**Лабораторна робота №4**

**з чисельних методів**

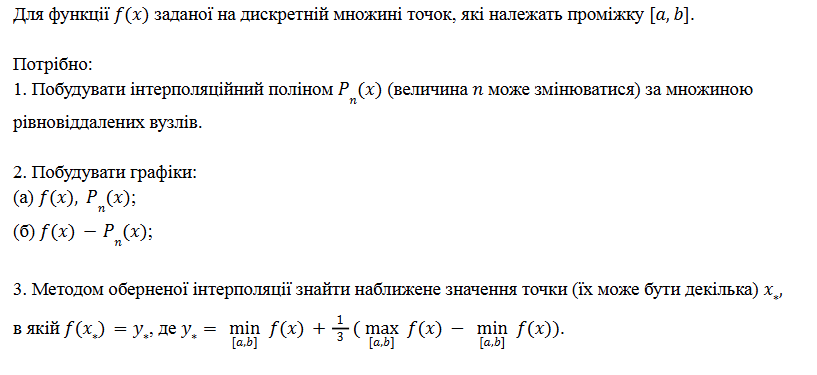
**Варіант №6**

**ТК-31**

**Луковського Максима Юрійовича**

## Постановка задачі

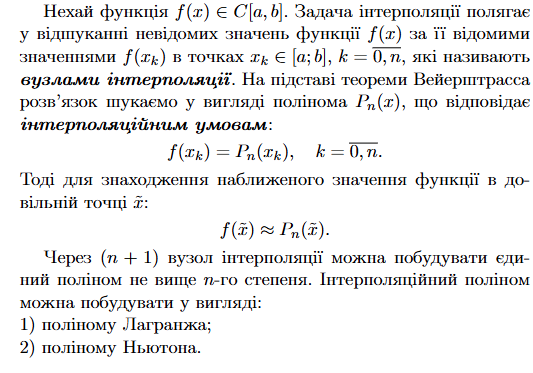
Методом Ньютона:

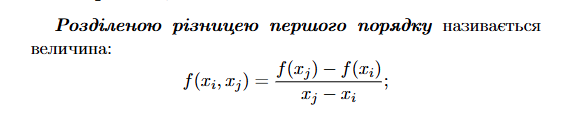


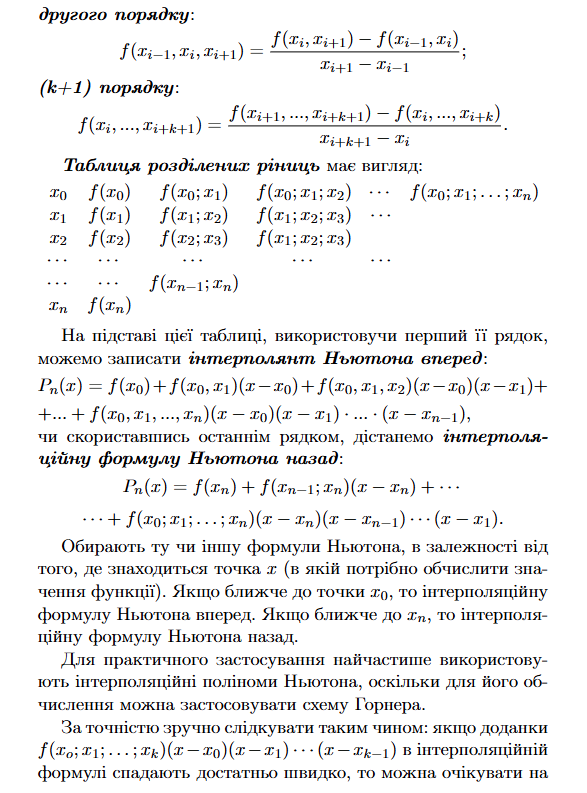


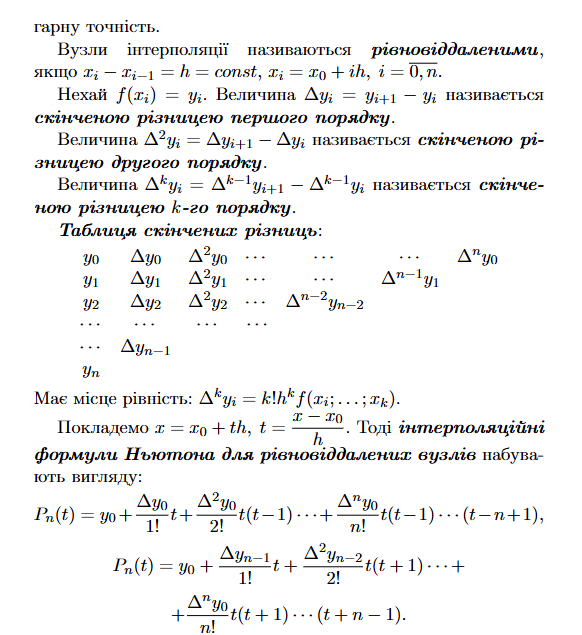
[a,b] = [-1,1]

## Теоретична частина









## Практична частина

Нехай n = 5;

Результат роботи програми:  
Введіть n:

5

x\_k = [-1.0, -0.5, 0.0, 0.5, 1.0]

f(x\_k) = [-0.41751872222600594, 0.14288305071182733, 1.0, 2.1123688797009343, 3.5036799918564934]

h = 1.0

Розділені різниці:

p.p. 1 п.: 1.120804 1.714234 2.224738 2.782622

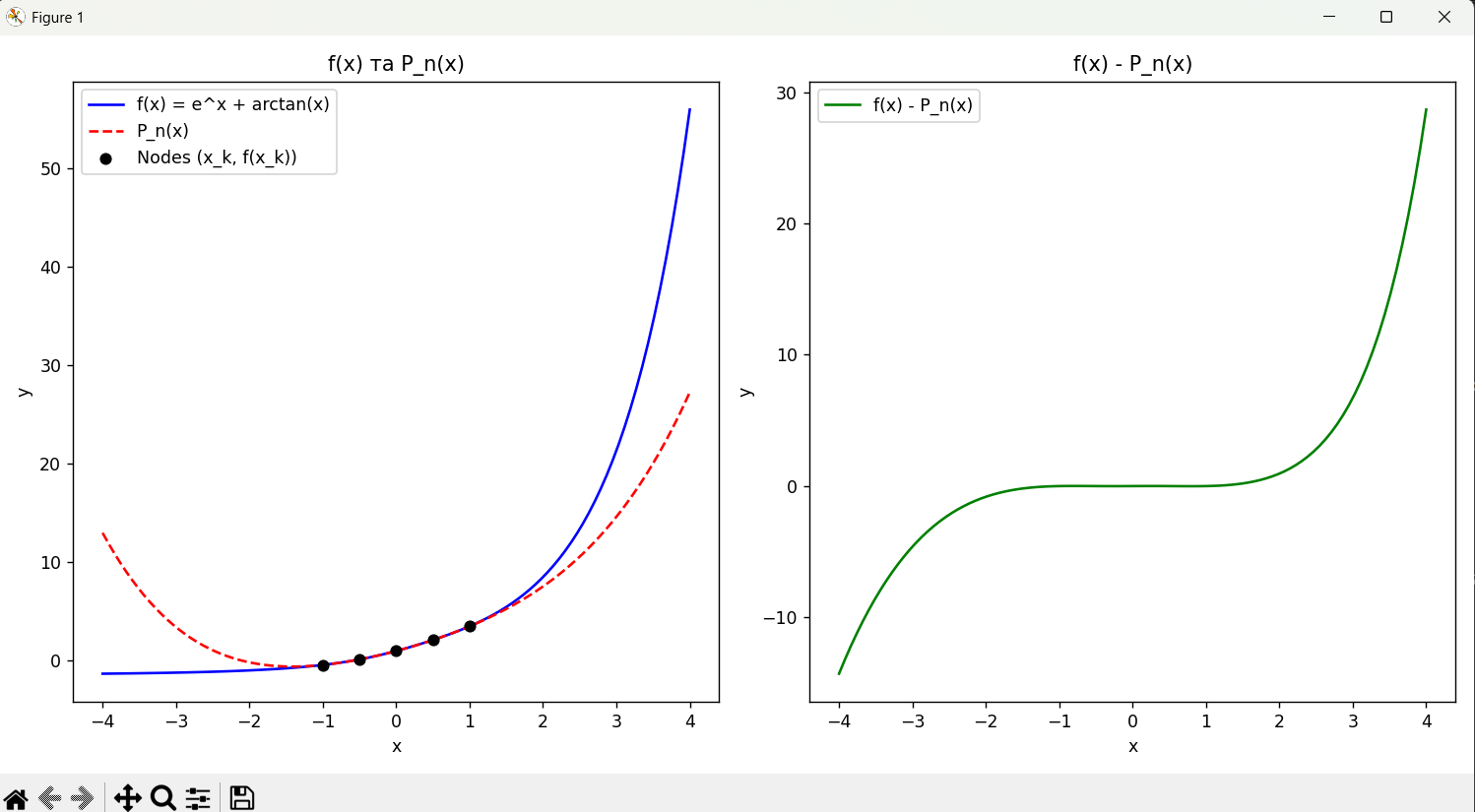
p.p. 2 п.: 0.593430 0.510504 0.557884

p.p. 3 п.: -0.055284 0.031587

p.p. 4 п.: 0.043436

Вигляд поліному Ньютона: -0.417519 + 1.120804\*(x + 1.0) + 0.593430\*(x + 1.0)\*(x + 0.5) + -0.055284\*(x + 0.5)\*(x + 1.0)\*(x - 0.0) + 0.043436\*(x - 0.0)\*(x + 0.5)\*(x + 1.0)\*(x - 0.5)

y\* = 0.889548

Графіки:  


Розділені різниці:

p.p. 1 п.: 0.892217 0.583351 0.449491 0.359373

p.p. 2 п.: -0.217892 -0.067967 -0.035994

p.p. 3 п.: 0.059262 0.009513

p.p. 4 п.: -0.012687

Вигляд поліному Ньютона: -1.000000 + 0.892217\*(x + 0.41751872222600594) + -0.217892\*(x + 0.41751872222600594)\*(x - 0.14288305071182733) + 0.059262\*(x - 0.14288305071182733)\*(x + 0.41751872222600594)\*(x - 1.0) + -0.012687\*(x - 1.0)\*(x - 0.14288305071182733)\*(x + 0.41751872222600594)\*(x - 2.1123688797009343)

3) Результат обчислення: -0.05452322525095825

## Висновки

У ході виконання лабораторної роботи було реалізовано метод Ньютона для побудови інтерполяційного полінома на основі розділених різниць. Поліном Ньютона був застосований до функції f(x)=e^x+arctan⁡(x)f(x) = e^x + \arctan(x)f(x)=ex+arctan(x), визначеної на відрізку [−1,1]. Було отримано точкові значення функції, розраховано розділені різниці, а також побудовано інтерполяційний поліном.

## Додатково:

import numpy as np

import matplotlib.pyplot as plt

def newton\_polynomial(x, x\_nodes, coef):

n = len(coef)

result = coef[0, 0]

product = 1.0

for i in range(1, n):

product \*= (x - x\_nodes[i - 1])

result += coef[0, i] \* product

return result

def divided\_differences(x\_nodes, y\_nodes):

n = len(x\_nodes)

coef = np.zeros([n, n])

coef[:, 0] = y\_nodes

for j in range(1, n):

for i in range(n - j):

coef[i][j] = (coef[i + 1][j - 1] - coef[i][j - 1]) / (x\_nodes[i + j] - x\_nodes[i])

return coef

def newton\_polynomial\_expression(x\_nodes, coef):

n = len(coef)

terms = [f"{coef[0, 0]:.6f}"]

for i in range(1, n):

if x\_nodes[i-1] < 0 :

term = f"(x + { -x\_nodes[i-1]})"

else :

term = f"(x - {x\_nodes[i-1]})"

for j in range(i-1):

if x\_nodes[j] < 0 :

term = f"(x + {-x\_nodes[j]})\*{term}"

else :

term = f"(x - {x\_nodes[j]})\*{term}"

terms.append(f"{coef[0, i]:.6f}\*{term}")

return " + ".join(terms)

a, b = -1, 1

f = lambda x: np.exp(x) + np.arctan(x)

print("Введіть n:")

n=int(input())

x\_nodes = np.linspace(a, b, n)

y\_nodes = f(x\_nodes)

print(f"x\_k = {[float(x) for x in x\_nodes]}")

print(f"f(x\_k) = {[float(f(x)) for x in x\_nodes]}")

h = (b - a)/2

print(f"h = {h}")

coef = divided\_differences(x\_nodes, y\_nodes)

print("\nРозділені різниці:")

for j in range(1, n):

print(f"p.p. {j} п.: " + " ".join([f"{coef[i, j]:.6f}" for i in range(n-j)]))

polynomial\_expression = newton\_polynomial\_expression(x\_nodes, coef)

print(f"1) Вигляд поліному Ньютона: {polynomial\_expression}")

x\_values = np.linspace(a - 3, b + 3, 100)

f\_values = f(x\_values)

P\_values = [newton\_polynomial(x, x\_nodes, coef) for x in x\_values]

difference\_values = f\_values - np.array(P\_values)

y\_min = np.min([f(a), f(b)])

y\_max = np.max([f(a), f(b)])

y\_star = y\_min + (1 / 3) \* (y\_max - y\_min)

print(f"y\* = {y\_star:.6f}")

coef = divided\_differences(y\_nodes, x\_nodes)

print("\nРозділені різниці:")

for j in range(1, n):

print(f"p.p. {j} п.: " + " ".join([f"{coef[i, j]:.6f}" for i in range(n-j)]))

polynomial\_expression = newton\_polynomial\_expression(y\_nodes, coef)

print(f"Вигляд поліному Ньютона: {polynomial\_expression}")

result = newton\_polynomial(y\_star, y\_nodes, coef)

print(f"3) Результат обчислення: {result}")

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)

plt.plot(x\_values, f\_values, label='f(x) = e^x + arctan(x)', color='blue')

plt.plot(x\_values, P\_values, label='P\_n(x)', linestyle='--', color='red')

plt.scatter(x\_nodes, y\_nodes, color='black', zorder=5, label='Nodes (x\_k, f(x\_k))')

plt.title('f(x) та P\_n(x)')

plt.xlabel('x')

plt.ylabel('y')

plt.legend()

plt.subplot(1, 2, 2)

plt.plot(x\_values, difference\_values, label='f(x) - P\_n(x)', color='green')

plt.title('f(x) - P\_n(x)')

plt.xlabel('x')

plt.ylabel('y')

plt.legend()

plt.tight\_layout()

plt.show()