ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет программной инженерии и компьютерной техники
Системное и прикладное программное обеспечение

Дисциплина «Вычислительная математика»

Отчет к лабораторной работе № 2
«Численное решение нелинейных уравнений и систем»
Вариант 17

Выполнил:
Студент группы P3213
Харламов Александр Сергеевич

Преподаватель:
Малышева Татьяна Алексеевна

Санкт-Петербург
2022 год

Цель работы

Изучить численные методы решения нелинейных уравнений и их систем, найти корни заданного нелинейного уравнения, выполнить программную реализацию методов.

Задание лабораторной работы

1. Найти корни нелинейного уравнения:

- Крайний левый методом хорд

- Крайний правый методом Ньютона

- Центральный методом простой итерации

2. Решить систему нелинейных уравнений методом простой итерации

3. Вычислить погрешности

Порядок выполнения работы

1. Отделить корни заданного нелинейного уравнения графически

2. Определить интервалы изоляции корней

3. Уточнить корни нелинейного уравнения различными методами

4. Вывести таблицу результатов и построить график

5. Решить систему нелинейныйх уравнений методом простой итерации

6. Вывести решения и построить график

Рабочие формулы используемых методов

Рабочая формула метода хорд:

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

Рабочая формула метода Ньютона:

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

Рабочая формула метода простой итерации:

$x_{i+1}=\varphi(x_i)$, где $x=\varphi(x)$ - эквивалентный вид уравнения $f(x)=0$

Листинг программы

https://github.com/itookyourboo/itmo_comp_math/tree/master/lab2_non-linear-equations

```
def horde_method(f, left, right, fix=-1, eps=10e-3):
    res = Result(
        header='№ a b x f(a) f(b) f(x) |a-b|'.split()
    )

    x0 = left if fix == -1 else right

    for i in range(1, LIMIT + 1):
        x1 = (left * f(right) - right * f(left)) / (f(right) - f(left))

        res.data.append([
            i, left, right, x1, f(left), f(right), f(x1), abs(left - right)
        ])

        if (
                abs(x1 - x0) <= eps or
                abs(f(x1)) <= eps
        ):
            res.root = x1
            res.error = abs(x1 - x0)
            break

        if f(x1) * f(left) < 0:
            right = x1
        else:
            left = x1

        x0 = x1

    return res
```

```python
def newton_method(f, df, x0, eps=10e-3):
    res = Result(
        header="№ x_k f(x_k) f'(x_k) x_{k+1} |x_k-x_{k+1}|".split()
    )

    for i in range(1, LIMIT + 1):
        x1 = x0 - f(x0) / df(x0)

        res.data.append([
            i, x0, f(x0), df(x0), x1, abs(x1 - x0)
        ])

        if (
                abs(x1 - x0) <= eps or
                abs(f(x1) / df(x1)) <= eps or
                abs(f(x1)) <= eps
        ):
            res.root = x1
            res.error = abs(x1 - x0)
            break

        x0 = x1

    return res
```

```python
def simple_iteration_method(f, phi, x0=1, eps=10e-3):
    res = Result(
        header="№ x_k f(x_k) x_{k+1} phi(x_k) |x_k-x_{k+1}|".split()
    )

    for i in range(1, LIMIT + 1):
        x1 = phi(x0)

        res.data.append([
            i, x0, f(x0), x1, phi(x0), abs(x1 - x0)
        ])

        if (
                abs(x1 - x0) <= eps
        ):
            res.root = x1
            res.error = abs(x1 - x0)
            break

        x0 = x1

    return res
```

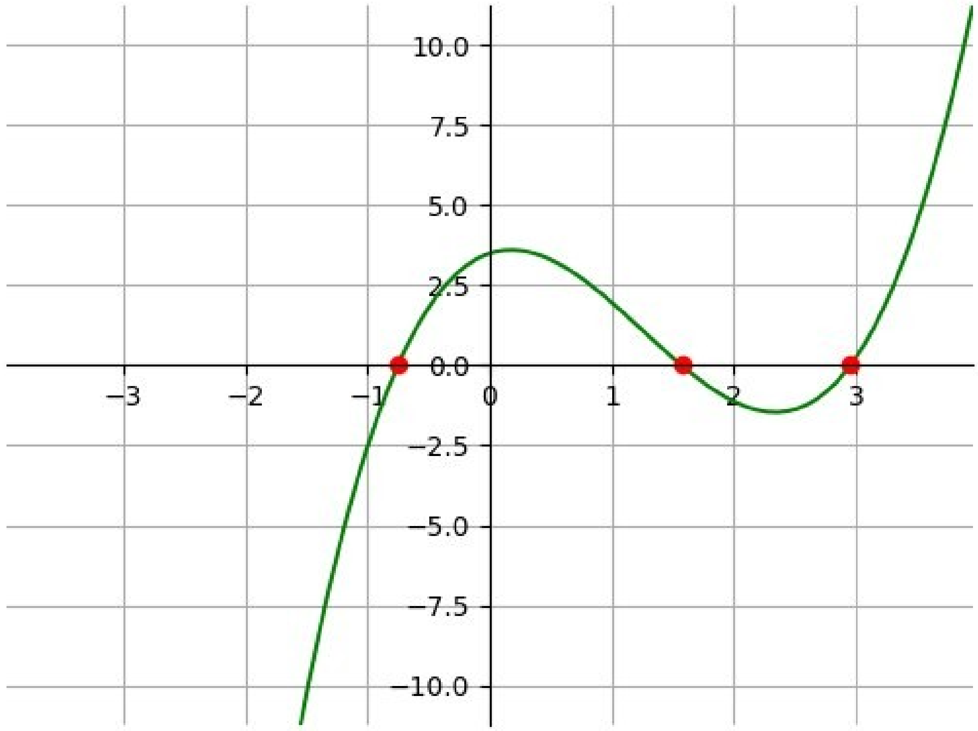
```python

```

```python
def system_simple_iteration_method(xs, x0=None, eps=1e-3):
    n = len(xs)
    if x0 is None:
        x0 = [1] * n
    x1 = x0[:]
    err = [0] * n

    for i in range(1, LIMIT + 1):
        converges = True
        for j in range(n):
            x1[j] = xs[j](*x0)
            err[j] = abs(x1[j] - x0[j])
            if err[j] > eps:
                converges = False

        if converges:
            return Result(i, True, x1, err)

        x0 = x1[:]

    return Result(i, False, [0] * n, [0] * n)
```

Изоляция корней

```
Интервалы
+--------+---------+--------+
|   x    |   f(x)  | f'(x)  |
+--------+---------+--------+
| -2.000 | -22.130 | 28.370 |
| -1.500 | -10.265 | 19.340 |
| -1.000 |  -2.540 | 11.810 |
| -0.500 |  1.795  | 5.780  |
| 0.000  |  3.490  | 1.250  |
| 0.500  |  3.295  | -1.780 |
| 1.000  |  1.960  | -3.310 |
| 1.500  |  0.235  | -3.340 |
| 2.000  |  -1.130 | -1.870 |
| 2.500  |  -1.385 | 1.100  |
| 3.000  |  0.220  | 5.570  |
+--------+---------+--------+
Корни
+--------+--------+---------------------+
|   x1   |   x2   | Единственный корень? |
+--------+--------+---------------------+
| -1.000 | -0.500 |         True        |
| 1.500  | 2.000  |         True        |
| 2.500  | 3.000  |         True        |
+--------+--------+---------------------+
```

Результаты работы программы



Функция: x^3 - 3.78x^2 + 1.25x + 3.49

Границы левого корня через пробел (-2 0):

Нулевое приближение правого корня (10):

Нулевое приближение центрального корня (-0.5):

Погрешность (0.01):

Вывод в файл:

Левый корень методом хорд: -0.744 (err: 0.005733230233266218)

№	a	b	x	f(a)	f(b)	f(x)	|a-b|
1	-2.000	0.000	-0.272	-22.130	3.490	2.849	2.000
2	-2.000	-0.272	-0.469	-22.130	2.849	1.967	1.728
3	-2.000	-0.469	-0.594	-22.130	1.967	1.202	1.531
4	-2.000	-0.594	-0.667	-22.130	1.202	0.680	1.406
5	-2.000	-0.667	-0.706	-22.130	0.680	0.368	1.333
6	-2.000	-0.706	-0.728	-22.130	0.368	0.194	1.294
7	-2.000	-0.728	-0.739	-22.130	0.194	0.101	1.272
8	-2.000	-0.739	-0.744	-22.130	0.101	0.052	1.261

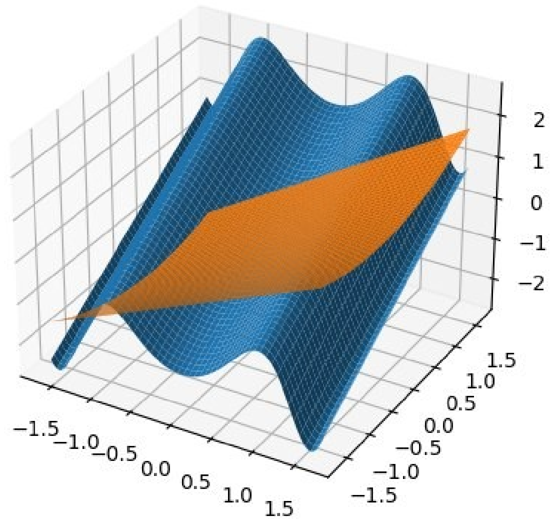
Центральный корень методом простой итерации: 1.576 (err: 0.0012443213660149333)

№	x_k	f(x_k)	x_{k+1}	phi(x_k)	|x_k-x_{k+1}|
1	-0.500	1.795	-1.243	-1.243	0.743
2	-1.243	-5.823	1.574	1.574	2.817
3	1.574	-0.009	1.576	1.576	0.001

Правый корень методом Ньютона: 2.959 (err: 0.014991073231303442)

№	x_k	f(x_k)	f'(x_k)	x_{k+1}	|x_k-x_{k+1}|
1	10.000	637.990	225.650	7.173	2.827
2	7.173	186.998	101.366	5.328	1.845
3	5.328	54.088	46.130	4.155	1.173
4	4.155	15.165	21.636	3.454	0.701
5	3.454	3.923	10.934	3.096	0.359
6	3.096	0.801	6.596	2.974	0.121
7	2.974	0.079	5.302	2.959	0.015

Корни: -0.744 1.576 2.959




```
Выберите функцию №1
1. f1(x1, x2) = 0.1 * x1^2 + x1 + 0.2 * x2^2 - 0.3
2. f1(x1, x2) = x1 - sin(2 * x2^2 + 3)
2
Выберите функцию №2
1. f2(x1, x2) = 0.2 * x1^2 + x2 + 0.1 * x1 * x2 - 0.7
2. f2(x1, x2) = exp(x1^3 - 8 * x1^2) + 4 * x2
1
Начальные приближения (1 1):
Погрешность (0.001):
Вывод в файл:
Решение: -0.659 0.656
Погрешности: 0.0005433345195254846 0.00043804837652694495
Количество итераций 15
```

Выводы

Я познакомился с такими методами решения нелинейных уравнений, как метод хорд, метод Ньютона и метод простых итераций. Последний мне не понравился, так как постоянно сходится не к тому корню, к которому задумывалось. Также я поработал с библиотеками numpy и matplotlib, поэтому лабораторная была полезной!