого отрезка:
Рассматриваемый отрезок: [-3.0, 5.0]
Введите эпсилон:
0.0001
Метод хорд, ответ - 1,9999999079
Метод касательных, ответ - 2,0000000001
Разница в ответах между методами: 0,0000000922
```

Метод простой итерации:

```
Выберите, что нужно сделать:
1) Решить нелинейное уравнение.
2) Решить систему нелинейных уравнений.
2
Выберите систему:
1) y = 0.1x^3 - 7,
   y = x^2 + 0.5;
2) y = 5x^3,
   y = -x^2 - 3;
2
Введите эпсилон:
0.0001
Решение системы:
x = -0,91564001, y = -3,83834752
```

Вывод:

Я рассмотрел и реализовал некоторые методы для решения нелинейных уравнений и их систем. Все рассмотренные методы являются итерационными.

В методах хорд и касательных рассматриваются функции на определенных отрезках. Главные условия существования корня – разные знаки у значений функций на концах отрезка, а также непрерывность и монотонность функции на этом отрезке. Для метода касательных также важно условие сохранения знака первой и второй производной, первая производная также не должна быть равна нулю. Метод хорд напоминает метод половинного деления, но исходный отрезок делится не напополам, а в зависимости от расстояния от концов отрезка до корня. В методе касательных функция заменяется производной, а приближенным значением корня является пересечение касательной оси Ох. Достоинства данных методов – простота реализации и близость итоговых значений к истинному решению, недостатки – необходимость каждый раз подсчитывать производную (в методе касательных).

Метод простой итерации для СНАУ ищет приближенные значения корней данной системы. Из недостатков можно отметить, что условие сходимости соблюдается нечасто, из-за чего данный метод редко может гарантировать нахождение корня.