, алеМіністерство освіти і науки України
Національний університет “Львівська політехніка”
Інститут комп’ютерних наук та інформаційних технологій
Кафедра програмного забезпечення

Звіт
Про виконання лабораторної роботи №7
На тему:
«Чисельні методи розв’язування систем нелінійних рівнянь»
з дисципліни «Чисельні Методи»

Лектор:
доцент каф. ПЗ
Мельник Н. Б.
Виконав:
ст. гр. ПЗ-11
Солтисюк Д.А.
Прийняла:
доцент каф. ПЗ
Мельник Н. Б.

« __ » ________ 2022 р.

∑ = _____ _________ .

Львів – 2022

Тема: Чисельні методи розв'язування систем нелінійних рівнянь.
Мета: Ознайомлення на практиці з методом ітерацій та методом Ньютона розв'язування систем нелінійних рівнянь.

Теоретичні відомості

Метод простої ітерації - суть методу полягає у перетворенні системи з двох нелінійних рівнянь до вигляду:

$$\begin{cases} x = \varphi_1(x, y), \\ y = \varphi_2(x, y), \end{cases}$$

Після цього ітераційний процес зводиться до такого вигляду:

$$\begin{cases} x_{n+1} = \varphi_1(x_n, y_n), \\ y_{n+1} = \varphi_2(x_n, y_n), \end{cases} \qquad n = 1, 2, \ldots$$

Для збіжності ітераційного процесу мають виконуватися такі умови:

1) функції $\varphi_1(x, y)$ та $\varphi_2(x, y)$ визначені та неперервно-диференційовані в області D ;
2) початкове наближення і всі наступні наближення належать області D ;
3) в області D виконуються нерівності:

$$\left|\frac{\partial \varphi_1}{\partial x}\right| + \left|\frac{\partial \varphi_2}{\partial x}\right| \le q_1 < 1 , \qquad \left|\frac{\partial \varphi_1}{\partial y}\right| + \left|\frac{\partial \varphi_2}{\partial y}\right| \le q_2 < 1 .$$

Ітераційний процес припиняється якщо $|x_{n+1} - x_n| + |y_{n+1} - y_n| < e$(точність).

Метод простої ітерації, який застосовують для знаходження розв'язку одного нелінійного рівняння або системи двох нелінійних рівнянь, має перший порядок збіжності (лінійну збіжність).

Метод Ньютона – суть методу полягає у перетворенні системи нелінійних рівнянь до вигляду:

$$\begin{cases} f_1(x+\Delta_x, y+\Delta_y)=0, \\ f_2(x+\Delta_x, y+\Delta_y)=0. \end{cases}$$

Після цього ми записуємо якобіан, складений складеної з частинних похідних функцій f_1 і f_2 в деякій точці:

$$\Delta(x_0,y_0)=\begin{vmatrix} \dfrac{\partial f_1(x_0,y_0)}{\partial x} & \dfrac{\partial f_1(x_0,y_0)}{\partial y} \\ \dfrac{\partial f_2(x_0,y_0)}{\partial x} & \dfrac{\partial f_2(x_0,y_0)}{\partial y} \end{vmatrix} \neq 0,$$

а поправки Δ_x і Δ_y визначимо за правилом Крамера із системи:

$$\Delta_x=-\frac{1}{\Delta(x_0,y_0)}\begin{vmatrix} f_1(x_0,y_0) & \dfrac{\partial f_1(x_0,y_0)}{\partial y} \\ f_2(x_0,y_0) & \dfrac{\partial f_2(x_0,y_0)}{\partial y} \end{vmatrix}, \qquad \Delta_y=-\frac{1}{\Delta(x_0,y_0)}\begin{vmatrix} \dfrac{\partial f_1(x_0,y_0)}{\partial x} & f_1(x_0,y_0) \\ \dfrac{\partial f_2(x_0,y_0)}{\partial x} & f_2(x_0,y_0) \end{vmatrix}.$$

Наступне наближення розв'язку системи отримаємо у вигляді:

$$\begin{cases} x_{n+1}=x_n+\Delta_x, \\ y_{n+1}=y_n+\Delta_y, \end{cases} \qquad n=0,1,2,\ldots$$

Індивідуальне завдання

Розв'язати систему нелінійних рівнянь з точністю $\varepsilon = 10^{-3}$ методом ітерацій та методом Ньютона.

$$24.\begin{cases}\cos x + y = 1{,}2\\ 2x - \sin(y - 0{,}5) = 2\end{cases}$$

Код програми

simple_iteration.py

```
from typing import Callable

from common.main import NonLinearEqSysOrientedMethod
from common.utils import print_header
from prettytable import PrettyTable


class SimpleIterationMethod(NonLinearEqSysOrientedMethod):

    def __init__(self, eqsys: list[Callable]):
        super().__init__("Simple iterations", eqsys)

    def execute_method(self, tolerance=1e-4):
        """
        1) Making an approximate guess
        2) Linearly iterating using equeation function to get next x and y
        3) Checking if can be considered solution:
           (|x_next - x| + |y_next - y| < eps)
        """
        x, y = 0, 0

        print_header("Initial guess")
        print([x, y])
        print("\n")

        x_next, y_next = x, y
        iterations = 0

        f1, f2 = self.eqsys
        t = PrettyTable(["Iteration", "X", "Y", "Precision"])
        result = None

        while True:
            iterations += 1
            x_next, y_next = f2(y), f1(x)
            precision = abs(x_next - x) + abs(y_next - y)
            x, y = x_next, y_next
```

```python
            t.add_row([iterations, x, y, precision])

            if precision < tolerance:
                result = (x, y)
                break

        print(t)

        return result
```

newton_iteration.py

```python
import math
from typing import Callable

import numpy as np
from common.gaussian import GaussianEliminationMethod
from common.main import NonLinearEqSysOrientedMethod
from prettytable import PrettyTable


class NewtonIterationMethod(NonLinearEqSysOrientedMethod):

    def __init__(self, eqsys: list[Callable]):
        super().__init__("Newton iteration", eqsys)

    def execute_method(self, tolerance=1e-4):
        """
        1) May (x0, y0) -> approximate solution
           delta_x, delta_y -> corrections
        2) Delta(x0, y0) can be found from Jakobian (det of the Jakobi matrix)
        3) delta_x, delta_y -> using gaussian elimintaion method
        4) x_next = x + delta_x     y_next = y + delta_y
        """
        X = np.array([0, 0], dtype=float)

        def f(X):
            x, y = X
            return np.array(
                [math.cos(x) + y - 1.2, 2 * x - math.sin(y - 0.5) - 2])

        def Jf(X):
            x, y = X
            return np.array([[-math.sin(x + 0.5), 1], [2, math.cos(y - 2)]])

        X_delta = X.copy()
        t = PrettyTable(["Iteration", "X", "Y", "Precision"])

        result = None
        iterations = 0
        while True:
            iterations += 1
            A = Jf(X)
            B = f(X)
            X_delta = GaussianEliminationMethod(A, B).compile(silent=True)
            X -= X_delta
            x, y = X
            norm = np.linalg.norm(B)

            t.add_row([iterations, x, y, norm])

            if norm < tolerance:
                result = (x, y)
```

```python
            break
    print(t)
    return result
```

Протокол роботи

>>> Simple iterations method

Initial guess
[0, 0]

Iteration	X	Y	Precision
1	0.7602872306978985	0.19999999999999996	0.9602872306978985
2	0.8522398966693302	0.4753618985438697	0.3673145645153014
3	0.9876821955854084	0.5417012987973607	0.20178169916956923
4	1.020844606716731	0.649373870289418	0.1408349826233798
5	1.0744095021362239	0.6773539314957908	0.0815449566258658
6	1.0882128159509048	0.7237484436245609	0.06019782594345102
7	1.110943088967333	0.7359308070843579	0.03491263647622522
8	1.1168740538762336	0.7561834044262912	0.026183562250833847
9	1.1266951843831754	0.7615060214596817	0.01514374754033232
10	1.1292678319226266	0.7703536119288203	0.011420238008589823
11	1.133536110445549	0.7726781247549311	0.0065927913490332335
12	1.1346557878768575	0.7765409542856905	0.004982506962067812
13	1.1365148336013218	0.7775555523721622	0.002873643810936022
14	1.1370027877802074	0.7792412996522058	0.002173701458929145
15	1.1378132079999983	0.7796840083842325	0.001253128951817617
16	1.1380259746556105	0.7804195036454868	0.0009482619168665885
17	1.1383793952475922	0.7806126452923352	0.000546562238830095
18	1.1384721913771836	0.780933509300789	0.00041366013804511326
19	1.1386263411052133	0.7810177658656403	0.00023840629288107706
20	1.1386668173421137	0.7811577379376979	0.00018044830895802555
21	1.1387340567245536	0.7811944930922596	0.00010399453700160599
22	1.138751712609734	0.7812555525056459	7.871529856662285e-05

Results
(1.138751712609734, 0.7812555525056459)

>>> Newton iteration method

Iteration	X	Y	Precision
1	0.8907603456272517	0.6270532584696111	1.6219766340998225
2	1.0531458091385146	0.730939866349515	0.3635068829639399
3	1.109852209483029	0.7618577249217229	0.12764024168248814
4	1.127624212843223	0.7729645630571234	0.04124638503952255
5	1.1340167877298677	0.7775749628843557	0.0153171957493659
6	1.1366634391487223	0.7796178201471612	0.006450134111898808
7	1.1378349988094905	0.7805451343650538	0.0028745486820954584
8	1.1383666842823914	0.7809697748322206	0.0013078526590731008
9	1.1386101507064772	0.7811648791749961	0.0005994718279190011
10	1.1387220145868784	0.7812546429191219	0.0002755457532436978
11	1.1387734815937602	0.7812959653419438	0.00012679652038496088
12	1.138797174410268	0.7813149928137072	5.8375156980959674e-05

Results
(1.138797174410268, 0.7813149928137072)

Висновки

Виконуючи лабораторну роботу №7, я навчився програмувати розв'язки систем нелінійних рівнянь методами простої ітерації та Ньютона.