

Sprawozdanie z laboratorium 4''

Hubert Miklas

08-04-2025

1 Wstęp

Tematem laboratorium 4'' była kontynuacja aproksymacji i interpolacji.

2 Zadania

Zadanie 1

Obliczyć 4–5 współczynników aproksymacji funkcji $|x|$ w przedziale $[-1, 1]$ wielomianami Czebyszewa. Obliczyć błąd aproksymacji w równoodległych punktach z odstępem 0.2, poczynając od punktu -0.8 . Narysować na papierze kratkowanym funkcję aproksymowaną i aproksymującą.

Zadanie 2

W eksperymencie uzyskano następujące dane dwuwymiarowe (x, y) :

$$(0.0, 0.0), \quad (0.5, 1.6), \quad (1.0, 2.0), \quad (6.0, 2.0), \quad (7.0, 1.5), \quad (9.0, 0.0).$$

Chcielibyśmy wykonać interpolację danych gładką krzywą celem znalezienia rozsądnych wartości y dla wartości x pomiędzy punktami, w których wykonano pomiary.

- Proszę określić wielomian stopnia piątego, który interpoluje podane dane i narysować wykres wielomianu w zakresie od 0 do 9.
- Proszę obliczyć funkcję sklejaną stopnia trzeciego, która interpoluje podane dane i narysować wykres w tymże zakresie.
- Proszę obliczyć funkcję interpolującą dla tychże danych przy pomocy wielomianów Czebyszewa i zrobić wykres w tymże zakresie.
- Która z wyżej obliczonych funkcji interpolujących daje lepsze wyniki pomiędzy zadanymi punktami? Proszę objaśnić, dlaczego ww. funkcje tak się zachowują.

Zadanie 1

Mamy kolejne wielomiany Czebyszewa [2]:

$$T_0(x) = 1$$

$$T_1(x) = x$$

$$T_2(x) = 2x^2 - 1$$

$$T_3(x) = 4x^3 - 3x$$

$$T_4(x) = 8x^4 - 8x^2 + 1$$

Na podstawie wzoru aproksymacyjnego Czebyszewa [2, 1], możemy obliczyć całki:
Funkcję $f(x)$ aproksymujemy szeregiem:

$$f(x) = \sum_{i=0}^n c_i T_i(x) \quad (1)$$

Gdzie współczynniki c_i można obliczyć ze wzoru:

$$c_0 = \frac{1}{\pi} \int_{-1}^1 \frac{f(x)T_0(x)}{\sqrt{1-x^2}} dx.$$

$$c_k = \frac{2}{\pi} \int_{-1}^1 \frac{f(x)T_k(x)}{\sqrt{1-x^2}} dx \quad \text{dla } k > 0,$$

Możemy zatem obliczyć te całki:

$$c_0 = \frac{1}{\pi} \int_{-1}^1 \frac{f(x) \cdot 1}{\sqrt{1-x^2}} dx,$$

$$c_1 = \frac{2}{\pi} \int_{-1}^1 \frac{f(x) \cdot x}{\sqrt{1-x^2}} dx,$$

$$c_2 = \frac{2}{\pi} \int_{-1}^1 \frac{f(x) \cdot (2x^2 - 1)}{\sqrt{1-x^2}} dx,$$

$$c_3 = \frac{2}{\pi} \int_{-1}^1 \frac{f(x) \cdot (4x^3 - 3x)}{\sqrt{1-x^2}} dx,$$

$$c_4 = \frac{2}{\pi} \int_{-1}^1 \frac{f(x) \cdot (8x^4 - 8x^2 + 1)}{\sqrt{1-x^2}} dx.$$

Obliczając wartości tych współczynników, otrzymujemy:

$$c_0 = \frac{2}{\pi}$$

$$c_1 = 0$$

$$c_2 = \frac{4}{3\pi}$$

$$c_3 = 0$$

$$c_4 = -\frac{4}{15\pi}$$

Najłatwiej jest policzyć całki, w których całkowana funkcja jest nieparzysta, ponieważ wynik takiego całkowania to 0. Podstawiając to do wzoru obliczone współczynniki: **(1)**:

$$f(x) \approx \frac{2}{\pi} + \frac{4}{3\pi} \cdot (2x^2 - 1) - \frac{4}{15\pi} \cdot (8x^4 - 8x^2 + 1) = \frac{2}{5\pi} + \frac{24x^2}{5\pi} - \frac{32x^4}{15\pi} \quad (2)$$

Wartość x	Wartość $ x $	Aproksymacja	Błąd względny
-0.800000	0.800000	0.827029	0.027029
-0.600000	0.600000	0.589357	0.010643
-0.400000	0.400000	0.354402	0.045598
-0.200000	0.200000	0.187353	0.012647
0.000000	0.000000	0.127324	0.127324
0.200000	0.200000	0.187353	0.012647
0.400000	0.400000	0.354402	0.045598
0.600000	0.600000	0.589357	0.010643
0.800000	0.800000	0.827029	0.027029
1.000000	1.000000	0.976150	0.023850

Table 1: Tabela wartości funkcji oryginalnej i aproksymacji błędów

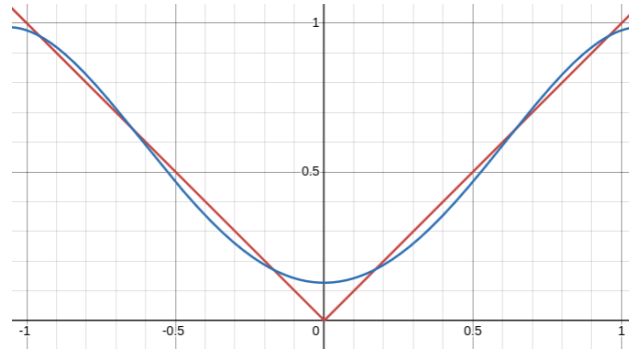


Figure 1: Wykres obrazujący aproksymację

Wnioski

Jak widać na wykresie 1 i w tabeli 1, aproksymacja jest naprawdę bliska oryginalnej funkcji. Jedynie w punkcie $x = 0$, gdzie błąd względny wynosi 10%, występuje znaczna różnica między funkcją a jej aproksymacją.

Zadanie 2

Dane:

Mamy dane:

$$(0.0, 0.0), \quad (0.5, 1.6), \quad (1.0, 2.0), \quad (6.0, 2.0), \quad (7.0, 1.5), \quad (9.0, 0.0).$$

(a) Wielomian interpolacyjny stopnia 5

Wiedząc, że dla $n + 1$ punktów (tutaj $n + 1 = 6$) istnieje jedyny wielomian stopnia co najwyżej $n = 5$, szukamy $P_5(x)$ w postaci:

$$P_5(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5.$$

Współczynniki a_0, a_1, \dots, a_5 wyznaczamy z układu równań wynikającego z warunku:

$$P_5(x_i) = y_i, \quad i = 1, \dots, 6.$$

Przykładowo:

$$\begin{cases} a_0 + a_1 \cdot 0.0 + a_2 \cdot 0.0^2 + \dots + a_5 \cdot 0.0^5 = 0.0, \\ a_0 + a_1 \cdot 0.5 + a_2 \cdot (0.5)^2 + \dots + a_5 \cdot (0.5)^5 = 1.6, \\ a_0 + a_1 \cdot 1.0 + a_2 \cdot (1.0)^2 + \dots + a_5 \cdot (1.0)^5 = 2.0, \\ \vdots \\ a_0 + a_1 \cdot 9.0 + a_2 \cdot (9.0)^2 + \dots + a_5 \cdot (9.0)^5 = 0.0. \end{cases}$$

Do rozwiązania tego układu równań skorzystałem z bibliotecznej funkcji `numpy.linalg.solve`.

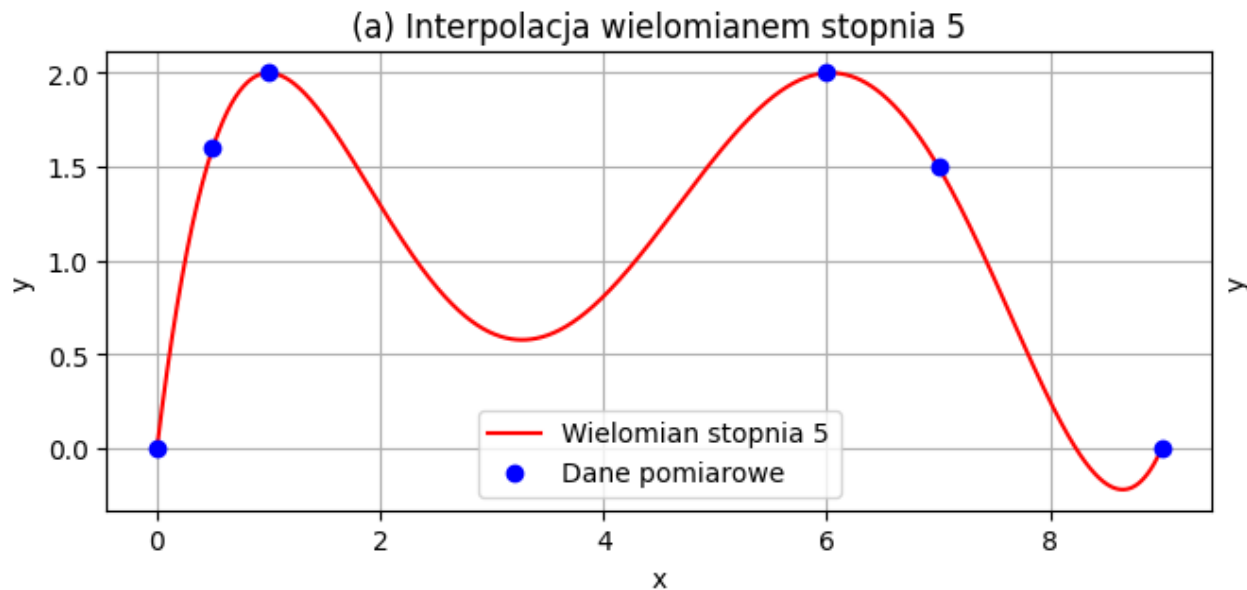


Figure 2: Interpolacja punktów wyniku eksperymentu wielomianem piątego stopnia

(b) Funkcja sklejana stopnia trzeciego (spline)

Kolejnym podejściem jest interpolacja przy użyciu sklejań kubicznych. [4, 3] W metodzie tej szukamy funkcji $S(x)$ złożonej z kawałków wielomianów trzecich:

$$S(x) = S_i(x), \quad x \in [x_i, x_{i+1}], \quad i = 1, \dots, 5,$$

gdzie każdy $S_i(x)$ ma postać:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3.$$

Warunki na sklejenie:

1. Warunki interpolacji:

$$S_i(x_i) = y_i, \quad S_i(x_{i+1}) = y_{i+1}.$$

2. Ciągłość pierwszej i drugiej pochodnej w punktach x_2, \dots, x_5 :

$$S'_i(x_{i+1}) = S'_{i+1}(x_{i+1}), \quad S''_i(x_{i+1}) = S''_{i+1}(x_{i+1}).$$

3. Dodatkowo przyjmujemy warunki brzegowe. Dla naturalnego splajn'u przyjmujemy:

$$S_1''(x_1) = 0, \quad S_5''(x_6) = 0.$$

Rozwiązując układ równań dla współczynników a_i, b_i, c_i, d_i , uzyskujemy funkcję sklejaną $S(x)$, która gładko interpoluje dane.

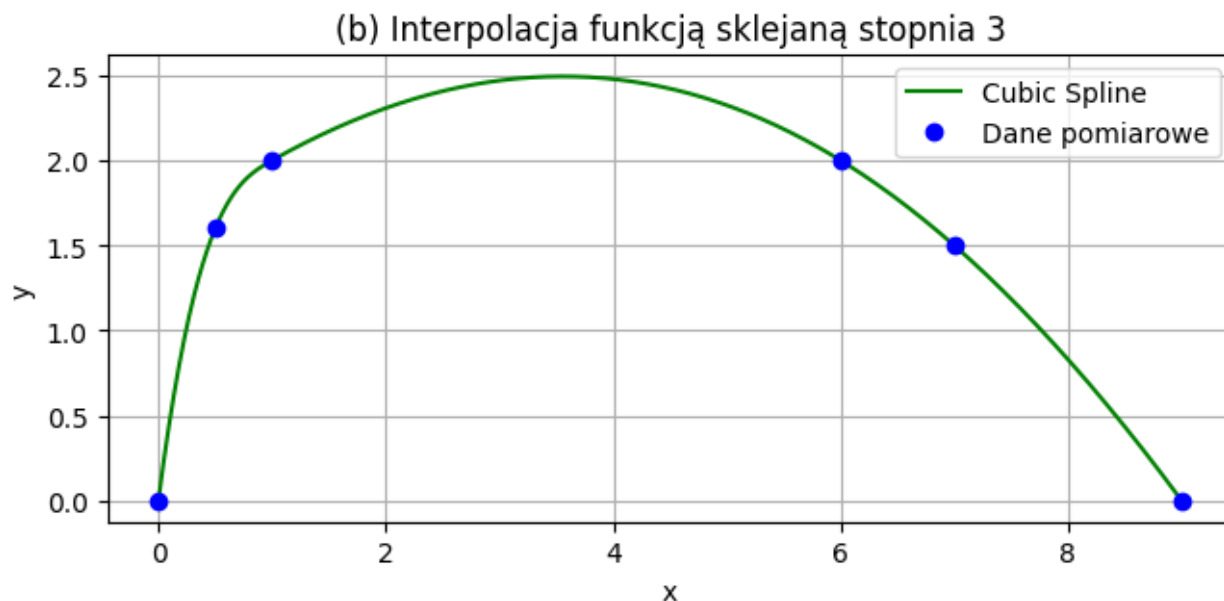


Figure 3: Spline wielomianami trzeciego stopnia

(c) Interpolacja przy pomocy wielomianów Czebyszewa

Interpolacja przy użyciu wielomianów Czebyszewa polega na aproksymacji funkcji (lub danych) w bazie wielomianów Czebyszewa $T_k(x)$. Rozważamy rozwinięcie funkcji $f(x)$ (lub interpolanta $P(x)$) w szereg:

$$P(x) = \sum_{k=0}^n c_k T_k(x),$$

gdzie współczynniki c_k są wyznaczone na podstawie równań wynikających z warunków interpolacji.

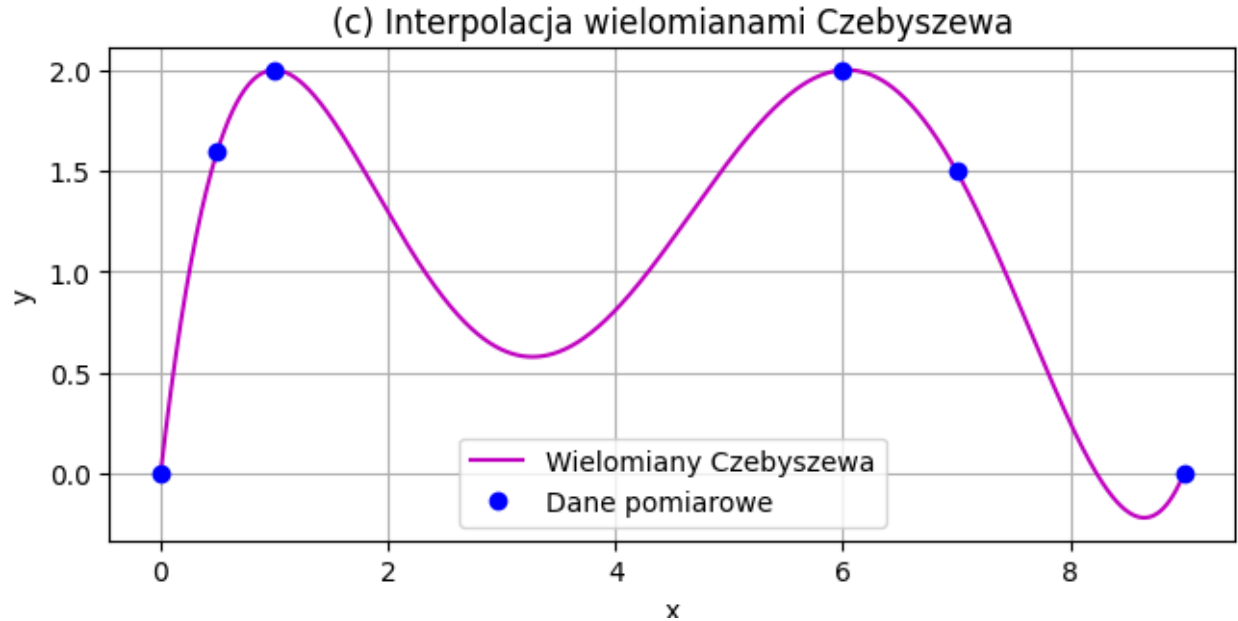


Figure 4: Wykres obrazujący interpolację punktów metodą Czebyszewa

(d) Porównanie wyników interpolacji

Wielomian stopnia 5: Interpolant ten dokładnie przechodzi przez wszystkie punkty. Jednakże, przy nierównomiernym rozkładzie węzłów, może wystąpić zjawisko oscylacyjne (efekt Rungego), zwłaszcza na krańcach przedziału, co może skutkować nienaturalnym przebiegiem między punktami.

Spline wielomianu stopnia trzeciego: Sklejana funkcja interpolacyjna jest składana kawałkami – dla każdego przedziału $[x_i, x_{i+1}]$ mamy oddzielny wielomian. Dzięki warunkom ciągłości pierwszej i drugiej pochodnej uzyskujemy gładką krzywą, która zazwyczaj nie wykazuje widocznych oscylacji między punktami. Zaletą tego podejścia jest lokalny charakter – zmiana wartości w jednym przedziale nie wpływa na całą funkcję.

Interpolacja z bazą wielomianów Czebyszewa: Gdybyśmy mieli możliwość wyboru węzłów, metoda Czebyszewa pozwala na konstrukcję interpolanta o mniejszych oscylacjach dzięki optymalnemu rozmieszczeniu punktów (węzłów Czebyszewa minimalizują tzw. stałą Lebesgue'a). W naszym zadaniu, o ile dokonamy odpowiedniej transformacji, możemy uzyskać interpolanta, który jest numerycznie stabilniejszy niż bezpośredni wielomian interpolacyjny wyższego stopnia, ale – przy zadanych eksperymentalnych punktach – niekoniecznie da znaczącą poprawę względem splajnów.

Wnioski: Najlepszym podejściem dla danych eksperymentalnych jest zazwyczaj interpolacja przy użyciu spline'ów (punkt (b)), ponieważ:

- **Lokalność:** Sklejany interpolant (spline) jest tworzony lokalnie – błąd w jednym przedziale nie przenosi się na inne.
- **Brak oscylacji:** W przeciwieństwie do globalnego wielomianu stopnia 5, spline nie generuje niepożądanych oscylacji między punktami.
- **Gładkość:** Zapewnienie ciągłości pierwszej i drugiej pochodnej prowadzi do bardziej naturalnego przebiegu krzywej, co jest często pożądane w interpretacji wyników eksperymentalnych.

Oczywiście metoda interpolacji przy użyciu wielomianów Czebyszewa ma duże zalety w aproksymacji funkcji (szczególnie gdy można wybrać optymalne węzły), jednak przy takich danych – gdy punkty są już danymi pomiarowymi – spline zazwyczaj daje bardziej „realistyczny” przebieg krzywej.

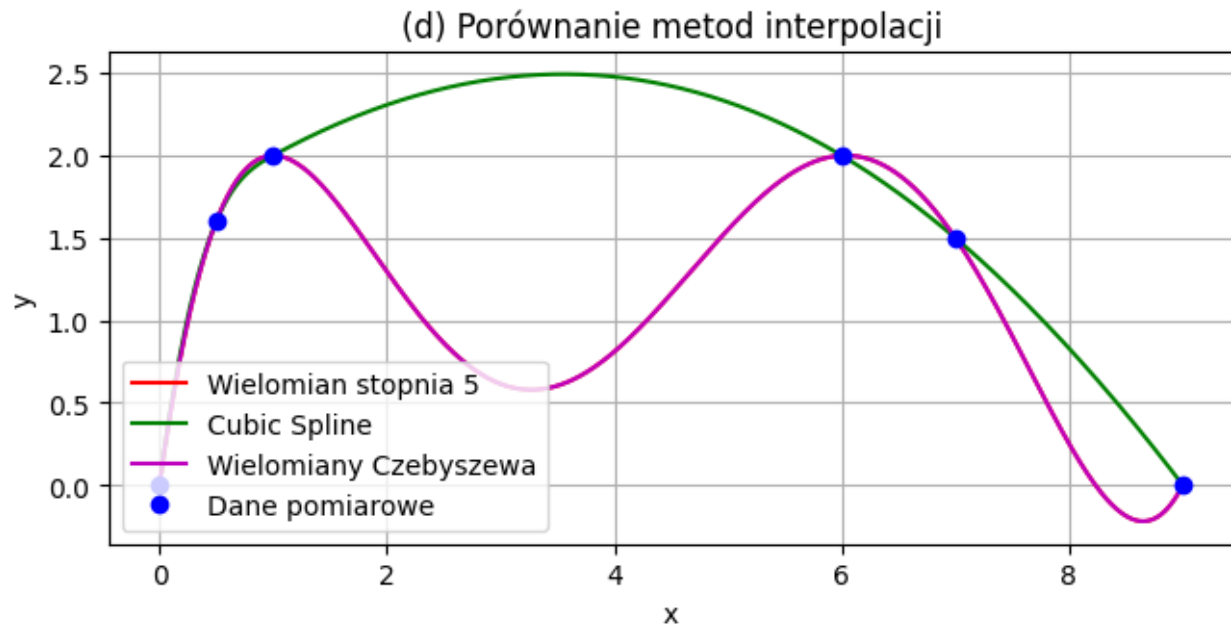


Figure 5: Wykres zestawiający wszystkie podejścia

Implementacja skryptu generująca powyższe wykresy

Poniżej znajduje się pełny skrypt generujący wykresy 2, 4, 3 i 5

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import CubicSpline
from numpy.polynomial.chebyshev import chebfit, chebval

# Experimental data
x_data = np.array([0.0, 0.5, 1.0, 6.0, 7.0, 9.0])
y_data = np.array([0.0, 1.6, 2.0, 2.0, 1.5, 0.0])

# Points to be plotted
x_plot = np.linspace(0, 9, 1000)

def polynomial_interpolation(x, y):
    A = np.vstack([x**i for i in range(6)]).T
    coeffs = np.linalg.solve(A, y)

    def poly_func(x_new):
        result = np.zeros_like(x_new, dtype=float)
        for i, coeff in enumerate(coeffs):
            result += coeff * (x_new ** i)
        return result

    return poly_func, coeffs
```

```

def cubic_spline_interpolation(x, y):
    spline = CubicSpline(x, y)
    return spline

def chebyshev_interpolation(x, y, degree=5):
    x_scaled = 2 * (x - np.min(x)) / (np.max(x) - np.min(x)) - 1

    coeffs = chebfit(x_scaled, y, degree)
    def cheb_func(x_new):
        x_new_scaled = 2 * (x_new - np.min(x)) / (np.max(x) - np.min(x)) - 1
        return chebval(x_new_scaled, coeffs)

    return cheb_func, coeffs

poly_func, poly_coeffs = polynomial_interpolation(x_data, y_data)
spline_func = cubic_spline_interpolation(x_data, y_data)
cheb_func, cheb_coeffs = chebyshev_interpolation(x_data, y_data)

y_poly = poly_func(x_plot)
y_spline = spline_func(x_plot)
y_cheb = cheb_func(x_plot)

plt.figure(figsize=(15, 10))

# Polynomial interpolation
plt.subplot(2, 2, 1)
plt.plot(x_plot, y_poly, 'r-', label='Wielomian stopnia 5')
plt.plot(x_data, y_data, 'bo', label='Dane pomiarowe')
plt.title('(a) Interpolacja wielomianem stopnia 5')
plt.xlabel('x')
plt.ylabel('y')
plt.grid(True)
plt.legend()

# Spline
plt.subplot(2, 2, 2)
plt.plot(x_plot, y_spline, 'g-', label='Cubic Spline')
plt.plot(x_data, y_data, 'bo', label='Dane pomiarowe')
plt.title('(b) Interpolacja funkcją sklejaną stopnia 3')
plt.xlabel('x')
plt.ylabel('y')
plt.grid(True)
plt.legend()

# Chebyshev interpolation
plt.subplot(2, 2, 3)
plt.plot(x_plot, y_cheb, 'm-', label='Wielomiany Czebyszewa')
plt.plot(x_data, y_data, 'bo', label='Dane pomiarowe')
plt.title('(c) Interpolacja wielomianami Czebyszewa')
plt.xlabel('x')
plt.ylabel('y')
plt.grid(True)
plt.legend()

```



```

# Method comparison
plt.subplot(2, 2, 4)
plt.plot(x_plot, y_poly, 'r-', label='Wielomian stopnia 5')
plt.plot(x_plot, y_spline, 'g-', label='Cubic Spline')
plt.plot(x_plot, y_cheb, 'm-', label='Wielomiany Czebyszewa')
plt.plot(x_data, y_data, 'bo', label='Dane pomiarowe')
plt.title('(d) Porównanie metod interpolacji')
plt.xlabel('x')
plt.ylabel('y')
plt.grid(True)
plt.legend()

plt.tight_layout()
plt.show()

print("Polynomial coefficients:")
for i, coeff in enumerate(poly_coeffs):
    x = f"x ^ {i}" if i > 0 else ''
    print(f"{round(coeff,10)} * {x} + ", end='')
print()

```

References

- [1] dr inż. Włodzimierz Funika. Aproksymacja. @misc{key,author={},title={},howpublished={\url{https://home.agh.edu.pl/~funika/mownit/lab4/aproksymacja.pdf}},year={},note={ [Accessed07-04-2025]}}, 2025. [Accessed 07-04-2025].
- [2] Wikipedia. Chebyshev polynomials — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Chebyshev%20polynomials&oldid=1284419238>, 2025. [Online; accessed 08-April-2025].
- [3] Wikipedia. Spline (mathematics) — Wikipedia, the free encyclopedia. [http://en.wikipedia.org/w/index.php?title=Spline%20\(mathematics\)&oldid=1280913089](http://en.wikipedia.org/w/index.php?title=Spline%20(mathematics)&oldid=1280913089), 2025. [Online; accessed 11-April-2025].
- [4] Wikiversity. Cubic Spline — Wikiversity. https://en.wikiversity.org/wiki/Cubic_Spline_Interpolation, 2025. [Online; accessed 11-April-2025].