|  |  |  |  |
| --- | --- | --- | --- |
|  |  | № варіанта | Значення аргумента х |
| 0.45  0.46  0.47  0.48  0.49  0.50  0.51  0.52  0.53  0.54  0.55 | 20.1946  19.6133  18.9425  18.1746  17.3010  16.3123  15.1984  13.9484  12.5508  10.9937  9.2647 | 20 | 0.455 0.533 0.473 0.551 0.461 0.543 |

Пальонка Анастасія,2-8,20 варіант

import numpy as np

from math import factorial

import matplotlib.pyplot as plt

# задані точки

x = np.array([0.45,0.46,0.47,0.48,0.49,0.50,0.51,0.52,0.53,0.54,0.55])

y = np.array([20.1946,19.6133,18.1746,18.1746,17.301,16.3123,15.1984,13.9484,12.5508,10.9937,9.2648])

# перша інтерполяційна формула

def first\_interpolation(x, y, x0):

n = len(x)

f = np.zeros((n, n))

f[:, 0] = y

for j in range(1, n):

for i in range(n - j):

f[i, j] = (f[i + 1, j - 1] - f[i, j - 1]) / (x[i + j] - x[i])

ans = 0

for j in range(n):

prod = f[0, j]

for i in range(j):

prod \*= (x0 - x[i])

ans += prod

return ans

# друга інтерполяційна формула

def second\_interpolation(x, y, x0):

n = len(x)

f = np.zeros((n, n))

f[:, 0] = y

for j in range(1, n):

for i in range(n - j):

f[i, j] = (f[i + 1, j - 1] - f[i, j - 1]) / (x[i + j] - x[i])

ans = f[0, 0]

for j in range(1, n):

prod = f[0, j]

for i in range(j):

prod \*= (x0 - x[i])

ans += prod

return ans

# обчислюємо значення функції в точках x = 0.473 та x = 0.551

x1 = 0.473

x2 = 0.551

y1 = first\_interpolation(x, y, x1)

y2 = second\_interpolation(x, y, x2)

print(f"f({x1}) = {y1}")

print(f"f({x2}) = {y2}")

# будуємо графік інтерполяційної функції

xx = np.linspace(np.min(x), np.max(x), 100)

yy = np.zeros\_like(xx)

for i in range(len(xx)):

yy[i] = second\_interpolation(x, y, xx[i])

plt.plot(x, y, 'o', label='Дані точки')

plt.plot(xx, yy, label='багаточлен Н\*ютона')

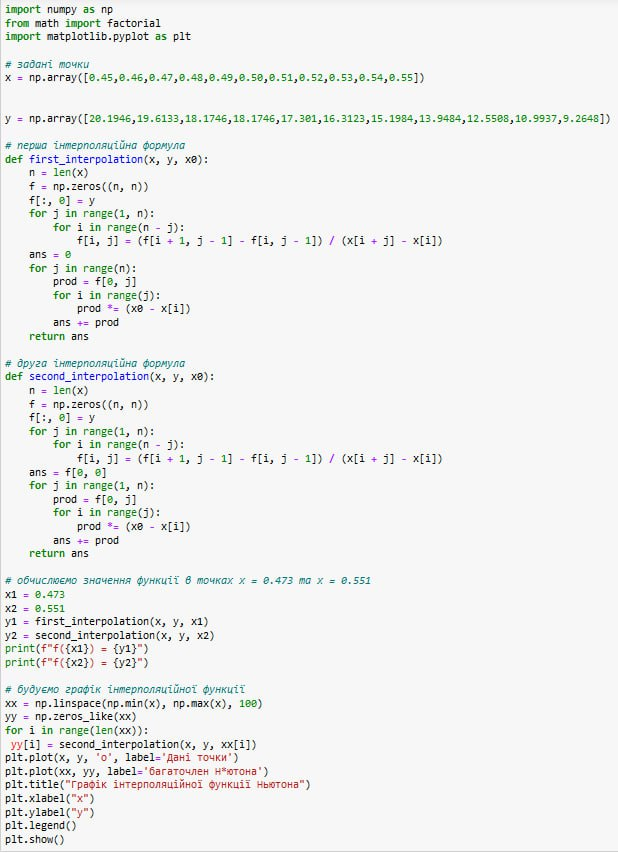
plt.title("Графік інтерполяційної функції Ньютона")

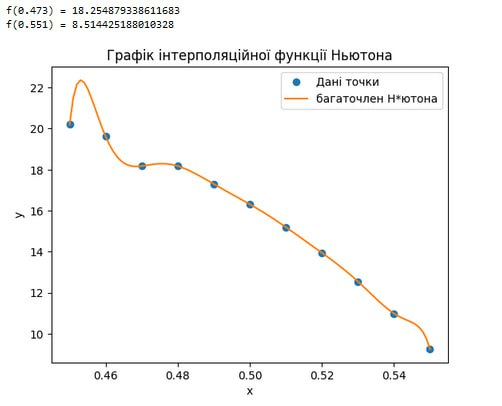
plt.xlabel("x")

plt.ylabel("y")

plt.legend()

plt.show()





https://github.com/nastiapal