Київський національний університет імені Тараса Шевченка

Звіт

До лабораторної роботи №4

з дисципліни “Чисельні методи”

на тему:

“Метод Ньютона. Природничий кубічний сплайн”

Виконав

студент 3 курсу

Групи ТТП-31

Факультету комп’ютерних наук та кібернетики

Сіхневич Святослав

Київ, 2024

Зміст

Постановка задачі

Реалізувати алгоритми інтерполяції з вашого варіанту для табличної функції, отриманої з вашої аналітичної функції. Для вашої аналітичної функції на проміжку обрати не менше 15 точок, за якими побудувати табличну функцію. У звіті навести всі можливі графіки.

А) Метод Ньютона

Б) Природний кубічний сплайн.

9) $1/x, x \in [1,5]$;

Хід роботи

Для реалізації цього завдання реалізуємо алгоритми інтерполяції для функції 9) $1/x, x \in [1,5]$;

Ось код реалізації:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import CubicSpline

def f(x):
    return 1/x

x = np.linspace(1, 5, 15)
y = f(x)

def divided_diff(x, y):
    n = len(x)
    coeffs = np.zeros([n, n])
    coeffs[:, 0] = y
    for j in range(1, n):
        for i in range(n-j):
            coeffs[i, j] = (coeffs[i+1, j-1] - coeffs[i, j-1]) / (x[i+j] - x[i])
    return coeffs[0, :]

def newton_interpolation(x, y, x_new):
    coeffs = divided_diff(x, y)
    n = len(x) - 1
    p = coeffs[n]
    for k in range(1, n+1):
        p = p*(x_new - x[n-k]) + coeffs[n-k]
    return p

f_spline = CubicSpline(x, y, bc_type='natural')
```

```python
f_spline = CubicSpline(x, y, bc_type='natural')

x_new = np.linspace(1, 5, 100)
y_newton = newton_interpolation(x, y, x_new)
y_spline = f_spline(x_new)

plt.plot(x, y, 'o', label='Дані')
plt.plot(x_new, f(x_new), label='Точна функція')
plt.plot(x_new, y_newton, label='Метод Ньютона')
plt.plot(x_new, y_spline, label='Кубічний сплайн')
plt.legend()
plt.xlabel('x')
plt.ylabel('y')
plt.title('Інтерполяція функції 1/x')
plt.grid(True)
plt.show()
```

1. Визначення функції та вибір вузлів інтерполяції

- Визначається функція f(x)=1/x, яка буде інтерпольована.
- Створюється набір вузлів для інтерполяції за допомогою рівномірного поділу відрізка [1,5] на 15 точок.
- Визначаються значення функції y=f(x) у цих вузлах.

2. Реалізація методу Ньютона

- Для методу Ньютона використовується **розділена різниця**, що є основою для обчислення коефіцієнтів інтерполяційного многочлена.
    - Будується таблиця розділених різниць, яка зберігає значення коефіцієнтів на кожному кроці.
    - На основі цієї таблиці обчислюється многочлен Ньютона, що поступово будується, додаючи кожен новий коефіцієнт та множники, залежні від різниці між x(нове) і вузлами інтерполяції.

3. Реалізація кубічного сплайну

- Використовується бібліотека SciPy для побудови **кубічного сплайну** з натуральними крайовими умовами (друга похідна на краях інтервалу дорівнює нулю).
- Кубічний сплайн апроксимує функцію, створюючи набір кубічних поліномів на кожному з інтервалів між сусідніми вузлами.

4. Побудова нової сітки

- Генерується більш щільна сітка точок x(нове) на тому ж інтервалі [1,5], щоб порівняти точні значення функції та результати обох методів інтерполяції.

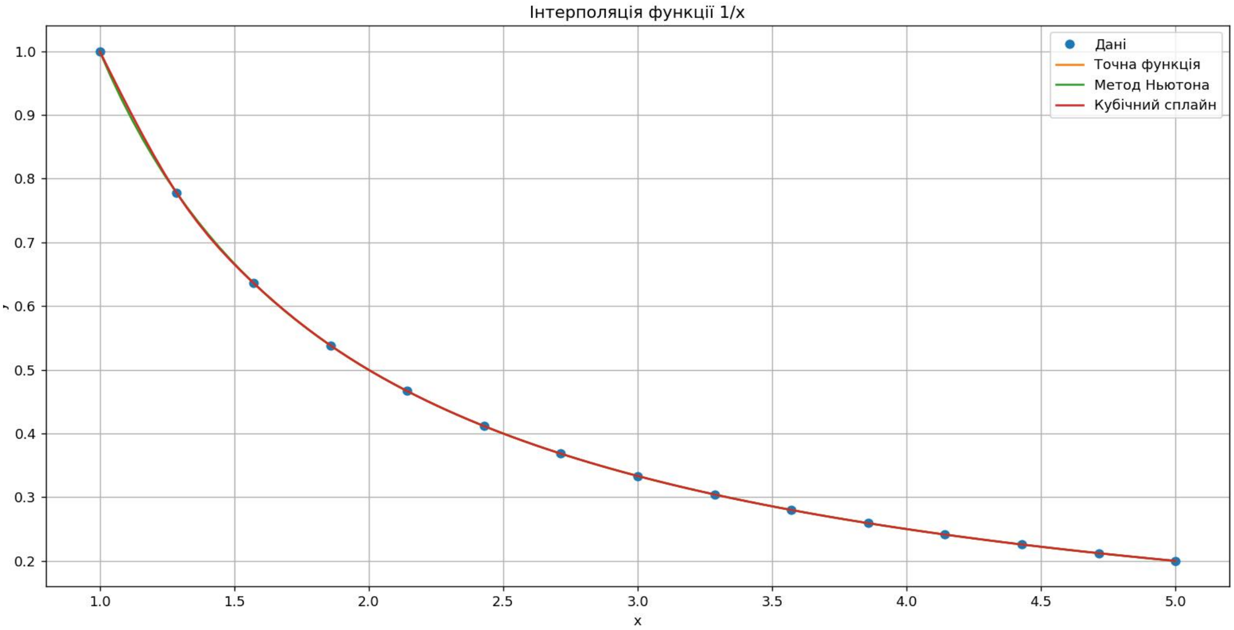
5. Обчислення інтерпольованих значень

- Для кожної точки з нової сітки обчислюються значення функції:
    - Методом Ньютона (на основі коефіцієнтів розділених різниць).
    - Кубічним сплайном (використовуючи функцію SciPy).

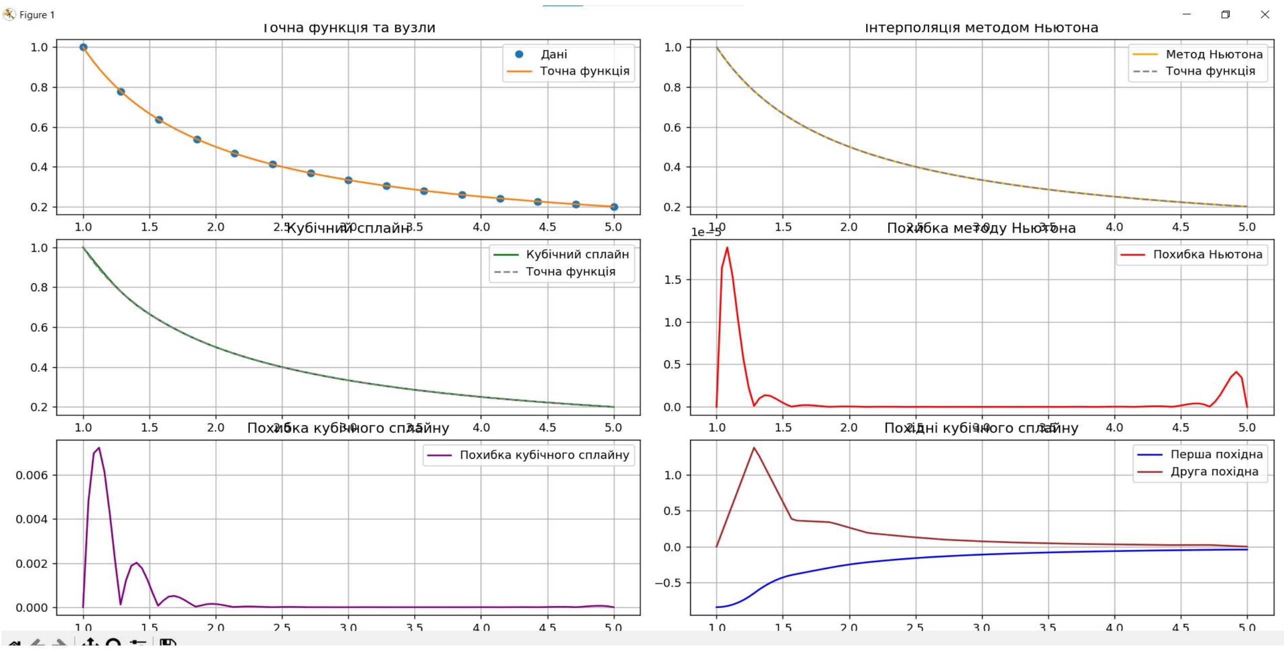
6. Візуалізація результатів

- Будуються графіки:
    - Точна функція $f(x)=1/x$.
    - Набір вихідних точок (вузлів інтерполяції), зображених у вигляді маркерів.
    - Інтерпольовані значення, обчислені методом Ньютона.
    - Інтерпольовані значення, обчислені кубічним сплайном.
- Додається легенда для пояснення кривих, а також підписи осей, заголовок та сітка для поліпшення сприйняття графіка.

Результати виконання:



Ось усі можливі графіки:



Висновок: метод Ньютона добре підходить для задач з невеликою кількістю вузлів, де потрібно швидко отримати інтерполяційний поліном. Кубічний сплайн є кращим вибором, коли важлива гладкість функції, точність у проміжках між вузлами та мінімізація похибки.

Додатки

Посилання на репозиторій з програмним кодом для обрахунку в даній лабораторній роботі:

https://github.com/Svyatoslavk27/NM_4lab