Лабораторна робота 3
Чисельні методи розв'язання нелінійних рівнянь з одним невідомим та систем нелінійних рівнянь

3.1. Чисельні методи розв'язання нелінійних рівнянь з одним невідомим

3.1.1. Мета роботи

Вивчення чисельних методів розв'язання нелінійних рівнянь, придбання навичок використання цих методів для розв'язання задач пошуку коренів нелінійних рівнянь із застосуванням комп'ютера.

3.1.2. Методичні рекомендації щодо організації самостійної роботи

За темою лабораторної роботи студент повинен: *знати* загальне формулювання задачі пошуку коренів нелінійного рівняння; *вміти* розв'язувати цю задачу з використанням чисельних методів [1 – 6].

Розглянемо рівняння виду:

$$f(x) = 0, \tag{3.1}$$

де $f(x)$ – нелінійна неперервна функція однієї змінної. Розв'язати рівняння – означає знайти таке x^*, при якому рівняння перетворюється в тотожність (x^* ще називають коренем рівняння). У загальному випадку рівняння може мати багато коренів. Розглянуті нижче чисельні методи розв'язання нелінійних рівнянь дозволяють знаходити один корінь на заданому відрізку $[a,b]$. При цьому на відрізку повинен існувати тільки один корінь.

Розглянемо кілька методів розв'язання нелінійного рівняння (3.1) на відрізку $[a,b]$.

Метод половинного ділення (метод дихотомії). При розв'язанні нелінійного рівняння методом половинного ділення задаються відрізок $[a,b]$, на якому існує тільки один розв'язок, і бажана точність $\varepsilon > 0$. На 0-й ітерації методу $[a_0, b_0] = [a,b]$, на k-й ітерації методу маємо поточний відрізок $[a_k, b_k]$. Далі визначається середина відрізка $x_k = \frac{a_k + b_k}{2}$ й перевіряється умова $f(a_k) \times f(x_k) < 0$. Якщо зазначена умова виконується, то $b_{k+1} = x_k$, $a_{k+1} = a_k$. Якщо умова не виконується, то $a_{k+1} = x_k$, $b_{k+1} = b_k$. Ділення відрізка навпіл припиняється при досягненні заданої точності, тобто $|b_k - a_k| \le \varepsilon$. Тут x_k є наближенням розв'язку на k-й ітерації методу. Очевидно, що

$$\left|x_{k+1} - x^*\right| \le \frac{1}{2}\left|x_k - x^*\right| \le \cdots \le \left(\frac{1}{2}\right)^k |b-a|,$$

тобто метод збігається з лінійною швидкістю з коефіцієнтом $q = 1/2$.

Метод хорд. При розв'язанні нелінійного рівняння методом хорд задаються відрізок $[a, b]$, на якому існує тільки один розв'язок, і точність ε. На 0-й ітерації методу $[a_0, b_0] = [a, b]$, на k-й ітерації методу маємо поточний відрізок $[a_k, b_k]$. Потім через дві точки з координатами $(a_k, f(a_k))$ й $(b_k, f(b_k))$ проводимо відрізок прямої лінії (хорду) і визначаємо точку перетину цієї лінії з віссю абсцис (точка x_k). Якщо при цьому $f(a_k) \times f(x_k) < 0$, то праву межу інтервалу переносимо в точку x_k (тобто $b_{k+1} = x_k$, $a_{k+1} = a_k$). Якщо зазначена умова не виконується, то в точку x_k переноситься ліва межа інтервалу ($a_{k+1} = x_k$, $b_{k+1} = b_k$). Пошук розв'язку припиняється при досягненні заданої точності, тобто $|f(x_k)| < \varepsilon$. Точка перетину хорди з віссю абсцис визначається за формулою:

$$x_{k+1} = a_k + \left|\frac{f(a_k)}{f(b_k) - f(a_k)}\right| \times (b_k - a_k). \tag{3.2}$$

Метод дотичних (метод Ньютона). При розв'язанні нелінійного рівняння методом дотичних задаються початкове наближення x_0 і точність ε. Потім у точці $(x_0, f(x_0))$ проводиться дотична до графіка $f(x)$ й визначається точка x_1 перетину дотичної з віссю абсцис. У точці $(x_1, f(x_1))$ знову будується дотична, обчислюється наступне наближення шуканого розв'язку x_2 і т. д. Зазначена процедура повторюється доти, поки $|f(x_i)| > \varepsilon$. Точка перетину $(k+1)$-ої дотичної з віссю абсцис визначається за формулою:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}. \qquad (3.3)$$

Умова збіжності методу дотичних: $f(x_0) \times f''(x_0) > 0$. При цьому швидкість збіжності буде квадратичною: $\left|x_{k+1} - x^*\right| \le M\left|x_k - x^*\right|^2$ для всіх $k > k_0$, де $k_0 > 0$, $M > 0$.

Метод простої ітерації. При розв'язанні нелінійного рівняння (3.1) методом ітерацій його потрібно записати у вигляді:

$$x = \phi(x). \qquad (3.4)$$

Задаються початкове наближення x_0 й точність ε. Перше наближення розв'язку x_1 знаходиться з виразу $x_1 = \phi(x_0)$, друге – $x_2 = \phi(x_1)$ і т. д. У загальному випадку $k+1$ наближення обчислюється за формулою $x_{k+1} = \phi(x_k)$. Зазначена процедура припиняється при досягненні заданої точності, тобто $\left|f(x_k)\right| = \left|x_k - \phi(x_k)\right| \le \varepsilon$. Умова збіжності методу ітерацій: $\left|\phi'(x)\right| \le q < 1$ для всіх $x \in [a, b]$. При цьому швидкість збіжності буде лінійною: $\left|x_k - x^*\right| \le Mq^k$ для всіх $k > 0$, де $M > 0$.

3.1.3. Контрольні приклади

Приклад 1. Розв'язати методом простої ітерації рівняння $x^3 - 5x = 0$ з точністю $\varepsilon = 10^{-4}$.

Розв'язання.

Спочатку визначимо відрізок, що містить розв'язок. Зробимо це графічно.

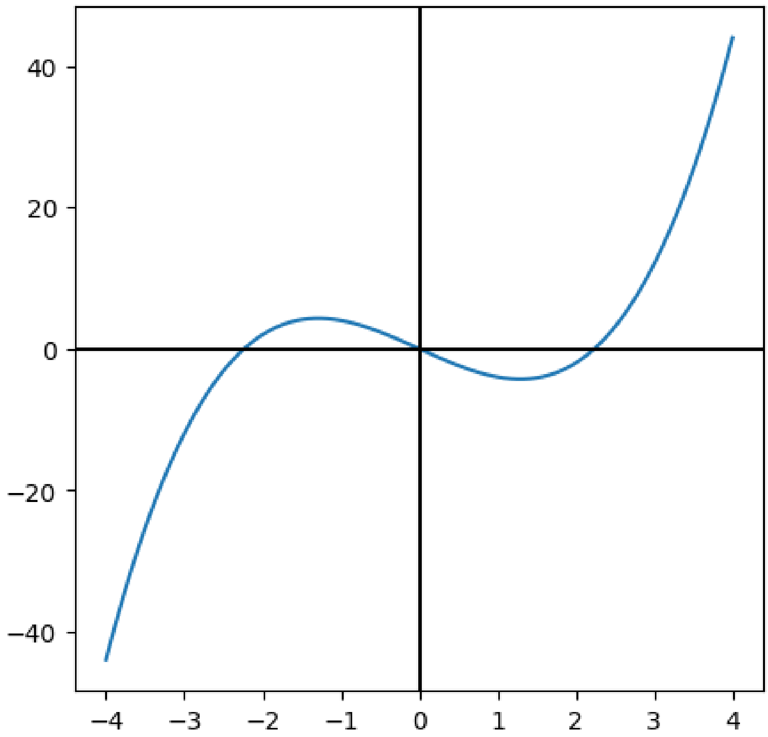
Вводимо функцію лівої частини рівняння:

```
def MyFun(x):
  y = x**3 - 5*x
  return y
```

Будуємо графік заданої функції:

```
import numpy as np
import matplotlib.pyplot as plt

a = -4
b = 4
```

```
    n = 100
    x = np.linspace(a, b, n)
    y = MyFun(x)

    plt.figure(figsize=(5,5))

    plt.plot(x, y)
    plt.axhline(0, color='k') #x-axis line
    plt.axvline(0, color='k') #y-axis line

    plt.show()
```



Із графіка видно, що коренів три. Знайдемо корінь на відрізку $[2,4]$. Спробуємо застосувати метод простої ітерації:

а) розглянемо рівняння $x=\frac{x^3}{5}$, тобто $\phi(x)=\frac{x^3}{5}$. Тоді $\phi'(x)=\frac{3x^2}{5}$.

Але $\left|\frac{3x^2}{5}\right|\geq\frac{12}{5}=2.4$ для всіх $x\in[2,4]$, тобто $|\phi'(x)|\geq 2.4$, тобто умова збіжності методу простої ітерації не виконується;

б) розглянемо рівняння $x=\sqrt[3]{5x}$, тобто $\phi(x)=\sqrt[3]{5x}$. Тоді $\phi'(x)=\frac{5\times(5x)^{-\frac{2}{3}}}{3}=\frac{5}{3(5x)^{\frac{2}{3}}}$, тому для всіх $x\in[2,4]$ $|\phi'(x)|\leq\frac{5}{3\times 10^{\frac{2}{3}}}<0.36$, тобто умова збіжності методу простої ітерації виконується $(q=0.36)$.

Візьмемо $x_0 = 4$, далі $x_1 = \sqrt[3]{5x_0}$, $x_2 = \sqrt[3]{5x_1}$ і т. д. до досягнення точності $|x_k - x_{k-1}| < \varepsilon$.

Вводимо функцію правої частини рівняння (3.4):

```
def g(x):
  y = (5*x)**(1/3)
  return(y)
```

Запрограмуємо в Python метод ітерацій для розв'язання нелінійних рівнянь. Процедура має такий вигляд:

```
def MetIter(x0, eps):
  while (True):
    x1 = g(x0)
    if np.absolute(x1-x0) <= eps:
      break
    x0 = x1
  return x1
```

Виклик процедури для розв'язання нелінійного рівняння:

```
x0 =4
eps = 1e-4
x = MetIter(x0, eps)
print(x)

2.2360900007100635
```

Перевірка розв'язку:

```
y = x - g(x)
print(y)
```

```
1.4682164283463806e-05
```

Тобто *y* майже дорівнює нулюі, відповідно до заданої точності eps = 0.0001.

Приклад 2. Траєкторія польоту снаряда в безповітряному просторі під дією тільки однієї сили тяжіння описується рівнянням (параболою) $y = xtg\,\theta_0 - \dfrac{gx^2}{2V_0^2 \cos^2 \theta_0}$, де V_0 – початкова швидкість у момент пострілу, θ_0 – кут відносно горизонту в момент пострілу (рад), *x, y* – відповідно абсциса і ордината в площині польоту снаряда, *g* – прискорення вільного падіння (9.81м/сек^2). Необхідно знайти, під яким кутом був зроблений постріл, якщо снаряд пролетів 3 600 м, а початкова швидкість снаряда дорівнювала 800 км/год.

Розв'язання.

Місце падіння снаряду має координати $(x_F, y_F) = (3600;0)$ тому для знаходження кута θ_0, під яким був зроблений постріл треба розв'язати нелінійне рівняння $y_F = x_F tg\theta_0 - \dfrac{gx_F^2}{2V_0^2 \cos^2 \theta_0}$. Перепишемо його у вигляді (3.1), тобто $f(\theta_0) = x_F tg\theta_0 - \dfrac{gx_F^2}{2V_0^2 \cos^2 \theta_0} - y_F = 0$, і розв'яжемо за допомогою процедури fsolve пакета scipy.optimize

```
import math

V0 = 800000/3600 # швидкість в м/сек
xF = 3600
yF = 0
g = 9.81

def Fun(Teta):
  y = xF*math.tan(Teta) - g*xF*xF/(2*V0*V0*math.cos(Teta)**2) - yF
  return y

a = 0
b = math.pi/3
print(Fun(a))
print(Fun(b))
```

```
-1287.2682
1086.3101072479594
```

Тут розв’язок рівняння шукається на відрізку [a, b] = [0, π/3], який задовольняє умові наявністі на ньому кореня.

```
from scipy.optimize import fsolve

sol = fsolve(Fun, (a+b)/2)
print(sol)
```

```
[0.39841863]
```

Відповідь. Постріл було зроблено під кутом 0.39841863 радіан або близько 23 градусів.

3.1.4. Порядок виконання роботи і варіанти завдань

3.1.4.1. Зміст звіту

У теоретичній частині роботи необхідно коротко описати:

- постановку задачі розв'язання нелінійного рівняння з однією змінною;
- чисельні методи розв'язання нелінійного рівняння з однією змінною.

У практичній частині роботи необхідно:

- запрограмувати у вигляді окремих модулів методи половинного ділення, хорд, дотичних і простої ітерації;
- привести тексти складених програм;
- розв'язати задачу з індивідуального завдання з застосуванням складених програм та засобів математичних пакетів Python;
- порівняти трудомісткість і швидкість збіжності методів.

3.1.4.2. Варіанти індивідуальних завдань

Варіант	**Рівняння**
1	$x^2 = \exp(-x^2) - 1$
2	$x = \cos(x)$
3	$x = x^2 - 1$
4	$x = 2\exp(-x)$
5	$x = 3\cos(x)$
6	$x = 2\exp(-x)$
7	$x = tg(2x) - 1$
8	$x = \ln(x) + 2$
9	$x = \cos(2x)$
10	$x = \exp(-3x)$
11	$x = \exp(-3x) + 1$
12	$x = \exp(-x^2)$
13	$x = \exp(-3x^2)$
14	$x^2 = \exp(-x^2)$
15	$x = tg(x) - 2$

3.1.5. Контрольні запитання

1. Сформулюйте постановку задачі розв'язання рівнянь з одним невідомим.

2. У чому полягає суть методу дихотомії? Яка у нього швидкість збіжності?

3. У чому полягає суть методу хорд? Яка у нього швидкість збіжності?

4. У чому полягає суть методу Ньютона? Яка у нього швидкість збіжності?

5. У чому полягає суть методу простої ітерації? Яка у нього швидкість збіжності?

3.2. Чисельні методи розв'язання систем нелінійних рівнянь

3.2.1. Мета роботи

Вивчення чисельних методів розв'язання систем нелінійних рівнянь, придбання навичок використання цих методів для пошуку розв'язків систем нелінійних рівнянь із застосуванням комп'ютера.

3.2.2. Методичні рекомендації щодо організації самостійної роботи

За темою лабораторної роботи студент повинен: *знати* загальне формулювання задачі розв'язання системи нелінійних рівнянь; *вміти* розв'язувати цю задачу з використанням чисельних методів [1 – 6].

Розглянемо систему нелінійних рівнянь виду:

$$\begin{cases} f_1(x_1, x_2, ..., x_n) = 0 \\ f_2(x_1, x_2, ..., x_n) = 0 \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ f_n(x_1, x_2, ..., x_n) = 0 \end{cases}, \qquad (4.1)$$

де $f_i(x_1, x_2, ..., x_n)$, $i = \overline{1, n}$ – деякі нелінійні функції n змінних. Якщо ввести позначення $x = (x_i)_{i=1}^{n}$ – вектор-стовпець розмірності n з елементами x_i, $F(x) = \begin{pmatrix} f_1(x) \\ \dots \\ f_n(x) \end{pmatrix}$ – векторна функція розмірності n, елементами якої є функції $f_i(x) = f_i(x_1, x_2, ..., x_n)$, $i = \overline{1,n}$, то систему (4.1) можна записати у векторному вигляді:

$$F(x) = 0, \tag{4.2}$$

Розв'язати систему (4.2) – означає знайти таке $x^* \in R^n$, для якого $F(x^*) \equiv 0$, тобто $f_i(x_1^*, x_2^*, ..., x_n^*) \equiv 0 \;\; \forall i = \overline{1,n}$.

Розглянемо три основні методи розв'язання систем нелінійних рівнянь виду (4.2).

Метод Ньютона. Метод будує ітераційну послідовність $\{x^k\}$, $k = 0, 1, 2, \ldots$ $(x^k \in R^n)$ наближень розв'язку x^*(початкове наближення x^0 задається) за такою ітераційною формулою:

$$x^{k+1} = x^k - \left[F'(x^k)\right]^{-1} F(x^k). \tag{4.3}$$

Тут $F'(x) = \begin{pmatrix} \dfrac{\partial f_1(x)}{\partial x_1} & \ldots & \dfrac{\partial f_1(x)}{\partial x_n} \\ \ldots & \ldots & \ldots \\ \dfrac{\partial f_n(x)}{\partial x_1} & \ldots & \dfrac{\partial f_n(x)}{\partial x_n} \end{pmatrix}$, тобто $F'(x^k)$ – матриця розмірності $n \times n$. Ітераційний процес (4.3) триває доти, поки не виконається умова $\left\| F(x^k) \right\| \le \varepsilon$, де ε – задана точність розв'язання задачі (4.2).

Якщо послідовність $\{x^k\}$, $k = 0, 1, 2, \ldots$ збігається, то швидкість її збіжності квадратична, тобто $\left\| x^{k+1} - x^* \right\| \le M \left\| x^k - x^* \right\|^2$ починаючи з якогось k. Тут M – деяка додатна константа [1; 2; 6].

Основним недоліком методу Ньютона є те, що він збігається тільки для достатньо близьких до розв'язку початкових наближень x^0.

Метод ітерацій. При розв'язанні системи нелінійних рівнянь (4.2) методом ітерацій її потрібно записати у вигляді $x = \Phi(x)$. Тут $\Phi(x)$ – векторна функція розмірності n від $x \in R^n$. Задаються початкове наближення x^0 й точність ε. Метод будує ітераційну послідовність $\{x^k\}$, $k = 0, 1, 2, \ldots$, наближень розв'язку x^* за такою ітераційною формулою:

$$x^{k+1} = \Phi(x^k). \tag{4.4}$$

Метод ітерацій збігається, якщо $\|\Phi'(x)\| \le q < 1$ для всіх x приналежних деякому околі $V(x^*)$ розв'язку x^* й $x^0 \in V(x^*)$. При цьому швидкість збіжності буде лінійною, тобто $\|x^{k+1} - x^*\| \le q\|x^k - x^*\|$ починаючи з якогось k.

Метод найменших квадратів. Для розв'язання системи нелінійних рівнянь (4.2) методом найменших квадратів треба ввести функцію виду

$$\Phi(x) \equiv \|F(x)\|^2 = \sum_{i=1}^{n} f_i^2(x). \tag{4.5}$$

Тоді розв'язком задачі (4.2) буде точка, в якій функція $\Phi(x)$ досягає мінімального значення.

Для пошуку точки мінімума функції $\Phi(x)$ можна застосувати функцію minimize(fn, par) пакету SciPy, де **fn** – імя цільвої фукції, **par** – початкове наближення точки мінімума. Для її виклику необхідно визначити функцію $\varphi(z)$ й задати початкове наближення $z^0 \in R^k$, тоді розв'язком є z* = minimize (φ, z^0).

3.2.3. Контрольний приклад

Приклад 1. Розв'язати методом Ньютона систему нелінійних рівнянь $\begin{cases} 2x_1^2 + x_2^2 - 1 = 0 \\ x_1^3 + 6x_1^2 x_2 - 1 = 0 \end{cases}$ з точністю $\varepsilon = 10^{-4}$ (початкове наближення $x_0 = (0.65, 0.35)^T$).

Розв'язання.

Для розв'язання поставленої задачі реалізуємо метод Ньютона в в Python:

```
import numpy as np
from numpy import linalg as LA

def Newton(Fx, FFx, x, eps):
  y = Fx(x)
  while (LA.norm(y) > eps):
    yy = FFx(x)
    x = x - np.dot(np.linalg.inv(yy), y)
    y = Fx(x)
  return x
```

Тут Fx – функція, що обчислює ліву частину системи (4.2), а FFx – функція, що обчислює похідну по x від лівої частини системи (4.2).

Введення початкових даних:

```
x0 = [0.65, 0.35]

def Fx(x):
  f = [0, 0]
  f[0] = 2*(x[0]**2 + x[1]**2) - 1
  f[1] = x[0]**3 + (6*x[0]**2)*x[1] - 1
  return f

def FFx(x):
  ff = np.array([[0, 0],
                 [0, 0]])
  ff[0,0] = 4*x[0]
  ff[0,1] = 2*x[1]
  ff[1,0] = 3*x[0]**2 + 12*x[0]*x[1]
  ff[1,1] = 6*x[0]**2
  return ff
```

Виклик запрограмованої процедури для отримання розв'язку та перевірка правильності розв'язку:

```
eps = 1e-5
x = Newton(Fx, FFx, x0, eps)
print(x)
```

```
[0.64061012 0.29935769]
```

```
Fx(x)
```

```
[-7.304363138782577e-06, -7.474274206220599e-07]
```

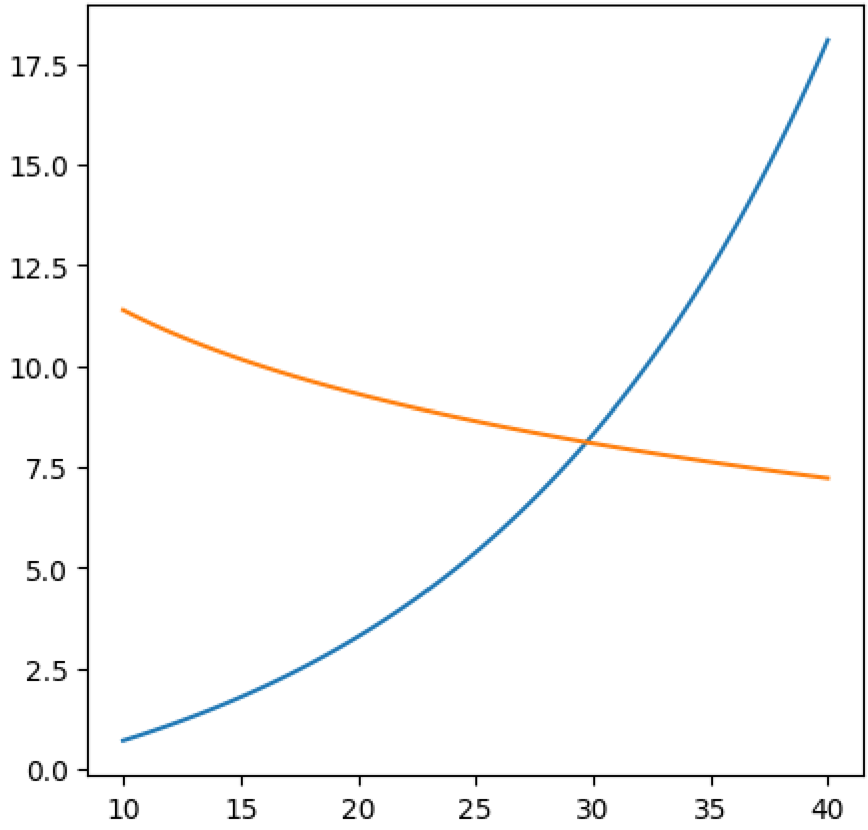
Приклад 2. Пропозиція на деякий товар описується функцією $S = \exp\left(\frac{P+5}{15}\right) - 2$, а попит на цей товар описується функцією $D = -3\ln\left(\frac{P}{3}\right) + 15$, де S – величина пропозиції товару, D – величина попиту товару, P – ціна товару. Знайдіть рівноважну ціну та кількість товару, при якій встановиться рівноважна ціна. (Рівноважна ціна – це ціна при якій попит дорівнює пропозиції).

Позначимо через x_1 ціну, а через x_2 кількість товару. Тоді систему перепишемо так:

$$\begin{cases} \exp\left(\dfrac{x_1+5}{15}\right) - 2 - x_2 = 0, \\ -3\ln\left(\dfrac{x_1}{3}\right) + 15 - x_2 = 0 \end{cases}.$$

Для вибору початкового наближення побудуємо графік:

```
import math

a = 10
b = 40
n = 100
x = np.linspace(a, b, n)
y1 = [0]*n
y2 = [0]*n
for i in range(n):
  y1[i] = math.exp((x[i]+5)/15) - 2
  y2[i] = -3*math.log(x[i]/3) + 15

plt.figure(figsize=(5,5))

plt.plot(x, y1)
plt.plot(x, y2)
```



Задамо початкове наближення таким: $x_0 = (28, 10)^T$.

Введемо початкові дані задачі:

```
# Вводимо вектор початкового наближення
x0 = [28, 10]
```

```python
# Вводимо векторну функцію F лівої частини системи
def Fx(x):
  f = [0, 0]
  f[0] = math.exp((x[0]+5)/15) - 2 - x[1]
  f[1] = -3*math.log(x[0]/3) + 15 - x[1]
  return f

# Вводимо матричну функцію частинних похідних від F
def FFx(x):
  ff = np.array([[0, 0],
                 [0, 0]])
  ff[0,0] = math.exp((x[0]+5)/15)*15
  ff[0,1] = -1
  ff[1,0] = -9/x[0]
  ff[1,1] = -1
  return ff
```

Скористаємося записаною в прикладі 1 процедурою і перевіримо отриманий результат:

```python
eps = 1e-5
x = Newton(Fx, FFx, x0, eps)
print(x)
print(Fx(x))
```

```
[29.71847585  8.12053007]
[-9.961652976286928e-06, -6.689505838153309e-09]
```

Таким чином, при ціні товару 8,12 грошових одиниць попит і пропозиція будуть майже однакові і становитимуть 29 або 30 одиниць товару.

3.2.4. Порядок виконання роботи і варіанти завдань

3.2.4.1. Зміст звіту

У теоретичній частині роботи необхідно коротко описати:

- постановку задачі розв'язання системи нелінійних рівнянь;
- чисельні методи розв'язання системи нелінійних рівнянь.

У практичній частині роботи необхідно:

- запрограмувати у вигляді окремих модулів методи Ньютона, ітерацій та найменших квадратів;
- привести тексти складених програм;
- розв'язати задачу з індивідуального завдання з застосуванням складених програм та засобів математичних пакетів Python
- порівняти трудомісткість і швидкість збіжності методів.

3.2.4.2. Варіанти індивідуальних завдань

Варіант	Система рівнянь	Початкове наближення
1	2	3
1	$\begin{cases} \ln\left(1+\dfrac{x_1+x_2}{5}\right)-\sin\left(\dfrac{x_2}{3}\right)-x_1+1.1=0, \\ \cos\left(\dfrac{x_1x_2}{6}\right)-x_2+0.5=0. \end{cases}$	$x^0=(1;1)$
2	$\begin{cases} \lg\left(\dfrac{x_2}{x_3}\right)-x_1+1=0, \\ 2x_1^2+x_2-x_3-0.4=0, \\ \dfrac{x_1x_2}{20}-x_3+2=0. \end{cases}$	$x^0=(1;2.2;2)$
3	$\begin{cases} x_1+x_1^2-2x_2x_3-0.1=0, \\ x_2-x_2^2+3x_1x_3+0.2=0, \\ x_3+x_3^2+2x_1x_2-0.3=0. \end{cases}$	$x^0=(0;0;0)$
4	$\begin{cases} x_1^3+x_2^3-6x_1+3=0, \\ x_1^3-x_2^3-6x_2+2=0. \end{cases}$	$x^0=(0.5;0.5)$
5	$\begin{cases} x_1+3\lg x_1-x_2^2=0, \\ 2x_1^2-x_1x_2-5x_1+1=0. \end{cases}$	$x^0=(3.4;2.2)$
6	$\begin{cases} 5x_1-6x_2+20\lg x_1+16=0, \\ 2x_1+x_2-10\lg x_2-4=0. \end{cases}$	$x^0=(0;0)$

1	2	3
7	$\begin{cases} 0.5\sin\left(\dfrac{x_2}{3}\right)-x_1+1=0, \\ 0.3\cos x_1-x_2=0. \end{cases}$	$x^0=(1;0)$
8	$\begin{cases} x_1^2-x_2^2-1=0, \\ x_2(x_2-1)-1=0. \end{cases}$	$x^0=(2;10)$
9	$\begin{cases} 2x_1-x_2-6\lg x_1-3=0, \\ 15x_1-10x_2-60\lg x_2-6=0. \end{cases}$	$x^0=(0;0)$

10	$\begin{cases} x_1^2 - x_2^2 = 0, \\ x_1 x_2 - 4 = 0. \end{cases}$	$x^0 = (0; 0)$
11	$\begin{cases} x_1 - x_2 - 6\lg x_1 - 1 = 0, \\ x_1 - 3x_2 - 6\lg x_2 - 2 = 0. \end{cases}$	$x^0 = (0.5; 0.2)$
12	$\begin{cases} 2x_1^3 - x_2^2 - 1 = 0, \\ x_1 x_2^3 - x_2 - 4 = 0. \end{cases}$	$x^0 = (0; 0)$
13	$\begin{cases} x_1 + x_2 + x_3 - 13 = 0, \\ x_1^2 + x_2^2 + x_3^2 - 91 = 0, \\ x_2^2 - x_1 x_3 = 0. \end{cases}$	$x^0 = (0; 0)$
14	$\begin{cases} \sin x_1 + 2x_2 - 2 = 0, \\ \cos x_1 + x_2 - 1.5 = 0. \end{cases}$	$x^0 = (0; 0)$
15	$\begin{cases} \cos(x_1 + 0.5) - x_2 - 2 = 0, \\ \sin x_2 - 2x_1 - 1 = 0. \end{cases}$	$x^0 = (0; 0)$

3.2.5. Контрольні запитання

1. Сформулюйте постановку задачі розв'язання системи нелінійних рівнянь.
2. У чому полягає суть методу Ньютона? Яка у нього швидкість збіжності?
3. У чому полягає суть методу ітерацій? Яка у нього швидкість збіжності?
4. У чому полягає суть методу найменших квадратів?