**Комп’ютерний практикум №5**

**Інтерполяція**

**Виконав:**

Студент 3 курсу ФТІ

групи ФІ-92

Поночевний Назар Юрійович

Варіант 12

**Завдання:**

Відрізок інтерполяції розбити не менш ніж на 10 вузлів. Використовуючи аналітичне задання функції, визначене варіантом (x^2 cos x), побудувати таблицю значень функції у вузлах на відповідному відрізку інтерполяції [-π/2, π]. Побудувати за таблично заданою функцією:

- інтерполяційний поліном Pn(x) у формі Ньютона або Лагранжа;

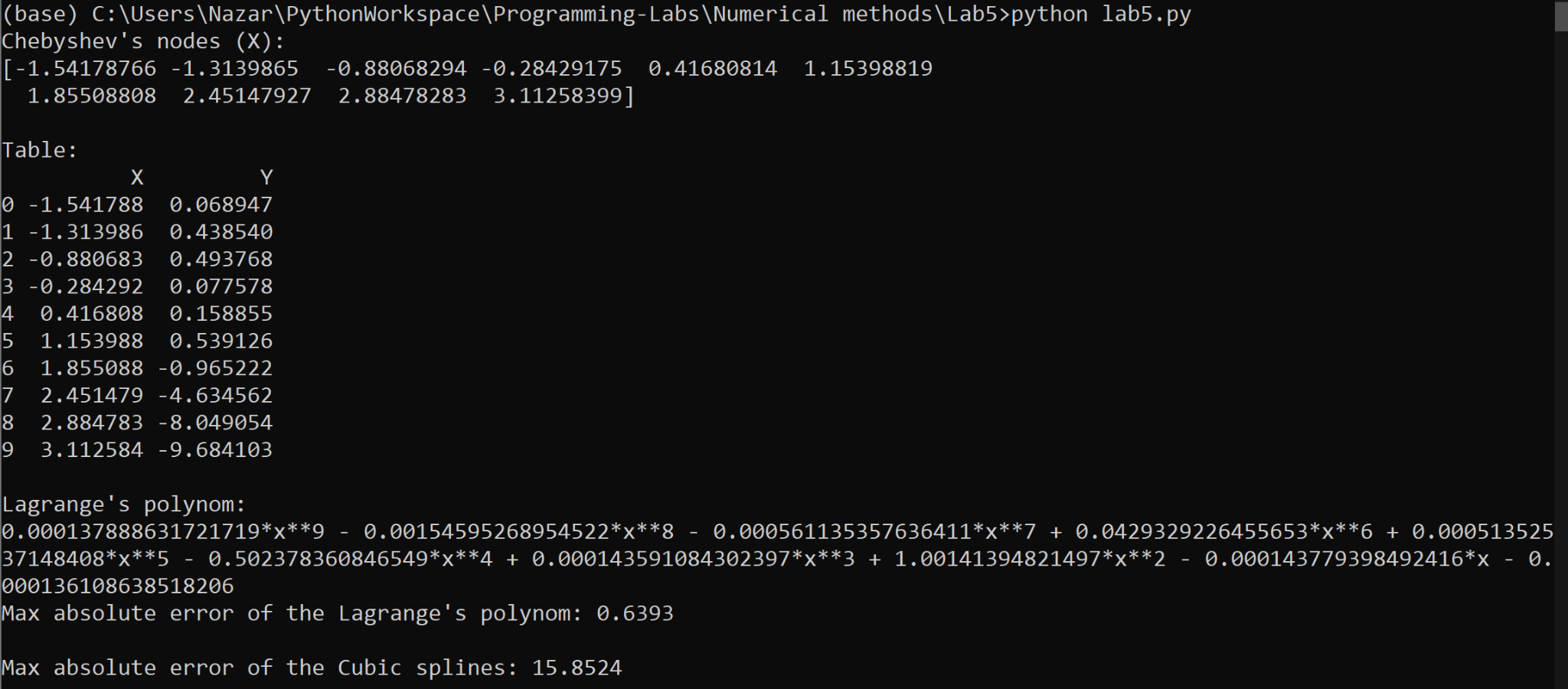
- здійснити інтерполяцію сплайнами (другого чи третього порядку);

- побудувати графік похибки інтерполяції.

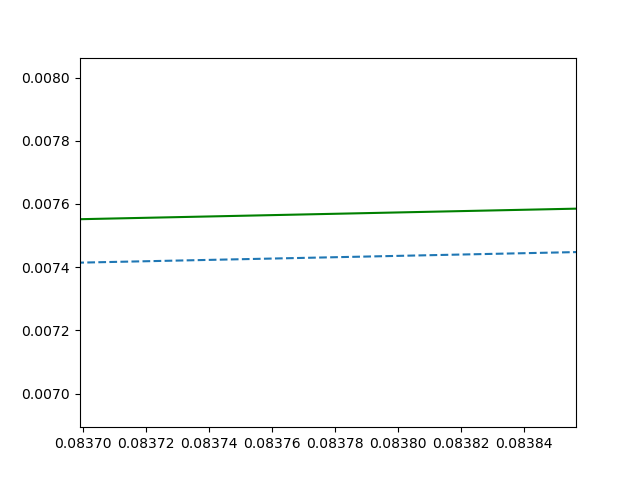
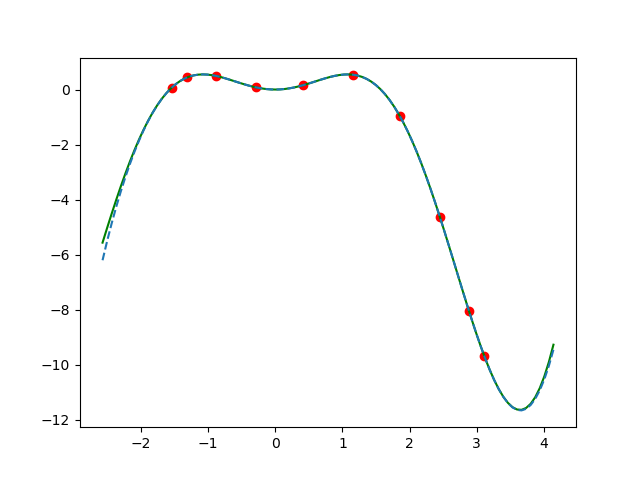
1. Реалізуємо програму

| """ Interpolation  """  import math import numpy as np import sympy as sp import pandas as pd import matplotlib.pyplot as plt   # ------------ Input ------------   def f(x):  return x\*\*2 \* math.cos(x)   A, B = -math.pi / 2, math.pi   # ------------ Code ------------   def chebyshev\_nodes(n, a, b):  nodes = []  for k in range(1, n + 1):  nodes.append((0.5 \* (a + b)) + ((0.5 \* (b - a)) \* math.cos((((2 \* k) - 1) \* math.pi) / (2 \* n))))  return nodes   def build\_lagrange\_polynom(X, y):  x = sp.Symbol('x')  L = 0  for i in range(len(y)):  nominator = y[i]  for j in range(len(y)):  if j != i:  nominator \*= (x - X[j])  denominator = 1  for j in range(len(y)):  if j != i:  denominator \*= (X[i] - X[j])  L += (nominator / denominator)  return sp.simplify(L)   def build\_cubic\_splines(x0, x, y):  x = np.asfarray(x)  y = np.asfarray(y)  size = len(x)  xdiff = np.diff(x)  ydiff = np.diff(y)  Li = np.empty(size)  Li\_1 = np.empty(size-1)  z = np.empty(size)  Li[0] = math.sqrt(2\*xdiff[0])  Li\_1[0] = 0.0  B0 = 0.0  z[0] = B0 / Li[0]   for i in range(1, size-1, 1):  Li\_1[i] = xdiff[i-1] / Li[i-1]  Li[i] = math.sqrt(2\*(xdiff[i-1]+xdiff[i]) - Li\_1[i-1] \* Li\_1[i-1])  Bi = 6\*(ydiff[i]/xdiff[i] - ydiff[i-1]/xdiff[i-1])  z[i] = (Bi - Li\_1[i-1]\*z[i-1])/Li[i]   i = size - 1  Li\_1[i-1] = xdiff[-1] / Li[i-1]  Li[i] = math.sqrt(2\*xdiff[-1] - Li\_1[i-1] \* Li\_1[i-1])  Bi = 0.0  z[i] = (Bi - Li\_1[i-1]\*z[i-1])/Li[i]   i = size-1  z[i] = z[i] / Li[i]  for i in range(size-2, -1, -1):  z[i] = (z[i] - Li\_1[i-1]\*z[i+1])/Li[i]   index = x.searchsorted(x0)  np.clip(index, 1, size-1, index)  xi1, xi0 = x[index], x[index-1]  yi1, yi0 = y[index], y[index-1]  zi1, zi0 = z[index], z[index-1]  hi1 = xi1 - xi0   f0 = zi0/(6\*hi1)\*(xi1-x0)\*\*3 + \  zi1/(6\*hi1)\*(x0-xi0)\*\*3 + \  (yi1/hi1 - zi1\*hi1/6)\*(x0-xi0) + \  (yi0/hi1 - zi0\*hi1/6)\*(xi1-x0)  return f0   def main():  X = np.array(sorted(chebyshev\_nodes(10, A, B)))  print(f"Chebyshev's nodes (X):\n{X}")   Y = np.array([f(x) for x in X])  df = pd.DataFrame({'X': X, 'Y': Y})  print(f"\nTable:\n{df}")    x = sp.Symbol('x')  L = build\_lagrange\_polynom(X, Y)  print(f"\nLagrange's polynom:\n{L}")    ls = np.linspace(A - 1, B + 1, 100)  ls\_Y\_true = np.array([f(i) for i in ls])  ls\_Y\_L = np.array([L.subs(x, i) for i in ls])  r = max(np.abs(ls\_Y\_true - ls\_Y\_L))  print(f"Max absolute error of the Lagrange's polynom: {round(r, 4)}")    plt.scatter(X, Y, color="red")  plt.plot(ls, ls\_Y\_true, color="green")  plt.plot(ls, ls\_Y\_L, linestyle='--')  plt.show()   ls\_Y\_S = build\_cubic\_splines(ls, X, Y)  r = max(np.abs(ls\_Y\_true - ls\_Y\_S))  print(f"\nMax absolute error of the Cubic splines: {round(r, 4)}")   plt.scatter(X, Y, color="red")  plt.plot(ls, ls\_Y\_true, color="green")  plt.plot(ls, ls\_Y\_S, linestyle='--')  plt.show()   if \_\_name\_\_ == "\_\_main\_\_":  main() |
| --- |

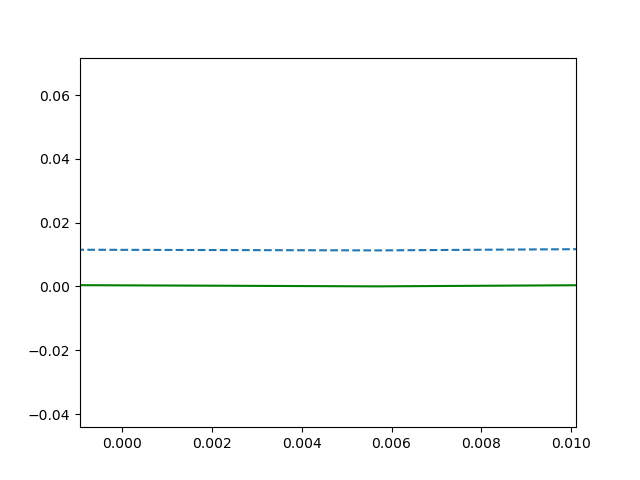
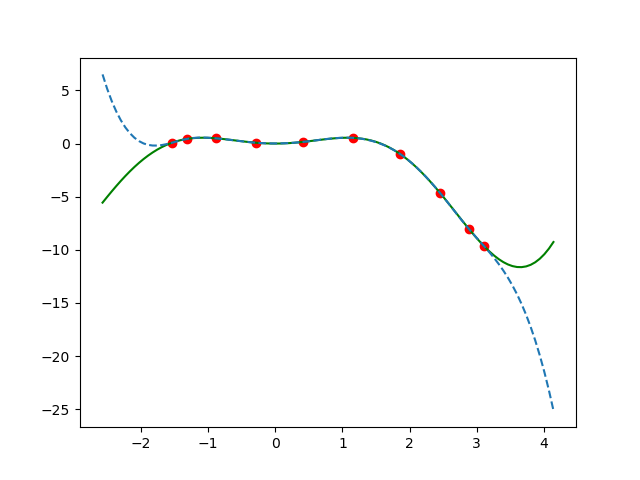
1. Результат



Поліном Лагранжа у звичайному та збільшеному масштабі (червоним вузли, зеленим функція, синім штрихом поліном Лагранжа):



Кубічні сплайни у звичайному та збільшеному масштабі (червоним вузли, зеленим функція, синім штрихом кубічні сплайни):



1. Контрольні запитання

**В чому перевага побудови полінома Ньютона у порівнянні з поліномом Лагранжа?**

Поліном Ньютона легше запрограмувати, бо є можливість обчислювати розділені різниці рекурентним способом.

**Яка кількість вузлів необхідна для побудови інтерполяційного полінома порядку n?**

Число вузлів інтерполяційного полінома має бути на одиницю більше його ступеня.