МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені ІГОРЯ СІКОРСЬКОГО»

Лабораторна робота №3

з дисципліни: «Чисельні методи»

16 Варіант

СПЕЦІАЛЬНОСТІ 121 – Інженерія програмного забезпечення

Виконав: Левак О. О.

Група: ІТ-91

Перевірила: Тимофєєва Ю.С.

Київ 2020

**Лабораторна робота №3. Інтерполяція функцій. Сплайн-**

**Інтерполяція**

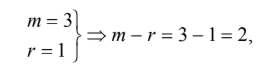
**Мета:** навчитися інтерполяції функцій за допомогою інтерполяції сплайнами. Ознайомитися з тригонометричною інтерполяцією.

**ТЕОРЕТИЧНІ ВІДОМОСТІ**

Розглянута в попередній лабораторній роботі інтерполяція, коли інтерполяційний поліном будується відразу за всіма вузлама інтерполяції називається глобальною інтерполяцією. При цьому збільшення числа вузлів автоматично призводить до підвищення степеня полінома, і як наслідок, до прояву його коливальних властивостей. Тому звичайну поліноміальну інтерполяцію здійснюють максимум за 3-4 вузлами. Інтерполяцію за декількома вузлами таблиці, називають локальною.

Однак, така локальна інтерполяція за допомогою Ln або Nn має ту ваду, що інтерполююча функція у вузлах стикування поліномів має неперервність тільки нульового порядку, тобто інтерполюючи функція належить класу C0 функцій (див. рис. 4.2 для L2 N2 , у вузлі \* x ). Цієї вади не має сплайн-інтерполяція, що забезпечує неперервність у вузлах стикування локальних многочленів за похідними відповідно порядку один, два тощо.

**Сплайном** степеня m дефекту r називається (m-r) раз неперервно диференційовна функція, що на кожному відрізку являє собою многочлен степеня m. Найпоширенішими в науці та техніці є сплайни 3-го степеня дефекту один, тобто



тобто двічі неперервно диференційовний многочлен 3-го степеня на кожному

відрізку. Сплайни, що задовольняють умові інтерполяції, називаються інтерполяційними. Основною перевагою інтерполяційного кубічного сплайна дефекту один є таке: цей сплайн має мінімум інтегральної кривизни на всьому даному відрізку [a,b] порівняно з іншими інтерполяційними функціями f (x).

Щоб визначити 4× n коефіцієнтів використаємо наступні умови у вузлах інтерполяції:

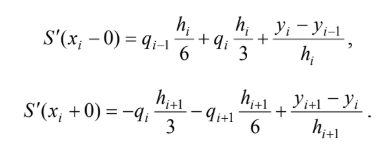
умова інтерполяції i i S(x ) = y , i = 0,n ;

неперервність сплайна ( − 0) = ( + 0) i i S x S x , i = 1,n −1;

неперервність похідних 1-го порядку ′( − 0) = ′( + 0) i i S x S x , i = 1,n −1;

неперервність похідних 2-го порядку ′′( − 0) = ′′( + 0) i i S x S x , i = 1,n −1.

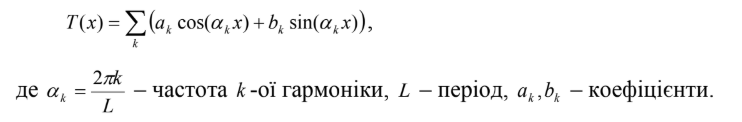
Обчислюючи від похідні першого порядку й підставляючи в них значення i x = x , отримаємо



Оскільки багато явищ у природі мають періодичний характер, на

практиці широко використовується інтерполяція дискретних періодичних

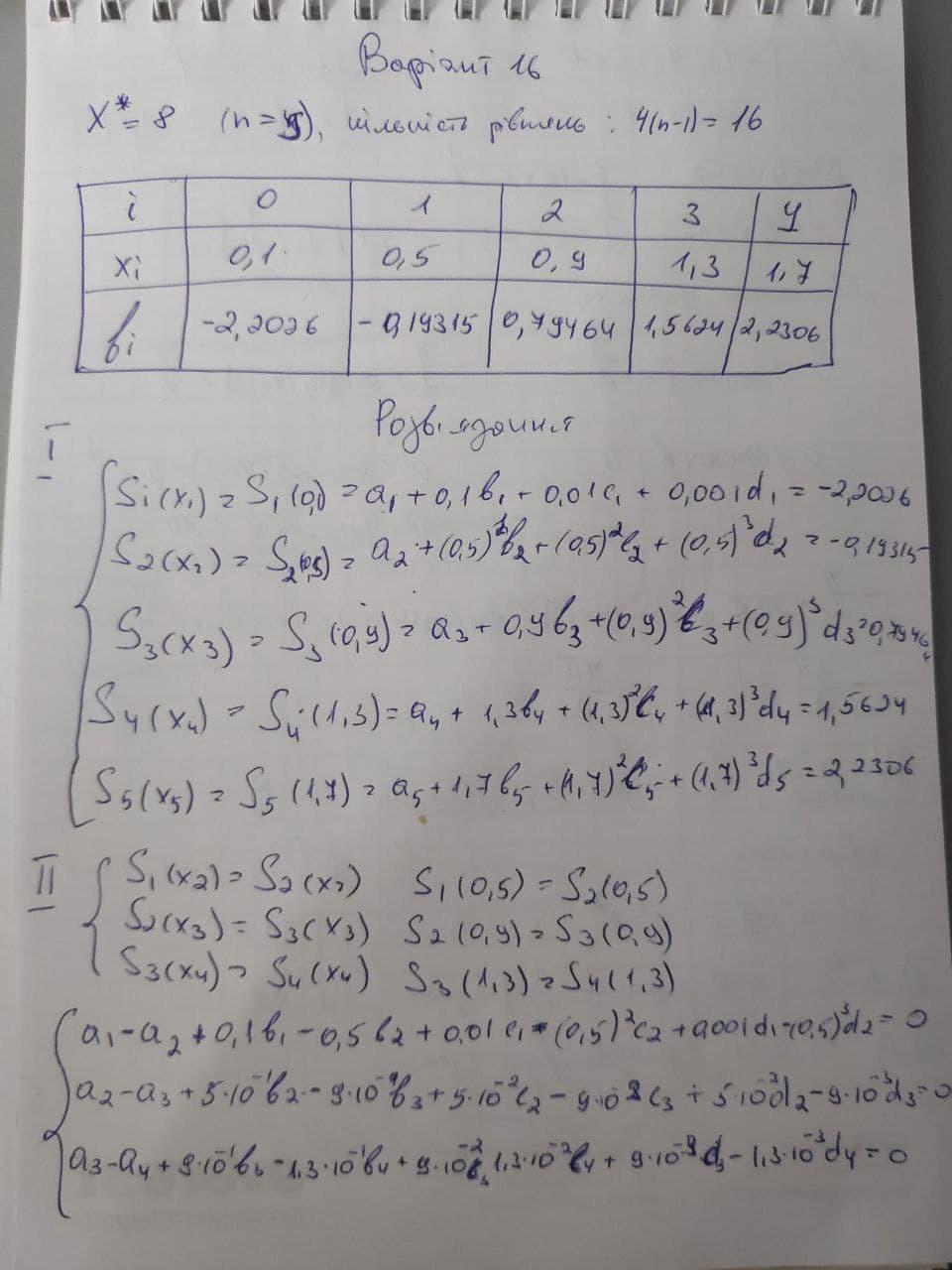
функцій **тригонометричними поліномами** вигляду:

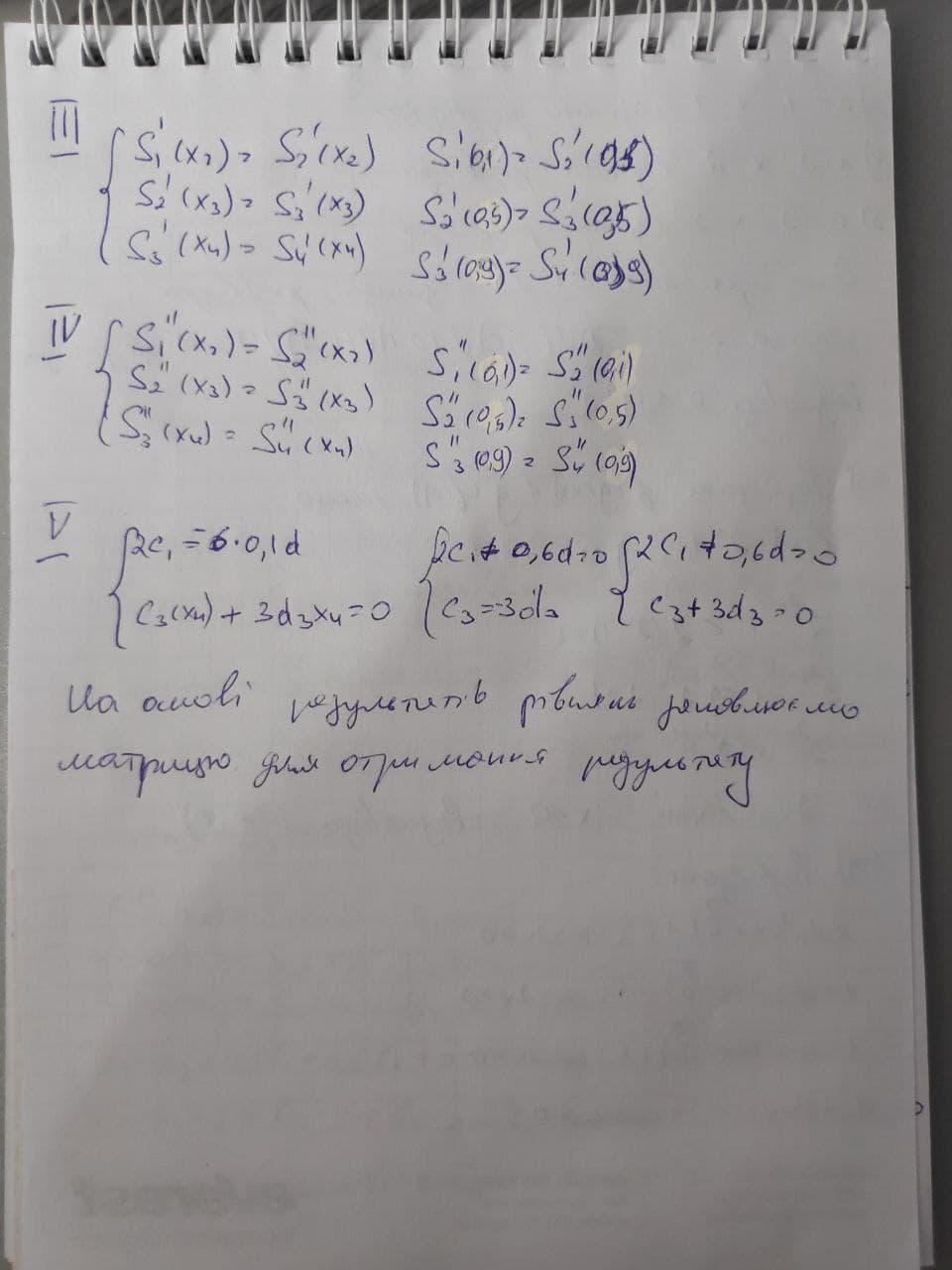


Такий підхід дозволяє подати складну періодичну функцію як

суперпозицію простих періодичних функцій (елементарних гармонік).

**РОЗВ’ЯЗОК ЗАДАЧІ В АНАЛІТИЧНІЙ ФОРМІ**



****

**ЛІСТИНГ ПРОГРАМИ РОЗВ’ЯЗУВАННЯ ЗАДАЧІ**

**import matplotlib.pyplot as plt**

**xPoints = [0.1, 0.5, 0.9, 1.3, 1.7]**

**yPoints = [-2.2026, -0.19315, 0.79464, 1.5624, 2.2306]**

**h = 1**

**x = 8**

**def FindCoefs(xPoints, yPoints, h):**

**n = len(xPoints)**

**resultCoefs = [0 for i in range(n)]**

**startCoefs = [[0, 0, 0, 0, 0, 0] for i in range(n)] # a b c d A B (0, 1, 2, 3, 4, 5)**

**startCoefs[1][0] = 0 # a1**

**startCoefs[n - 2][2] = 0 # c(n-2)**

**for i in range(2, n - 1):**

**startCoefs[i][0] = h / 6 # a**

**for i in range(1, n - 1):**

**startCoefs[i][1] = (2 \* h) / 3 # b**

**for i in range(1, n - 2):**

**startCoefs[i][2] = h / 6 # c**

**for i in range(1, n - 1):**

**startCoefs[i][3] = ((yPoints[i + 1] - yPoints[i]) / h) - ((yPoints[i] - yPoints[i - 1]) / h) # d**

**startCoefs[1][4] = startCoefs[1][2] / startCoefs[1][1] # A = c / b**

**for i in range(2, n - 1):**

**startCoefs[i][4] = startCoefs[i][2] / (startCoefs[i][1] + startCoefs[i][0] \* startCoefs[i - 1][4]) # A**

**startCoefs[1][5] = startCoefs[1][3] / startCoefs[i][1] # B = d / b**

**for i in range(2, n - 1):**

**startCoefs[i][5] = (startCoefs[i][3] - startCoefs[i][0] \* startCoefs[i - 1][5]) # B = d - a \* B**

**startCoefs[i][5] /= (startCoefs[i][1] + startCoefs[i][0] \* startCoefs[i - 1][4]) # B /= b + a \* A**

**for i in range(n - 2, 0, -1):**

**resultCoefs[i] = startCoefs[i][4] \* resultCoefs[i + 1] + startCoefs[i][5] # q = A \* q0 + B**

**return resultCoefs**

**def SplineX(coefs, xPoints, yPoints, x, h):**

**intervals = [(xPoints[i - 1], xPoints[i]) for i in range(1, len(xPoints))]**

**intervalLen = len(intervals)**

**if x < intervals[0][0]:**

**return FinishFn(coefs, xPoints, yPoints, x, 0, h)**

**if x > intervals[intervalLen - 1][1]:**

**return FinishFn(coefs, xPoints, yPoints, x, len(xPoints) - 2, h)**

**for i in range(0, intervalLen):**

**if x >= intervals[i][0] and x <= intervals[i][1]:**

**return FinishFn(coefs, xPoints, yPoints, x, i, h)**

**def FinishFn(coefs, xPoints, yPoints, x, i, h):**

**f = (((coefs[i] \* (xPoints[i + 1] - x) \*\* 3) / 6 \* h))**

**f += ((coefs[i + 1] \* (x - xPoints[i]) \*\* 3) / 6 \* h)**

**f += (((yPoints[i] / h) - coefs[i] \* (h / 6)) \* (xPoints[i + 1] - x))**

**f += (((yPoints[i + 1] / h) - coefs[i + 1] \* (h / 6)) \* (x - xPoints[i]))**

**f \*= (-1)**

**return f**

**coefs = FindCoefs(xPoints, yPoints, h)**

**print(coefs)**

**valuesPlotX = [x \* 0.1 for x in range(10, 50)]**

**valuesPlot = [SplineX(coefs, xPoints, yPoints, i, h) for i in valuesPlotX]**

**print([SplineX(coefs, xPoints, yPoints, i, h) for i in valuesPlotX])**

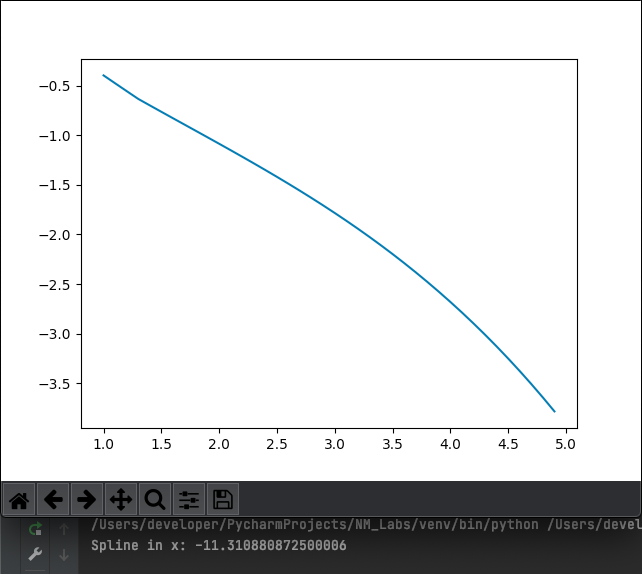
**plt.plot(valuesPlotX, valuesPlot)**

**plt.show()**

**s = SplineX(coefs, xPoints, yPoints, x, h)**

**print("Spline in x:", s)**

**РЕЗУЛЬТАТИ ВИКОНАННЯ ПРОГРАМИ**

****

**Висновки :**

В ході лабораторної роботи ми навчилися інтерполяції функцій за допомогою інтерполяції сплайнами та знайомилися з тригонометричною інтерполяцією.