Obliczenia naukowe Lista 4

Radosław Wojtczak

Numer indeksu: 254607

Spis treści

1 Zadanie 1. 2
1.1 Krótki opis problemu . . . 2
1.2 Rozwiazanie . . . 2

2 Zadanie 2. 3
2.1 Krótki opis problemu . . . 3
2.2 Rozwiazanie . . . 3

3 Zadanie 3. 4
3.1 Krótki opis problemu . . . 4
3.2 Rozwiazanie . . . 4

4 Zadanie 4. 5
4.1 Krótki opis problemu . . . 5
4.2 Rozwiazanie . . . 5

5 Zadanie 5. 6
5.1 Krótki opis problemu . . . 6
5.2 Rozwiazanie . . . 6
5.3 Wyniki oraz ich interpretacje . . . 6
5.4 Wnioski . . . 10

6 Zadanie 6. 11
6.1 Krótki opis problemu . . . 11
6.2 Rozwiazanie . . . 11
6.3 Wyniki oraz ich interpretacje . . . 11
6.4 Wnioski . . . 15

Zadanie 1.

1.1 Krótki opis problemu

Głównym celem tego zadania było zaimplementowanie funkcji *ilorazyRoznicowe*, która oblicza ilorazy różnicowe.

1.2 Rozwiazanie

Rozwiązanie znajduje się w pliku *header.jl*, w module *Functions*. Jest to dosłowna interpretacja algorytmu znajdują się w książce autorstwa **David'a Kincaid'a oraz Ward'a Cheney'a** pod tytułem **Analiza numeryczna** w rodziale **Ilorazy różnicowe**. Korzystając z tego sposobu, na samym początku wypełniamy tablicę wartościami funkcji w węzłach. Tablicę tworzymy kolumnami, z dołu do góry. Wykonując obliczenia w takiej kolejności tablica *fx* w każdym momencie zawiera ilorazy różnicowe. Ów rozwiązanie nie wykorzystuje tablicy dwuwymiarowej (zgodnie z poleceniem).

```
for i = 0 to n do
    d_i ← f(x_i)
end do
for j = 1 to n do
    for i = n to j step −1 do
        d_i ← (d_i − d_{i−1})/(x_i − x_{i−j})
    end do
end do
```

Figure 1.1: Pseudokod algorytmu zaimplementowanego w funkcji *ilorazyRoznicowe*

Zadanie 2.

2.1 Krótki opis problemu

Głownym celem tego zadania było zaimplementowanie funkcji *warNewton*, która oblicza wartość wielomianu interpolacyjnego stopnia n w postaci Newtona $N_n(x)$ w punkcie $x = t$ za pomocą uogólnionego algorytmu Hornera **W czasie O(n)!**

2.2 Rozwiazanie

Rozwiązanie znajduje się w pliku *header.jl*, w module *Functions*. Korzystając z zadania 8 z listy nr 4 (ćwiczenia) autorstwa Profesora **Pawła Zielińskiego** skorzystamy z uogólnionych wzorów Hornera w celu zaimplementowania podanej funkcjonalności. W funkcji znajduje się tylko jedna pętla, wykonująca się maksymalnie tyle razy, jak długi jest wektor, na którym operuje, czyli $n + 1$ razy, co daje nam złożoność obliczeniową $O(n)$ (zgodnie z poleceniem).

$$\begin{aligned} w_n(x) &:= f[x_0, x_1, \ldots, x_n] \\ w_k(x) &:= f[x_0, x_1, \ldots, x_k] + (x - x_k)w_{k+1}(x) \; (k = n-1, \ldots, 0) \\ N_n(x) &:= w_0(x). \end{aligned}$$

Figure 2.1: Wzory wykorzystane w implementacji funkcji *warNewton*

Zadanie 3.

3.1 Krótki opis problemu

Głównym celem tego zadania było zaimplementowanie funkcji *naturalna*, która oblicza współczynniki postaci naturalnej znając współczynniki wielomianu interpolacyjnego w Postaci Newtona oraz węzły (dane z oznaczeniami dokładnie opisane są w pliku *header.jl*, w komentarzu dokumentującym funkcję) **W czasie** $O(n^2)$**!**

3.2 Rozwiazanie

Rozwiązanie znajduje się w pliku *header.jl*, w module *Functions.* W wielomianie interpolującym, współczynnikiem stojącym przy najwyższej potędze (x^n) jest c_n. Korzystając z ów wiedzy możemy obliczyć kolejne współczynniki zaczynając od najwyższej potęgi i przy kolejnych iteracjach zmieniać współczynniki w taki sposób, aby były one w każdym momencie w postaciach naturalnych. W kolejnych iteracjach będziemy korzystać z obliczonych wcześniej współczynników postaci naturalnej do zaktualizowania ich o nowe potęgi. Z racji tego, że w funkcji znajdują się dwie zagnieżdżone pętle, która maksymalnie mogą wykonać się n razy otrzymujemy złożoność na poziomie $\mathbf{O}(n^2)$.

Zadanie 4.

4.1 Krótki opis problemu

Głównym celem tego zadania było zaimplementowanie funkcji rysującej wielomian interpolacyjny i interpolowaną funkcję na zadanym przedziale $[a, b]$

4.2 Rozwiazanie

Rozwiązanie znajduje się w pliku *header.jl*, w module *Functions*. Dzielimy przedział na n części i obliczamy wartości funkcji przekazanej jako argument dla wyznaczonych w wyniku podziału wartości x. Wyniki przechowujemy w specjalnych tablicach. Kolejnym krokiem jest wyznaczenie ilorazów różnicowych na podstawie otrzymanych wyników. Następnie dla otrzymania lepszego graficznego przedstawienia dzielimy przedział na $n * 10$ części, gdzie 10 jest z góry ustaloną dodatkową gęstością- w ten sposób łatwiej nam będzie zauważyć zjawiska, które zachodzą w trakcie interpolacji wielomianem. Powtarzamy poprzednie czynności, ponownie otrzymane wyniki tablicujemy (oczywiście bez wyznaczenia kolejnych ilorazów różnicowych), a otrzymane tablice przedstawiamy graficznie przy pomocy funkcji *plot* z pakietu *PyPlot*

Zadanie 5.

5.1 Krótki opis problemu

Celem tego zadania było przetestowanie wcześniej zaimplementowanych funkcji na dwóch przykładach:

1. e^x na przedziale $[0, 1]$ dla $n = 5, 10, 15$
2. $x^2 * sin(x)$ na przedziale $[-1, 1]$ dla $n = 5, 10, 15$

5.2 Rozwiazanie

Rozwiązanie znajduje się w pliku o nazwie *zad5.jl*. Funkcje **a i b** implementują odpowiednio funkcje e^x i $x^2 * sin(x)$. Funkcja *main()* odpowiada za rozruch programu.

5.3 Wyniki oraz ich interpretacje

Wykresy przedstawiające otrzymane rozwiązania:

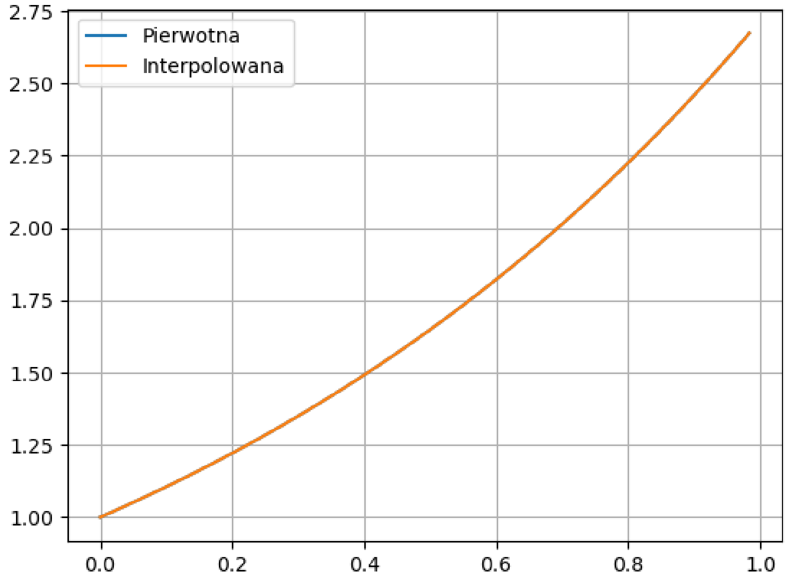

Figure 5.1: Wykres funkcji e^x dla $n = 5$

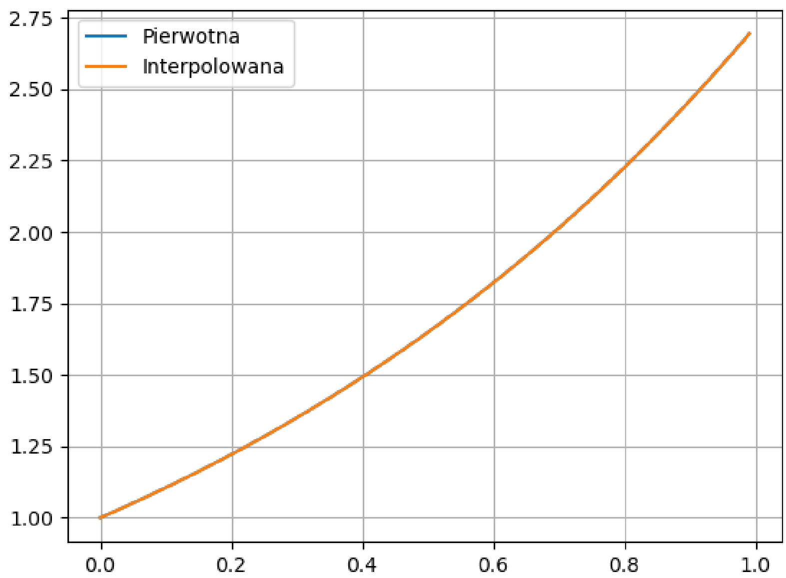

Figure 5.2: Wykres funkcji e^x dla $n = 10$

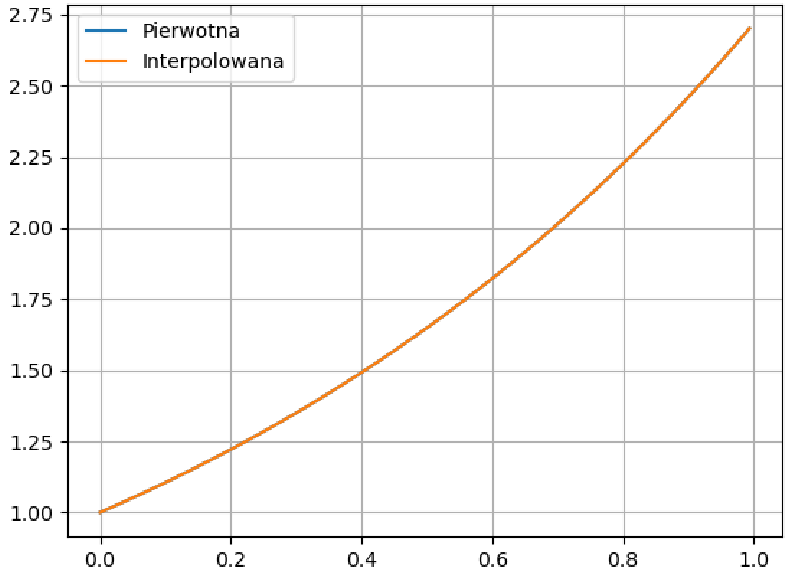

Figure 5.3: Wykres funkcji e^x dla $n = 15$

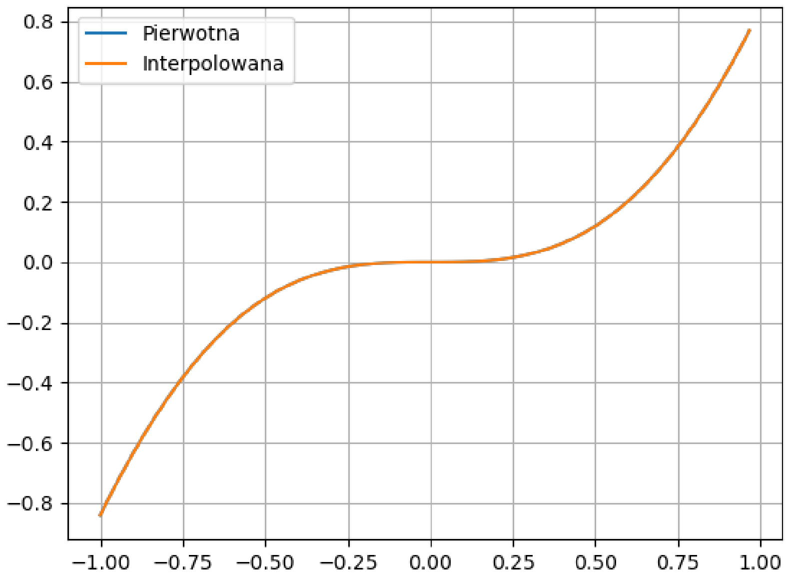

Figure 5.4: Wykres funkcji $x^2 * sin(x)$ dla $n = 5$

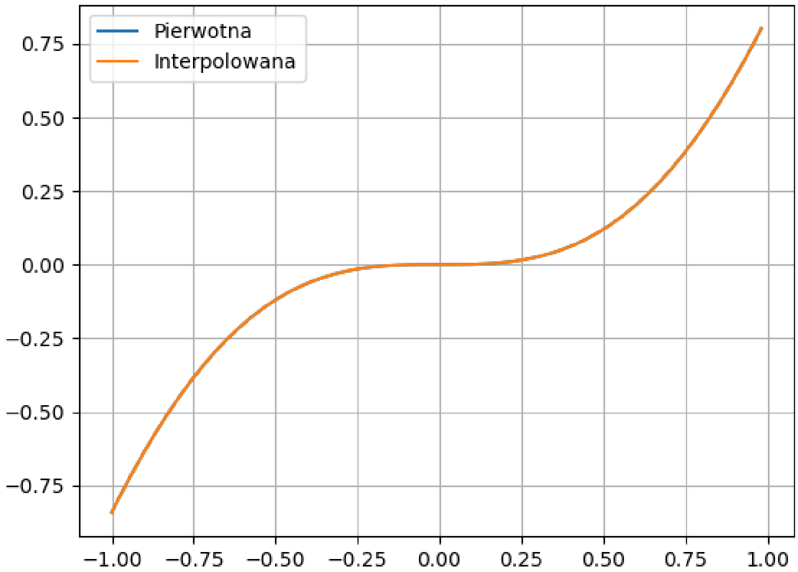

Figure 5.5: Wykres funkcji $x^2 * sin(x)$ dla $n = 10$

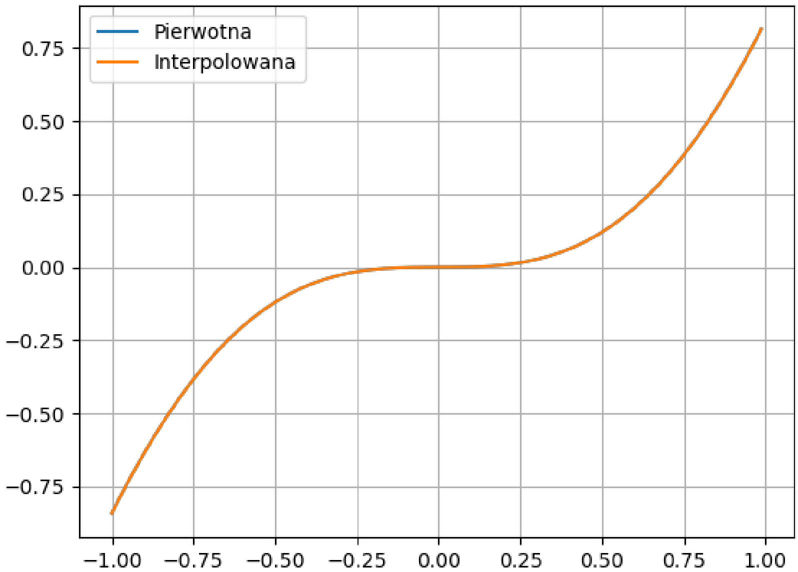

Figure 5.6: Wykres funkcji $x^2 * sin(x)$ dla $n = 15$

Interpretacja: Zauważamy, iż dana funkcja oraz jej interpolacja wielomianem są wręcz tożsame. Wynika to z faktu, iż ów funkcje mają stałą pochodną oraz niewiele różnią się między kolejnymi sprawdzanymi argumentami

5.4 Wnioski

Ze względu na stały znak pochodnej oraz niewielkie różnice w wartościach zwracanych przez funkcję, wielomian interpolacyjnych bardzo dokładnie interpoluje podane funkcje w podanych zakresach.

Zadanie 6.

6.1 Krótki opis problemu

Celem tego zadania było przetestowanie wcześniej zaimplementowanych funkcji na dwóch przykładach:

1. $|x|$ na przedziale $[-1, 1]$ dla $n = 5, 10, 15$
2. $\frac{1}{1+x^2}$ na przedziale $[-5, 5]$ dla $n = 5, 10, 15$

6.2 Rozwiazanie

Rozwiązanie znajduje się w pliku o nazwie *zad6.jl*. Funkcje **c i d** implementują odpowiednio funkcje $|x|$ i $\frac{1}{1+x^2}$. Funkcja *main()* odpowiada za rozruch programu.

6.3 Wyniki oraz ich interpretacje

Wykresy przedstawiające otrzymane rozwiązania:

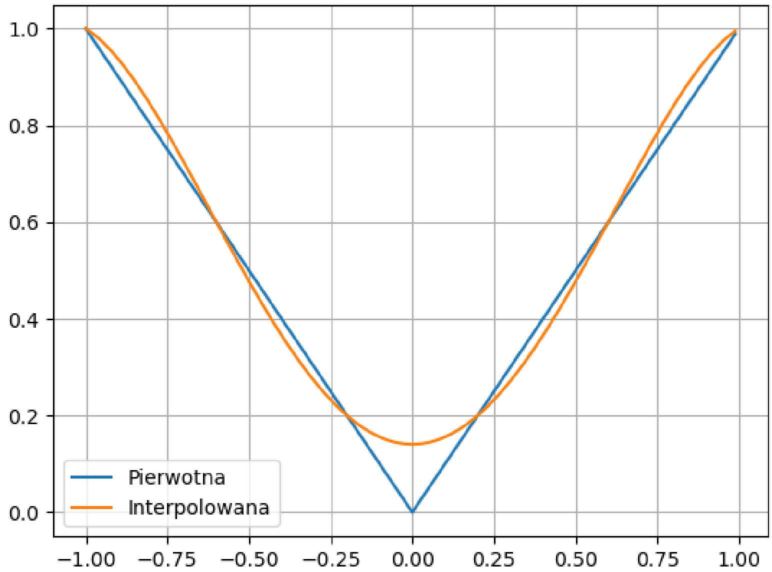

Figure 6.1: Wykres funkcji $|x|$ dla $n = 5$

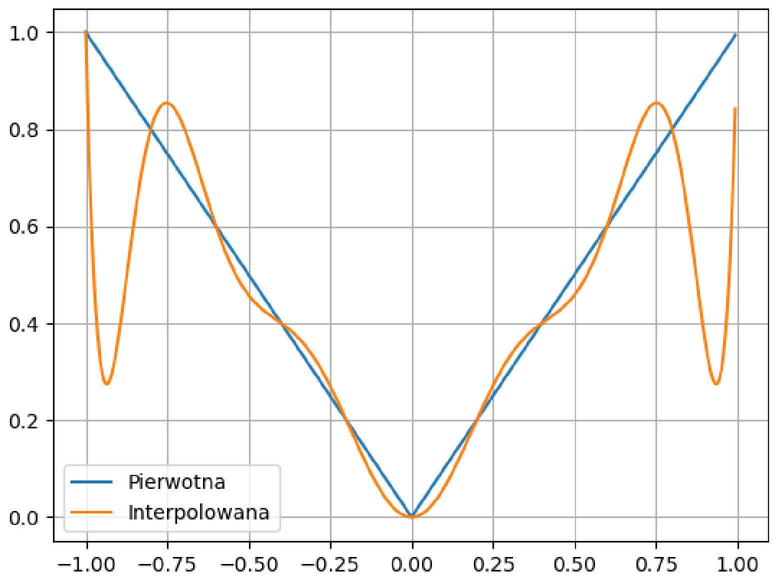

Figure 6.2: Wykres funkcji $|x|$ dla $n = 10$

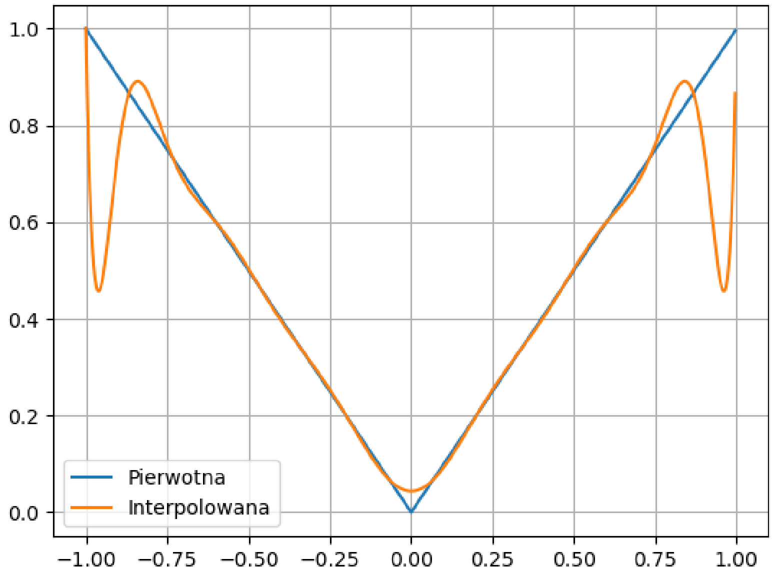

Figure 6.3: Wykres funkcji $|x|$dla $n = 15$

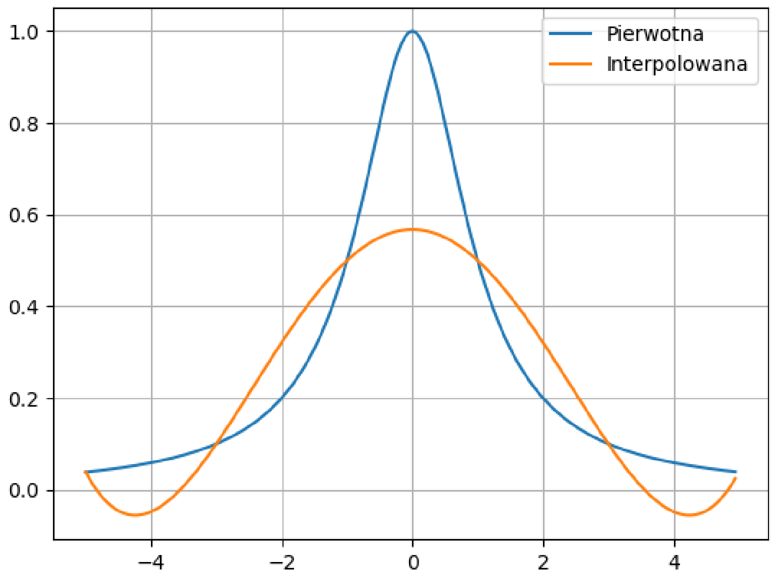

Figure 6.4: Wykres funkcji $\frac{1}{1+x^2}$ dla $n = 5$

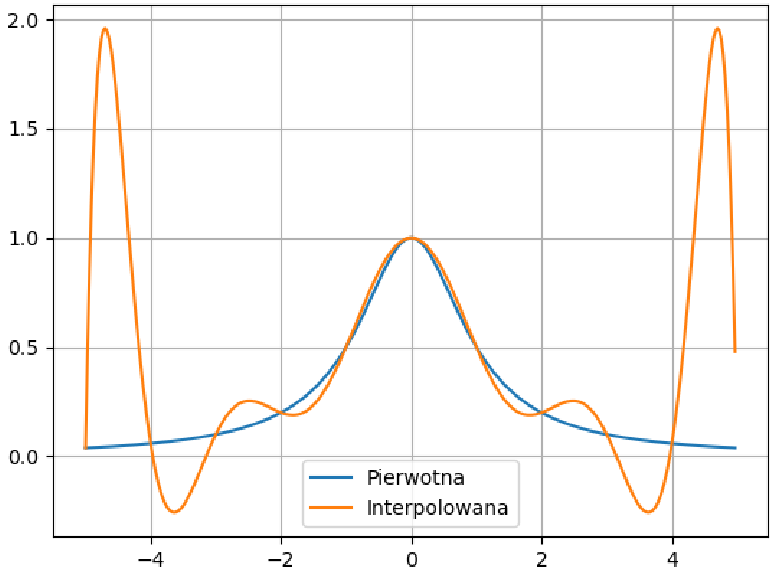

Figure 6.5: Wykres funkcji $\frac{1}{1+x^2}$ dla $n = 10$

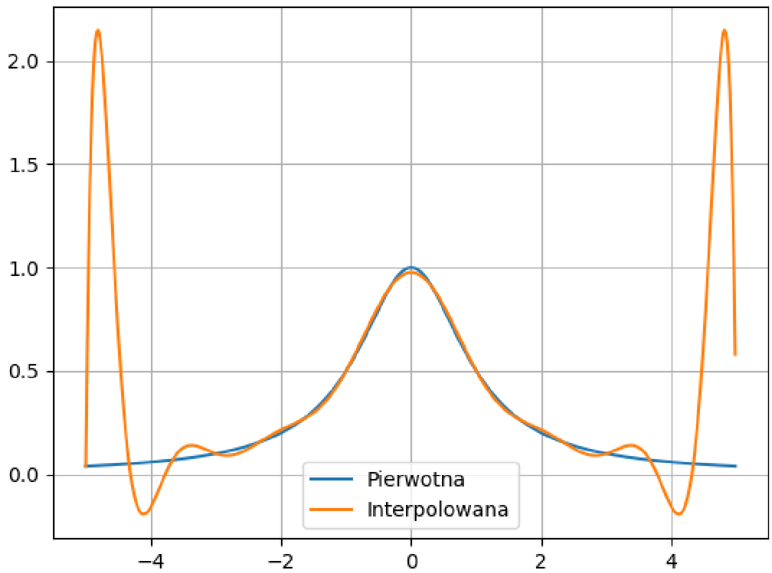

Figure 6.6: Wykres funkcji $\frac{1}{1+x^2}$ dla $n = 15$

Interpretacja: Zauważamy, iż w odróżnieniu od poprzedniego przypadku wykres funkcji jak i jej interpolacja wielomianem zdecydowanie się różnią. To co jest charakterystyczne to fakt, iż im mniejsze n tym otrzymujemy mniejszą niedokładność w środkowej partii przedziału, a większą na jej krańcach, co dynamicznie zmienia się ze wzrostem n (już dla 10, czy 15 widzimy większą dokładność w środku, aniżeli na krańcach przedziału, gdzie funkcja zdecydowanie odbiega od pierwowzoru). Zauważamy duże odchylenia w miejscach, w których pochodna zmienia znak oraz wartości funkcji między kolejnymi węzłami są odpowiednio duże.

6.4 Wnioski

Zauważamy, iż ze wzrostem stopnia wielomianu interpolacyjnego zyskujemy dokładność w środkowych partiach przedziału na rzecz dokładności na jej krańcach. Wynika to ze swego rodzaju paradoksu opisanego przez niemieckiego matematyka **Carla Rungego**, który zauważył, iż zachodzi pogorszenie jakości interpolacji wielomianowej, mimo zwiększenia liczby jej węzłów. Dodatkowo stwierdził, iż wielomiany wysokich stopniów nie nadają się do interpolacji, gdy w interpolacji korzystamy z węzłów równoodległych. Aby uniknąć tego efektu stosuje się interpolacje z węzłami gęściej upakowanymi na krancach przedziałów. (Jednym z rozwiązań jest interpolacja, której węzły są miejscami zerowymi wielomianu Czebyszewa n-tego stopnia).