# КОМП’ЮТЕРНИЙ ПРАКТИКУМ № 1. РОЗВ’ЯЗАННЯ НЕЛІНІЙНИХ РІВНЯНЬ

Роботу виконав:

Грунда Ярослав

Студент 3-го курсу

Групи Фі-21

## Вихідні дані

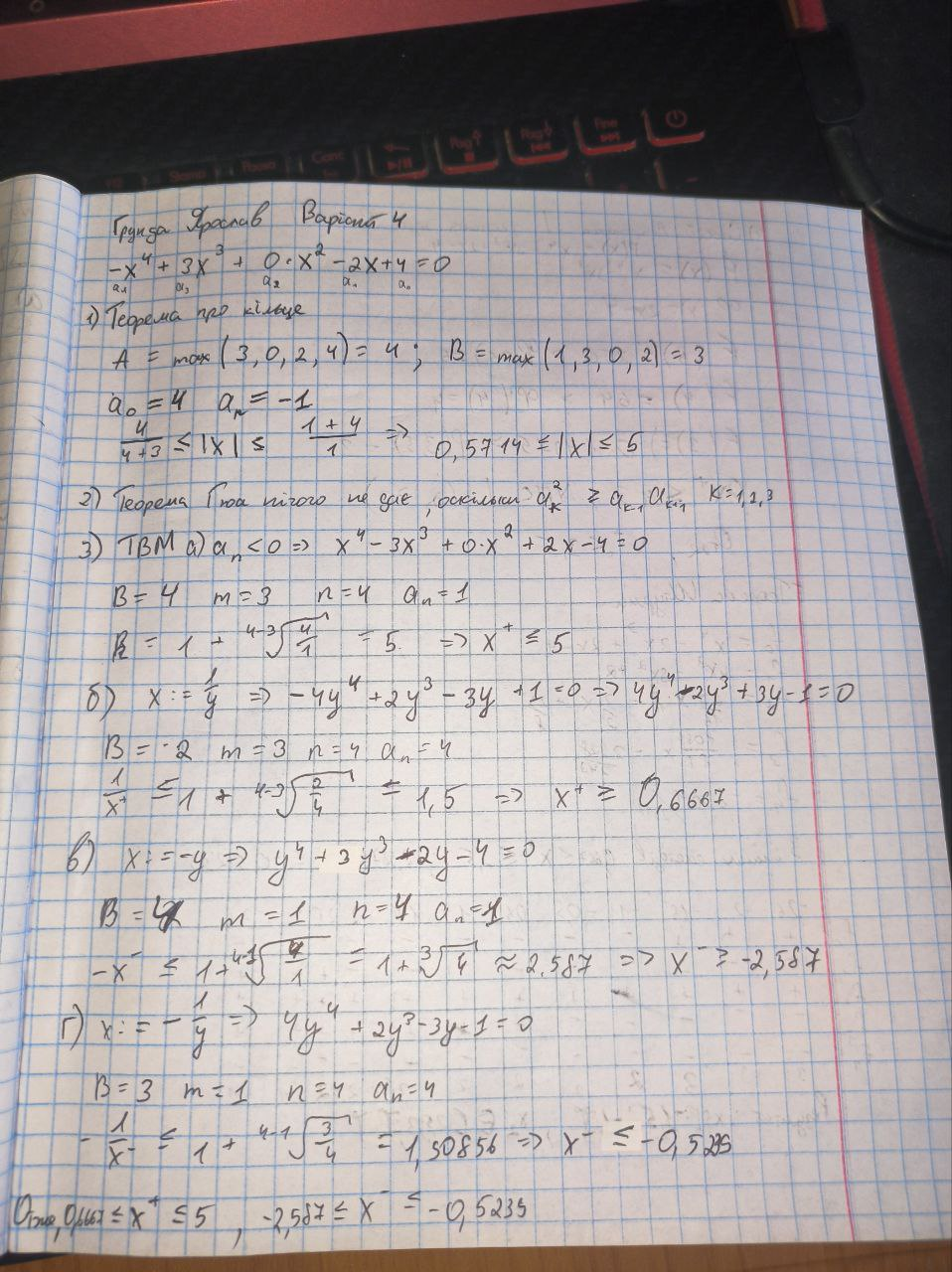
Поліном:

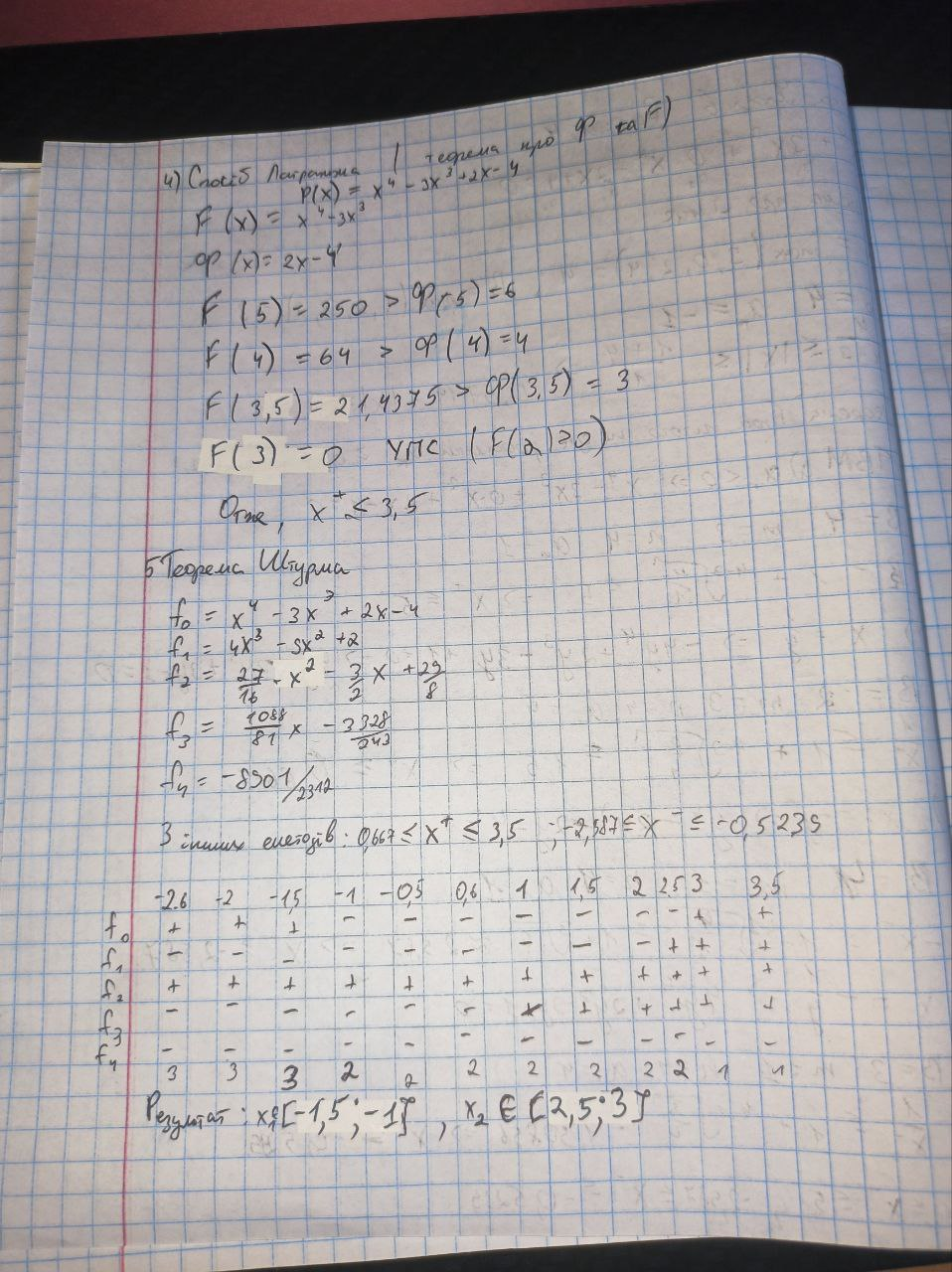
f(x)=-x^4+3x^3-2x+4

Обрані проміжки (за теоремою Штурма):

[a1, b1] = [2.5, 3]  
[a2, b2] = [-1.5, -1]

## Допрограмний етап





## **Лістинг програми уточнення коренів**

Програма реалізована на мові Python. Вона реалізує три методи: бісекції, хорд і дотичних (Ньютона). На кожній ітерації виводяться: номер ітерації, наближене значення кореня, похибка (критерій завершення).

x = sp.Symbol('x')

# Заданий поліном

f\_expr = x\*\*4 - 3\*x\*\*3 + 2\*x - 4

f = sp.lambdify(x, f\_expr, modules='numpy')

f\_prime\_expr = sp.diff(f\_expr, x)

f\_prime = sp.lambdify(x, f\_prime\_expr, modules='numpy')

def bisection\_method(f, a, b, eps=1e-6):

steps = []

iteration = 1

while abs(b - a) > eps:

c = (a + b) / 2

error = abs(b - a)

steps.append((iteration, c, error))

if f(a) \* f(c) <= 0:

b = c

else:

a = c

iteration += 1

c = (a + b) / 2

error = abs(b - a)

steps.append((iteration, c, error))

return c, steps

def secant\_method(f, a, b, eps=1e-6):

steps = []

c\_prev = None

iteration = 1

while True:

c = (a \* f(b) - b \* f(a)) / (f(b) - f(a))

error = abs(c - c\_prev) if c\_prev is not None else None

steps.append((iteration, c, error))

if f(a) \* f(c) <= 0:

b = c

else:

a = c

if c\_prev is not None and error < eps:

break

c\_prev = c

iteration += 1

return c, steps

def newton\_method(f, f\_prime, x0, eps=1e-6, max\_iter=100):

steps = []

iteration = 1

for \_ in range(max\_iter):

fx = f(x0)

fpx = f\_prime(x0)

if fpx == 0:

break

x1 = x0 - fx / fpx

error = abs(x1 - x0)

steps.append((iteration, x1, error))

if error < eps or abs(fx) < eps:

break

x0 = x1

iteration += 1

return x0, steps

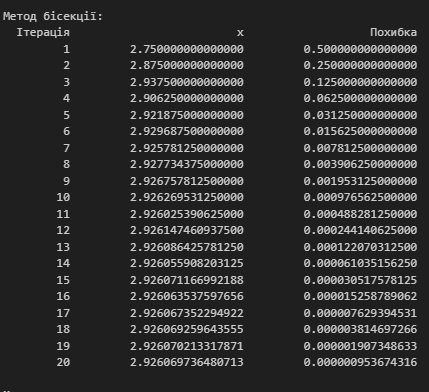
a, b = 2.5, 3

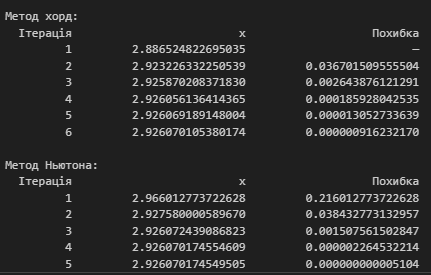
x0 = (a+b)/2

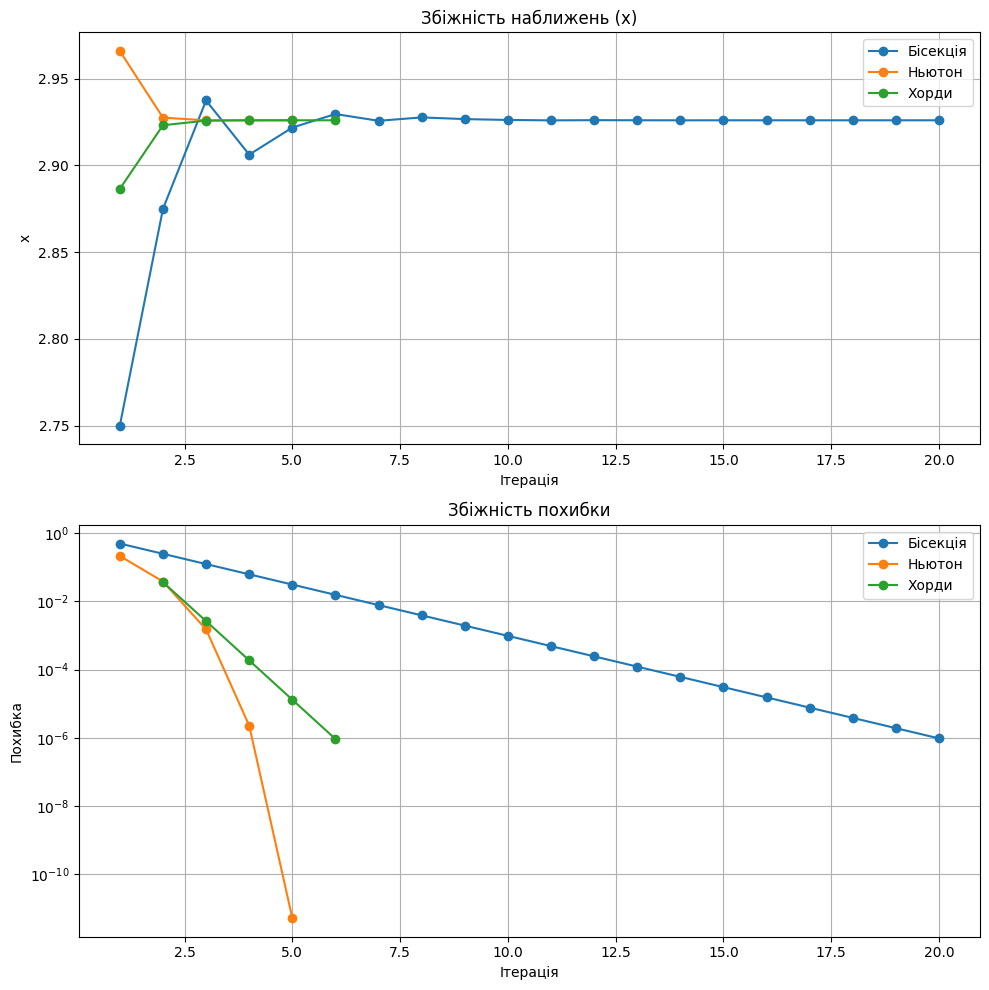
eps = 1e-6

## Результати

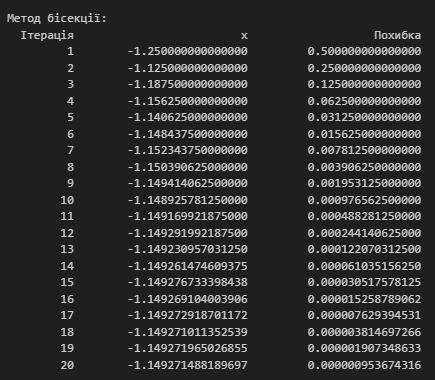
### Перший корінь

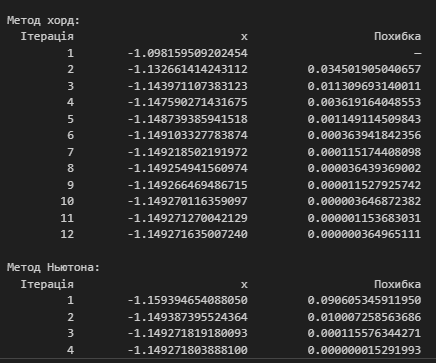






### Другий корінь





## Висновки

**Метод Ньютона** показав **найшвидше збіження** — лише 5 та 4 ітерацій.

Це зумовлено тим, що метод використовує **інформацію про похідну**, а отже, враховує кривизну графіка функції.

**Метод хорд** також є ефективним — 6 та 12 ітерацій.  
Його збіжність **слабша, ніж у Ньютона**, але краща за бісекцію. Він не потребує обчислення похідної, але використовує наближений напрямок січної.

**Метод бісекції** хоч і найпростіший , але потребував **найбільше ітерацій (20)**.