**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

Інститут **КНІТ**
Кафедра **ПЗ**

# ЗВІТ

До лабораторної роботи № 2
**На тему**: "*Розв'язування нелінійних рівнянь методом дотичних та методом послідовних наближень*"
**З дисципліни**: "Чисельні методи"

**Лектор**:
доцент кафедри ПЗ
Мельник Н.Б.

**Виконав**:
студент групи ПЗ-16
Коваленко Д.М.

**Прийняла**:
асистент кафедри ПЗ
Гарматій Г.Ю.

«_____» _____ 2022 р.
$\sum$ = _____.....

**Львів — 2022**

**Тема.** Розв'язування нелінійних рівнянь методом дотичних та методом послідовних наближень.

**Мета.** Ознайомлення на практиці з методом дотичних та методом послідовних наближень для розв'язування нелінійних рівнянь.

## Теоретичні відомості

### Метод Ньютона (метод дотичних)

Запишемо рівняння дотичної до кривої $y = f(x)$ в точці $(x_i; f(x_i))$, де $f(x)$ є неперервною монотонною нелінійною функцією, яка на кінцях відрізку $[a, b]$ приймає значення різних знаків, причому її похідні $f'(x)$ та $f''(x)$ є неперервними та монотонними.

$$y - f(x_i) = f'(x_i)(x - x_i)$$

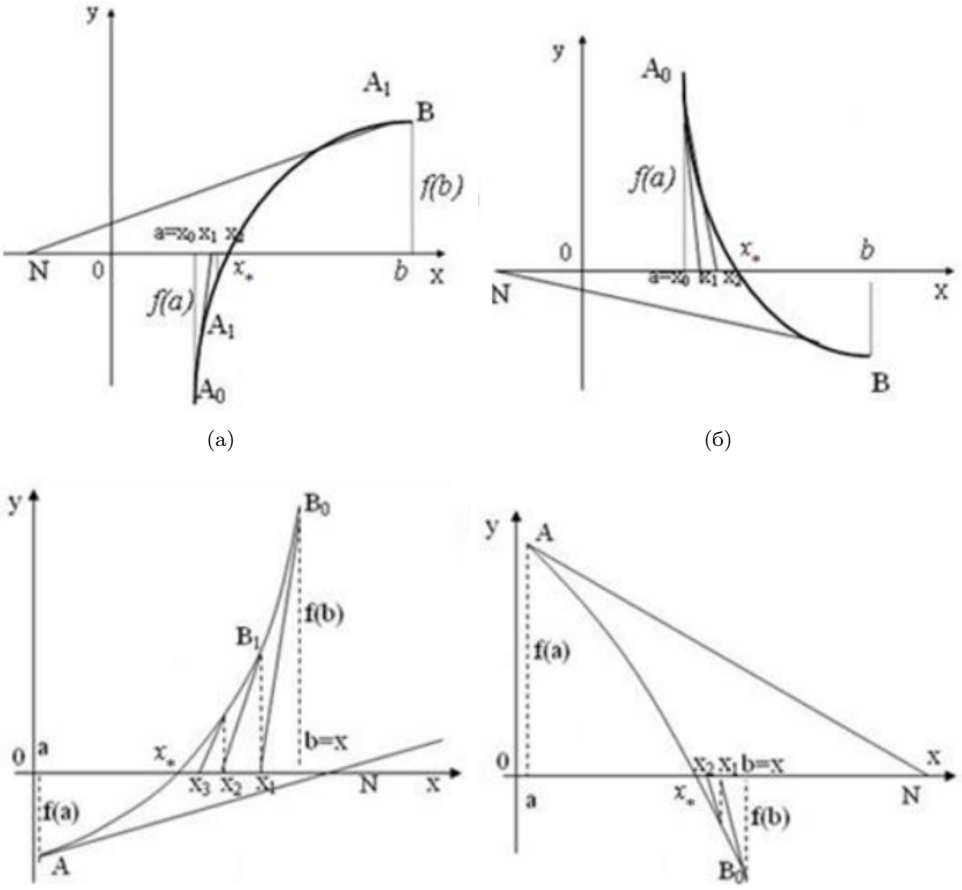


Рис. 1: Геометричний зміст методу Ньютона:
а) графік функції $y = f(x)$ є вгнутим $(f'(x) > 0; f''(x) > 0)$;
б)графік функції $y = f(x)$ є опуклим $(f'(x) < 0; f''(x) < 0)$;
в)графік функції $y = f(x)$ є опуклим $(f'(x) > 0; f''(x) < 0)$;
г)графік функції $y = f(x)$ є вгнутим $(f'(x) < 0; f''(x) > 0)$.

Геометричний зміст (рис. 1) методу Ньютона полягає в тому, що дугу кривої $y = f(x)$ на відрізку $[a, b]$ замінюють дотичною до цієї кривої, а наближене значення кореня визначають як абсцису точки перетину дотичної з віссю $Ox$, проведеної через один із кінців відрізка.

Тоді ітераційні формули запишемо у вигляді

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}, \qquad i = 1, 2, 3...$$

Для вибору початкового наближення кореня рівняння $f(x) = 0$ необхідно керуватися таким правилом: за початкову точку слід вибирати той кінець відрізка $[a, b]$ , в якому знак функції $y = f(x)$ співпадає зі знаком її другої похідної $y = f''(x)$.

### Метод простої ітерації (метод послідовних наближень)

Запишемо у рівняння $f(x) = 0$ канонічній формі

$$x = \phi(x)$$

Довільним способом визначимо найближче значення $x_0$ кореня рівняння і підставимо його у праву частину співвідношення. У результаті одержимо

$$x_1 = \phi(x_0)$$

Підставимо у праву частину рівняння замість $x_0$ значення $x_1$, тоді $x_2$. Повторюючи цей процес, отримаємо ітераційні формули

$$x_i = \phi(x_{i-1}), \qquad i = 1, 2, 3...$$

Кожний дійсний корінь $x_*$ рівняння є абсцисою точки перетину кривої $y = \phi(x)$ з прямою $y = x$. Доведено що ітераційний процес, визначений формулами, збігається до єдиного кореня рівняння $f(x) = 0$, якщо на відрізку $[a, b]$, який містить цей корінь, виконано умову:

$$|\phi'(x)| \leq q = \max_{x \in [a,b]} |\phi'(x)| < 1$$

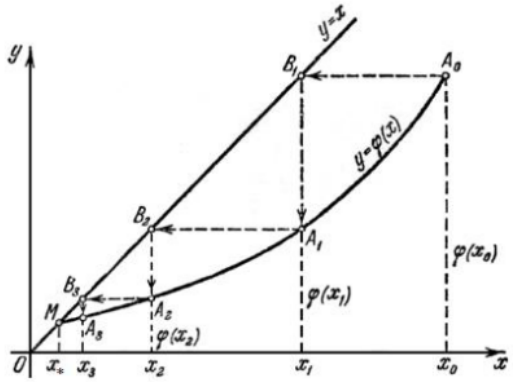


Рис. 2: Геометричний зміст методу ітерацій

## Лабораторне завдання

1. Ознайомитись з теоретичними відомостями.
2. Скласти програму розв'язування нелінійного рівняння $x^3 - 6x^2 - 7 = 0$ методом дотичних та методом простої ітерації.

## Хід роботи

### Графічний метод

За графіком функції (рис. 3) можна зробити висновок, що корінь рівняння належить проміжку $[6, 7]$.

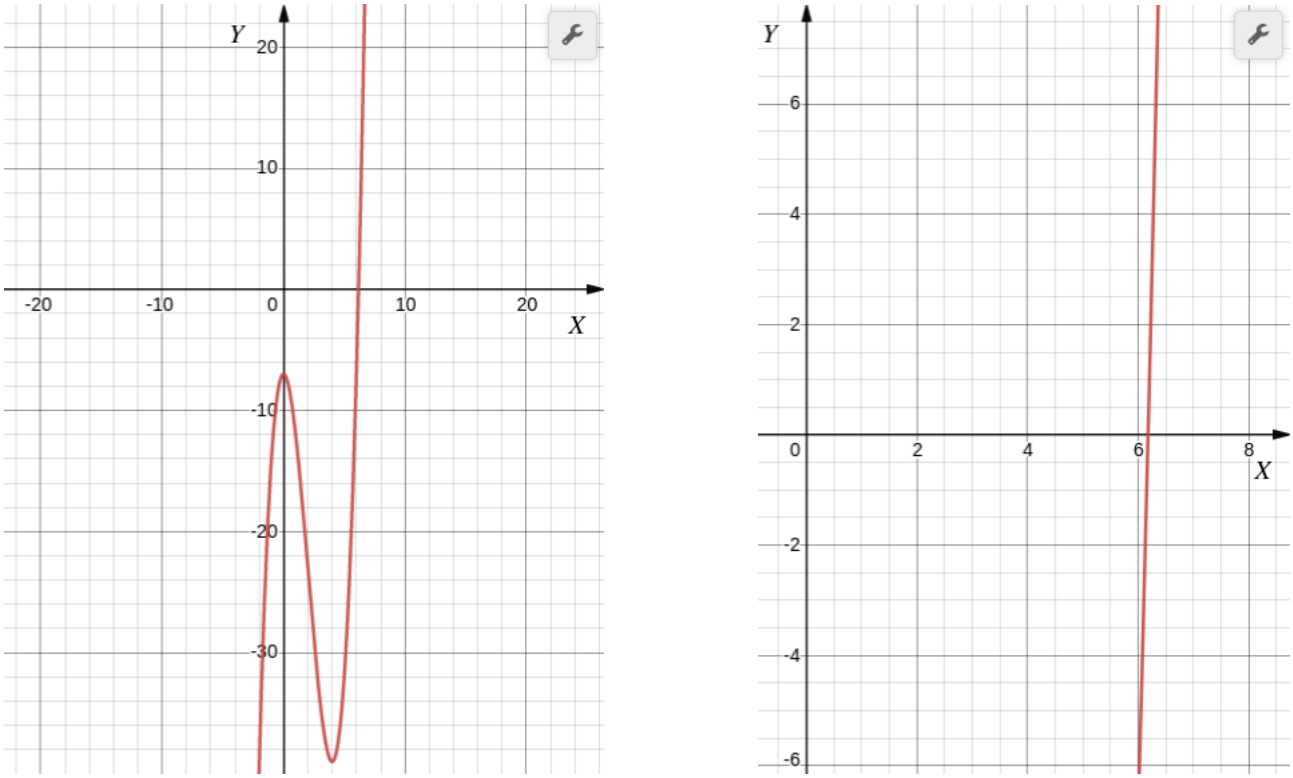


Рис. 3: Графік функції $y = x^3 - 6x^2 - 7$

**Аналітичний метод**

Для аналітичного розв'язку визначимо монотонність функції f(x), для цього розв'яжемо рівняння $f'(x) = 0$ та знайдемо інтервали монотонності.

$3x^2 - 12x = 0$

Інтервалами монотонності є $(-\infty; 0)$, $(0; 4)$, $(4; +\infty)$. Виберемо лише ті інтервали, де функція змінює знак $f(x) = f(x^3 - 6x^2 - 7) = +\infty$, $f(x) = f(x^3 - 6x^2 - 7) = -\infty$

$f(4) < 0$, $f(+\infty) > 0$, Отже, $(4; +\infty)$ єдиний відрізок, де є корінь, бо на всіх інших знак не змінюється і функція є монотонною. Це значить, що корінь є і він лежить в цьому інтервалі. Знайдемо відрізок, де є корінь рівняння, для цього перевіримо знак функції у цілочисельних точках інтервалу.

$f(6) = -7 < 0$ та $f(7) = 42 > 0$, отже корінь належить відрізку $[6; 7]$.

**Метод дотичних**

***Код програми*** *(файл lab_21.py):*

```
class TangentMethod:
    def __init__(self, a, b, eps, f, first_der, second_der):
        self.a = a # Ліва межа
        self.b = b # Права межа
        self.x = 1 # Шукане значення
        self.i = 0 # Кількість ітерацій
        self.eps = eps # Точність
        self.f = f # Функція
        self.first_der = first_der # Перша похідна функції
        self.second_der = second_der # Друга похідна функції
        if a >= b:
            return print("Помилка, 'a' більше або рівне 'b'")
        elif eps > b - a:
```

```
            return print("Помилка, 'eps' більше ніж інтервал між 'a'
    та 'b'")
        elif self.calc_f(f, a) * self.calc_f(self.second_der, a)>0:
            x = a
        else:
            x = b

    def calculate(self):
        print(f"x: {self.x} f(x): {self.calc_f(self.f, self.x)} f'(x
    ): {self.calc_f(self.first_der, self.x)}")
        if abs(self.calc_f(self.f, self.x)) < self.eps:
            return f"Відповідь: {self.x} \n Кількість ітерацій: {self.i}"
        else:
            self.i += 1
            self.x -= self.calc_f(self.f, self.x) / self.calc_f(self
    .first_der, self.x)
            return self.calculate()

    def calc_f(self, f, x):
        return eval(f.replace("x", str(x)))
a = float(input("Ліва межа: "))
b = float(input("Права межа: "))
eps = float(input("Точність: "))
f = input("Функція: ") or "x*x*x - 6*x*x - 7"
first_der = input("Перша похідна функції: ") or "3*x*x - 12*x"
second_der = input("Друга похідна функції: ") or "6*x - 12"
print(TangentMethod(a,b,eps, f, first_der, second_der).calculate())
```

```
Ліва межа: 6
Права межа: 7
Точність: 0.001
Функція: x**3-6*x**2-7
Перша похідна функції: 3**2-12*x
Друга похідна функції: 6*x-12
x: 1 f(x): -12 f'(x): -9
x: -0.3333333333333326 f(x): -7.703703703703703 f'(x): 4.333333333333332
x: 1.4444444444444449 f(x): -16.504801097393695 f'(x): -11.074074074074076
x: -0.04595565465130669 f(x): -7.012768587933599 f'(x): 0.5578036223989707
x: 12.526154504428469 f(x): 1016.983314007985 f'(x): 320.3997859532992
x: 9.352047159187551 f(x): 286.1726794784364 f'(x): 150.15779229275321
x: 7.446234113398336 f(x): 73.18847871955074 f'(x): 76.98439805383126
x: 6.495541757937249 f(x): 13.907928935780149 f'(x): 48.6296870920726
x: 6.209545080445829 f(x): 1.0797335293468961 f'(x): 41.16080935291704
x: 6.183313003226716 f(x): 0.008671989297653226 f'(x): 40.50032304889717
x: 6.183098881742125 f(x): 5.753799428021011e-07 f'(x): 40.49494876329665
Відповідь: 6.183098881742125
Кількість ітерацій: 10
```

Рис. 4: Робота програми

**Метод простої ітерації**

Для методу простої ітерації виконаємо деякі перетворення рівняння.

$$x^3 - 6x^2 - 7 = 0;$$
$$x^3 = 6x^2 + 7;$$
$$x = (6x^2 + 7)^{1/3};$$
$$\phi(x) = (6x^2 + 7)^{1/3};$$
$$\phi'(x) = \frac{4x}{(6x^2 + 7)^{2/3}}.$$

$|\phi'(x)| = |\frac{4x}{(6x^2+7)^{2/3}}| \leq 0.65$ для всіх $x \in [6, 7]$. Тому ітераційний процес є збіжним.

***Код програми*** *(файл lab_22.py):*

```
class IterationMethod:
    def __init__(self, a, b, eps, f, phi):
        self.a = a # Ліва межа
        self.b = b # Права межа
        self.x = 6.5 # Шукане значеня
        self.i = 0 # Кількість ітерацій
        self.eps = eps # Точність
        self.f = f # Функція
        self.phi = phi # Канонічні форма функції

        if a >= b:
            print("Помилка, 'a' більше або рівне 'b'")
            exit(1)
        elif eps > b - a:
            print("Помилка, 'eps' більше ніж інтервал між 'a' та 'b'")
            exit(1)

    def calculate(self):
        print(f"x: {self.x} f(x): {self.calc_f(self.f, self.x)}")
        if abs(self.calc_f(self.f, self.x)) < self.eps:
            return f"Відповідь: {self.x} \n Кількість ітерацій: {self.i}"
        else:
            self.i += 1
            self.x = self.calc_f(self.phi, self.x)
            return self.calculate()

    def calc_f(self, f, x):
        return eval(f.replace("x", str(x)))

a = float(input("Ліва межа: "))
b = float(input("Права межа: "))
eps = float(input("Точність: "))
f = input("Функція: ") or "x*x*x - 6*x*x - 7"
phi = input("Функція у канонічній формі: ") or "(6*x*x + 7)**(1/3)"

print(IterationMethod(a, b, eps, f, phi).calculate())
```

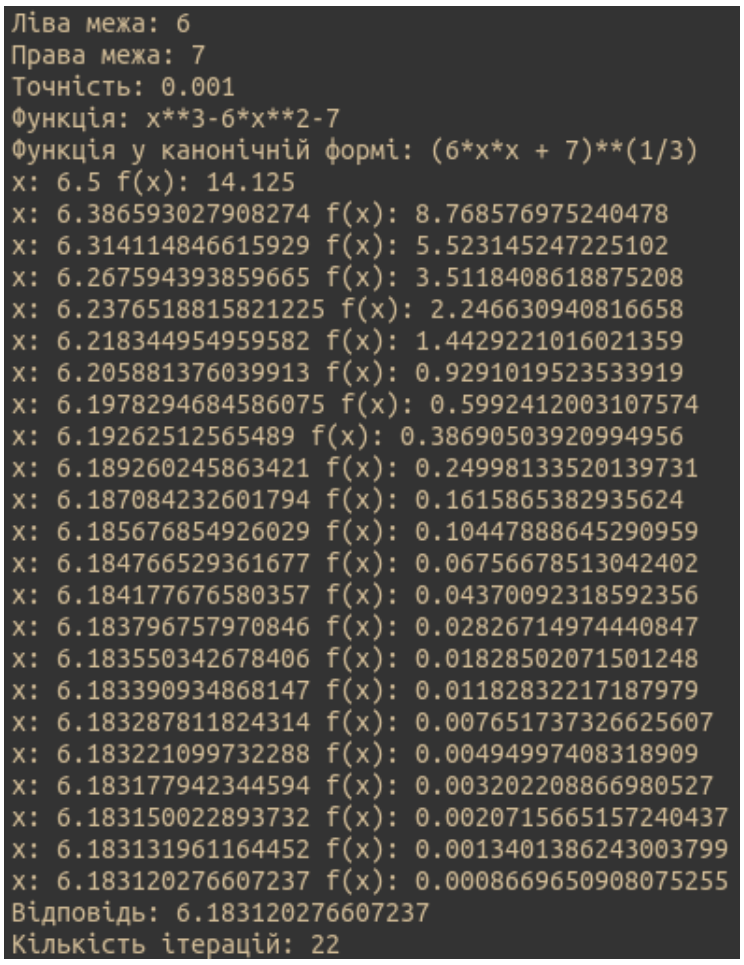


Рис. 5: Робота програми

## Висновок

На лабораторній роботі я засвоїв практичні навички використання методу дотичних та методу послідовних наближень та розробив функції для розв'язку нелінійного рівняння $x^3 - 6x^2 - 7$ з точністю 0.001 за допомогою цих методів. Корінь рівняння 6.183.