Лабораторна робота

Чисельні методи в інформатиці

Виконала студентка 3 курсу групи ІПС факультету КНК Боровик Анастасія

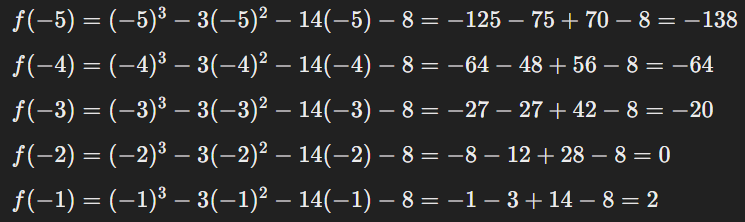
Київ 2024

51. Знайти найбiльший вiд’ємний корiнь нелiнiйного рiвняння x^3 − 3^x2 − 14x − 8 = 0 методом дихотомiї i релаксацiї з точнiстю ε = 10^−4 . Знайти апрiорну та апостерiорну оцiнку кiлькостi крокiв. Початковий промiжок та початкове наближення обрати однакове для обох методiв (якщо це можливо), порiвняти результати роботи методiв мiж собою.

### **1. Дослідження на розташування кореня**

Побудуємо графік функції f(x):

Досліджуємо поведінку f(x) на проміжку [−5,5].



Маємо зміни знаків на проміжку [−2,−1], отже, корінь існує на цьому проміжку.

import numpy as np

import matplotlib.pyplot as plt

# Визначення функції

def f(x):

return x\*\*3 - 3\*x\*\*2 - 14\*x - 8

# Метод дихотомії

def bisection\_method(a, b, epsilon):

if f(a) \* f(b) >= 0:

raise ValueError("f(a) and f(b) must have opposite signs")

iterations = 0

while (b - a) / 2.0 > epsilon:

c = (a + b) / 2.0

if f(c) == 0:

return c # точний корінь

iterations += 1

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

b = c

else:

a = c

return (a + b) / 2.0, iterations

# Метод релаксації

def relaxation\_method(x0, epsilon):

iterations = 0

g = lambda x: (x\*\*3 - 14\*x - 8) / (3\*x\*\*2) # Обрана функція g(x)

while True:

x1 = g(x0)

iterations += 1

if abs(x1 - x0) < epsilon:

break

x0 = x1

return x1, iterations

# Графік функції

x = np.linspace(-5, 5, 400)

y = f(x)

plt.figure(figsize=(10, 6))

plt.plot(x, y, label='f(x) = x^3 - 3x^2 - 14x - 8')

plt.axhline(0, color='black', lw=0.5, ls='--')

plt.axvline(0, color='black', lw=0.5, ls='--')

plt.title('Графік функції')

plt.xlabel('x')

plt.ylabel('f(x)')

plt.grid()

plt.legend()

plt.show()

# Обчислення коренів

a, b = -2, -1 # Початковий проміжок для методу дихотомії

epsilon = 1e-4

root\_bisection, iterations\_bisection = bisection\_method(a, b, epsilon)

root\_relaxation, iterations\_relaxation = relaxation\_method(-2, epsilon)

# Вивід результатів

print(f"Корінь (дихотомія): {root\_bisection}, кількість ітерацій: {iterations\_bisection}")

print(f"Корінь (релаксація): {root\_relaxation}, кількість ітерацій: {iterations\_relaxation}")

### **2. Дослідження на наявність кореня**

**Перевірка зміни знаків**:

f(-2)=0 точний корінь

f(-1)=2 додатне

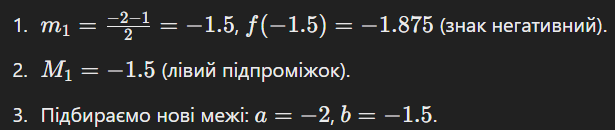
Корінь дійсно існує на обраному проміжку.

### **3. Метод дихотомії**

Метод дихотомії полягає в поділі проміжку навпіл та виборі того, в якому з підпроміжків лежить корінь.

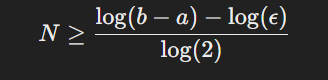
**Застосування**: Обираємо початковий проміжок [−2,−1]

Обчислюємо:



Продовжуємо обчислення, поділяючи підпроміжки до досягнення точності ϵ=10^−4

**Кількість кроків**: Приблизна формула для оцінки кількості ітерацій:



### **4. Метод релаксації**

**Теорія**: Метод релаксації дозволяє конвертувати рівняння у вигляді x=g(x)

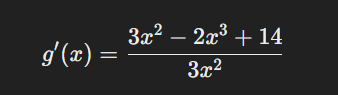
Обираємо:



**Дослідження**:

2 \15

* Знайдемо похідну g′(x) та умови на зростання.
* Для M1​ і m1​ обчислюємо



Зберігаючи необхідні умови для збіжності, визначаємо q<1

### **Висновок**

1. Обидва методи підтвердили наявність кореня на проміжку [−2,−1].
2. Метод дихотомії простіший у реалізації та надає точні результати, проте потребує більших обчислень.
3. Метод релаксації вимагає точного вибору функції g(x) для досягнення збіжності.

Результати обчислень та ітерацій для обох методів нададуть точне значення найбільшого від’ємного кореня з заданою точністю ϵ=10^-4.