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
«Системы нелинейных алгебраических уравнений»
Вариант «1бв»

Группа: P32312

Выполнил:
Воронин И.А.

Проверила:
Перл О.В.

Санкт-Петербург
2023

Описание методов

Метод хорд – сначала выбираются две точки на графике функции, которые находятся по разные стороны от корня. Затем проводится прямая через эти точки, которая пересекает ось абсцисс в точке, близкой к искомому корню. Далее одна из выбранных точек заменяется точкой пересечения хорды с осью абсцисс, и процедура повторяется до тех пор, пока не будет достигнута заданная точность.

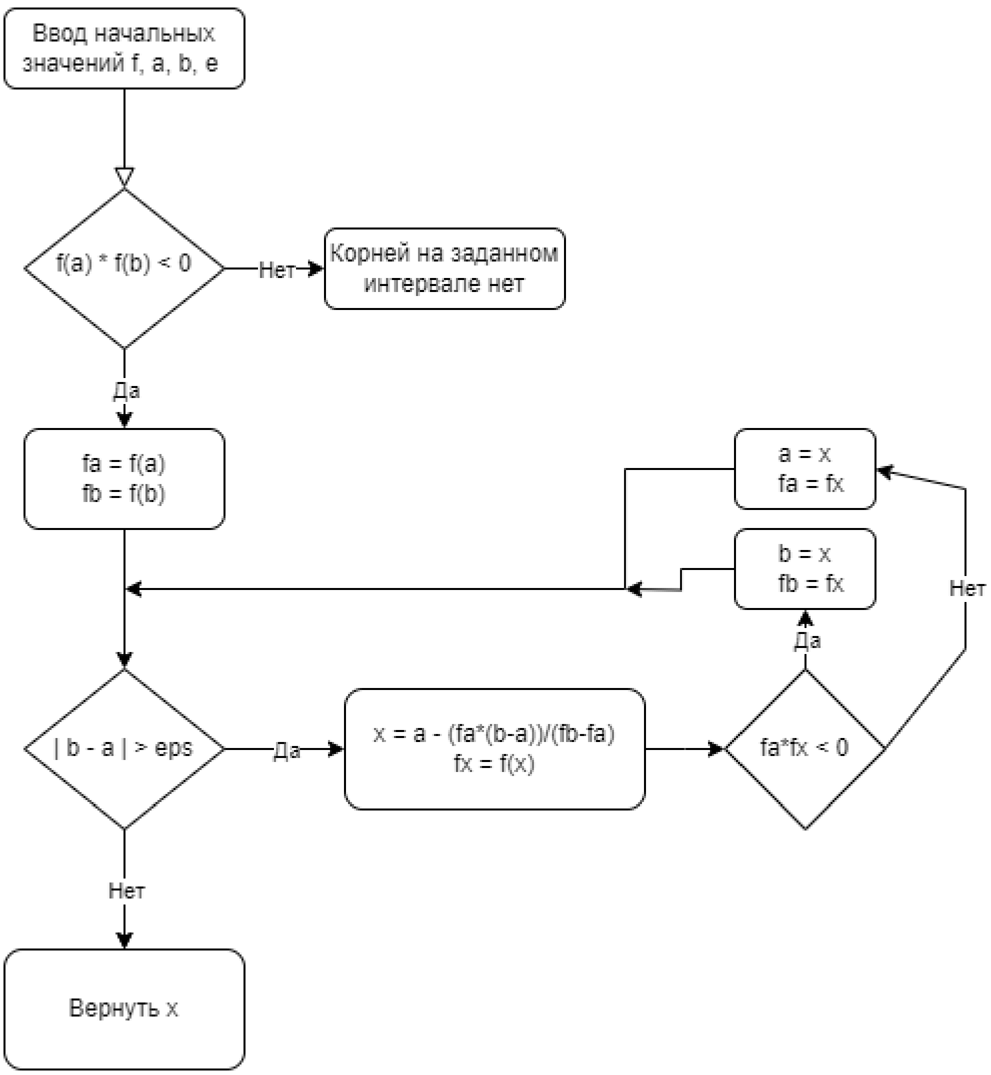
Рабочая формула метода:

$$x_{i+1} = x_i - \frac{f(x_i) \cdot (x_i - x_0)}{f(x_i) - f(x_0)}$$

Критерий окончания итерационного процесса:

$$|x_{i+1} - x_i| < \varepsilon$$

Блок-схема:



Листинг реализованного численного метода:

```
def solve(f,a,b, eps=1e-6, max_iterations = 1000):
    fa = f(a)
    fb = f(b)

    if fa * fb > 0:
        print("Метод хорд не применить: корни на заданном интервале отсутствуют")
        return 0

    i = 0
    while abs(b - a) > eps and i < max_iterations:
        x = a - (fa*(b-a))/(fb-fa)
        fx = f(x)
        if fa*fx < 0:
            b = x
            fb = fx
        else:
            a = x
            fa = fx
        i += 1
    return x
```

Результат работы:

```
=== 1 function ===
2x^4 - 3x^3 - 5x^2 - 8x = 0
================

=== 2 function ===
x^3 - 2x + 6 = 0
================

=== 3 function ===
ln(x) + (x + 1)^3 = 0
================

Введите номер функции: 3
Введите левую границу: 0.1
Введите правую границу: 0.2
Метод хорд: 0.18743892873054832
Метод касательных: 0.1874388122752963
Разница между методами: 1.1645525202497176e-07
```

Метод касательных - начинается с выбора начального приближения для корня уравнения. Затем производится вычисление производной функции в этой точке, и строится касательная к кривой графика функции в этой точке. Пересечение касательной с осью абсцисс дает новое приближенное значение корня уравнения. Этот процесс повторяется до тех пор, пока не будет достигнута требуемая точность.

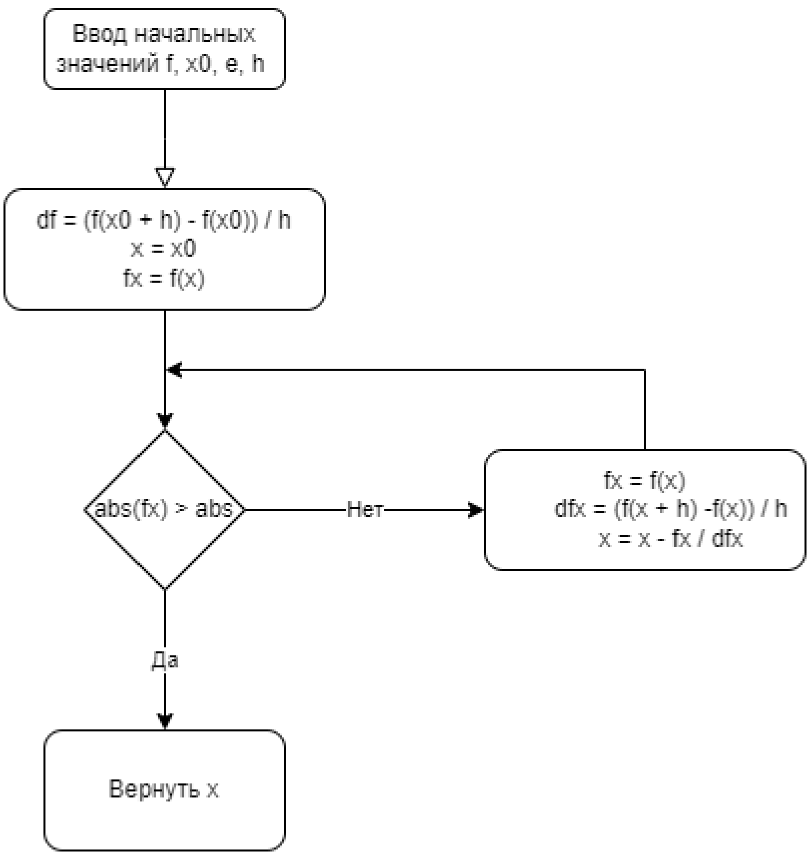
Рабочая формула метода:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Критерий окончания итерационного процесса:

$$\left|x_{n+1} - x_n\right| < \varepsilon$$

Блок-схема:



Листинг численного метода:

```
def solve(f, x0, eps=1e-5, max_iterations = 1000, h =
0.00001):
    x = x0
    for i in range(max_iterations):
        fx = f(x)
        dfx = (f(x + h) - f(x)) / h
        if abs(fx) < eps:
            return x
        x = x - fx / dfx
```

Результат работы:

```
=== 2 function ===
x^3 - 2x + 6 = 0
================

=== 3 function ===
ln(x) + (x + 1)^3 = 0
================

Введите номер функции: 2
Введите левую границу: -5
Введите правую границу: 5
Метод хорд: -2.179981072158157
Метод касательных: -2.179981072146664
Разница между методами: 1.149302875091962e-11
```

Метод Ньютона для СНАУ - аппроксимирует систему линейной системой и решает ее. На каждой итерации метод вычисляет значение функции и матрицу Якоби в текущей точке, решается СЛАУ и полученное решение используется для получения нового приближения.

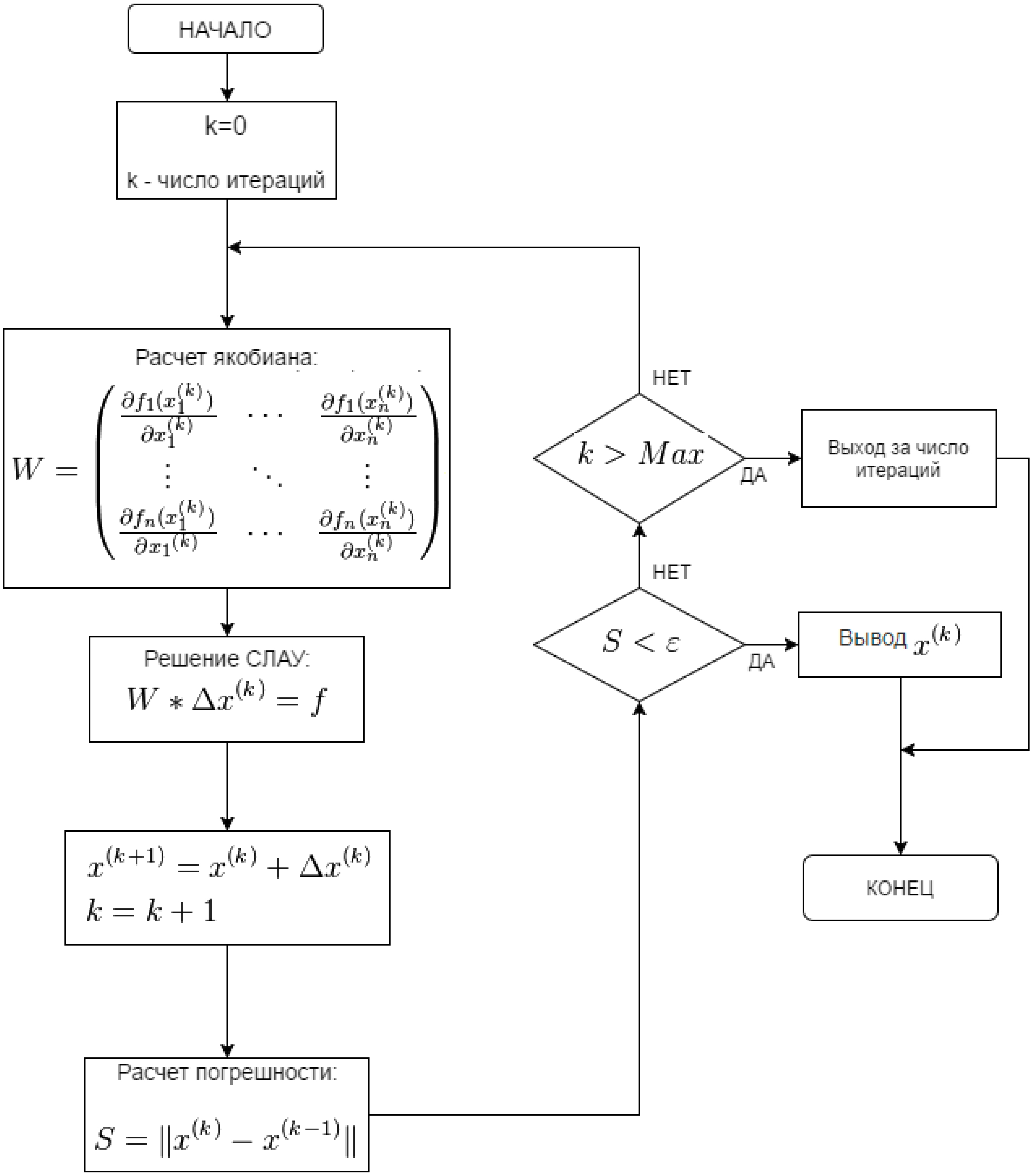
Рабочая формула метода:

$$\boldsymbol{F(xi)+\mathrm{J}(xi)(xi+1-xi) = 0}$$

Критерий окончания итерационного процесса:

$$\boldsymbol{|\, x_i \ - x_{i-1} \,| < eps}$$

Блок-схема:



Листинг численного метода:

```
def solve(funcs, start, eps=1e-5):
    f1 = funcs[0]
    f2 = funcs[1]
    x0 = start[0]
    y0 = start[1]
    while True:
```

```
        f1x0 = f1([x0, y0])
        f2x0 = f2([x0, y0])

        jac = create_jacobi_matrix(f1,f2,[x0,y0])

        det = jac[0][0]*jac[1][1] - jac[0][1]*jac[1][0]
        if det == 0:
            raise ValueError("Метод Ньютона неприменим")
        dx, dy = (jac[1][1]*f1x0 - jac[0][1]*f2x0)/det, (-jac[1][0]*f1x0 + jac[0][0]*f2x0)/det

        x1, y1 = x0 - dx, y0 - dy

        if abs(x1 - x0) < eps and abs(y1 - y0) < eps:
            break

        x0, y0 = x1, y1

    return [x0,y0]
```

Результат работы:

```
=== 1 system ===
sin(x)
x * y / 2
================

=== 2 system ===
tan(x*y + 0.4) - x^2
0.9 * x^2 + 2 * y^2 - 1
================

=== 3 system ===
tan(x*y) - x^2
0.5 * x^2 + 2 * y^2 - 1
================

Введите номер системы: 1
Введите начальное приближение x: 1
Введите начальное приближение y: 2
x = 2.4841722645954634e-13, y = 3.488363826228493
```

Вывод

Метод хорд:

Достоинства:

Простота реализации

Недостатки:

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

Выбор начального приближения

Метод касательных:

Достоинства:

Квадратичная сходимость.

Недостатки:

Вычисления производной на каждой итерации.

Выбор начального приближения

Метод Ньютона:

Достоинства:

Метод Ньютона имеет более быструю сходимость, чем метод простой итерации, особенно для систем уравнений с высокой степенью нелинейности.

Недостатки:

Метод Ньютона требует, чтобы матрица Якоби была невырожденной и ее обратная матрица была легко вычислимой.

Метод Ньютона обычно предпочтительнее метода простой итерации для систем нелинейных уравнений с большой степенью нелинейности и хорошо обусловленными матрицами Якоби. Однако метод простой итерации может быть более устойчивым и применимым для систем уравнений с малой степенью нелинейности и непрерывно дифференцируемыми функция