Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”

Факультет ПИиКТ

ITMO UNIVERSITY

ОТЧЁТ

По лабораторной работе №2

По предмету: Вычислительная математика

Вариант: 1АГ

Студент:

Андрейченко Леонид Вадимович

Группа Р3230

Преподаватель: Перл Ольга Вячеславовна

Санкт – Петербург

2022

Описание методов

В данной лабораторной работе мной были реализованы три метода решения нелинейных уравнений. Опишем алгоритм каждого из них:

Метод половинного деления

Идея метода: начальный интервал делим пополам, получаем начальное приближение к корню: $\boldsymbol{x_0} = \frac{\boldsymbol{a_0}+\boldsymbol{b_0}}{\mathbf{2}}$

Вычисляем f(x0). В качестве нового интервала выбираем ту половину отрезка, на концах которого функция имеет разные знаки: $[a_0, x_0]$ либо $[b_0, x_0]$. Другую половину отрезка $[a_0, b_0]$, на которой функция f(x) знак не меняет, отбрасываем. Новый интервал вновь делим пополам, получаем очередное приближение к корню: $x_1=(a_1+b_1)/2$.

Основная формула метода: $x_i = \frac{a_i+b_i}{2}$. Окончания итерационного процесса: $|b_n - a_n| \leq \varepsilon$

Метод простой итерации

Уравнение $f\ (x) = 0$ приведем к эквивалентному виду: $x = \varphi(x)$, выразив x из исходного уравнения. Зная начальное приближение: $x0 \in a, b$, найдем очередные приближения: $x1 = \varphi(x0) \rightarrow x2 = \varphi\ x1\ \ldots$

Рабочая формула метода: $\mathbf{x_{i+1}} = \boldsymbol{\varphi}(\boldsymbol{x_i})$

Условия сходимости метода простой итерации определяются следующей теоремой. Теорема. Если на отрезке локализации a, b функция $\varphi(x)$ определена, непрерывна и дифференцируема и удовлетворяет неравенству: $\varphi\ '\ x < q$, где $0 \leq q < 1$, то независимо от выбора начального приближения $x0 \in a, b$ итерационная последовательность метода будет сходится к корню уравнения.

Метод Ньютона (для решения систем)

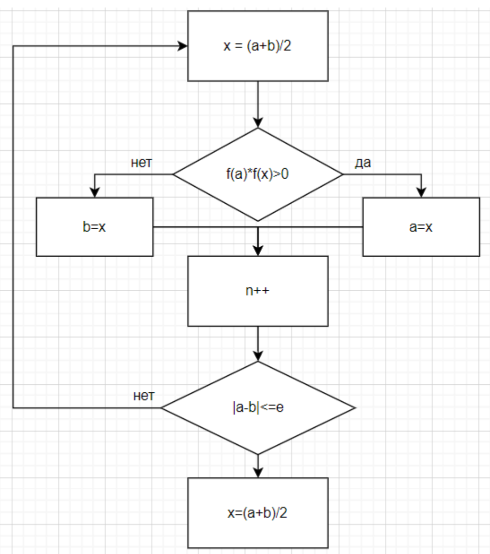
Метод решения систем нелинейных уравнений является обобщением метода Ньютона решения нелинейных уравнений, который основан на идее линеаризации. Пусть $F(x): R^1 \rightarrow R^1$ - дифференцируемая функция и необходимо решить уравнение F(x)=0. Взяв некоторое x0 в качестве начального приближения решения, мы можем построить линейную аппроксимацию F(x) в окрестности x0:$F(x0+h) \approx F(x0)+F'(x0)h$ и решить получающееся линейное уравнение $F(x0)+F'(x0)h=0$.

Таким образом получаем итеративный метод : $\mathbf{x_{k+1}=x_k-F'(x_k)^{-1}F(x_k),\ k=0,1,\ldots}$

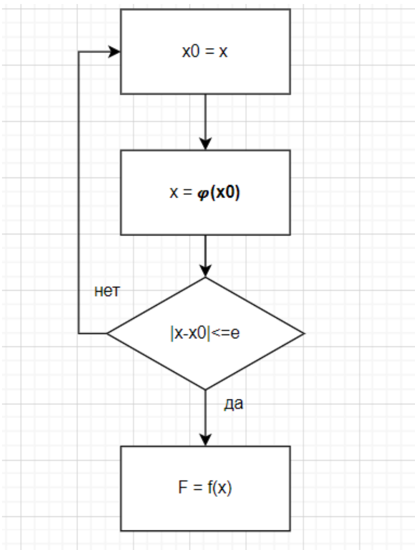
Итерационный процесс решения систем нелинейных уравнений методом Ньютона состоит в определении приращений $\Delta x1$, $\Delta x2$, … , Δxn к значениям неизвестных на каждой итерации. Критерий окончания итерационного процесса: $\max|\Delta x_i<=\varepsilon|$

Блок-схемы методов

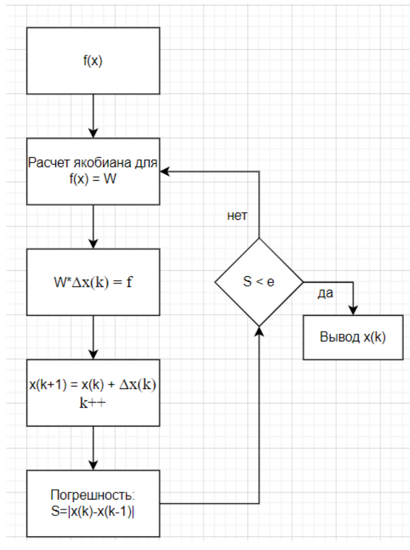
Метод половинного деления



Метод простой итерации



Метод Ньютона



Листинг программы

Метод половинного деления

```
def bisection_method(equation, a, b,
sigma):
    print("Метод деления пополам:")
    while b - a > sigma:
        middle = equation(a) *
equation((a + b) / 2)
        if middle > 0:
            a = (a + b) / 2
        elif middle < 0:
            b = (a + b) / 2
        else:
            break
    print("Корень: " + str((a + b) /
2) + "\n")
    return (a + b) / 2
```

Метод простой итерации

```
def
fixed_point_iteration_method(equation,
a, sigma):
    print("Метод простой итерации:")
    dx = float(1)
    while dx * dx > sigma:
        b = a
        a = equation(b)
        dx = a - b
    print("Корень: " + str(a) + "\n")
    return a
```

Метод Ньютона

```
def newton_method(kind, x, y,
sigma=0.001):
    X = np.array([x, y])
    dx = [1, 1]
    while abs(dx[0]) > sigma or
abs(dx[1]) > sigma:
        X_last = X
        dx =
np.dot(np.linalg.inv(df(X[0], X[1],
kind)), f(X_last[0], X_last[1],
kind))
        X += dx
    print("Вычисленные корни
системы: \nx1 = " + str(X[0]) +
"\nx2 = " + str(X[1]))
```

Пример работы

```
Выберите что вы хотите решить:
(1) Нелинейное уравнение
(2) Систему нелинейных уравнений
>> 1
Здравствуйте, выберите уравнение:
(1) : x*sin(x)=0
(2) : x-5-lg(x)=0
(3) 2^(x)-x^(2) =0
(4) Пользовательский ввод
>> 1
Метод деления пополам:
Корень: 3.1417968750000003

Метод простой итерации:
Корень: 3.1351031136815775

Разница в полученных решения составляет: 0.006693761318422808
```

```
Выберите что вы хотите решить:
(1) Нелинейное уравнение
(2) Систему нелинейных уравнений
>> 2
Здравствуйте, выберите уравнение:
(1)
x^2+xy-10
y+3xy^2-57
(2)
x+xy^3-9
xy+xy^2-6
>> 1
Вычисленные корни системы:
x1 = 1.9057377049180326
x2 = 3.0655737704918034
```

Вывод

В ходе выполнения данной лабораторной работы я узнал какими способами можно решать системы нелинейных уравнений и обычные нелинейные уравнения. Если обобщить реализованные мной методы, то можно сказать следующее: *Метод половинного деления* прост и надежен, обладает абсолютной сходимостью надо применять, когда требуется высокая надежность счета, а скорость несущественна. Также если интервал содержит несколько корней, то неизвестно к какому относится вычислительный процесс, имеет линейную сходимость. *Метод простой итерации* – довольно прост, однако недостатком этого метода является его сходимость 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. Если $\varphi' x \approx 1$, то сходимость может быть очень медленной. *В методе Ньютона* важен удачный выбор начального приближения для обеспечения хорошей сходимости. Сходимость ухудшается с увеличением числа уравнений системы.