

# Zadanie Numeryczne 07

Mateusz Wojtyna

## 1. Wstęp

Należało znaleźć z dokładnością do czterech miejsc po przecinku wartości współczynników wielomianu interpolacyjnego opartego na funkcji przedstawionej w tabelce, oraz należało zrobić wykres tego wielomianu w przedziale  $-2 \leq x \leq 1.25$ .

$x_i$	$y_i = f(x_i)$
-1.00	6.000000000000000
-0.75	3.04034423828125
-0.50	1.74218750000000
-0.25	1.26361083984375
0.25	0.75982666015625
0.50	0.63281250000000
0.75	0.85809326171875
1.00	2.000000000000000

Wykorzystano interpolację Lagrange'a do wyznaczenia tego wielomianu.

## 2. Opis

### 2.1. Interpolacja Lagrange'a

Wielomian interpolacyjny Lagrange'a dla funkcji o  $n$  węzłach dany jest następującym wzorem:

$$P(x) = \sum_{i=1}^n l_i(x)y_i$$

gdzie  $l_i(x)$  jest wielomianem bazowym Lagrange'a:

$$l_i(x) = \prod_{\substack{j=1 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}$$

## 2.2. Wyznaczenie współczynników wielomianu interpolacyjnego

Możemy wyznaczyć współczynnik  $a_0$  obliczając:

$$a_0 = P(0) = \sum_{i=1}^n l_i(0)y_i$$

Zajmie to  $\mathcal{O}(n^2)$  czasu, ponieważ dla każdego  $i$  musimy policzyć jeszcze  $l_i(0)$  co zajmuje  $\mathcal{O}(n)$  czasu. Po wyliczeniu każdego  $l_i(0)$  możemy zapisać go w pamięci do wykorzystania w następnych współczynnikach.

Współczynniki  $a_{k>0}$  można obliczyć z następującego wzoru:

$$a_k = \sum_{i=1}^n l_i(0)y_i^{(k)}$$

gdzie:

$$y_i^{(k)} = \begin{cases} y_i & k = 0 \\ \frac{y_i^{(k-1)} - a_{k-1}}{x_i} & k \in \{1, \dots, n-1\} \end{cases}$$

Wyliczamy je w czasie  $\mathcal{O}(n)$ , ponieważ mamy już zapamiętany  $l_i(0)$ , a  $y_i^{(k)}$  możemy wyliczyć w czasie stałym dla każdego  $i$ .

## 3. Kod

Program napisano w Pythonie z użyciem pakietu *NumPy* oraz *Matplotlib*.

```
import numpy as np
import matplotlib.pyplot as plt
from numpy.typing import NDArray

# Dla przejrzystości kodu
array = NDArray[np.float64]
vector = NDArray[np.float64]
matrix = NDArray[np.float64]

def lagrange_basis(j: int, x: float, nodes: vector) -> float:
    N = len(nodes)
    nominator = 1.0
    denominator = 1.0

    for i in range(0, j):
        nominator *= x - nodes[i]
        denominator *= nodes[j] - nodes[i]
    for i in range(j + 1, N):
        nominator *= x - nodes[i]
        denominator *= nodes[j] - nodes[i]

    return nominator / denominator
```

```

def lagrange_polynomial(nodes: vector, values: vector) -> vector:
    """
    Zwraca współczynniki wielomianu interpolacyjnego Lagrange'a
    """
    N = len(nodes)
    a = np.zeros(N, dtype=np.float64)
    l = np.zeros(N, dtype=np.float64)
    values = values.copy()

    # k = 0
    for j in range(N):
        l[j] = lagrange_basis(j, 0, nodes)
        a[0] += l[j] * values[j]

    # k > 0
    for k in range(1, N):
        for j in range(N):
            values[j] = (values[j] - a[k - 1]) / nodes[j]
        a[k] += l[j] * values[j]

    return a

def P(x: float, a: vector):
    N = len(a)

    y = 0
    tmp = 1
    for i in range(N):
        y += a[i] * tmp
        tmp *= x

    return y

def main():
    np.set_printoptions(suppress=True)

    nodes = np.array([-1, -0.75, -0.5, -0.25, 0.25, 0.5, 0.75, 1], dtype=np.
                     float64)
    values = np.array(
        [
            6,
            3.04034423828125,
            1.7421875,
            1.26361083984375,
            0.75982666015625,
            0.6328125,
            0.85809326171875,
            2,
        ],
        dtype=np.float64,
    )

```

```

)
a: vector = lagrange_polynomial(nodes, values)
print(a)

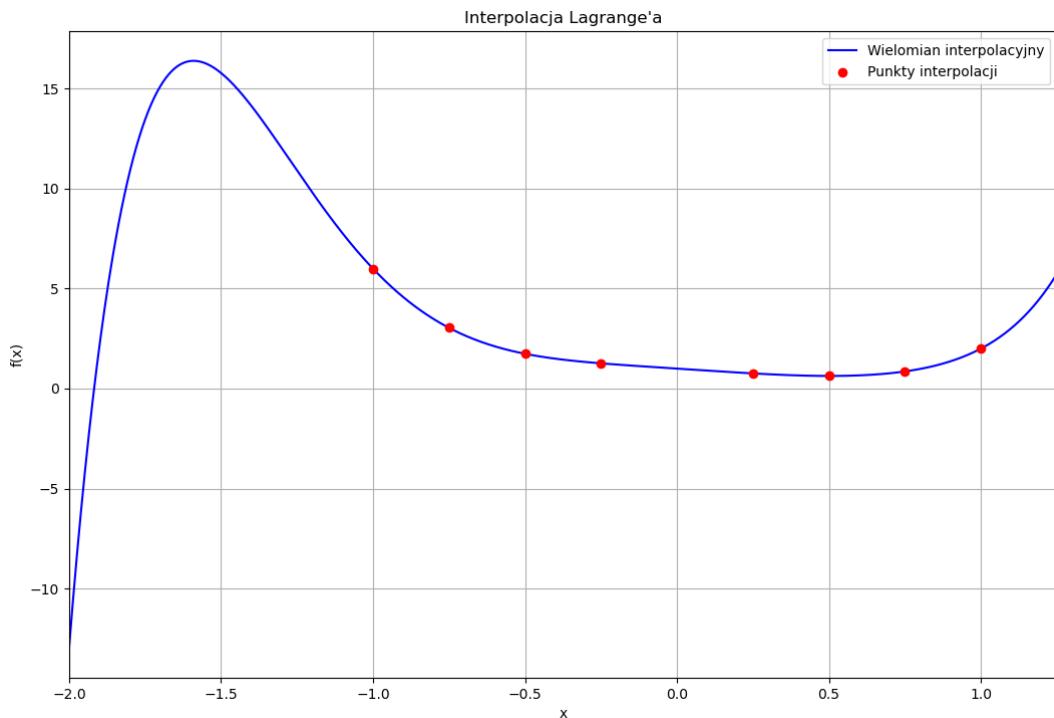
x_plot = np.linspace(-2, 1.25, 1000)
y_plot = np.array([P(x, a) for x in x_plot])
plt.plot(x_plot, y_plot, label="Wielomian interpolacyjny", color="blue")
plt.scatter(nodes, values, color="red", label="Punkty interpolacji",
            zorder=5)
plt.xlabel("x")
plt.ylabel("f(x)")
plt.title("Interpolacja Lagrange'a")
plt.grid(True)
plt.legend()
plt.xlim(-2, 1.25)
plt.show()

if __name__ == "__main__":
    main()

```

## 4. Wyniki

$$a_0 = 1, \quad a_1 = -1, \quad a_2 = 0, \quad a_3 = 0, \quad a_4 = 3, \quad a_5 = -2, \quad a_6 = 0, \quad a_7 = 1$$



Rysunek 1: Wielomian interpolacyjny funkcji  $f$