**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**

**«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»**

Кафедра інформатики та інтелектуальної власності

**ЗВІТИ**

про виконання лабораторних робіт з дисципліни

«Методи та засоби обчислювальної математики»

Варіант 18

Група КН-321в

Виконавець Дмитро ХОМА

Викладач Дмитро ЄЛЬЧАНІНОВ

Харків 2023

**4 ІНТЕРПОЛЯЦІЯ КУБІЧНИМИ СПЛАЙНАМИ**

**4.1 Завдання**

Для заданих варіантів значень функції в певних точках провести її інтерполяцію кубічними сплайнами. Крайові умови (значення другої похідної) для всіх варіантів прийняти рівними 0.

Значення функції для заданого варіанта подані у табл. 4.1.

Таблиця 4.1 – Значення функції в певних точках

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 0.5 | 1 | 2 | 3.5 | 4 | 6 |
|  | -15.351 | 1.396 | 34.206 | 82.917 | 57.901 | 98.128 | 123.178 |

**4.2 Метод інтерполяції кубічними сплайнами**

У загальному випадку значення функції умовно можуть бути подані у табл. 4.2.

Таблиця 4.2 – Значення функції у загальному випадку

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | … |  |  |  | … |  |
|  |  | … |  |  |  | … |  |

Інтерполяційну функцію шукають у вигляді інтерполяційного полінома

Поліноми , …, , …, називаються кубічними сплайнами.

Кубічні сплайни мають задовольняти таким умовам:

– відповідність значенням функції : ; ;

– відповідність перших похідних сусідніх сплайнів: ;

– відповідність других похідних сусідніх сплайнів: ;

– відповідність крайовим умовам: ; .

Це призводить до такої системи з лінійних рівнянь для визначення коефіцієнтів кубічних сплайнів:

**4.3 Код програми**

Інтерполяція кубічними сплайнами реалізована у коді програми мовою Python, як показано на рис. 4.1.

from gaussMethod import Gauss

x = [0, 0.5, 1, 2, 3.5, 4, 6]

f = [-15.351, 1.396, 34.206, 82.917, 57.901, 98.128, 123.178]

d = len(x)

s = [0]\*4\*(d-1)

A = []

for i in range(4\*(d-1)):

    A.append(s[:])

b = s[:]

for j in range(d-1):

    for i in range(4):

        A[j][i+4\*j] = (x[j]-x[j+1])\*\*i

    b[j] = f[j]

for j in range(d-1, 2\*(d-1)):

    A[j][4\*(j-(d-1))] = 1

    b[j] = f[1+(j-(d-1))]

for j in range(2\*(d-1), 3\*d-4):

    for i in range(4):

        if i == 0:

            A[j][1+4\*(j-2\*(d-1))] = -1

        else:

            A[j][i+4+4\*(j-2\*(d-1))] = i\*(x[1+j-2\*(d-1)]-x[2+j-2\*(d-1)])\*\*(i-1)

for j in range(3\*d-4, 4\*d-6):

    A[j][2+4\*(j-(3\*d-4))] = -2

    A[j][6+4\*(j-(3\*d-4))] = 2

    A[j][7+4\*(j-(3\*d-4))] = 6\*(x[1+(j-(3\*d-4))]-x[2+(j-(3\*d-4))])

A[4\*d-6][2] = 2

A[4\*d-6][3] = 6\*(x[0]-x[1])

A[4\*d-5][-2] = 2

print('Матриця А')

for i in range(len(A)):

    print(A[i])

print('Вектор b')

print(b)

c = Gauss(A,b)

print()

print('Вектор коефіцієнтів')

print(c)

print()

print('Коефіцієнти сплайнів')

for i in range(d-1):

    print(c[4\*i:4\*i+4])

print()

print('Сплайни')

for j in range(d-1):

    spline = 'S\_' + str(j+1) + '(x) = ' + str(c[0+4\*j]) + '+'

    for i in range(1, 4):

        spline = spline + str(c[i+4\*j]) + '\*(x-' + str(x[j+1]) + ')^' + str(i) + '+'

    spline = spline[:-1].replace('+-', '-').replace('^1', '')

    spline = spline + ', на [' + str(x[j]) + ';' + str(x[j+1]) + ']'

    print(spline)

Рисунок 4.1 – Код програми, що реалізує інтерполяцію кубічними сплайнами

**4.4 Тестування програми**

Результат роботи програми, що реалізує інтерполяцію кубічними сплайнами, показано на рис. 4.2.

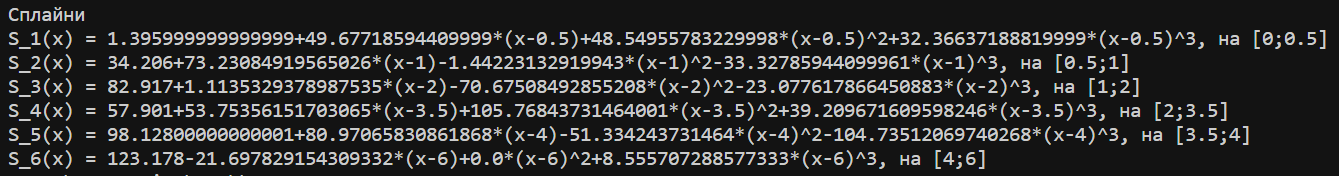


Рисунок 4.2 – Результат роботи програми, що реалізує інтерполяцію кубічними сплайнами

Застосуємо засоби графічного онлайн-калькулятора Desmos задля візуалізації кубічних сплайнів. Графіки цих сплайнів «плавно» і без «викидів» з'єднують задані точки, як показано на рис. 4.3.

Коефіцієнти кубічних сплайнів отримуються у результаті розв'язання системи лінійних рівнянь із великою кількістю невідомих. При розв’язку цієї системи можуть бути допущені як помилки обчислення, так і помилки округлення. Тому необхідним етапом побудови кубічних сплайнів є перевірка їхньої відповідності заданим умовам.

Цю перевірку можна зробити засобами сервісів Wolfram Cloud як показано на рис. 4.4-4.21.

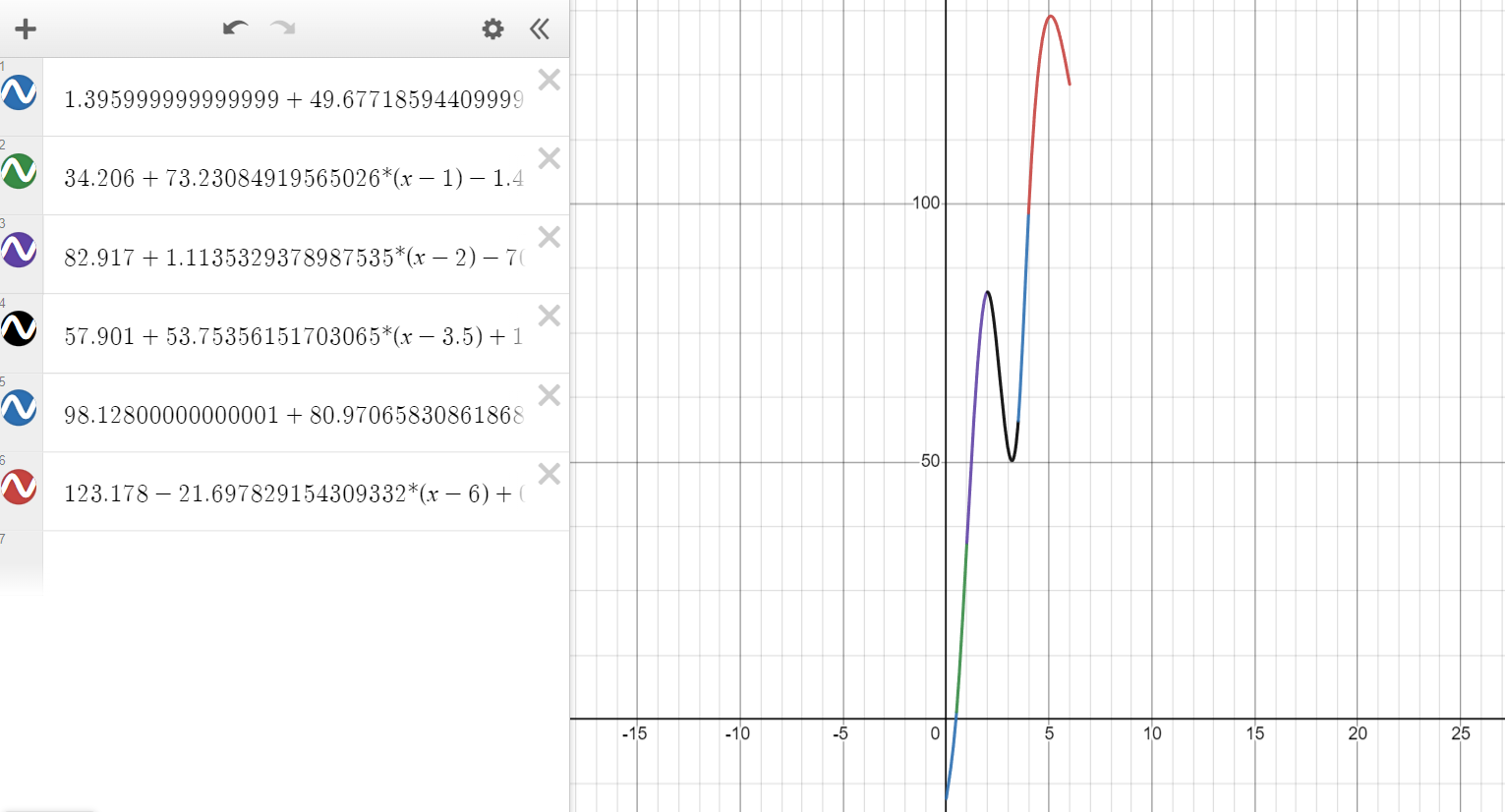


Рисунок 4.3 – Графіки кубічних сплайнів

**4.4.1 Перевірка відповідності сплайнів значенням функції**

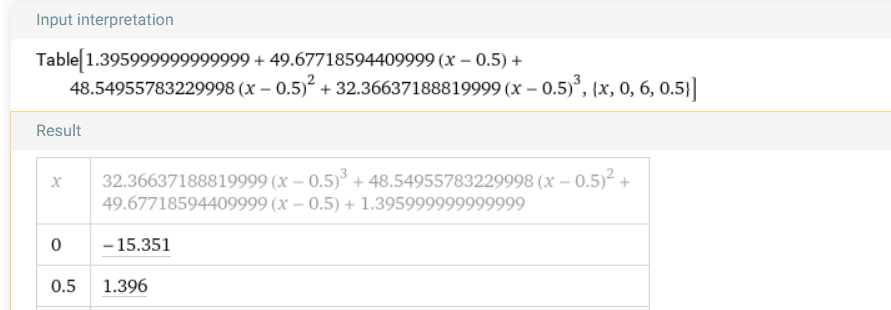


Рисунок 4.4 – Перевірка відповідності сплайну

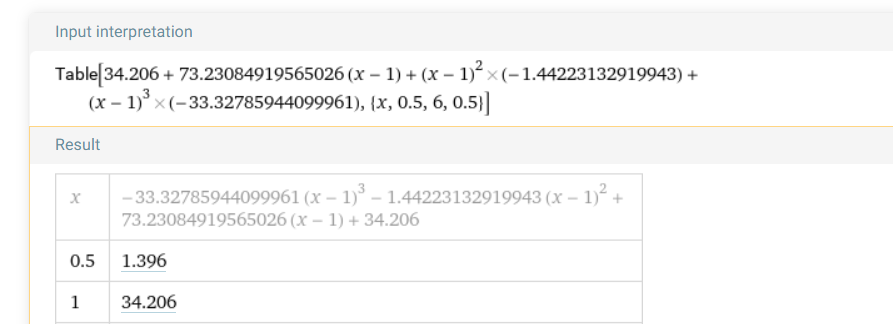


Рисунок 4.5 – Перевірка відповідності сплайну

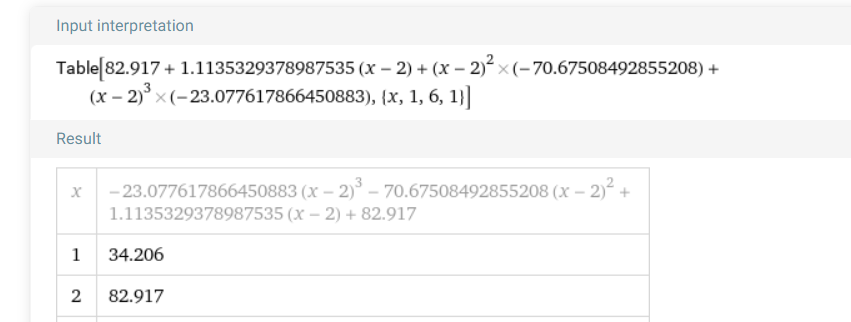


Рисунок 4.6 – Перевірка відповідності сплайну

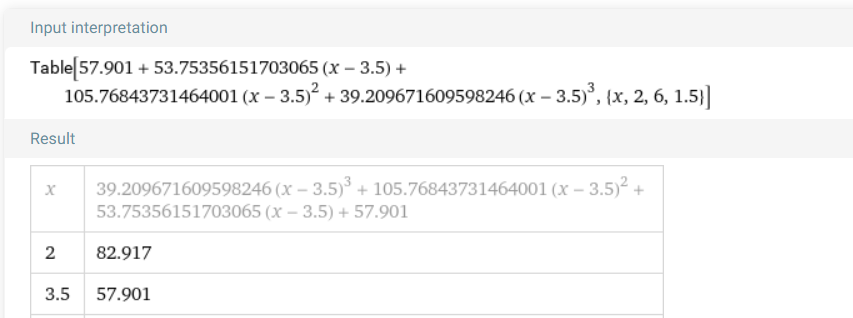


Рисунок 4.7 – Перевірка відповідності сплайну

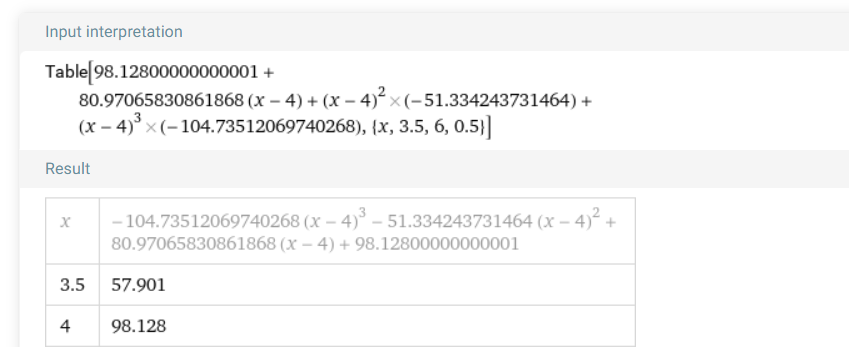


Рисунок 4.8 – Перевірка відповідності сплайну

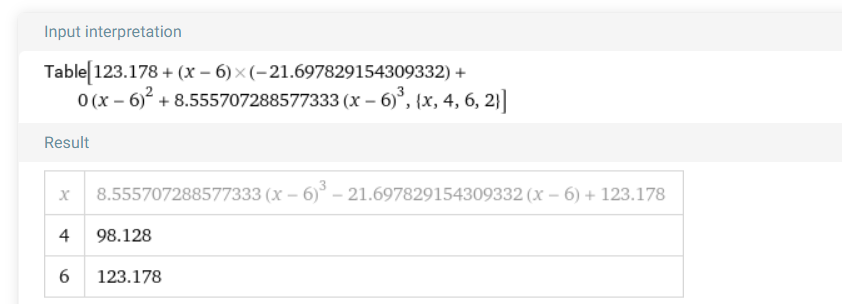


Рисунок 4.9 – Перевірка відповідності сплайну

Отже, значення сплайнів відповідають значенням функції.

**4.4.2 Перевірка відповідності значень перших похідних сусідніх сплайнів**

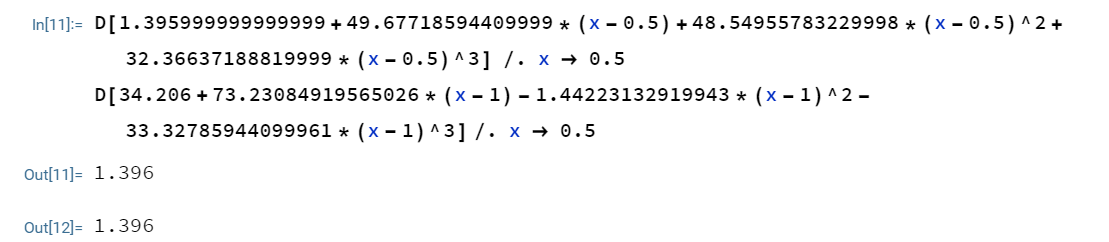


Рисунок 4.10 – Перевірка рівності

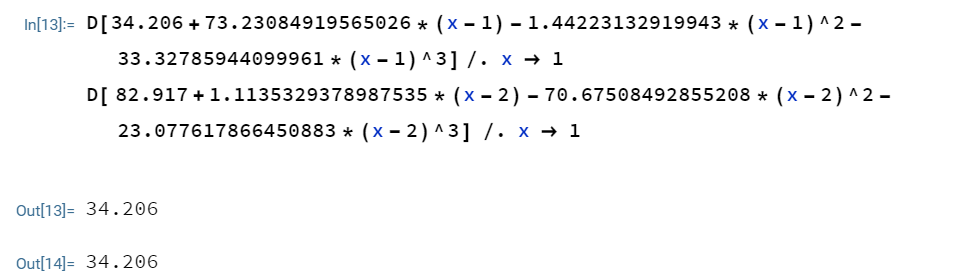


Рисунок 4.11 – Перевірка рівності

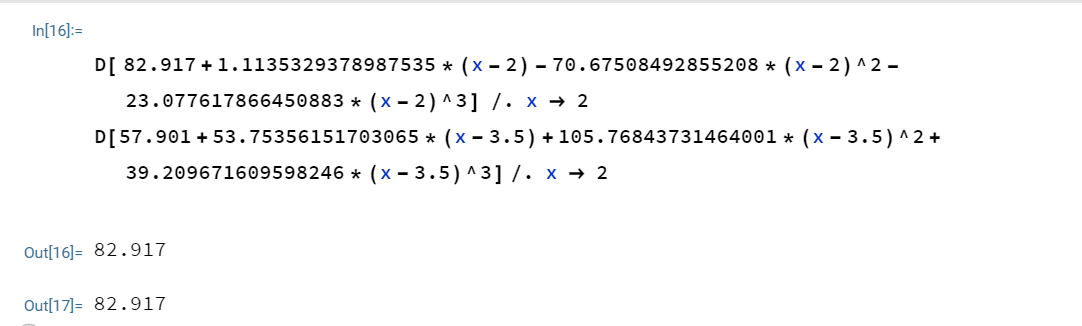


Рисунок 4.12 – Перевірка рівності

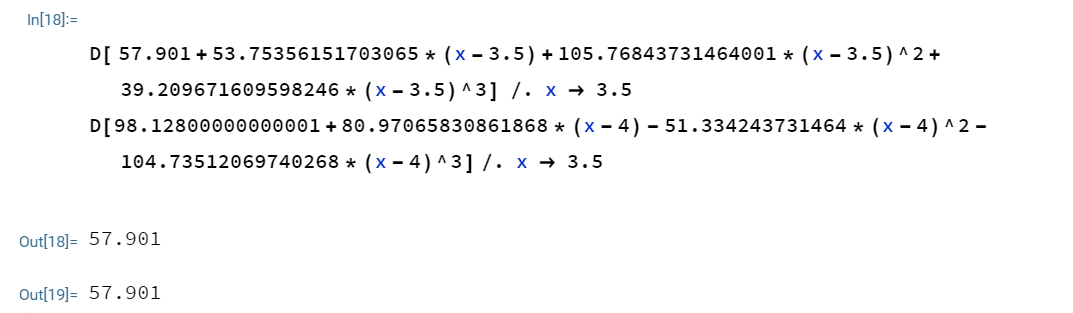


Рисунок 4.13 – Перевірка рівності

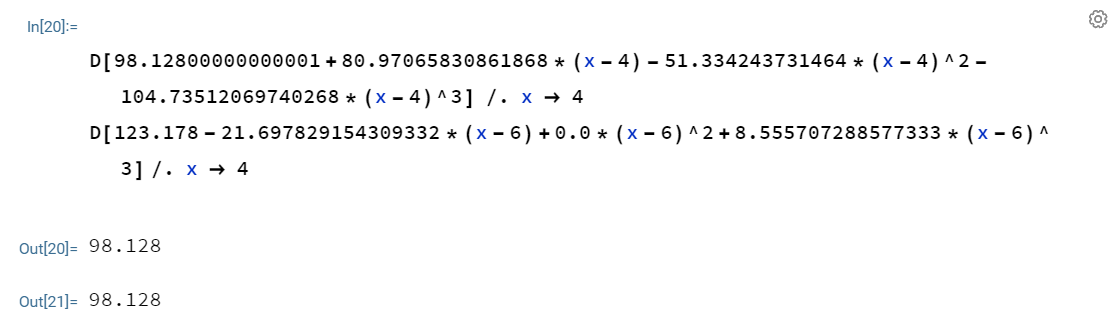


Рисунок 4.14 – Перевірка рівності

**4.4.3 Перевірка відповідності значень других похідних сусідніх сплайнів**

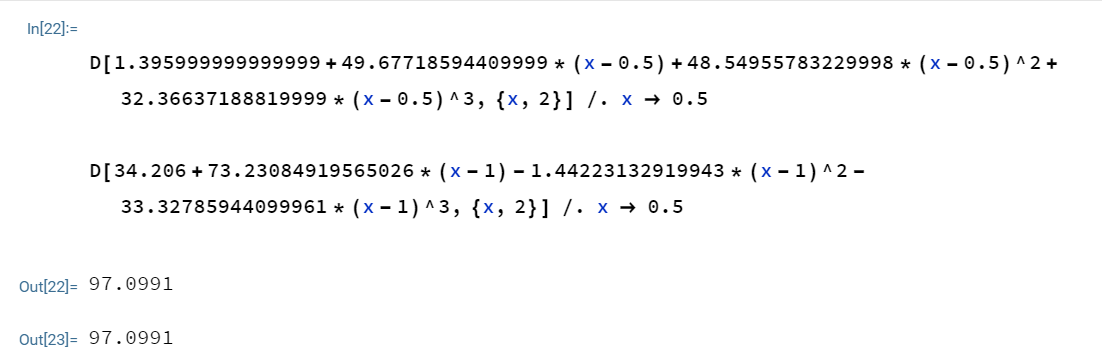


Рисунок 4.15 – Перевірка рівності

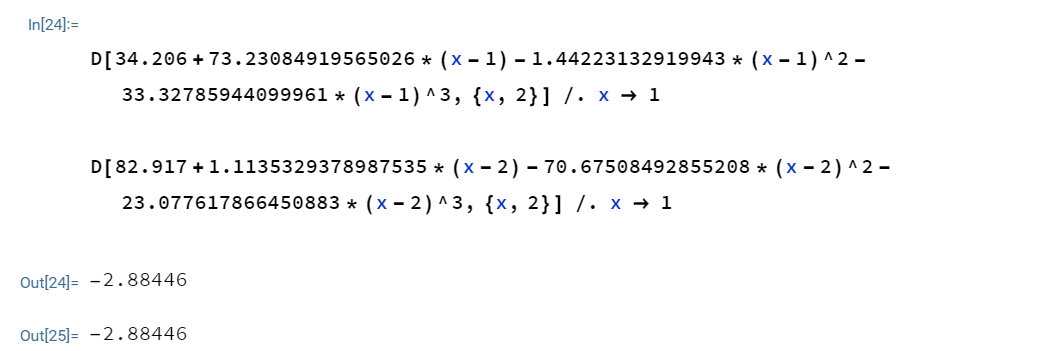


Рисунок 4.16 – Перевірка рівності

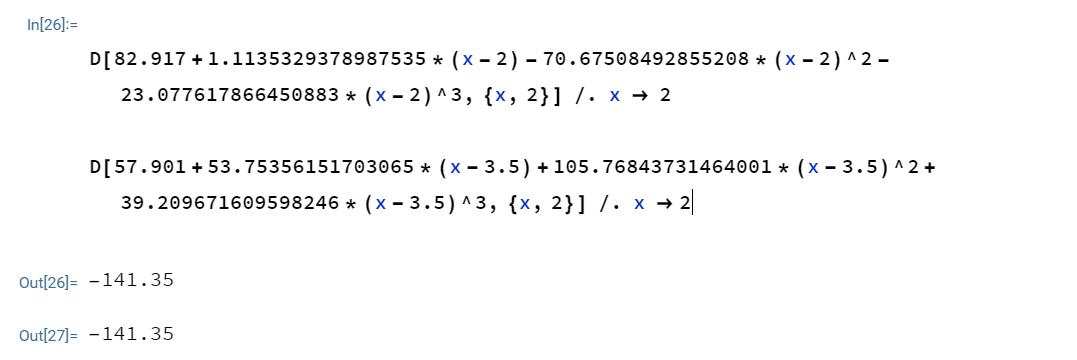


Рисунок 4.17 – Перевірка рівності

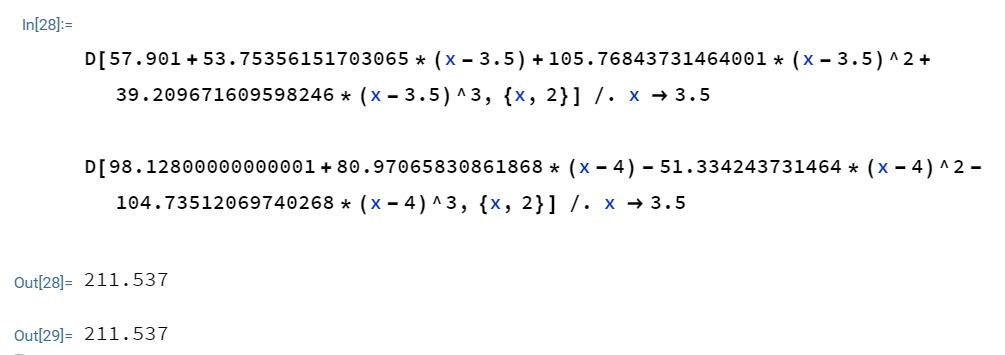


Рисунок 4.18 – Перевірка рівності

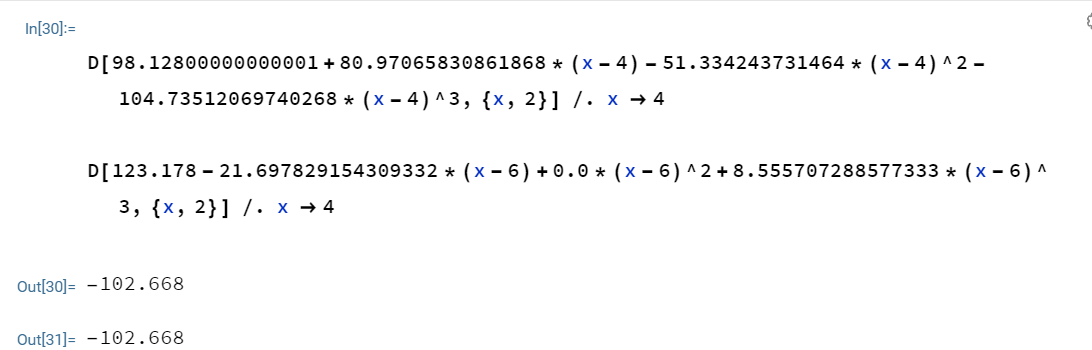


Рисунок 4.19 – Перевірка рівності

**4.4.4 Перевірка відповідності значень других похідних сплайнів крайовим умовам**

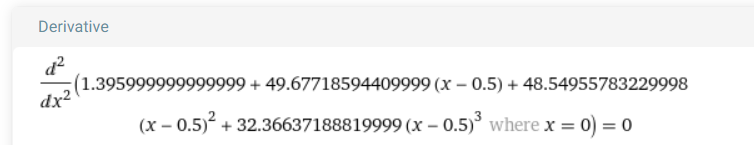


Рисунок 4.20 – Перевірка рівності

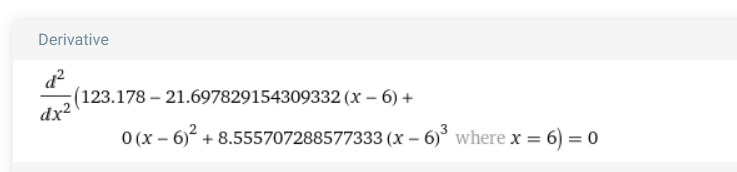


Рисунок 4.21 – Перевірка рівності

Отже, побудовані кубічні сплайни відповідають усім заданим умовам.