# Laboratorium z Metod Numerycznych: Interpolacja i analiza błędów

Patryk Blacha, Radosław Szepielak

28 marca 2025

## Spis treści



## 1 Wprowadzenie

W pierwszym zadaniu analizujemy średnią geometryczną odległości punktów na osi rzędnych dla trzech różnych rozkładów punktów: punktów Czebyszewa, punktów Legendre'a oraz punktów rozmieszczonych równomiernie. W drugim zadaniu przeprowadzamy interpolację funkcji przy użyciu różnych metod, takich jak wielomiany Lagrange'a i funkcje sklejane, a także analizujemy dokładność tych metod.

## 2 Zadanie 1: Średnia geometryczna odległości punktów

Zadaniem jest obliczenie średniej geometrycznej odległości punktów na osi rzędnych dla trzech rozkładów punktów: Czebyszewa, Legendre'a oraz rozmieszczonych równomiernie na przedziale [-1, 1]. Obliczamy tę odległość dla różnych wartości $n = 10, 20, 50$.

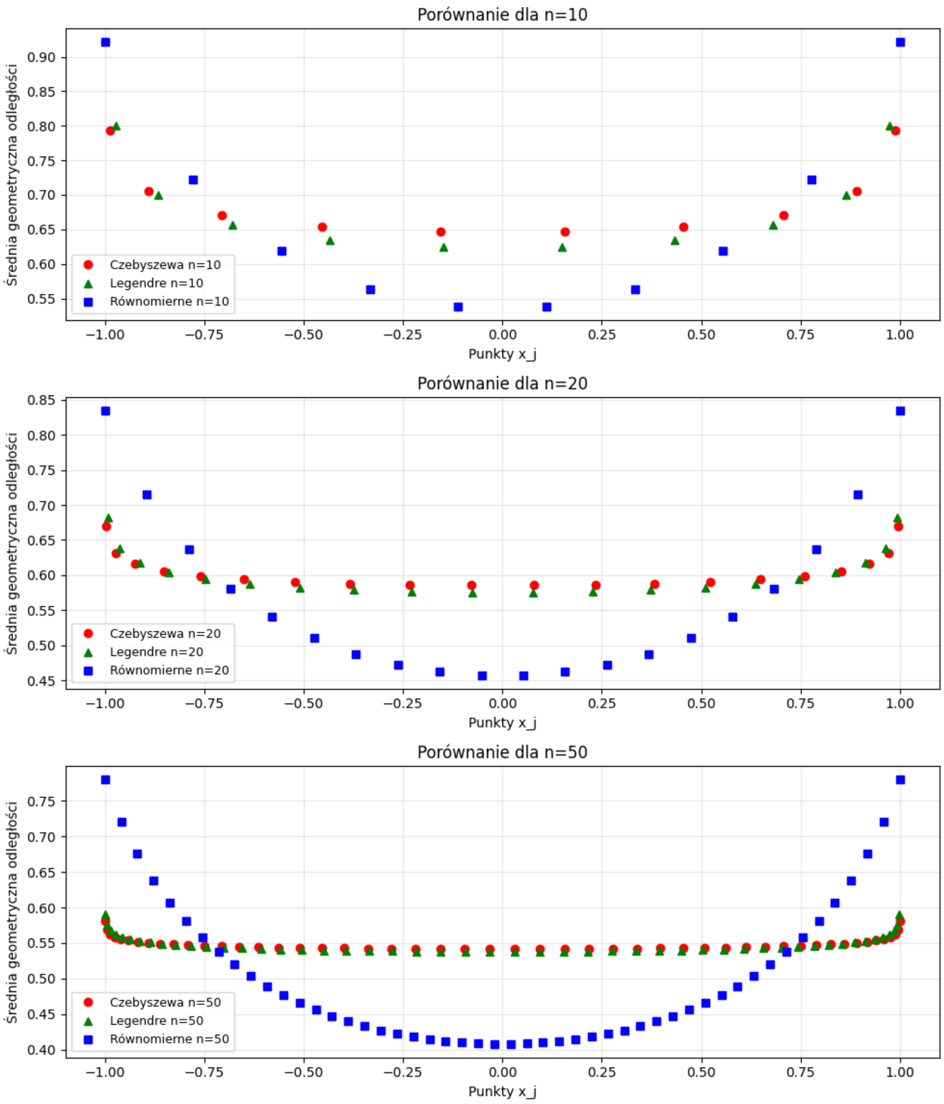
### 2.1 Obliczenia i wykresy

Do obliczenia średniej geometrycznej odległości każdego punktu do pozostałych użyto funkcji `geometric_mean_distance`, której kod jest poniżej:

```python
import numpy as np

def geometric_mean_distance(points):
    n = len(points)
    result = np.zeros(n)
    for i in range(n):
        distances = np.abs(points[i] - np.delete(points, i))
        result[i] = np.exp(np.mean(np.log(distances)))
    return result
```

Wyniki dla punktów Czebyszewa, Legendre'a oraz punktów równomiernych zostały przedstawione na wykresach poniżej.



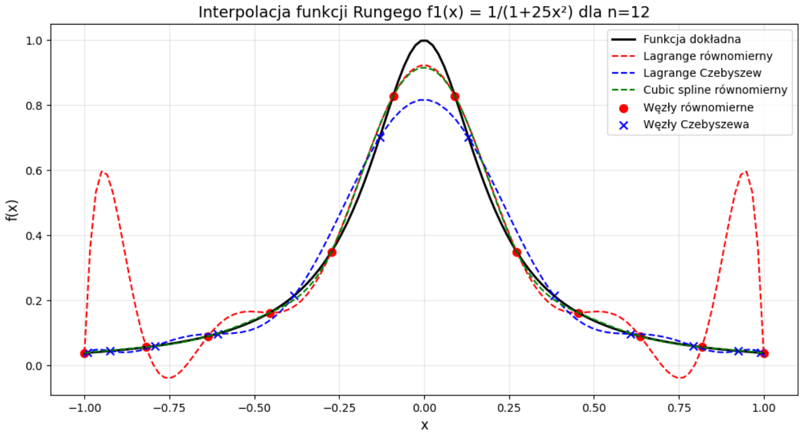

Rysunek 1: Średnia geometryczna odległości punktów na osi rzędnych dla różnych rozkładów punktów.

# 3 Zadanie 2: Interpolacja funkcji

W drugim zadaniu dokonaliśmy interpolacji funkcji $f_1(x)$ oraz $f_2(x)$ na różnych przedziałach. Do interpolacji użyto wielomianów Lagrange'a oraz funkcji sklejanych. Dodatkowo, przeprowadziliśmy analizę błędów interpolacji dla różnych liczby węzłów.

## 3.1 Interpolacja funkcji Rungego

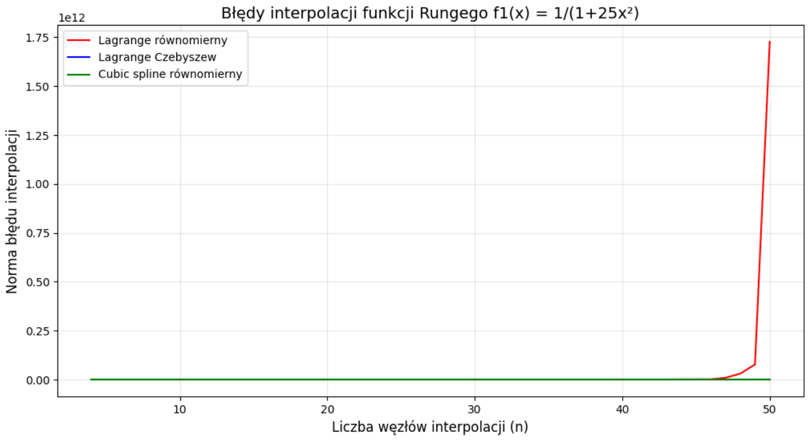
Funkcja $f_1(x) = \frac{1}{1+25x^2}$ na przedziale $[-1, 1]$ została zinterpolowana przy użyciu węzłów równomiernych oraz węzłów Czebyszewa. Wyniki interpolacji przedstawiono na poniższym wykresie.



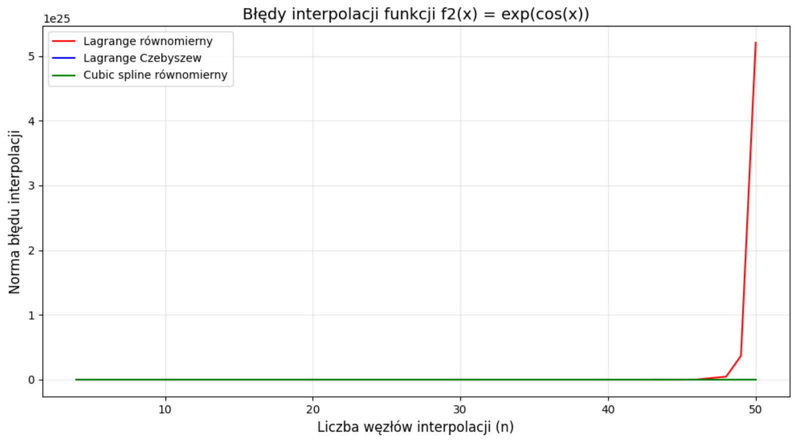

Rysunek 2: Interpolacja funkcji Rungego przy użyciu różnych metod dla $n = 12$.

## 3.2 Analiza błędów interpolacji

Dla różnych wartości $n = 4, 5, \ldots, 50$ obliczono błędy interpolacji dla funkcji $f_1(x)$ oraz $f_2(x)$. Błędy interpolacji przedstawiono na poniższych wykresach.




Rysunek 3: Błędy interpolacji funkcji Rungego $f_1(x)$ w zależności od liczby węzłów $n$.




Rysunek 4: Błędy interpolacji funkcji $f_2(x)$ w zależności od liczby węzłów $n$.

# 4 Wnioski

W analizie zadania 1 zauważono, że punkty Czebyszewa i Legendre'a zapewniają lepszą równomierność rozkładu, co wpływa na zmniejszenie średniej geometrycznej odległości. Z kolei w zadaniu 2 metoda interpolacji przy użyciu funkcji sklejanych okazała się bardziej dokładna niż metoda wielomianów Lagrange'a, szczególnie w przypadku większej liczby węzłów. Interpolacja funkcjami sklejanymi wykazała się również mniejszymi błędami w porównaniu do interpolacji wielomianami Lagrange'a, zwłaszcza dla funkcji Rungego.

- Interpolacja wielomianowa na węzłach równoodległych nie jest stabilna dla dużych n, co prowadzi do efektu Rungego.
- Węzły Czebyszewa poprawiają jakość interpolacji, zmniejszając oscylacje na końcach przedziału.
- Interpolacja splajnowa jest znacznie bardziej stabilna i skuteczna w aproksymacji funkcji, szczególnie dla większej liczby węzłów.
- Dla funkcji o dużych zmianach krzywizny wielomiany interpolacyjne mogą prowadzić do dużych błędów, co sugeruje preferowanie metod splajnowych.

## 4.1 Ogólne wnioski

- **Efekt Rungego** jest szczególnie widoczny dla interpolacji z węzłami równomiernymi - błędy rosną wykładniczo (np. do $1.03 \times 10^3$ dla $n = 25$).
- **Węzły Czebyszewa** znacząco poprawiają dokładność (błąd $7.73 \times 10^{-2}$ dla $n = 25$), ale **splajny kubiczne** są jeszcze lepsze ($1.14 \times 10^{-2}$).
- Dla małych $n$ ($\leqslant 10$) wszystkie metody dają podobne wyniki, ale dla większych $n$ należy bezwzględnie unikać węzłów równomiernych.
- Najlepsze wyniki daje połączenie:
  1. Węzłów Czebyszewa dla interpolacji wielomianowej
  2. Splajnów kubicznych dla większej stabilności

$$\boxed{\text{Dla funkcji typu Rungego: } \begin{cases} \text{małe } n \leqslant 15 & \text{- węzły Czebyszewa} \\ \text{duże } n > 15 & \text{- splajny kubiczne} \\ \text{równomierne} & \text{lepiej nie stosować} \end{cases}}$$

W przypadku użycia wbudowanej metody `scipy.interpolate.lagrange` błędy przechowywane w słownikach `errors_f1` oraz `errors_f2` rosną gwałtownie około $n = 15$, wbudowana funkcja staje się niestabilna numerycznie.

# 5 Literatura


- Dokumentacja Python: numpy, scipy, matplotlib.