

ЛАБОРАТОРНА РОБОТА №6

Тема: «Наближення функцій. Інтерполяційний багаточлен Лагранжа»

Трубчанінов Андрій Сергійович

ФІТ 2-8

В-29

Код:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate.interpolated import lagrange

x = np.array([-3.0, -1.5, -1.0, 1.5], dtype=float)
y = np.array([-6.0, 4.0, -2.0, 4.0], dtype=float)
x_test = 1.2 # Точка, в якій потрібно обчислити значення

def lagrange_interpolation(x, y, x_test):
    n = len(x)
    p = np.zeros(n)
    for i in range(n):
        p_i = 1
        for j in range(n):
            if i != j:
                p_i *= (x_test - x[j]) / (x[i] - x[j])
        p[i] = p_i
    return np.dot(y, p)

f_interp = lagrange_interpolation(x, y, x_test)

print("Значення функції у точці x_test =", f_interp.round(4))

xnew = np.linspace(np.min(x), np.max(x), 100)
ynew = [lagrange_interpolation(x, y, i) for i in xnew]
```

```
        if i != j:
            p_i *= (x_test - x[j]) / (x[i] - x[j])
        p[i] = p_i
    return np.dot(y, p)

f_interp = lagrange_interpolation(x, y, x_test)

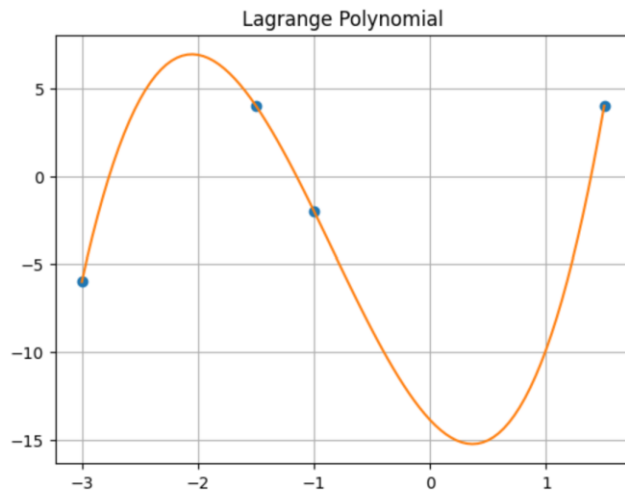
print("Значення функції у точці x_test =", f_interp.round(4))

xnew = np.linspace(np.min(x), np.max(x), 100)
ynew = [lagrange_interpolation(x, y, i) for i in xnew]

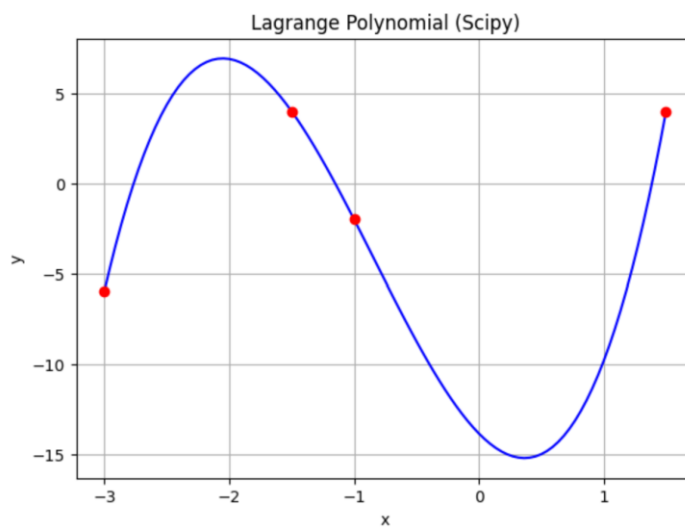
plt.plot(x, y, "o", xnew, ynew)
plt.title("Lagrange Polynomial")
plt.grid(True)
plt.show()

f_scipy = lagrange(x, y)
fig = plt.figure(figsize=(7, 5))
plt.plot(xnew, f_scipy(xnew), "b", x, y, "ro")
plt.title("Lagrange Polynomial (Scipy)")
plt.grid()
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```

Значення функції у точці $x_{\text{test}} = -5.4848$



Lagrange Polynomial (Scipy)



Код зі скрінів:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate.interpolate import lagrange

x = np.array([-3.0, -1.5, -1.0, 1.5], dtype=float)
y = np.array([-6.0, 4.0, -2.0, 4.0], dtype=float)
x_test = 1.2 # Точка, в якій потрібно обчислити значення

def lagrange_interpolation(x, y, x_test):
    n = len(x)
```

```

p = np.zeros(n)
for i in range(n):
    p_i = 1
    for j in range(n):
        if i != j:
            p_i *= (x_test - x[j]) / (x[i] - x[j])
    p[i] = p_i
return np.dot(y, p)

```

```

f_interp = lagrange_interpolation(x, y, x_test)

```

```

print("Значення функції у точці x_test =", f_interp.round(4))

```

```

xnew = np.linspace(np.min(x), np.max(x), 100)
ynew = [lagrange_interpolation(x, y, i) for i in xnew]

```

```

plt.plot(x, y, "o", xnew, ynew)
plt.title("Lagrange Polynomial")
plt.grid(True)
plt.show()

```

```

f_scipy = lagrange(x, y)
fig = plt.figure(figsize=(7, 5))
plt.plot(xnew, f_scipy(xnew), "b", x, y, "ro")
plt.title("Lagrange Polynomial (Scipy)")
plt.grid()
plt.xlabel("x")

```

```
plt.ylabel("y")
```

```
plt.show()
```