Sprawozdanie z Listy nr 4

Sara Żyndul
279686

7 grudnia 2025

Spis treści

0 Wstęp

0.1 Opis problemu interpolacji

Problem interpolacji polega na skonstruowaniu wielomianu $P_n(x)$ stopnia co najwyżej n, który w zadanych węzłach $x_0, x_1, \ldots, x_n$ przyjmuje wartości funkcji f, tzn.

$$P_n(x_k) = f(x_k) \quad \text{dla } k = 0, \ldots, n.$$

W praktyce często korzysta się z postaci Newtona wielomianu interpolacyjnego:

$$N_n(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \cdots + c_n \prod_{j=0}^{n-1}(x - x_j),$$

gdzie współczynniki $c_k = f[x_0, \ldots, x_k]$ są tzw. ilorazami różnicowymi. Postać Newtona jest wygodna z punktu widzenia obliczeń: ilorazy różnicowe wyliczamy rekurencyjnie (bez użycia macierzy 2-wymiarowej), a wartość wielomianu w dowolnym punkcie można efektywnie obliczyć algorytmem Hornera w czasie $O(n)$.

Istotnym aspektem jest dobór węzłów interpolacji: węzły równoodległe są proste w zastosowaniu, lecz dla dużych n mogą prowadzić do zjawiska Runge'go (duże oscylacje na końcach przedziału). Alternatywnie stosuje się węzły będące zerami wielomianu Czebyszewa, które zmniejszają błąd interpolacji.

0.2 Moduł i testy

Wszystkie funkcje zostały umieszczone w module module.jl. Dla każdej funkcji zostały wykonane testy, które potwierdziły poprawność implementacji.

```
Test Summary:         | Pass  Total  Time
Testy Modułu          |   25     25  6.7s
```

1 Zadanie 1

1.1 Idea algorytmu

Celem jest obliczenie ilorazów różnicowych

$$f[x_0],\ f[x_0, x_1],\ f[x_0, x_1, x_2],\ \ldots,\ f[x_0, \ldots, x_n]$$

dla zadanych węzłów $x_0, x_1, \ldots, x_n$ i wartości funkcji $f(x_0), \ldots, f(x_n)$. Definicja rekurencyjna ilorazów różnicowych brzmi:

$$\begin{aligned} f[x_k] &= f(x_k), \\ f[x_k, \ldots, x_{k+m}] &= \frac{f[x_{k+1}, \ldots, x_{k+m}] - f[x_k, \ldots, x_{k+m-1}]}{x_{k+m} - x_k}. \end{aligned}$$

Bezpośrednia implementacja tej definicji przy użyciu macierzy dwuwymiarowej jest prosta, ale zużywa pamięć rzędu $O(n^2)$. Algorytm *in-place* wykorzystuje fakt, że do obliczenia ilorazów rzędu m potrzebne są jedynie wartości ilorazów rzędu $m-1$. Zamiast przechowywać wszystkie rzędy w macierzy, nadpisujemy jeden wektor fx — na początku jest to kopia wektora wartości funkcji (ilorazy rzędu 0), a w kolejnych krokach jego elementy są zamieniane na ilorazy wyższych rzędów.

Istotna sztuczka polega na iterowaniu indeksów od końca do początku w pętli wyznaczającej dany rząd różnic. Dzięki temu przy obliczeniu nowego $fx[i]$ nadal mamy dostęp do potrzebnych wartości $fx[i]$ i $fx[i-1]$ (obie rzędu $m-1$). W tej procedurze zajmujemy pamięć $O(n)$ i wykonujemy $O(n^2)$ operacji arytmetycznych.

Powyższy algorytm wymaga pamięci $O(n)$ i wykonuje $O(n^2)$ operacji — jest to standardowa, efektywna implementacja ilorazów różnicowych używana do konstrukcji współczynników w postaci Newtona.

2 Zadanie 2

2.1 Idea algorytmu

Mamy wielomian interpolacyjny w postaci Newtona:

$$N_n(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \cdots + c_n \prod_{j=0}^{n-1}(x - x_j),$$

gdzie $c_k = f[x_0, \ldots, x_k]$ są ilorazami różnicowymi. Z zadania 8/L4 z ćwiczeń wiemy, że wartość $N_n(t)$ w punkcie t można obliczyć rekurencyjnie za pomocą uogólnionego algorytmu Hornera:

$$\begin{aligned} w_n(t) &:= c_n, \\ w_k(t) &:= c_k + (t - x_k)\, w_{k+1}(t), \qquad k = n-1, \ldots, 0, \\ N_n(t) &= w_0(t) \end{aligned}$$

Algorytm wykorzystuje tę rekurencję i liczy wartości w_k od $k = n$ w dół do $k = 0$. Dla oszczędności pamięci trzymamy tylko jedną skalarną zmienną w (wersja *in-place*): najpierw ustawiamy $w = c_n$, a następnie dla kolejnych k nadpisujemy $w \leftarrow c_k + (t - x_k)\, w$. Dzięki temu ewaluacja wymaga pamięci $O(1)$ i czasu $O(n)$.

3 Zadanie 3

3.1 Idea algorytmu

Mając współczynniki Newtona $c_0, \ldots, c_n$ (wektor fx) oraz węzły $x_0, \ldots, x_n$ chcemy otrzymać współczynniki w postaci naturalnej

$$a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n.$$

Zadanie 9/L4 z ćwiczeń sugeruje użycie idei z zadania 8: zamiast ewaluować wielomian w punkcie, traktujemy $w_k(x)$ jako wielomian i budujemy go iteracyjnie. Zaczynamy od wielomianu stałego $p(x) = c_n$. W kroku dla danego k mnożymy bieżący wielomian $p(x)$ przez $(x - x_k)$ i dodajemy c_k do wyrazu wolnego:

$$p_{\text{new}}(x) = c_k + (x - x_k)\, p(x).$$

Jeżeli $p(x) = \sum_{i=0}^{m} a_i x^i$ to mnożenie przez $(x - x_k)$ daje nowe współczynniki:

$$\begin{aligned} b_0 &= -x_k a_0, \\ b_j &= a_{j-1} - x_k a_j \quad (j = 1, \ldots, m), \\ b_{m+1} &= a_m, \end{aligned}$$

a następnie $b_0 \mathrel{+}= c_k$.

Powtarzając ten krok dla $k = n-1, n-2, \ldots, 0$ otrzymamy ostateczne współczynniki $a_0, \ldots, a_n$. Można to zrobić *in-place* w wektorze współczynników a, aktualizując go od końca (żeby nie nadpisać wartości, które jeszcze będą potrzebne). Złożoność algorytmu to $O(n^2)$ czasu i $O(n)$ pamięci.

4 Zadanie 4

4.1 Idea algorytmu

Celem zadania jest zinterpolowanie zadanej funkcji $f(x)$ na przedziale $[a, b]$ wielomianem Newtona stopnia n oraz porównanie tego wielomianu z oryginalną funkcją na wykresie. Algorytm składa się z następujących etapów:

1. **Walidacja danych wejściowych.**

2. **Wyznaczenie węzłów interpolacji.** W zależności od trybu wybieramy:

   - węzły równoodległe: dla $k = 0, \ldots, n$,

     $$x_k = a + k\,h, \qquad h = \frac{b-a}{n},$$

   - węzły Czebyszewa (zera T_{n+1}) w przedziale $[-1, 1]$:

     $$t_k = \cos\Big(\frac{2k-1}{2(n+1)}\pi\Big), \quad k = 1, \ldots, n+1,$$

     a następnie skalujemy je do $[a, b]$ przez

     $$x_k = \frac{a+b}{2} + \frac{b-a}{2}\,t_k.$$

   Węzły Czebyszewa zmniejszają efekt Runge'a w przypadku dużych n.

3. **Obliczenie wartości funkcji w węzłach.** Dla każdego węzła x_k obliczamy $f(x_k)$ i zapisujemy w wektorze wartości.

4. **Obliczenie ilorazów różnicowych.** Korzystamy z funkcji `ilorazyRoznicowe(x,f)`, która zwraca wektor współczynników Newtona $c_0, \ldots, c_n$ (ilorazy różnicowe). Ten etap kosztuje $O(n^2)$ operacji.

5. **Ewaluacja wielomianu na gęstej siatce.** Tworzymy gęstą siatkę punktów t_j na przedziale $[a, b]$ (np. $m = 400$ punktów). Dla każdego punktu t_j obliczamy wartość wielomianu Newtona za pomocą funkcji (`warNewton`), która działa w czasie $O(n)$ dla jednego punktu.

6. **Przygotowanie danych do rysowania.** Obliczamy wektor wartości oryginalnej funkcji $f(t_j)$ na tej samej siatce oraz wektor wartości wielomianu $p(t_j)$.

7. **Wizualizacja.** Rysujemy na jednym wykresie $f(t)$ i wielomian $N_n(t)$ na osi $[a, b]$, oraz zaznaczamy punkty węzłów $(x_k, f(x_k))$.

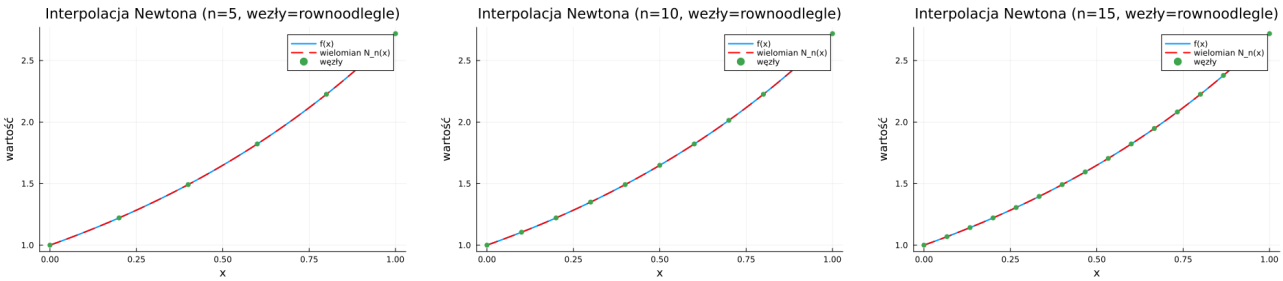
5 Zadanie 5

5.1 Opis zadania

Celem zadania było przetestowanie funkcji `rysujNnfx` dla węzłów równoodległych na funkcjach klasy C^∞ w niewielkich przedziałach. Eksperyment przeprowadzono dla wielomianów stopnia $n \in \{5, 10, 15\}$.

5.2 Podpunkt (a): Funkcja wykładnicza

Interpolowano funkcję $f(x) = e^x$ w przedziale $[0, 1]$.

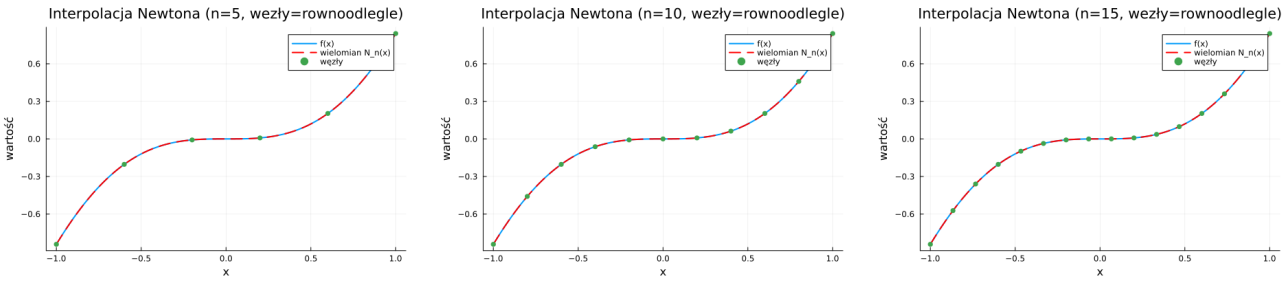


(a) $n = 5$ (b) $n = 10$ (c) $n = 15$

Rysunek 1: Interpolacja funkcji $f(x) = e^x$ na węzłach równoodległych.

5.3 Podpunkt (b): Funkcja trygonometryczno-wielomianowa

Interpolowano funkcję $f(x) = x^2 \sin(x)$ w przedziale $[-1, 1]$.



(a) $n = 5$ (b) $n = 10$ (c) $n = 15$

Rysunek 2: Interpolacja funkcji $f(x) = x^2 \sin(x)$ na węzłach równoodległych.

5.4 Wnioski

Zarówno dla funkcji e^x, jak i $x^2 \sin(x)$, interpolacja wielomianowa z użyciem węzłów równoodległych daje bardzo dobre rezultaty. Już dla najniższego rzędu $n = 5$ wielomian interpolacyjny idealnie pokrywa się z funkcją interpolowaną (krzywe na wykresach nakładają się). Wynika to z faktu, że badane funkcje są gładkie, a przedziały interpolacji stosunkowo wąskie, co sprzyja zbieżności wielomianu interpolacyjnego.

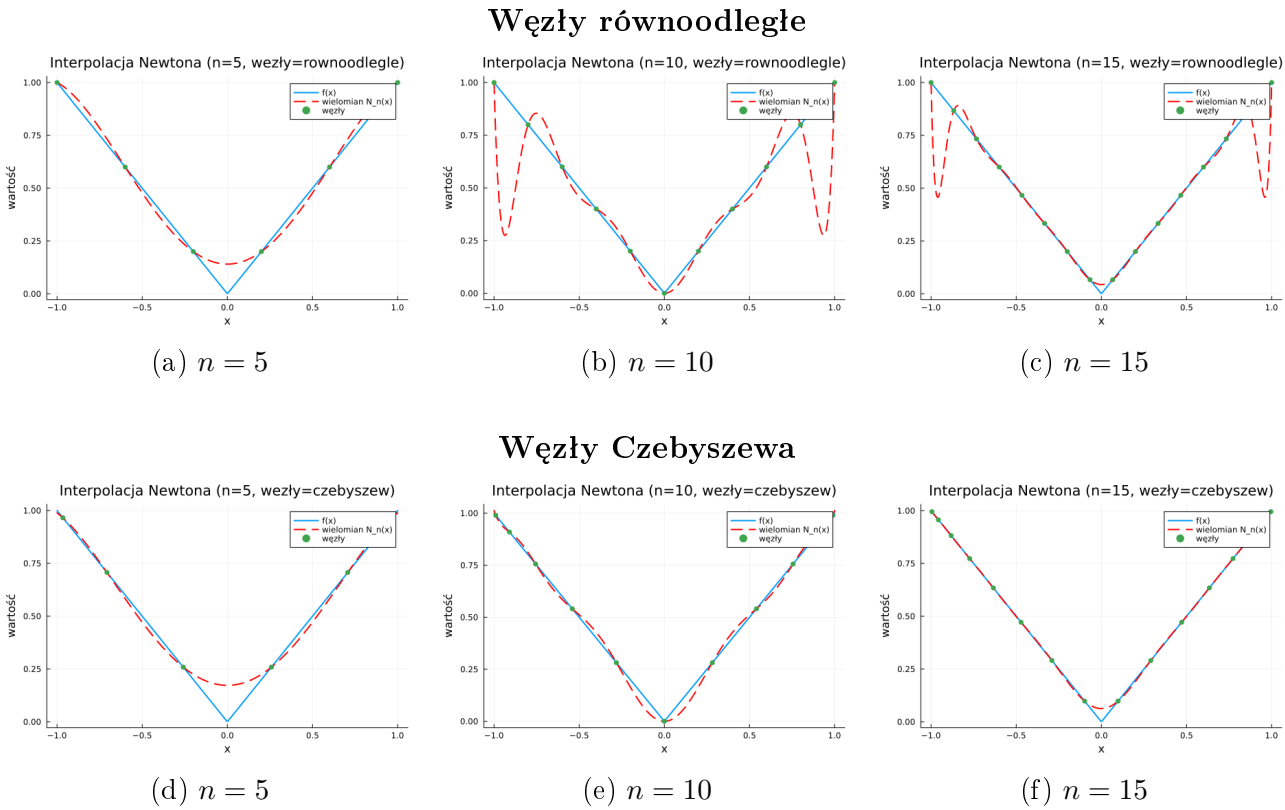
6 Zadanie 6

6.1 Opis zadania

W tym zadaniu analizowano wpływ doboru węzłów (równoodległe vs. Czebyszewa) na jakość interpolacji dla funkcji trudniejszych w aproksymacji tj. $f(x) = |x|$ oraz $f(x) = \frac{1}{1+x^2}$.

6.2 Podpunkt (a): Funkcja $f(x) = |x|$

Badano funkcję $f(x) = |x|$ w przedziale $[-1, 1]$. Funkcja ta w punkcie $x = 0$ jest "ostra"(zmienia gwałtownie kierunek).



(a) $n = 5$ (b) $n = 10$ (c) $n = 15$

(d) $n = 5$ (e) $n = 10$ (f) $n = 15$

Rysunek 3: Porównanie interpolacji funkcji $|x|$ dla węzłów równoodległych (góra) i Czebyszewa (dół).

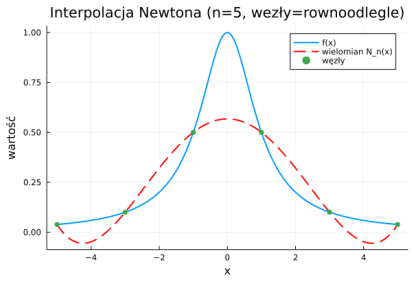
6.2.1 Analiza wyników

- **Węzły równoodległe:** Dla $n = 5$ wierzchołek w $x = 0$ jest zaokrąglony (wielomian nie jest w stanie oddać ostrego przegięcia). Wraz ze wzrostem stopnia wielomianu ($n = 10, 15$) odwzorowanie w okolicy zera się poprawia, ale pojawiają się duże błędy interpolacji na krańcach przedziału.
- **Węzły Czebyszewa:** Jakość interpolacji jest znacznie wyższa. Dla wyższych stopni ($n = 10, 15$) węzły te pozwalają na lepsze odwzorowanie funkcji przy zerze, jednocześnie eliminując problem dużych błędów na krańcach przedziału.

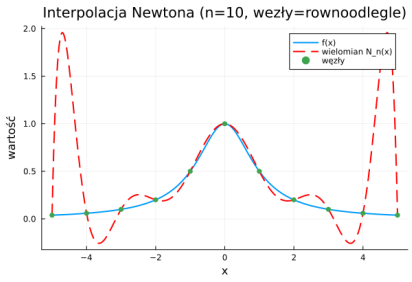
6.3 Podpunkt (b): Zjawisko Runge'go

Badano funkcję $f(x) = \frac{1}{1+x^2}$ w szerokim przedziale $[-5, 5]$.

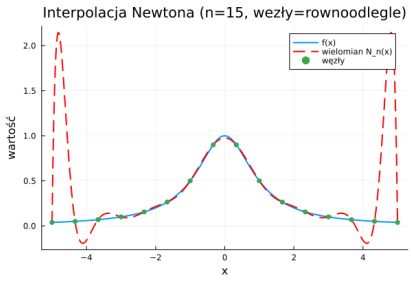
Węzły równoodległe



(a) $n = 5$

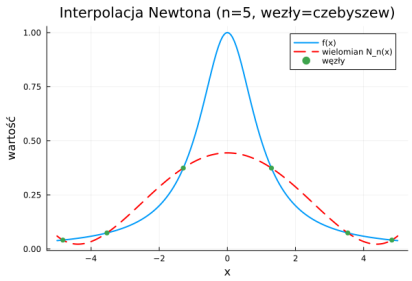


(b) $n = 10$

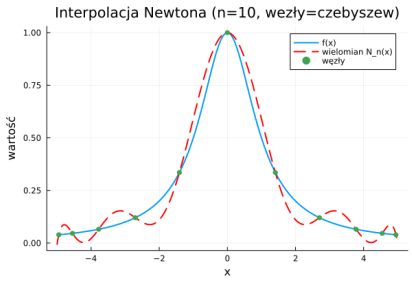


(c) $n = 15$

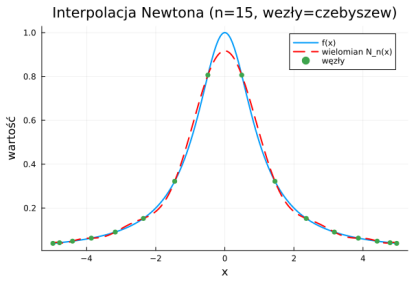
Węzły Czebyszewa



(d) $n = 5$



(e) $n = 10$



(f) $n = 15$

Rysunek 4: Ilustracja zjawiska Runge'go dla funkcji $1/(1 + x^2)$.

6.3.1 Analiza wyników

Na funkcji $f(x) = \frac{1}{1+x^2}$ wyraźnie widać zjawisko Runge'go, co stanowi klasyczny przykład, dla którego interpolacja wielomianowa na węzłach równoodległych zawodzi.

- **Węzły równoodległe:** Obserwujemy drastyczny wzrost błędu interpolacji na krańcach przedziału wraz ze wzrostem stopnia wielomianu n. Jest to tzw. **zjawisko Runge'go**. Gdy obszar interpolacji jest szeroki, węzły równoodległe powodują, że wielomian "nie nadąża"za funkcją przy brzegach, wpadając w mocne oscylacje. Dla żadnego n nie uzyskano zadowalającego przybliżenia.

- **Węzły Czebyszewa:** Zastosowanie węzłów, które są gęściej rozmieszczone na krańcach przedziału (zera wielomianu Czebyszewa), skutecznie eliminuje zjawisko Runge'go.

  - Dla $n = 5$ odwzorowanie jest słabe (zbyt mały stopień, by oddać kształt funkcji w zerze).
  - Dla $n = 10$ uzyskujemy idealne odwzorowanie w punkcie $x = 0$ (gdzie wypada węzeł) oraz dosyć stabilne zachowanie na brzegach.
  - Dla $n = 15$ wielomian dosyć wiernie odtwarza kształt funkcji w całym przedziale.

O zjawisku Runge'a. Runge zauważył, że interpolacja wielomianowa funkcji o dużych wariacjach (szczególnie z ostrymi końcami) przy użyciu równoodległych węzłów może prowadzić

do narastających oscylacji na brzegach przedziału wraz ze wzrostem stopnia wielomianu. Przyczyną jest to, że wielomiany wyższego stopnia mają duże wartości współczynników odpowiadających wysokim potęgom x^k, co w połączeniu z równoodległymi węzłami powoduje wzmacnianie błędu brzegowego. W praktyce jedną z najskuteczniejszych metod przeciwdziałania zjawisku Runge'go jest użycie węzłów Czebyszewa — rozmieszczenie tych węzłów skupia je gęściej przy końcach przedziału i redukuje oscylacje na całym przedziale.

6.4 Wnioski

- Dla problemów, w których funkcja ma ostrzejsze kształty lub szybkie zmiany na krańcach przedziału, stosowanie równoodległych węzłów może być bardzo niebezpieczne (efekt Runge'go).
- Węzły Czebyszewa znacząco poprawiają zachowanie interpolacji na końcach przedziału i tam, gdzie równoodległe węzły dają duże oscylacje.
- Jednakże dla bardzo „stromych" lokalnych struktur (np. w podpunkcie 6b dla $n = 5$) może być konieczne zwiększenie n lub lokalne zagęszczenie węzłów, by dobrze odtworzyć kształt funkcji.