Міністерство освіти і науки України

Львівський національний університет імені Івана Франка

Факультет електроніки та комп’ютерних технологій

Звіт

про виконання лабораторної роботи №9

З курсу “Методи обчислень”

на тему **:**

«Чисельні методи розв’язування нелінійних рівнянь з одним невідомим»

Виконав

студент групи ФеС-21

Шавало Андрій

Львів 2025 р.

Хід роботи

1. **Табуляція функції та визначення інтервалів із коренями**

Задав трансцендентну функцію на відрізку a,b з кроком h. Побудував таблицю значень (x, F(x)), яку зберіг у файл tab.txt. На основі змін знаку функції визначено інтервали.

```
p = 8
a = p - 3.01
b = p + 7.01
h = 0.1


def F(x):
    return x ** 3 - p * x ** 2
```

```
Перші 5 x: [4.99 5.09 5.19 5.29 5.39]
Перші 5 y: [np.float64(-74.949301), np.float64(-75.392571), np.float64(-75.69
Останні 5 x: [14.69 14.79 14.89 14.99 15.09]
Останні 5 y: [np.float64(1443.675908999986), np.float64(1485.2724389999853),
```

2. **Розв'язок рівняння з точністю ε=10^−10**

Я вибрав середину кожного інтервалу як початкове наближення і обчислив корінь рівняння з високою точністю ε=10^−10

```
def F_prime(x):
    return 3 * x ** 2 - 2 * p * x


def F_double_prime(x):
    return 6 * x - 2 * p


eps = 1e-10

print(F_prime(1))
print(F_double_prime(1))
-13
-10
```

3. **Реалізація методів знаходження коренів**

Я реалізував і застосував до кожного інтервалу такі методи:

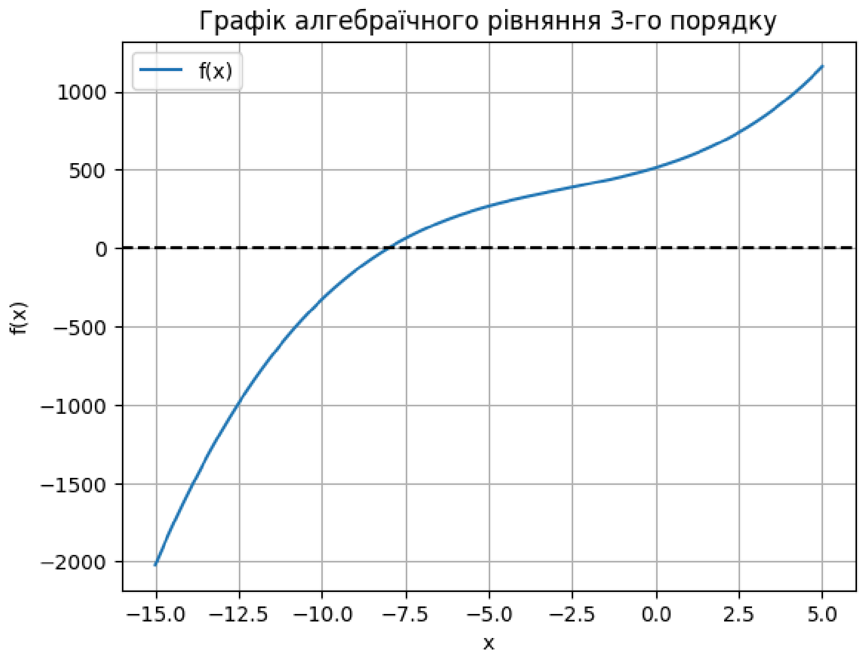
- просту ітерацію
- Ньютона
- Чебишева
- хорд
- парабол
- зворотну інтерполяцію

```
[Інтервал 1: 7.990000 – 8.090000]
  Проста ітерація:        x ≈ -0.0012202068, ітерацій: 1000
  Метод Ньютона:          x ≈ 8.0000000000, ітерацій: 4
  Метод Чебишева:         x ≈ 8.0000000000, ітерацій: 3
  Метод хорд:             x ≈ 8.0000000000, ітерацій: 5
  Метод парабол:          x ≈ 5.3333333897, ітерацій: 9
  Зворотна інтерполяція:  x ≈ 8.0000000000, ітерацій: 5
```

Я зберіг кількість ітерацій, що знадобилися кожному методу для досягнення заданої точності.

4. **Побудував графік алгебраїчного рівняння 3-го порядку**

```python
def f_poly(x):
    return x**3 + p * x**2 + p**2 * x + p**3
```



5. **Оцінка значення алгебраїчного рівняння**

Я реалізував функції для:

- зчитування коефіцієнтів з файлу
- обчислення значення полінома в точці x

```
def read_coeffs(filepath):
    return np.loadtxt(filepath)

def eval_poly(coeffs, x):
    result = 0
    for i, a in enumerate(reversed(coeffs)):
        result += a * x**i
    return result
print(eval_poly(read_coeffs("poly_coeffs.txt"), 2.0))
680.0
```

6. **Метод Ньютона з використанням схеми Горнера**

Я використав метод Ньютона, оптимізований схемою Горнера, щоб знайти дійсний корінь алгебраїчного рівняння.

```
coeffs = read_coeffs("poly_coeffs.txt")
real_root, iter_newton = newton_gorner(coeffs, -p)
print(f"[Горнер] Дійсний корінь ≈ {real_root:.10f}, ітерацій: {iter_newton}")

[Горнер] Дійсний корінь ≈ -8.0000000000, ітерацій: 1
```

7. **Метод Ліна для знаходження комплексних коренів**

Я реалізував метод Ліна для знаходження комплексно спряжених коренів квадратного множника:

```
D = p_quad**2 - 4 * q_quad
if D >= 0:
    r1 = (-p_quad + np.sqrt(D)) / 2
    r2 = (-p_quad - np.sqrt(D)) / 2
    print(f"[Лін] Два дійсних корені: {r1:.10f}, {r2:.10f}")
else:
    real = -p_quad / 2
    imag = np.sqrt(-D) / 2
    print(f"[Лін] Комплексні корені: {real:.10f} ± {imag:.10f}i")

[Лін] Два дійсних корені: 8.0000000000, -8.0000000000
```

Висновок:У ході лабораторної роботи я реалізував чисельні методи для розв’язання нелінійного рівняння F(x)=0. Я здійснив табуляцію функції на заданому відрізку, визначив інтервали зі зміною знаку та застосував шість чисельних методів: просту ітерацію, метод Ньютона, Чебишева, хорд, парабол і зворотної інтерполяції. Я оцінив точність кожного методу та кількість ітерацій до досягнення заданої точності, я побудував алгебраїчне рівняння третього порядку з одним дійсним та двома комплексними коренями. Я зберіг його коефіцієнти у файл, обчислював значення полінома за заданим x, реалізував метод Ньютона по схемі Горнера для знаходження дійсного кореня та метод Ліна — для визначення комплексно спряжених коренів.