

Efekt Rungego

Jan Sarba, Dariusz Rozmus

2.04.2025

1 Treść zadań

Napisz funkcję, która przyjmuje jako parametr wektor punktów x_0, \dots, x_n z przedziału $[-1, 1]$ i tworzy wykres z punktami x_j na osi odciętych i średnią geometryczną odległości do pozostałych punktów na osi rzędnych. Wyświetl wyniki dla:

- punktów Czebyszewa dla $n = 10, 20, 50$
- punktów Legendre’a dla $n = 10, 20, 50$
- punktów równomiernie rozmieszczonych od $x_0 = -1$ do $x_n = 1$ dla $n = 10, 20, 50$.

Punkty Legendre’a to zera wielomianów Legendre’a – pomocna może być funkcja `numpy.polynomial.legendre.legroots`.

Zadanie 2. Wyznacz wielomiany interpolujące funkcje

$$f_1(x) = \frac{1}{1 + 25x^2} \text{ na przedziale } [-1, 1],$$

$$f_2(x) = \exp(\cos(x)) \text{ na przedziale } [0, 2\pi],$$

używając:

- wielomianów Lagrange’a z równoodległymi węzłami
- kubicznych funkcji sklejanych z równoodległymi węzłami
- wielomianów Lagrange’a z węzłami Czebyszewa

$$x_j = -\cos(\theta_j) \quad \theta_j = \frac{2j-1}{2n}\pi, \quad 1 \leq j \leq n. \quad (1)$$

1. Dla funkcji Rungego, $f_1(x)$, z $n = 12$ węzłami interpolacji przedstaw na wspólnym wykresie funkcję $f_1(x)$ oraz wyznaczone wielomiany interpolacyjne i funkcję sklejaną. W celu stworzenia wykresu wykonaj próbkowanie funkcji $f_1(x)$ i wielomianów interpolacyjnych na 10 razy gęstszym zbiorze (próbkowanie jednostajne w x dla węzłów równoodległych, jednostajne w θ dla węzłów Czebyszewa). Pamiętaj o podpisaniu wykresu i osi oraz o legendzie.
2. Wykonaj interpolację funkcji $f_1(x)$ i $f_2(x)$ z $n = 4, 5, \dots, 50$ węzłami interpolacji, używając każdej z powyższych trzech metod interpolacji. Ewaluację funkcji, wielomianów interpolacyjnych oraz funkcji sklepanych przeprowadź na zbiorze 500 losowo wybranych punktów z dziedziny funkcji. Stwórz dwa rysunki, jeden dla $f_1(x)$, drugi dla $f_2(x)$. Na każdym rysunku przedstaw razem wykresy normy wektora błędów (czyli długości wektora) na tym zbiorze punktów w zależności od liczby węzłów interpolacji, n , dla każdej z trzech metod interpolacji.

Która metoda interpolacji jest najbardziej dokładna, a która najmniej?

Uwagi.

1. Interpolację funkcjami sklejanymi można zaimplementować funkcją `scipy.interpolate.CubicSpline`.
Nie należy używać funkcji `scipy.interpolate.interp1d`.

2 Argumentacja

2.1 Interpolacja i aproksymacja funkcji

Interpolacja wielomianowa jest klasycznym podejściem do aproksymacji funkcji, w którym poszukuje się wielomianu $P_n(x)$ stopnia co najwyżej $n - 1$, który dokładnie przechodzi przez n wybranych punktów $(x_i, f(x_i))$. Jednak w przypadku nierównomiernie rozmieszczonych punktów interpolacyjnych, zwłaszcza na końcach przedziału, może dojść do efektu Rungego – oscylacji funkcji interpolacyjnej, szczególnie dla dużych wartości n . W celu redukcji tego zjawiska stosuje się węzły Czebyszewa.

2.1.1 Węzły interpolacyjne

Węzły Czebyszewa są definiowane jako:

$$x_i = \cos\left(\frac{(2i-1)\pi}{2n}\right), \quad i = 1, \dots, n. \quad (2)$$

Dzięki temu rozmieszczeniu węzły są gęściej skupione na krańcach przedziału, co minimalizuje maksymalny błąd interpolacji, który zależy od iloczynu:

$$\prod_{i=1}^n (x - x_i). \quad (3)$$

Obok węzłów Czebyszewa można stosować również węzły Legendre’a, które są pierwiastkami wielomianów ortogonalnych w sensie iloczynu skalarnego na przedziale $[-1, 1]$. Ich zastosowanie znajduje szczególne uzasadnienie w metodach kwadratur Gaussa.

2.1.2 Interpolacja Lagrange’a

Interpolacja Lagrange’a polega na wyznaczeniu wielomianu postaci:

$$P_n(x) = \sum_{i=1}^n f(x_i) l_i(x), \quad (4)$$

gdzie wielomiany bazowe $l_i(x)$ są zdefiniowane jako:

$$l_i(x) = \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}. \quad (5)$$

Wadą tej metody jest konieczność wyznaczenia wielomianów bazowych dla każdego zestawu węzłów oraz podatność na oscylacje przy równomiernym rozkładzie punktów.

2.1.3 Interpolacja funkcjami sklejonymi (spline)

Alternatywną metodą interpolacji, eliminującą problemy związane z wielomianami wysokiego stopnia, jest zastosowanie funkcji sklejanych (spline'ów). W szczególności, często stosuje się interpolację funkcjami sklejonymi trzeciego stopnia (spline cubiczny), które są wielomianami stopnia 3 na każdym z podprzedziałów $[x_i, x_{i+1}]$, spełniającymi warunki gładkości pierwszej i drugiej pochodnej w węzłach.

Spline cubiczny $S(x)$ definiowany jest jako kawałkowy wielomian postaci:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, \quad x \in [x_i, x_{i+1}], \quad (6)$$

gdzie współczynniki są wyznaczone na podstawie wartości funkcji w węzłach oraz warunków ciągłości pierwszej i drugiej pochodnej. Metoda ta pozwala uniknąć nadmiernych oscylacji, charakterystycznych dla interpolacji wielomianowej.

By wykonać **zadanie 1**, definiujemy funkcje potrzebne do generacji potrzebnych punktów

```
# Generowanie punktów Czebyszewa
def chebyshev_points(n):
    return np.cos((2 * np.arange(n) + 1) / (2 * n) * np.pi)

# Generowanie punktów Legendre'a
def legendre_points(n):
    return np.polynomial.legendre.legroots([0] * n + [1])

# Generowanie punktów równomiernych
def uniform_points(n):
    return np.linspace(-1, 1, n)
```

Rozwiązanie **zadania 2** sprowadza się do wykonania następującej pętli:

```
for n in n_values:
    x_uniform = np.linspace(a, b, n)
    y_uniform = f(x_uniform)

    x_chebyshev = chebyshev_nodes(n, a, b)
    y_chebyshev = f(x_chebyshev)

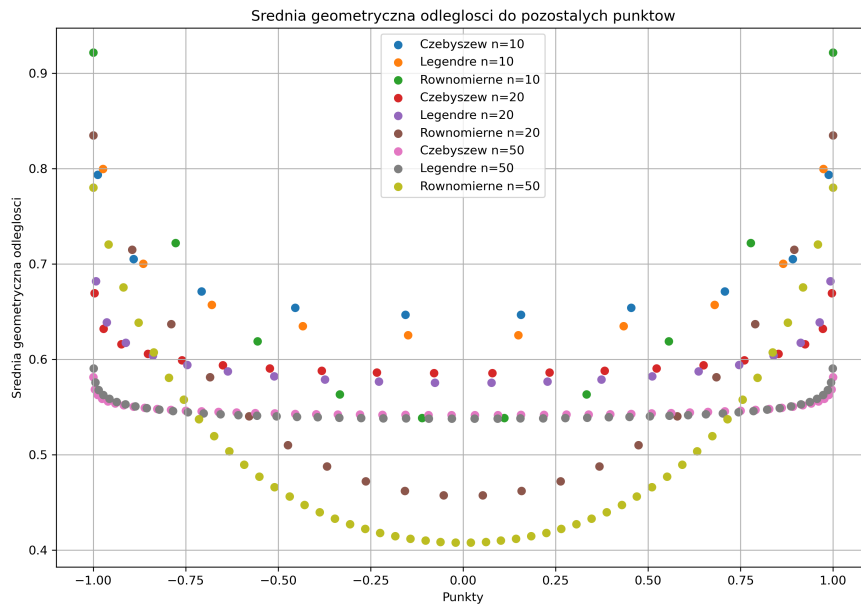
    # Lagrange (uniform)
    lagrange_uniform = BarycentricInterpolator(x_uniform, y_uniform)
    y_lagrange_uniform = lagrange_uniform(x_dense)
    #(...) rysowanie wykresu...

    # Cubic spline
    spline = CubicSpline(x_uniform, y_uniform)
    y_spline = (x_dense)
    #(...) rysowanie wykresu...

    # Lagrange (Chebyshev)
    lagrange_chebyshev = BarycentricInterpolator(x_chebyshev, y_chebyshev)
    y_lagrange_chebyshev = lagrange_chebyshev(x_dense)
    #(...) rysowanie wykresu...
```

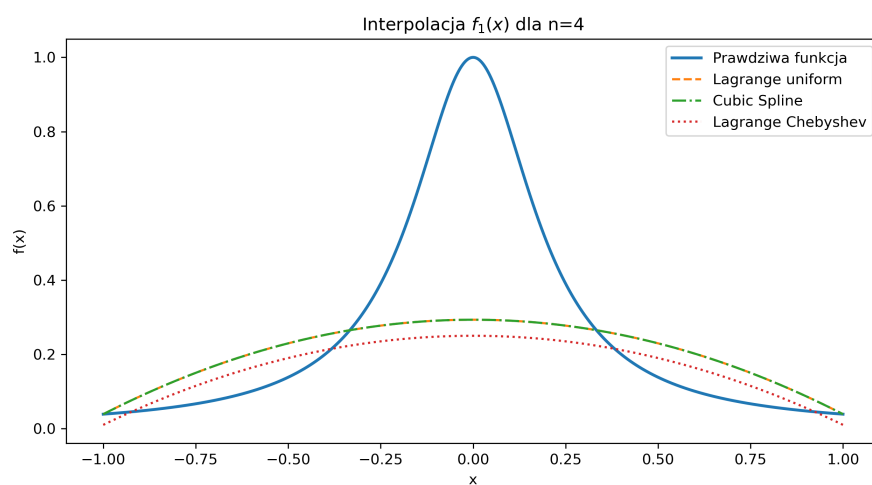
3 Wyniki

Kod z zadania 1 generuje wykres:

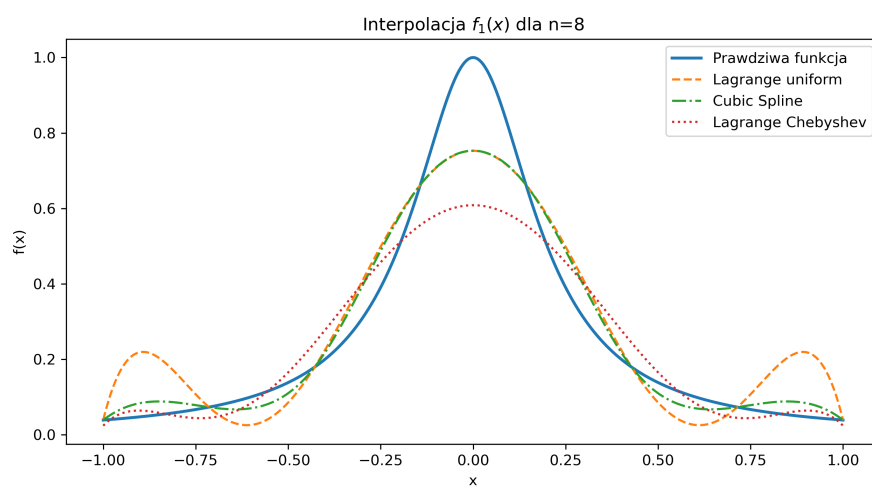


Rysunek 1: Zadanie 1

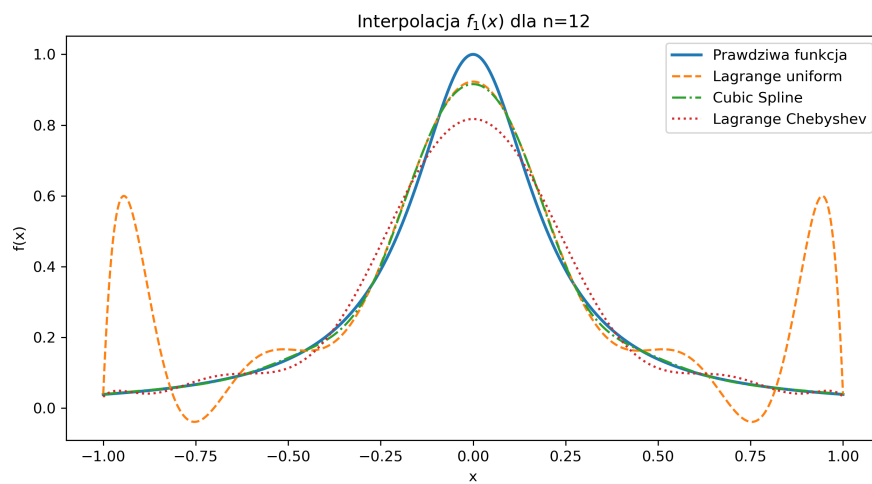
Poniżej znajduje się zbiór wybranych wykresów dla różnych funkcji i n z **zadania 2**:



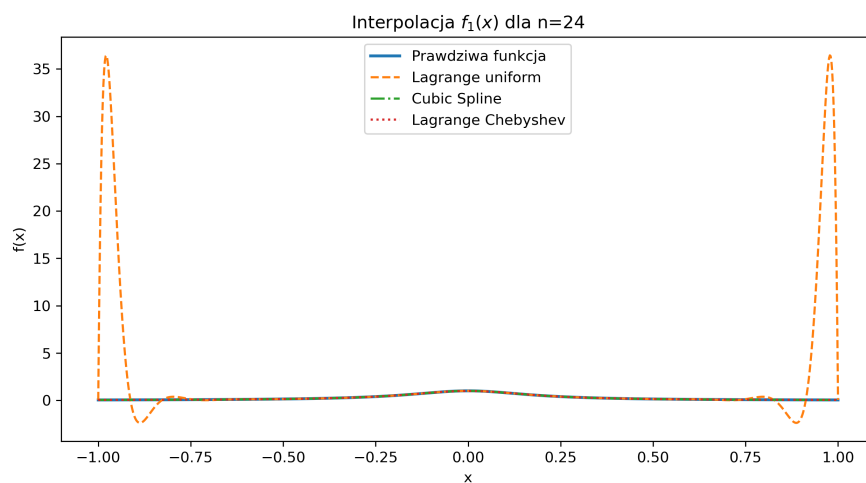
Rysunek 2: Zadanie 2, funkcja 1, $n=4$



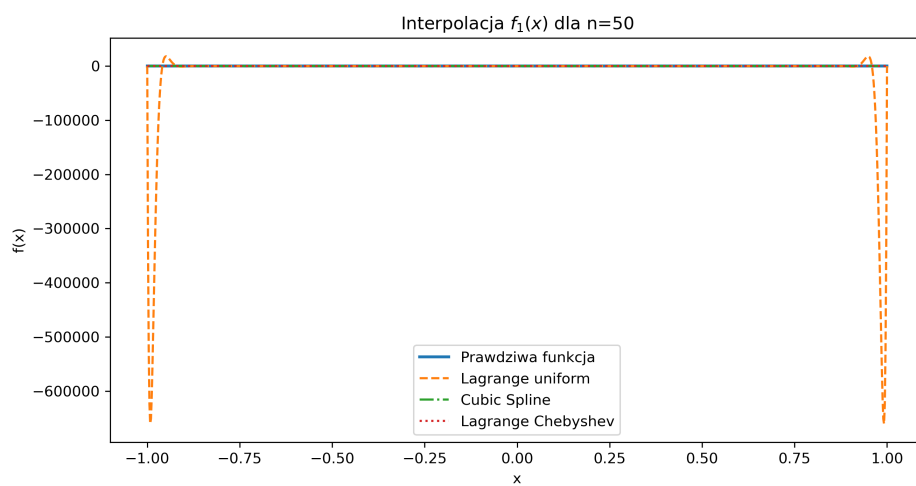
Rysunek 3: Zadanie 2, funkcja 1, $n=8$



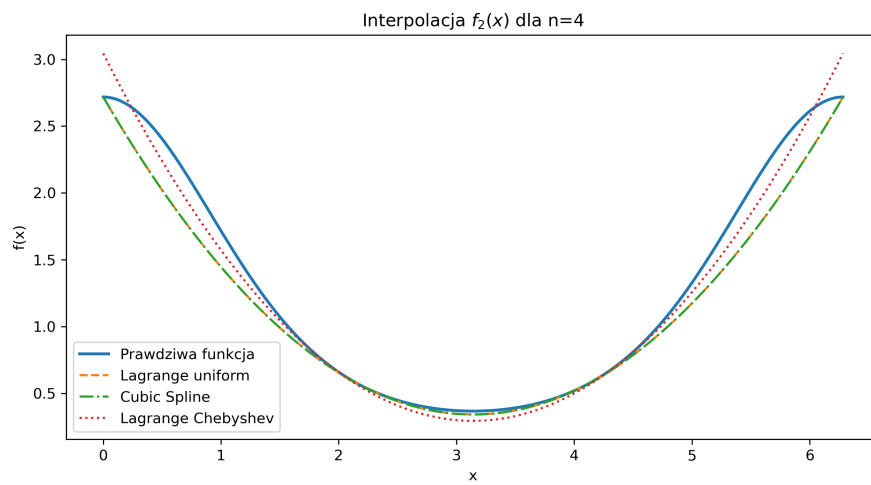
Rysunek 4: Zadanie 2, funkcja 1, $n=12$



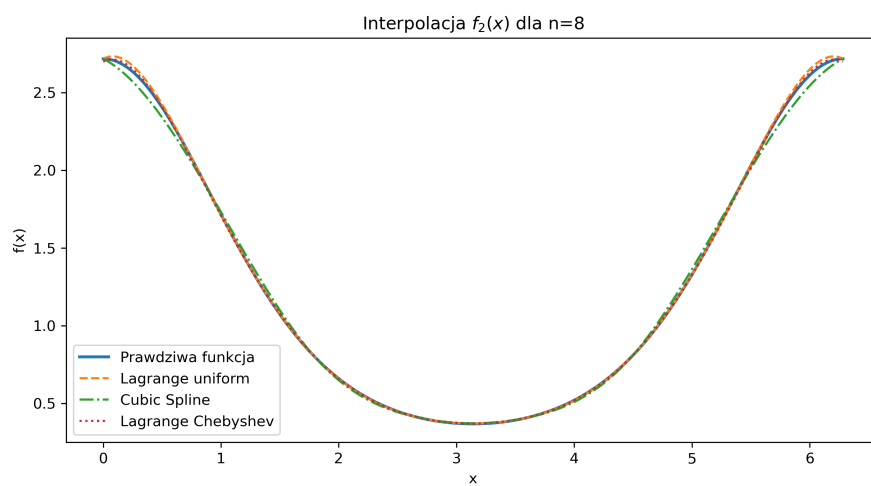
Rysunek 5: Zadanie 2, funkcja 1, $n=24$



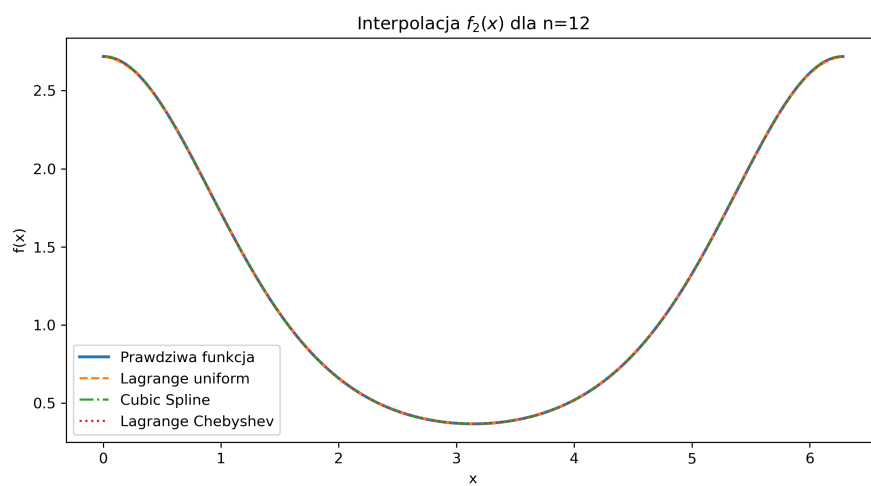
Rysunek 6: Zadanie 2, funkcja 1, $n=50$



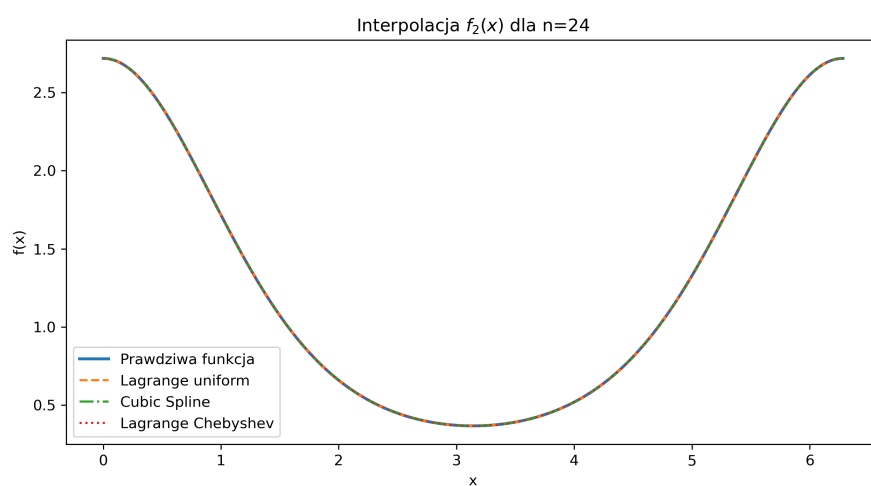
Rysunek 7: Zadanie 2, funkcja 2, $n=4$



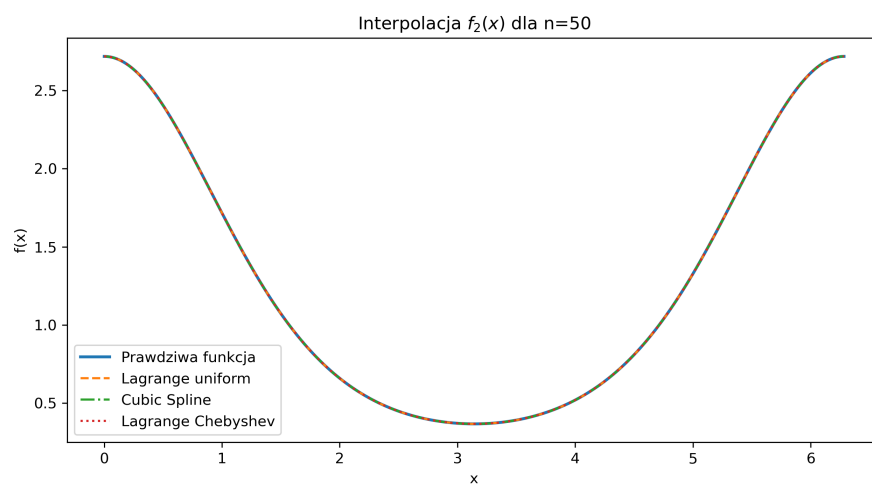
Rysunek 8: Zadanie 2, funkcja 2, $n=8$



Rysunek 9: Zadanie 2, funkcja 2, $n=12$



Rysunek 10: Zadanie 2, funkcja 2, $n=24$



Rysunek 11: Zadanie 2, funkcja 2, $n=50$

4 Wnioski

Dla funkcji $f_1(x)$ konieczne było zastosowanie większej liczby węzłów interpolacyjnych, aby uzyskać dobrą aproksymację. Jednak interpolacja Lagrange'a z węzłami Czebyszewa wykazuje duże oscylacje na brzegach przedziału, co jest przykładem zjawiska Rungego.

W przypadku funkcji $f_2(x)$ wszystkie metody interpolacyjne szybko i dokładnie aproksymują funkcję, co wynika z jej regularności i braku szybkich zmian wartości na rozważanym przedziale.

5 Bibliografia

- dr Rycerz K. – Wykład 2 – interpolacja (MOWNiT)
- prof. Heath M. T. – CS 450—numerical analysis, Chapter 7