Lista 4

Wojciech Rymer

Grudzień 2022

Zadanie 1

Opis problemu Zaimplementować funkcję obliczającą ilorazy różnicowe według podanego interfejsu.
Dane:
x - wektor długośći $n + 1$ zawierający węzły $x_0, ..., x_n$
f - wektor długości $n + 1$ zawierający wartości interpolowanej funkcji w węzłach $f(x_0), ..., f(x_n)$
Wyniki:
fx - wektor długości $n + 1$ zawierający obliczone ilorazy różnicowe f[1]=$f[x_0]$, f[2] = $f[x_0, x_1]$,... f[n+1] = $f[x_0, ..., x_n]$.

Rozwiązanie Stworzony algorytm oblicza ze złożonością obliczeniową $O(n^2)$ i złożonością pamięciową $O(n)$ ilorazy różnicowe. Algorytm rozpoczyna swoje wykonywanie na tablicy f i jako jeden krok, oblicza ilorazy wyższych rzędów na wszystkich komórkach, które nie miały już obliczonej finalnej wartości.
Przykład:

$$[f[x_0],\ f[x_1],\ f[x_2]] \rightarrow [f[x_0],\ f[x_0, x_1],\ f[x_1, x_2]] \rightarrow [f[x_0],\ f[x_0, x_1],\ f[x_0, x_1, x_2]]$$

Wyniki Algorytm został przetestowany dla danych wejściowych:
x = [3, 1, 5, 6]
y = [1, -3, 2, 4]
Otrzymano wynik:
fx = [1.0, -2.0, -0.375, 0.175]
Obliczone wartości zgadzają się.

Wnioski Udało się stworzyć algorytm obliczający ilorazy różnicowe w złożoności pamięciowej $o(n^2)$.

Zadanie 2

Opis problemu Zaimplementować funkcję obliczającą wartość interpolowanego wielomianu stopnia n w postaci Newtona w punkcie $x = t$ za pomocą uogólnionego algorytmu Hornera.
Dane:
x - wektor długości $n + 1$ zawierający węzły $x_0, ..., x_n$
fx - wektor długości $n + 1$ zawierający obliczone ilorazy różnicowe f[1]=$f[x_0]$, f[2] = $f[x_0, x_1]$,... f[n+1] = $f[x_0, ..., x_n]$.
t - punkt, w którym należy obliczyć wartość wielomianu
Wyniki:
nt - wartość wielomianu w punkcie t

Rozwiązanie Zauważmy że wielomian interpolacyjny w postaci Newtona można przekształcić w następujący sposób:

$$\sum_{k=0}^{n} f[x_0, ..., x_k] \prod_{j=0}^{k-1} (x - x_j) = f[x_0] + (x - x_0)(f[x_0, x_1] + (x - x_1)(...(x - x_{n-1} f[x_0, ..., x_n])...)$$

Dzięki takiemu przekształceniu, obliczenie wartości wielomianu dla zadanego x zajmuje jedynie $O(n)$ operacji. Zaimplementowany algorytm bazuje na tym przekształceniu, i ma złożoność operacji $O(n)$.

input : x, fx, t
output: r

$r \leftarrow fx[len(x)]$;
for $i \leftarrow len(x) - 1$ **to** 1 **do**
 | $r \leftarrow fx[i] + r * (t - x[i])$;
end

Algorithm 1: Algorytm wyliczający wartość wielomianu w punkcie x = t

Wyniki Algorytm został przetestowany dla danych wejściowych przy $f(x) = sin(x^2)$:
x = range(-1,1,10)
y = map(f,x)
t = 0.981
Otrzymano wynik:
nt = 0.8205948258134824
Obliczona wartość jest bliska faktycznej: $|f(x) - nt| =$ 5.1825e-6

Wnioski Przy użyciu pewnych przekształceń algebraicznych udało się opracować algorytm wyliczający wartość wielomianu w punkcie ze złożonością obliczeniową $O(n)$. Pozwala to na szybkie obliczanie wartości funkcji interpolacyjnej przy użyciu ilorazów różnicowych.

Zadanie 3

Opis problemu Celem zadania jest obliczenie współczynników postaci naturalnej wielomianu interpolacyjnego znając jego współczynniki w postaci Newtona oraz węzły w czasie $O(n^2)$. Dane:
x - wektor długości $n + 1$ zawierający węzły $x_0, ..., x_n$
fx - wektor długości $n + 1$ zawierający obliczone ilorazy różnicowe f[1]=$f[x_0]$, f[2] = $f[x_0, x_1]$,... f[n+1] = $f[x_0, ..., x_n]$.
Wyniki:
a - wektor długości $n + 1$ zawierający obliczone współczynniki postaci naturalnej a[1]=a_0 a[2]=a_1, ..., a[n+1] = a$_n$

Rozwiązanie W tym zadaniu również opieramy się na przekształceniu zawartym w zadaniu 2, zapiszmy je w sposób uproszczony:

$$\sum_{k=0}^{n} f[x_0, ..., x_k] \prod_{j=0}^{k-1}(x - x_j) = f[x_0] + (x - x_0)(f[x_0, x_1] + (x - x_1)(...(x - x_{n-1} f[x_0, ..., x_n])...) = f[x_0] + (x - x_0)w_1(x)$$

Założmy że dla $w_1(x)$ mamy tablicę $[a_{w_1 0}, a_{w_1 1}, \dots , a_{w_1 n-1}]$ dobrze obliczonych współczynników postaci naturalnej $w_1(x)$. Wtedy tablicę współczynników postaci naturalnej $f[x_0] + (x - x_0)w_1(x)$, nazwijmy ją T_0, można otrzymać w następujący sposób

$$T_0 = -x_0 * [a_{w_1 0}, a_{w_1 1}, \dots , a_{w_1 n-1}, 0] + [0, a_{w_1 0}, a_{w_1 1}, \dots , a_{w_1 n-1}] + [f[x_0], 0, ..., 0]$$

Poczynając od $T_n = [f[x_0, x_1, ..., x_n]]$ i używając przedstawionej wyżej metody możemy obliczyć T_0 w czasie $O(n^2)$. Algorytm wygląda następująco:

input : x, fx
output: T

$T \leftarrow zeroes(len(x))$
$T[1] = fx[len(x)$
for $i \leftarrow len(x) - 1$ **to** 2 **do**
 | $T_{partial} \leftarrow T$
 | **for** $j \leftarrow 1$ **to** $len(x) - 1 + i$ **do**
 | | $T_{partial}[j] \leftarrow T_{partial}[j] + T[j] * (-x[i - 1])$
 | | $T_{partial}[j + 1] \leftarrow T_{partial}[j + 1] + T[j]$
 | **end**
 | $T \leftarrow T_{partial}$
 | $T[1] \leftarrow T[1] + fx[i - 1]$
end

Algorithm 2: Algorytm wyliczający współczynniki postaci naturalnej wielomianu

Wyniki Algorytm został przetestowany dla danych wejściowych przy $f(x) = x^2$:

x = range(-1,1,10)

y = map(f,x)

Otrzymano wynik:

a = [0.0, 0.0, 0.999999 9.12120e-31, 1.32052e-15, 1.57772e-30, -2.70311e-15, 3.94430e-31, 1.50906e-15, 0.0]

Wnioski Otrzymane wyniki pokazują że algorytm bardzo dobrze odtworzył funkcję wielomianową co pokazuje jego poprawność. Można też zauważyć że $a_n = 0$ co zgadza się ze współczynnikami wielomianu interpolacyjnego.

Zadanie 4

Opis problemu Celem zadanie była implementacja funkcji, która zinterpoluje funkcję $f(x)$ na zadanym przedziale $[a, b]$ wielomianem stopnia n w postaci Newtona. Następnie narysuje wielomian interpolacyjny i interpolowaną funkcję.

Dane:

f - funkcja $f(x)$ zadana jako funkcja anonimowa

a,b - przedział interpolacji

n - stopień wielomianu interpolacyjnego

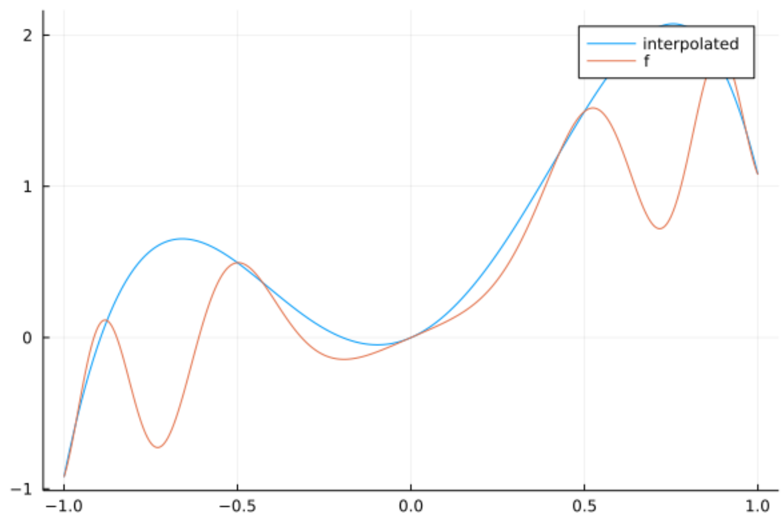
Wyniki:

- wykres funkcji interpolowanej i wielomianu interpolacyjnego na przedziale $[a, b]$

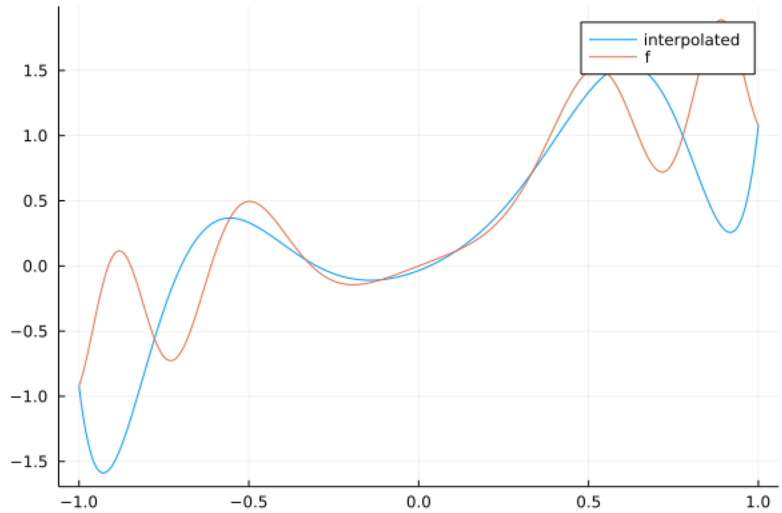
Rozwiązanie Używając funkcji zaimplementowanych w poprzednich zadaniach obliczone zostały ilorazy różnicowe interpolowanej funkcji dla równoodległych węzłów na zadanym przedziale, a następnie narysowane zostały wykresy funkcji interpolowanej i interpolacyjnych. Do rysowania wykresu użyte zostało 1000 próbek wartości funkcji.

Wyniki Przykładowe działanie programu dla funkcji $f(X) = x + sin((6x^2)^2)$, $a = -1$, $b = 1$, $n = [5, 10, 15]$

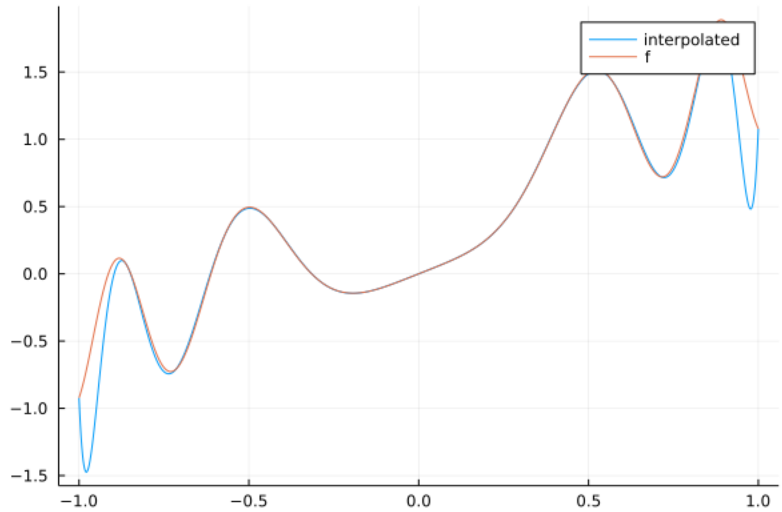
Rysunek 1: Wykres dla $n = 5$



Rysunek 2: Wykres dla $n = 10$



Rysunek 3: Wykres dla $n = 15$



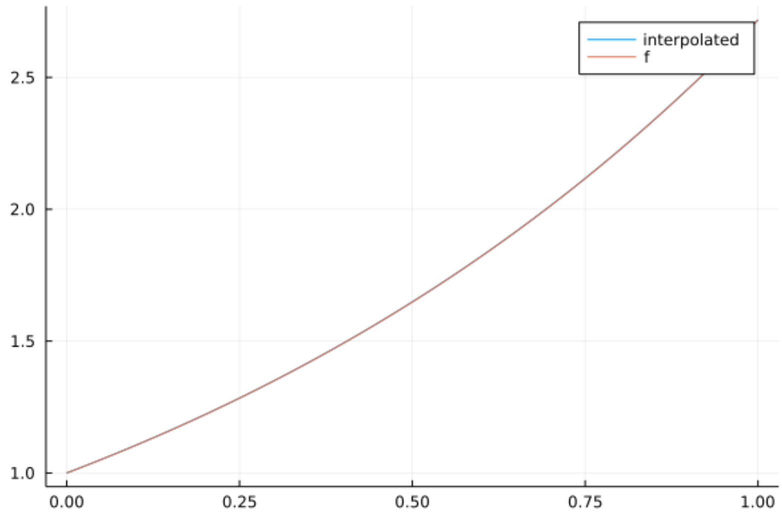
Zadanie 5

Opis problemu Zadanie polegało na przetestowaniu funkcji z zadania 4. na następujących przykładach:

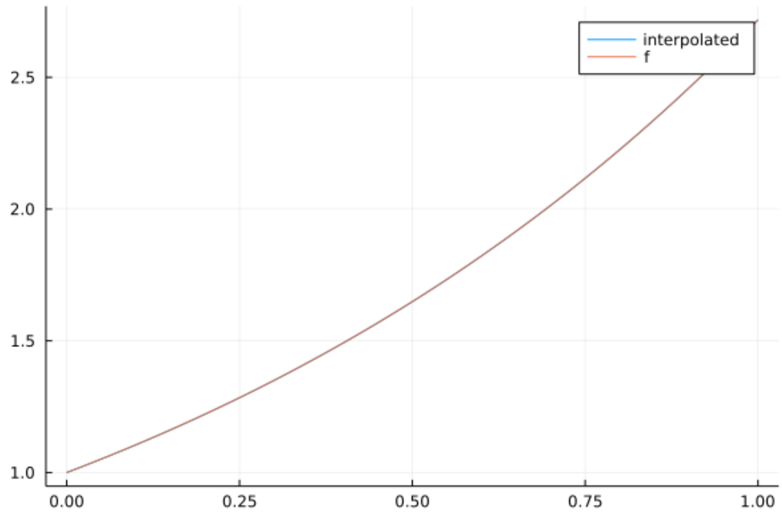
- (a) e^x, $[0, 1]$, $n = 5, 10, 15$
- (b) $x^2 sin(x)$, $[-1, 1]$, $n = 5, 10, 15$

Wyniki Zadane funkcje prezentują się następująco:

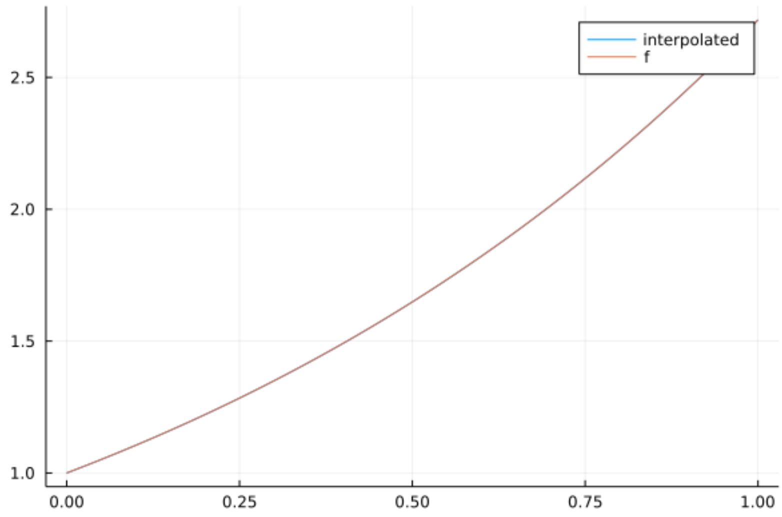
Rysunek 4: Wykres dla e^x, $n = 5$



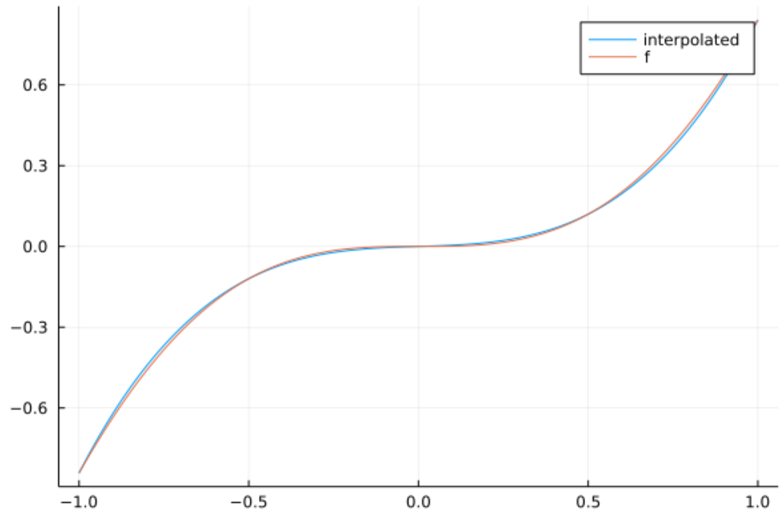
Rysunek 5: Wykres dla e^x, $n = 10$



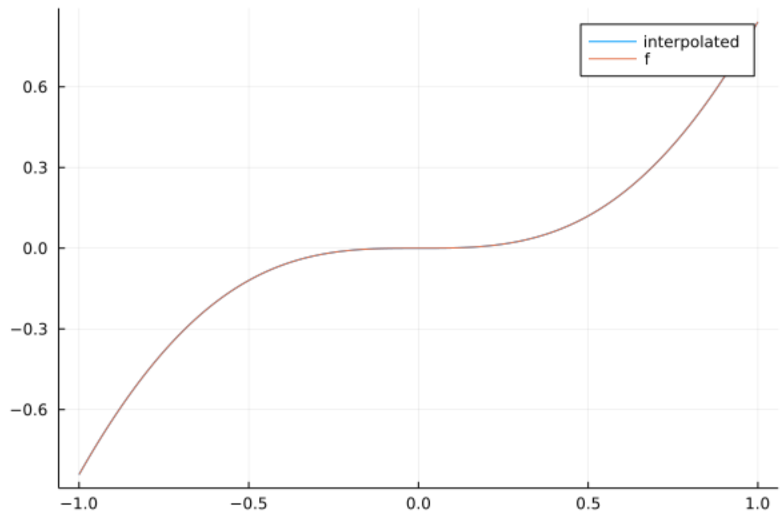
Rysunek 6: Wykres dla e^x, $n = 15$



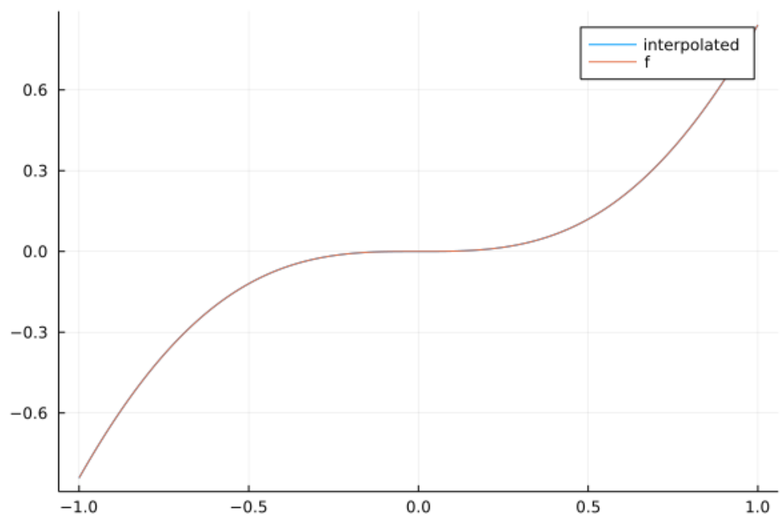
Rysunek 7: Wykres dla $x^2 sin(x)$, $n = 5$



Rysunek 8: Wykres dla $x^2 sin(x)$, $n = 10$



Rysunek 9: Wykres dla $x^2 sin(x)$, $n = 15$



Wnioski Zaimplementowana metoda interpolacji dobrze radzi sobie z interpolowaniem zadanych funkcji. Oceniając graficznie, dodawanie kolejnych węzłów w tym przypadku nie przynosiło znacząco lepszych rezultatów. Dzieje się tak, ponieważ zadane przedziały funkcji są łatwe w aproksymacji i już dla małych n otrzymywane są dosyć dobre przybliżenia funkcji interpolowanej.

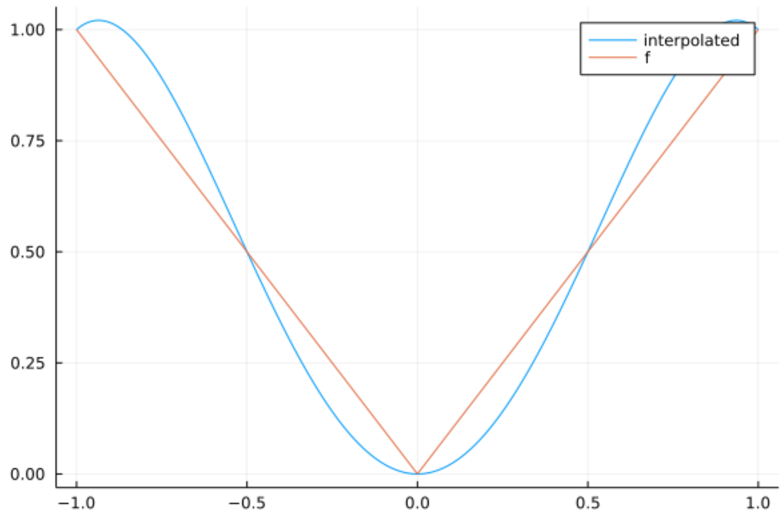
Zadanie 6

Opis problemu Zadanie polegało na przetestowaniu funkcji z zadania 4. na następujących przykładach i zbadać zjawisko rozbieżności.

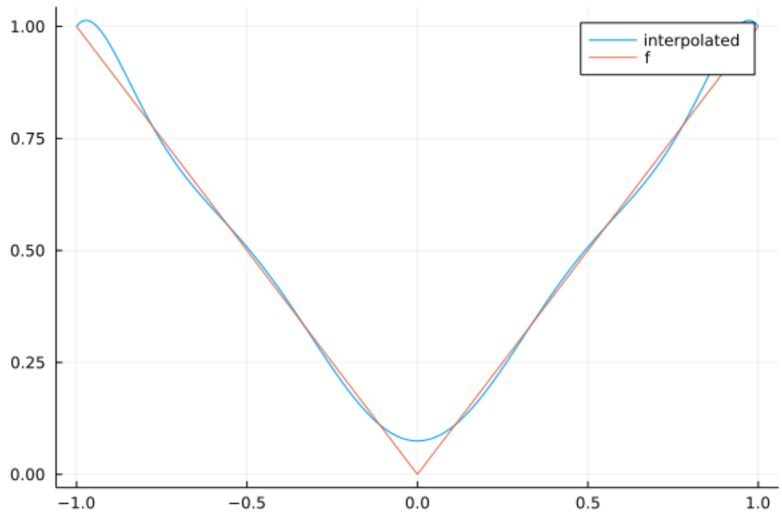
- (a) $|x|$, $[-1, 1]$, $n = 5, 10, 15$
- (b) $\frac{1}{1+x^2}$, $[-5, 5]$, $n = 5, 10, 15$

Wyniki Zadane funkcje prezentują się następująco:

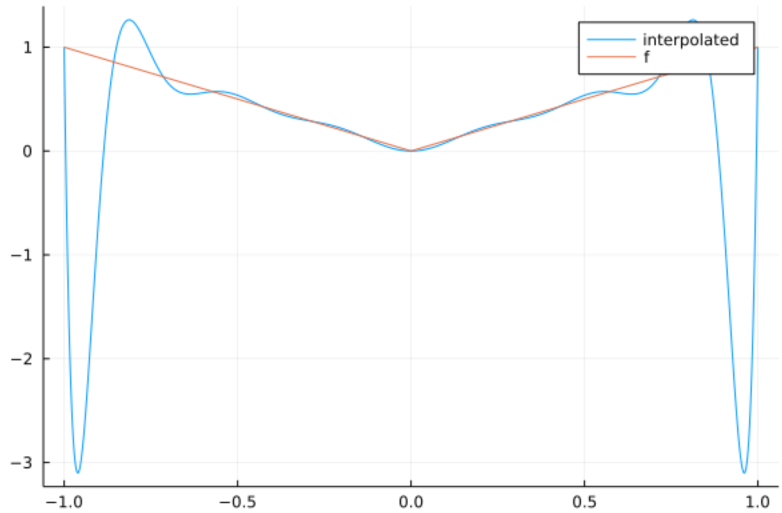
Rysunek 10: Wykres dla $|x|$, $n = 5$



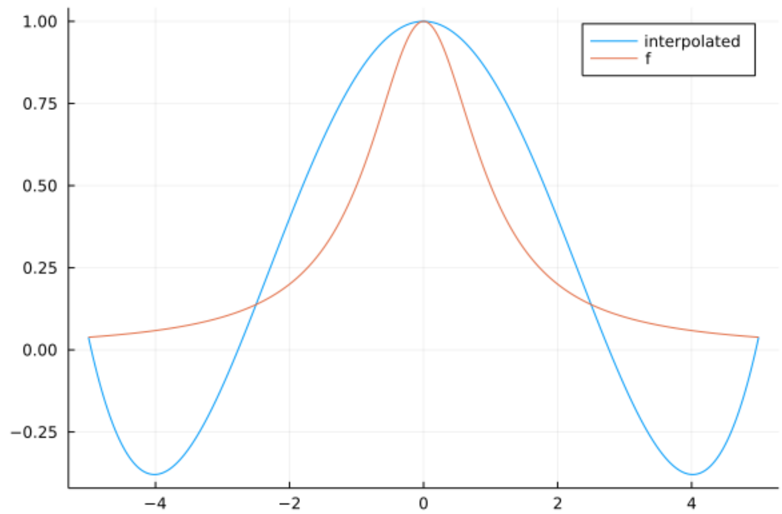
Rysunek 11: Wykres dla $|x|$, $n = 10$



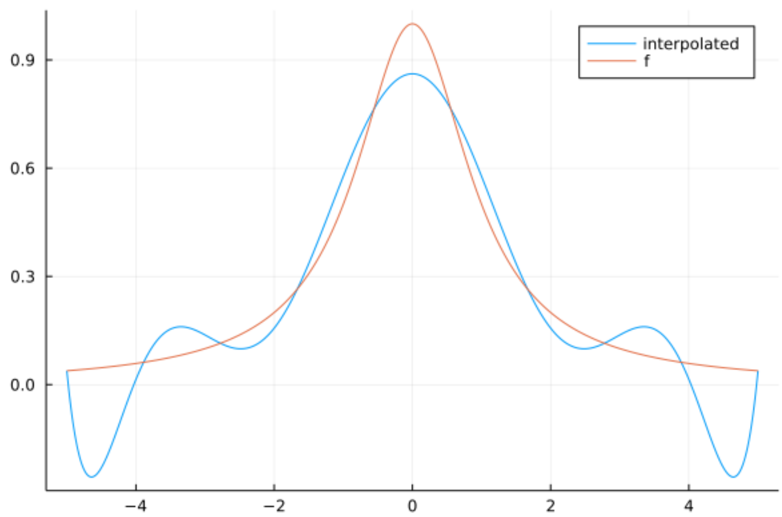
Rysunek 12: Wykres dla $|x|$, $n = 15$



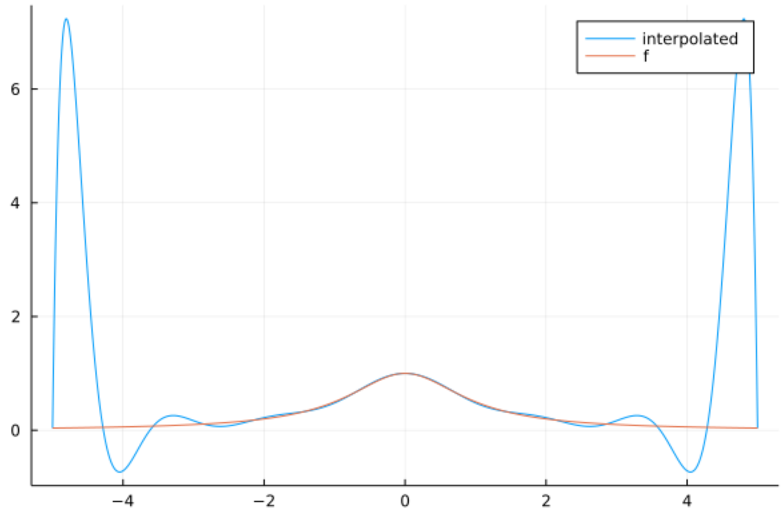
Rysunek 13: Wykres dla $\frac{1}{1+x^2}$, $n = 5$



Rysunek 14: Wykres dla $\frac{1}{1+x^2}$, $n = 10$



Rysunek 15: Wykres dla $\frac{1}{1+x^2}$, $n = 15$



Wnioski W tym zadaniu wybrane funkcje nie dają się poprawnie interpolować. W obydwu przypadkach zwiększenie ilości węzłów początkowo przynosi pozytywne efekty i lepszą aproksymację zadanych funkcji, lecz po kolejnym zwiększeniu liczby węzłów następuje znaczący spadek dokładności wielomianu interpolacyjnego. Jest to szczególnie widoczne na krańcach przedziałów. Ten efekt nazywa się zjawiskiem Runge'go i wynika z równomiernego rozmieszczenia węzłów na zadanym przedziale.