УНИВЕРСИТЕТ ИТМО

Факультет программной
инженерии и компьютерной техники
Вычислительная математика

Лабораторная работа № 2
Вариант 2ав

Преподаватель: Перл Ольга Вячеславовна
Выполнил: Геллер Л. А.
Группа: P3230

Санкт-Петербург, 2021

Описание метода, расчетные формулы

Решение нелинейных уравнений

- ### Метод деления пополам

1) Дано уравнение f(x) = 0, функция f(x) непрерывна на [a, b], f(a)*f(b) < 0
2) Поделим отрезок [a, b] пополам; Если $f\left(\frac{a+b}{2}\right) = 0$, то корень уравнения найден
3) В противном случае продолжаем рассматривать ту из половин отрезка, на концах которой сохранились разные знаки и повторяем пункты 2), 3)
4) Итерации продолжаем, пока значение функции не станет меньше по модулю, чем ε – небольшое положительное число – точность, заданная на входе

- ### Метод касательных

1) Дано уравнение f(x) = 0, функция f(x) непрерывна на [a, b], f(a)*f(b) < 0
2) Построим касательную к f(x), за следующее приближение x принимается точка пересечения касательной с Ox : $x_{к+1} = x_к - \dfrac{F(x_к)}{F'(x_к)}$,
3) Вычисления (п.2) продолжаем, пока $\left|F(x_{k+1})\right|$ не станет < ε – точности, или пока не выполняется $\left|x_{k+1} - x_k\right| < \varepsilon$

- ### Метод простых итераций

1) Дана система, преобразуемая к виду:

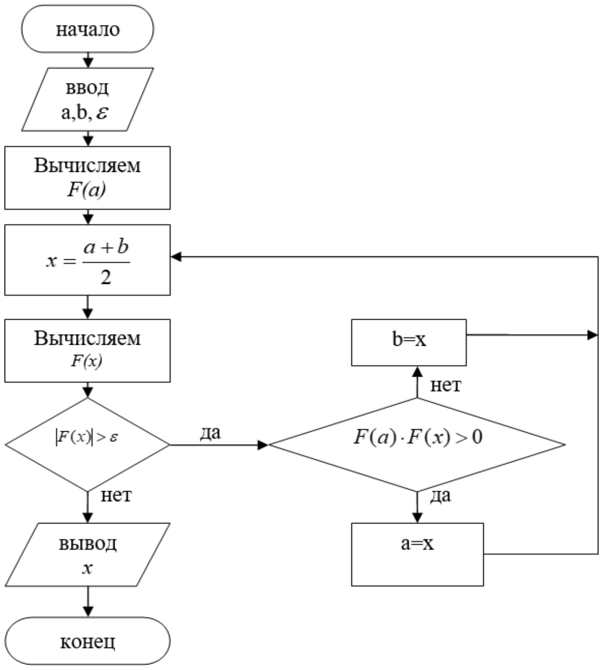
$$\begin{aligned} x_1 &= f_1(x_1, x_2, \ldots., x_n) \\ x_2 &= f_2(x_1, x_2, \ldots., x_n) \\ &\ldots\ldots\ldots\ldots\ldots\ldots \\ x_n &= f_n(x_1, x_2, \ldots., x_n) \end{aligned}$$

, причем функции определены и непрерывны вместе со своими частными производными в некоторой области D, которой принадлежит точное решение рассматриваемой системы

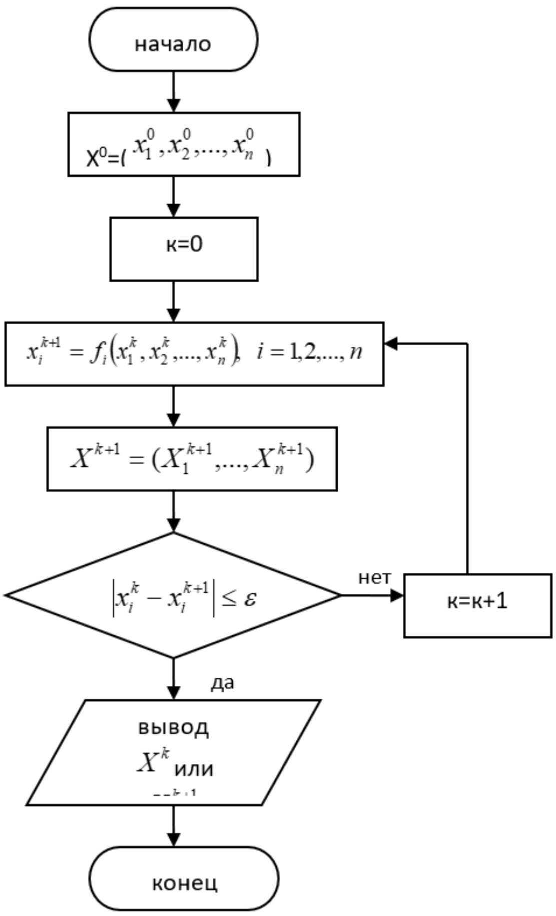
2) Выберем начальное приближение $X^0 = (x_1^0, x_2^0, \ldots, x_n^0)$
3) Подставим в уравнения (п.2) значения из X^0, таким образом получив новое приближение решений системы
4) Повторяем пункт 3, пока не выполнится: $\left|x_i^k - x_i^{k+1}\right| \le \varepsilon$

Блок-схема численного метода

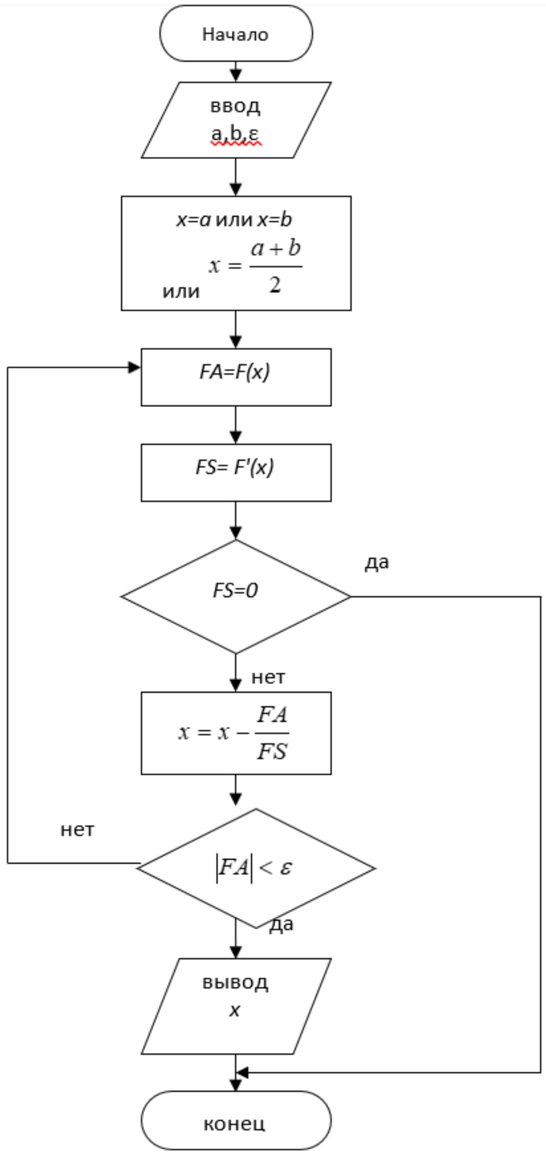
- Метод деления пополам



- Метод простых итераций (для систем)



- Метод касательных



Листинг реализованного численного метода программы

Метод бисекций:

```
public void solve() {
    execTime = System.nanoTime();
    yA = equation.getFunc().apply(a);
    iterations = 0;
    do {
        iterations++;
        x = (a + b) / 2;
        yX = equation.getFunc().apply(x);
        if (yA * yX > 0) a = x;
        else b = x;
    } while (Math.abs(yX) > epsilon);
    execTime = (System.nanoTime() - execTime)/1000;
}
```

Метод касательных:

```
public void solve() throws EquationException {
    execTime = System.currentTimeMillis();
    DoubleFunction<Double> fDerive = derive(equation.getFunc());
    x = (a+b)/2;
    double yX;
    double dX = fDerive.apply(x);
    do{
        iterations++;
        if (dX == 0) throw new EquationException("Производная = 0");
        yX = equation.getFunc().apply(x);
        dX = fDerive.apply(x);
        x -= (yX/dX);
    }
    while(Math.abs(yX) >= epsilon);
    execTime = System.currentTimeMillis() - execTime;
}
```

Метод простых итераций:

```
public void solve() throws EquationException {
    do {
        iterations++;
        previousVars = Arrays.copyOf(vars, vars.length);
        vars[0] = sof.getFuncs().get(0).apply(previousVars[1]);
        vars[1] = sof.getFuncs().get(1).apply(previousVars[0]);
        if (Arrays.stream(vars).anyMatch(x -> new Double(x).isInfinite() || new Double(x).isNaN()))
            throw new EquationException("Метод простых итераций не сходятся");
    } while (!isConverged(previousVars, vars));
}

private boolean isConverged(double[] previousX, double[] x){
    int n = previousX.length;
    for(int i=0; i<n; i++){
        if(Math.abs(x[i] - previousX[i]) > eps){ return false; }
    }
    return true;
}
```

Примеры и результаты работы программы на разных данных

1) Решение нелинейного уравнения методами биекций и касательных

```
Решить уравнение или систему уравнений?
1: Решить уравнение
2: Решить систему уравнений
1
Выберите тип решаемого уравнения:
1: Степенное
2: Логарифмическое
3: Тригонометрическое
1
Выберите степенное уравнение:
1: 2^(x+1) + 2^(x-1) + 2^x - 28 = 0
2: x^2 - 8x + 12 = 0
3: x^3 - 4x^2 + 2x - 8 = 0
2
Введите левую границу отрезка поиска корней:
3
Введите правую границу отрезка поиска корней:
7
Введите точность:
0,0001
Корень, найденный с помощью метода касательных:
x = 5.9999999999999885 +- 1.0E-4
Решение найдено за 5 итераций.
Время выполнения = 3.0 мс

Корень, найденный с помощью метода бисекций:
x = 6.0 +- 1.0E-4
Решение найдено за 2 итераций.
Время выполнения = 29.0 мкс
```

2) Решение системы нелинейных уравнений методом простых итераций

```
Решить уравнение или систему уравнений?
1: Решить уравнение
2: Решить систему уравнений
2
Выберите систему уравнений:
1:        y = x^2
          x = (y^2)/4
2:        y = x^2 - 2
          y = 2^x
3:        x + y = 0.5
          y + cos(x) = 0.5
3
Введите точность:
0,0001
x: 0.7390495805952085 +- 1.0E-4
y: -0.23913017652967106 +- 1.0E-4
Решение найдено за: 45 итераций
```

Вывод

Метод Бисекций является самым простым методом решения нелинейных уравнений. Методы Хорд и Касательных являются модификацией метода Бисекций. Они быстрее, но требуют более сложных вычислений (метод касательных также требует ввод начального приближения и поиск производной на каждой итерации). Вышеперечисленные методы обязательно сходятся, если найден отрезок, значения функции на концах которого обладают противоположными знаками (что гарантирует наличие на нем хотя бы одного корня).

Метод Ньютона обладает более высокой сходимостью, чем метод простых итераций за счет использования Якобиана (матрица частных производных), в результате чего повышается сложность вычислений.