МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Інститут **КНІТ**
Кафедра **ПЗ**

ЗВІТ

До лабораторної роботи № 1
На тему: "*Розв'язування нелінійних рівнянь методом дихотомії та методом хорд*"
З дисципліни: "Чисельні методи ПЗ"

Лектор:
доцент кафедри ПЗ
Мельник Н.Б.

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

Прийняв:
асистент кафедри ПЗ
Гарматій Г.Ю.

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

Львів — 2022

Тема. Розв'язування нелінійних рівнянь методом дихотомії та методом хорд.

Мета. Ознайомлення на практиці з методами відокремлення дійсних ізольованих коренів нелінійних рівнянь. Вивчення методу дихотомії та методу хорд уточнення коренів.

Індивідуальне завдання

1. Ознайомитися з теоретичним матеріалом.
2. Скласти програму розв'язування рівняння $x^3 - 6x^2 - 7 = 0$ методом дихотомії та методом хорд.

Теоретичні відомості

Наступні методи розв'язування нелінійних рівнянь дозволяються знайти розв'язок для наступної задачі: Розглянемо рівняння $f(x) = 0$, у якому $f(x)$ є неперервною нелінійною функцією. На відрізку $[a, b]$ дана функція є монотонною та диференційованою, на ньому міститься єдиний корінь x заданого рівняння, тобто $f(a)f(b) < 0$. Потрібно знайти значення кореня x із заданою похибкою ϵ.

Метод поділу відрізка навпіл

Покладемо $a_0 = a$, $b_0 = b$ і обчислимо $x_0 = \frac{(a_0+b_0)}{2}$. Якщо $f(x_0) = 0$, то $x = x_0$, у протилежному випадку, якщо $f(x_0) \neq 0$, то чинимо так:

$a_{n+1} = \{x_n$, якщо $signf(a_n) = signf(x_n)\}$, (1)

$b_{n+1} = \{x_n$, якщо $signf(b_n) = signf(x_n)\}$, (2)

$x_{n+1} = \frac{a_{n+1}+b_{n+1}}{2}$, $n = 1, 2, 3, ...$, (3)

і обчислюємо $f(x_{n+1})$. Якщо $f(x_{n+1}) = 0$, то ітераційний процес завершуємо і вважаємо, що $x \approx x_{n+1}$, а коли $f(x_{n+1}) \neq 0$, то продовжуємо ітераційний процес (1)-(3).

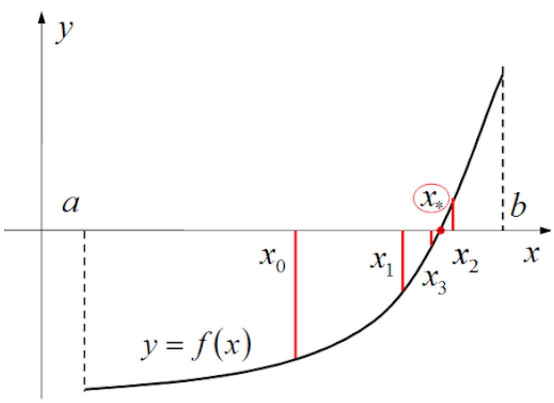

Рис. 1: Геометрична інтерпретація методу дихотомії

Графічна інтерпретація методу поділу відрізку навпіл

Зі співвідношень (1), (2) видно, що $signf(a_{n+1}) = signf(a_n)$ і $signf(a_{n+1}) = signf(a_n)$. Тому $f(a_{n+1})f(b_{n+1}) < 0$, а отже шуканий корінь x знаходиться на відрізку $[a_{n+1}, b_{n+1}]$.

Точність знаходиться за формулою $\epsilon = \frac{b-a}{2^{n+1}}$ тобто виконується нерівність $|x_n - x| \leq \frac{b-a}{2^{n+1}}$ (4).

Звідси випливає, що кількість ітерацій. які необхідно провести для знаходження наближеного кореня рівняння $f(x) = 0$ з заданою точністю ϵ задовольняє співвідношенню $n = [log_2 \frac{b-a}{\epsilon} - 1]$, де $[\xi]$ - ціла частина числа ξ (5)

Серед переваг даного методу потрібно відзначити простоту реалізації та надійність. Недоліком наведеного методу є невелика швидкість його збіжності.

Метод хорд

Суть методу хорд полягає в тому, що на відрізку $[a, b]$ малої довжини дугу функції $f(x)$ замінюють хордою ab , яка її стягує. За наближене значення кореня приймають абсцису точки перетину хорди з віссю Ox.

Для довільного $(i+1)$-го наближення точного значення кореня x для заданого рівняння використовують формулу $x_{i+1} = x_i - \frac{f(x_i)(b-x_i)}{f(b)-f(x_i)}$, $i = 1, 2, 3...$, де $x_0 = a$. (6)

Дугу кривої стягують хордою доти, поки шуканий наближений корінь не досягне точності ϵ, тобто $|x_{i+1} - x_i| < \epsilon$, (7)

де x_i, x_{i+1} - наближені значення кореня рівняння $f(x) = 0$, відповідно на i-му та $(i+1)$-му ітераційному кроці.

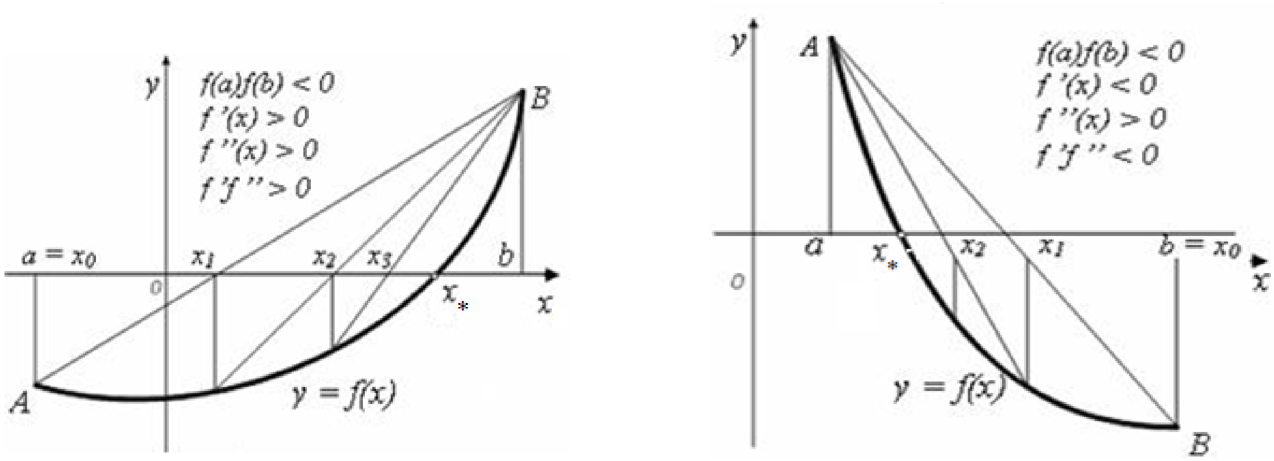

Рис. 2: Геометрична інтерпретація методу хорд

Для автоматизованого вибору рухомого кінця хорди і відповідно визначення співвідношення для обчислення наближеного значення кореня існує певне правило: нерухомим кінцем відрізка є той, для якого знак функції $f(x)$ співпадає зі знаком її другої похідної $f''(x)$. Якщо $f(b)f''(b) > 0$, то нерухомим є кінець $a(x_0 = b)$

Хід роботи

Графічний метод

Рис. 3: Графічний метод

За графіком функції розв'язок даної функції розташований на відрізку $[6; 7]$.

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 дихотомії

Код програми (файл *lab_11.py*):

```
class DichotomyMethod:
    """ Метод Дихотомії """


    def __init__(self, a, b, eps, f):
        self.a = a # Ліва межа
        self.b = b # Права межа
        self.x = 0 # Шуканий корінь
        self.eps = eps # Точність
        self.f = f # Функція


    def calculate(self):
        self.x = (self.a + self.b)/2
        print(f"a: {self.a}; b: {self.b}; x: {self.x}; f(x):
            {self.calc_func(self.x, self.f)}")
        if abs(self.calc_func(self.x, self.f)) < self.eps:
            return
        elif self.a == self.b:
            return print("No roots")
        elif self.calc_func(self.a, self.f) *
            self.calc_func(self.x, self.f) < 0:
            self.b = self.x
            return self.calculate()
        else:
            self.a = self.x
            return self.calculate()


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


a = float(input("Введіть ліву межу: "))
b = float(input("Введіть праву межу: "))
eps = float(input("Введіть точність: "))
f = input("Введіть функцію: ")

d = DichotomyMethod(a, b, eps, f)
d.calculate()
```

```
a: 6; b: 7; x: 6.5; f(x): 14.125
a: 6; b: 6.5; x: 6.25; f(x): 2.765625
a: 6; b: 6.25; x: 6.125; f(x): -2.310546875
a: 6.125; b: 6.25; x: 6.1875; f(x): 0.178466796875
a: 6.125; b: 6.1875; x: 6.15625; f(x): -1.078216552734375
a: 6.15625; b: 6.1875; x: 6.171875; f(x): -0.4529304504394531
a: 6.171875; b: 6.1875; x: 6.1796875; f(x): -0.13799715042114258
a: 6.1796875; b: 6.1875; x: 6.18359375; f(x): 0.02004331350326538
a: 6.1796875; b: 6.18359375; x: 6.181640625; f(x): -0.05902477353811264
a: 6.181640625; b: 6.18359375; x: 6.1826171875; f(x): -0.019502696581184864
a: 6.1826171875; b: 6.18359375; x: 6.18310546875; f(x): 0.00026731647085398436
a: 6.1826171875; b: 6.18310546875; x: 6.182861328125; f(x): -0.009618438009056263
a: 6.182861328125; b: 6.18310546875; x: 6.1829833984375; f(x): -0.004675747763030813
a: 6.1829833984375; b: 6.18310546875; x: 6.18304443359375; f(x): -0.002204262395252954
a: 6.18304443359375; b: 6.18310546875; x: 6.183074951171875; f(x): -0.0009684846495758848
a: 6.183074951171875; b: 6.18310546875; x: 6.1830902099609375; f(x): -0.00035058701121215563
```

Рис. 4: Робота програми за допомогою методу дихотомії

Метод хорд

Код програми (файл *lab_12.py*):

```python
class SecantMethod:
    """ Метод Хорд """

    def __init__(self, a, b, eps, f, f2):
        self.a = a # Ліва межа
        self.b = b # Права межа
        self.x = 0 # Шуканий корінь
        self.c = 0 # Нерухомий кінець
        self.eps = eps # Точність
        self.f = f # Функція
        self.f2 = f2 # Друга похідна функції

        if self.calc_func(self.a, self.f) * self.calc_func(self.a, self.f2) > 0:
            self.x = self.b
            self.c = self.a
        else:
            self.x = self.a
            self.c = self.b

    def calculate(self):
        print(f"c: {self.c}; x: {self.x}; f(x):
            {self.calc_func(self.x, self.f)}")
        if abs(self.calc_func(self.x, self.f)) < self.eps:
            return
        self.x = self.x - (self.calc_func(self.x, self.f) * (self.c - self.x))
            / (self.calc_func(self.c, self.f) - self.calc_func(self.x, self.f))
        return self.calculate()

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

a = float(input("Введіть ліву межу: "))
b = float(input("Введіть праву межу: "))
eps = float(input("Введіть точність: "))
f = input("Введіть функцію: ")
f2 = input("Введіть другу похідну функції: ")
```

```
    s = SecantMethod(a, b, eps, f, f2)
    s.calculate()
```

```
c: 7; x: 6; f(x): -7
c: 7; x: 6.142857142857143; f(x): -1.6093294460641516
c: 7; x: 6.174488567990374; f(x): -0.3477439036638259
c: 7; x: 6.181267360469596; f(x): -0.07412469461252158
c: 7; x: 6.182709774478561; f(x): -0.015754403360915603
c: 7; x: 6.183016229046821; f(x): -0.003346355551968827
c: 7; x: 6.183081317150865; f(x): -0.0007106979714990302
c: 7; x: 6.183095140308644; f(x): -0.00015093360147488966
```

Рис. 5: Робота програми за допомогою методу хорд

Висновок

На лабораторній роботі я за допомогою хорд та дихотомії знайшов корені нелінійного рівняння $x^3 - 6x^2 - 7$ з точністю 0.001, та розробив функції для розв'язку цих рівнянь.