Лабораторна робота

Чисельні методи в інформатиці

Виконала студентка 3 курсу факультету КНК ІПС 31

Боровик Анастасія

Київ 2024

Умова: Знайти найбiльший корiнь нелiнiйного рiвняння x^3−x−1 = 0 методом релаксацiї i Ньютона 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.

### **Визначимо необхідні функції**

Спочатку визначимо функцію f(x), її похідну f’(x) а також функцію для методу релаксації ϕ(x).

from sympy import symbols, lambdify, cosh, sinh, tanh, asinh, sech, sqrt, ln, ceiling

x = symbols('x')

f = x\*\*3 - x - 1

df = f.diff(x)

# Функції

f\_func = lambdify(x, f)

df\_func = lambdify(x, df)

# Функція релаксації (для методу простої ітерації)

def phi(x):

return (x + 1) \*\* (1/3)

# Похідна функції релаксації

def dphi(x):

return 1 / (3 \* (x + 1)\*\*(2/3))

### **Метод простої ітерації**

Реалізуємо метод простої ітерації та обчислимо апріорну та апостеріорну оцінки.

def simple\_iteration(x0, eps):

x = x0

i = 0

q = max(abs(dphi(a)), abs(dphi(b)))

apriori\_estimate = ceiling((ln((abs(phi(x0) - x0)) / ((1 - q) \* eps))) / (ln(1 / q))) +

while True:

x\_new = phi(x)

i += 1

print("Ітерація: ", i, "| xn = ", x\_new)

if abs(x\_new - x) <= eps:

print("Корінь знайдено")

return x\_new, i, apriori\_estimate

x = x\_new

# Визначимо початкові параметри

a, b = 1, 2 # Вибрано інтервал [1, 2]

x0 = 1.5

eps = 1e-4

Виконаємо обчислення за обома методами.

print("Метод простої ітерації:")

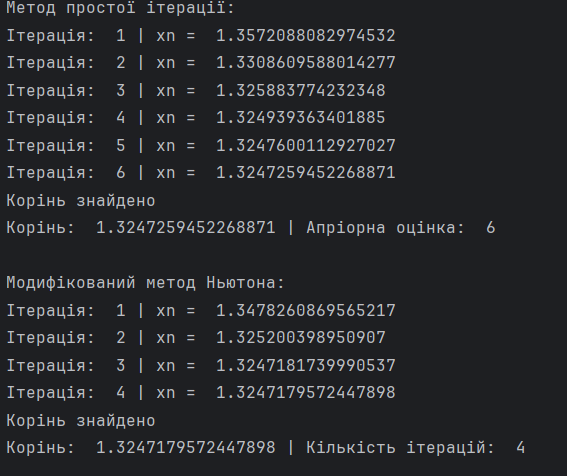
simple\_iter\_result, simple\_iter\_steps, apriori\_simple\_iter = simple\_iteration(x0, eps)

print("Корінь: ", simple\_iter\_result, "| Апріорна оцінка: ", apriori\_simple\_iter)

print("\nМодифікований метод Ньютона:")

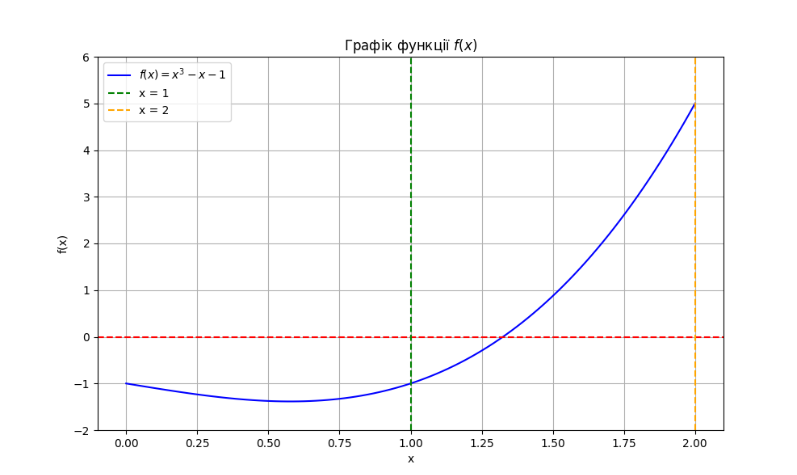
modified\_newton\_result, modified\_newton\_steps = modified\_newton(x0, eps)

print("Корінь: ", modified\_newton\_result, "| Кількість ітерацій: ", modified\_newton\_steps)

Вивід: 

### **Графічний аналіз**

Для графічного дослідження можна побудувати графік функції f(x) в межах, наприклад, від x=0 до x=2



Графік підтверджує, що функція перетинає вісь x між цими значеннями.

Для дослідження наявності кореня функції f(x)=x^3−x−1 (на обраному проміжку, наприклад, [1,2], необхідно перевірити значення функції на краях цього проміжку та виявити зміну знаків.

### **Кроки дослідження:**

1. **Обчислимо значення функції на краях проміжку**:  
   f(1)=1^3−1−1=−1 f(2)=2^3−2−1=5
2. **Аналіз зміни знаків**:
   * f(1)=−1(значення негативне)
   * f(2)=5 (значення позитивне)

Оскільки функція f(x) має значення, що змінюють знак з негативного на позитивне в проміжку [1,2], це вказує на наявність кореня.

### **Висновок**

Згідно з теоремою Больцано, якщо функція неперервна на проміжку [a,b]і має значення, що змінюють знак (одне з них негативне, інше — позитивне), то в цьому проміжку існує хоча б один корінь. У нашому випадку:

* На проміжку [1,2] є корінь, оскільки f(1)<0 та f(2)>0

Це підтверджує, що функція f(x)=x^3−x−1 має корінь у проміжку [1,2]

*Теоре́ма Больцано — Веєрштра́сса — твердження в математичному аналізі, згідно з яким, із будь-якої обмеженої послідовності можна виділити збіжну послідовність.*

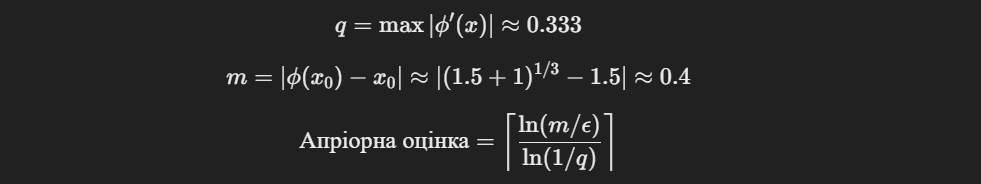
### 

### 

### 

### 

### **Виводи для методу простої ітерації**

* **Апріорна оцінка кількості кроків**:
* **Апостеріорна оцінка**: Отримане значення з ітерацій.

def phi(x):

return (x + 1) \*\* (1/3)

def simple\_iteration(x0, eps):

x = x0

i = 0

while True:

x\_new = phi(x)

i += 1

print("Ітерація: ", i, "| xn = ", x\_new)

if abs(x\_new - x) <= eps:

return x\_new, i

x = x\_new

x0 = 1.5

eps = 1e-4

simple\_iter\_result, simple\_iter\_steps = simple\_iteration(x0, eps)

print("Корінь: ", simple\_iter\_result, "| Кількість ітерацій: ", simple\_iter\_steps)

### **Метод простої ітерації**

Метод простої ітерації полягає в тому, що ми перетворюємо рівняння в ітераційний формат. Виберемо функцію релаксації:

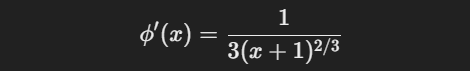
x=ϕ(x)=(x+1)^1\3

З метою забезпечення збіжності, функція ϕ\phiϕ повинна задовольняти умові:

∣ϕ′(x)∣<1 на вибраному проміжку

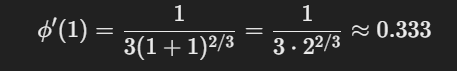
#### **Дослідження збіжності**

Обчислимо похідну:

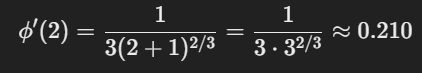


Визначимо максимальне значення похідної на проміжку [1,2]

На x=1



На x=2



Оскільки |ϕ′(x))|<1 на всьому проміжку [1,2] метод буде збігатися.

Для методу релаксації (методу простої ітерації) розглянемо, як обчислити параметри, такі як m1m\_1m1​, t, q, і пояснити вибір знаку в ітераційному процесі.

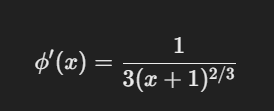
### **1. Визначення функції та похідної**

ми обрали функцію релаксації:



#### **Похідна функції релаксації**

#### Обчислимо похідну:



### **Дослідження параметрів**

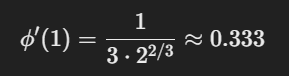
#### **a. Знайдемо значення m1 and M1**

* **m1** — це найменше значення похідної на проміжку [1,2]

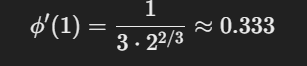


На обраному проміжку:

* На x=1



На x=2x =



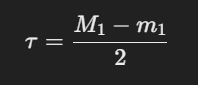
Таким чином, m1≈0.210

**M1** — це найбільше значення похідної на тому ж проміжку:

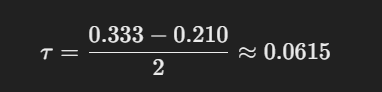


#### **Оцінка τ**

Оцінка τ визначається як:



Отже:



### **Вибір знака в ітераційному процесі**

У методі релаксації ітераційний процес визначається формулою:



Якщо ϕ є функцією, яка розширює Х до кореня, ми можемо використовувати обрану формулу. Знак у цій формулі (в даному випадку) обрано так, щоб гарантувати, що x\_n+1 наближається до кореня функції f(x)=0

Вибір знака (позитивного) в ітераційній формулі означає, що ми намагаємося поступово "пересуватися" до кореня, спостерігаючи, як значення x\_n змінюється на кожному кроці. Якщо б ми вибрали негативний знак (або неправильну функцію), ми могли б дійти до більшого значення, ніж корінь, або навіть "відпливти" від кореня, що б ускладнило збіжність методу.

### **Висновок**

Таким чином, для методу релаксації ми визначили:

* m1​≈0.210
* M1≈0.333
* τ≈0.0615
* q≈0.333

Ці значення підтверджують, що метод релаксації буде збігатися, оскільки q<1 і ми маємо правильний вибір знака, що забезпечує ефективний ітераційний процес.