Sprawozdanie 4 - Obliczenia Naukowe

Michał Kallas

7 grudnia 2024

1 Zadanie 1

1.1 Opis problemu

Napisać funkcję obliczającą ilorazy różnicowe.

1.2 Opis zagadnienia i rozwiązania

Ilorazy różnicowe są kluczowym narzędziem w numerycznej analizie funkcji, szczególnie przy konstrukcji interpolacyjnego wielomianu Newtona. Służą do efektywnego obliczania współczynników wielomianu bez potrzeby rozwiązywania układów równań liniowych.

Dla danych węzłów $x_0, x_1, \ldots, x_n$ oraz odpowiadających im wartości $f(x_0), f(x_1), \ldots, f(x_n)$, ilorazy różnicowe definiujemy rekurencyjnie:

$$f[x_i] = f(x_i),$$

$$f[x_i, x_{i+1}, \ldots, x_{i+k}] = \frac{f[x_{i+1}, \ldots, x_{i+k}] - f[x_i, \ldots, x_{i+k-1}]}{x_{i+k} - x_i}, \quad k \geqslant 1.$$

Algorytm polega na stopniowym aktualizowaniu tablicy wartości f_x, która na początku zawiera wartości funkcji $f(x_i)$. W procesie aktualizacji obliczamy kolejne ilorazy różnicowe, przechowując je w miejscu wyjściowych wartości. Złożoność czasowa algorytmu to $O(n^2)$.

1.3 Pseudokod

Algorytm 1 ilorazyRoznicowe

Dane: x – wektor długości $n+1$ zawierający węzły $x_0, \ldots, x_n$, f – wektor długości $n+1$ zawierający wartości funkcji w węzłach
Wynik: f_x – wektor długości $n+1$ zawierający obliczone ilorazy różnicowe

```
1: f_x ← f
2: for i ← 2 to n do
3:     for j ← n downto i do
4:         f_x[j] ← (f_x[j] − f_x[j − 1])/(x[j] − x[j − i + 1])
5:     end for
6: end for
7: return f_x
```

2 Zadanie 2

2.1 Opis problemu

Napisać funkcję obliczającą 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 Opis zagadnienia i rozwiązania

Dla danych węzłów $x_0, x_1, \ldots, x_n$ oraz odpowiadających im wartości $f(x_0), f(x_1), \ldots, f(x_n)$, wielomian interpolacyjny Newtona definiujemy jako:

$$N_n(x) = f[x_0]+f[x_0,x_1](x-x_0)+f[x_0,x_1,x_2](x-x_0)(x-x_1)+\cdots+f[x_0,\ldots,x_n](x-x_0)(x-x_1)\cdots(x-x_{n-1}),$$

Lub, w zwartej formie:

$$N_n(x) = \sum_{i=0}^{n} f[x_0, x_1, \ldots, x_i] \prod_{j=0}^{i-1} (x - x_j),$$

gdzie $f[x_0, x_1, \ldots, x_i]$ to i-tego rzędu ilorazy różnicowe, a przy $i = 0$ zakładamy, że pusty iloczyn wynosi 1.

Aby obliczyć wartość wielomianu interpolacyjnego w postaci Newtona $N_n(x)$ w punkcie $x = t$, wykorzystujemy uogólniony algorytm Hornera. Algorytm Hornera umożliwia obliczanie wartości takiego wielomianu w sposób efektywny, ze złożonością czasową $O(n)$. Zamiast wyliczać poszczególne składniki wielomianu (ilorazy różnicowe i iloczyny $(t - x_i)$) niezależnie i mnożyć je krok po kroku, algorytm Hornera przekształca wzór na $N_n(x)$ w formę umożliwiającą obliczenia w sposób „odwrócony".

Rozpoczynamy od współczynnika $f[x_0, x_1, \ldots, x_n]$, który odpowiada najwyższemu stopniowi wielomianu, i rekurencyjnie obliczamy wartość wielomianu, dodając kolejne składniki od końca. W każdej iteracji wynik obliczeń z poprzedniego kroku jest mnożony przez $(t - x_i)$, a następnie dodawany jest kolejny współczynnik.

2.3 Pseudokod

Algorytm 2 warNewton

Dane: x – wektor długości $n+1$ zawierający węzły $x_0, \ldots, x_n$, f_x – wektor długości $n+1$ zawierający ilorazy różnicowe, t – punkt, w którym należy obliczyć wartość wielomianu
Wynik: nt – wartość wielomianu w punkcie t

```
1: nt ← f_x[n]
2: for i ← n − 1 downto 1 do
3:     nt ← nt · (t − x[i]) + f_x[i]
4: end for
5: return nt
```

3 Zadanie 3

3.1 Opis problemu

Znając współczynniki wielomianu interpolacyjnego w postaci Newtona

$$c_0 = f[x_0], \quad c_1 = f[x_0, x_1], \quad c_2 = f[x_0, x_1, x_2], \quad \ldots, \quad c_n = f[x_0, \ldots, x_n] \quad \text{(ilorazy różnicowe)}$$

oraz węzły $x_0, x_1, \ldots, x_n$, napisać funkcję obliczającą, w czasie $O(n^2)$, współczynniki jego postaci naturalnej

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0.$$

3.2 Opis zagadnienia i rozwiązania

Wielomian interpolacyjny w postaci Newtona można przekształcić do postaci naturalnej, czyli:

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = \sum_{i=0}^{n} a_i x^i.$$

Bardzo istotną obserwacją jest to, że współczynnik $c_n = f_x[n]$ z postaci Newtona jest równy współczynnikowi a_n przy najwyższej potędze w postaci naturalnej. Kolejne współczynniki możemy wyliczać na podstawie poprzednich oraz wartości ilorazów różnicowych i węzłów, przy okazji aktualizując współczynniki przy wyższych potęgach poprzez odjęcie aktualnego węzła. Algorytm działa ze złożonością czasową $O(n^2)$.

Dla pierwszych 3 iteracji wyniki wyglądają następująco:

$$a_n = c_n$$

$$a_{n-1} = c_{n-1} + c_n \cdot \sum_{i=0}^{n-1} -x_i$$

$$a_{n-2} = c_{n-2} + c_{n-1} \cdot \sum_{i=0}^{n-2} -x_i + c_n \cdot \sum_{i=j=0, i<j}^{n-1} (-x_i)(-x_j)$$

3.3 Pseudokod

Algorytm 3 naturalna

Dane: x – wektor długości $n+1$ zawierający węzły $x_0, \ldots, x_n$, f_x – wektor długości $n+1$ zawierający ilorazy różnicowe

Wynik: a – wektor długości $n+1$ zawierający obliczone współczynniki postaci naturalnej

```
1: a ← wektor zer o rozmiarze n
2: a[n] ← f_x[n]
3: for i ← n − 1 downto 1 do
4:     a[i] ← f_x[i] − x[i] · a[i + 1]
5:     for j ← i + 1 to n − 1 do
6:         a[j] ← a[j] − x[i] · a[j + 1]
7:     end for
8: end for
9: return a
```

4 Zadanie 4

4.1 Opis problemu

Napisać funkcję `rysujNnfx`, która zinterpoluje zadaną funkcję $f(x)$ w przedziale $[a, b]$ za pomocą wielomianu interpolacyjnego stopnia n w postaci Newtona. Następnie narysuje wielomian interpolacyjny i interpolowaną funkcję

4.2 Rozwiązanie

W funkcji `rysujNnfx` na początku wyznaczyłem węzły interpolacyjne x_k, które są równomiernie rozmieszczone w przedziale $[a, b]$ z określoną odległością $h = \frac{b-a}{n}$. Węzły są obliczane w następujący sposób:

$$x_k = a + k \cdot h \quad \text{dla} \quad k = 0, 1, \ldots, n.$$

Po obliczeniu węzłów, obliczyłem wartości funkcji w tych punktach($f(x_k)$). Wartości te zostały następnie użyte do wyznaczenia ilorazów różnicowych za pomocą funkcji `ilorazyRoznicowe`.

Kolejnym krokiem było obliczenie wartości wielomianu interpolacyjnego oraz funkcji f w punktach, które zostały zadane do rysowania wykresu. Dla x_i, gdzie x_i to punkty w przedziale $[a, b]$, obliczane są wartości wielomianu interpolacyjnego za pomocą funkcji `warNewton`, a także $f(x_i)$. W celu otrzymania dokładniejszego wykresu, obliczam te wartości dla $100n$ punktów w przedziale $[a, b]$.

Po wyliczeniu wartości wielomianu interpolacyjnego oraz funkcji f, rysowany jest wykres.

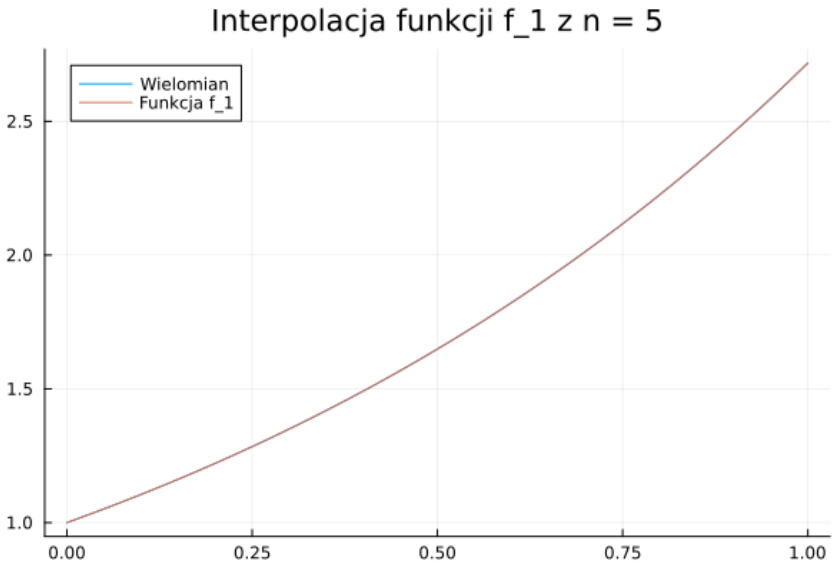
5 Zadanie 5

5.1 Opis problemu

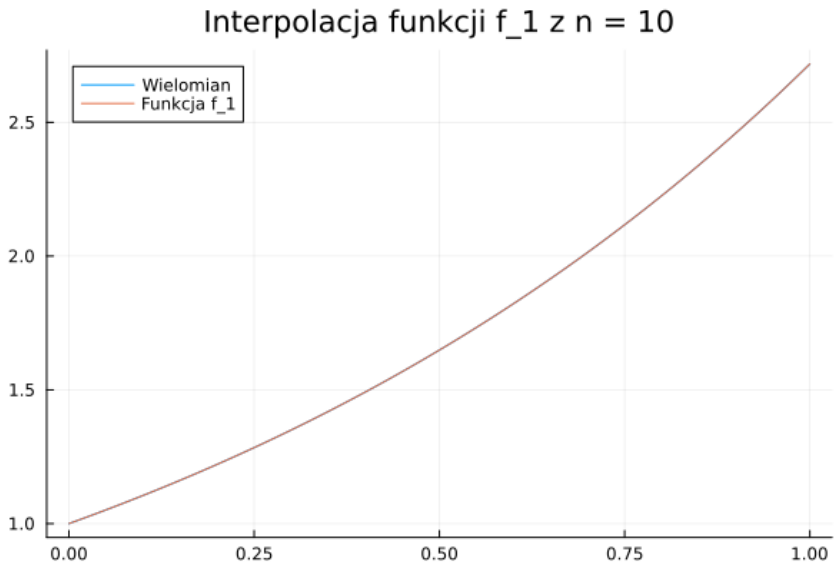
Przetestować funkcję `rysujNnfx(f,a,b,n)` na następujących przykładach:

- (a) $f_1(x) = e^x$, $[0, 1]$, $n = 5, 10, 15$,
- (b) $f_2(x) = x^2 \sin x$, $[-1, 1]$, $n = 5, 10, 15$.

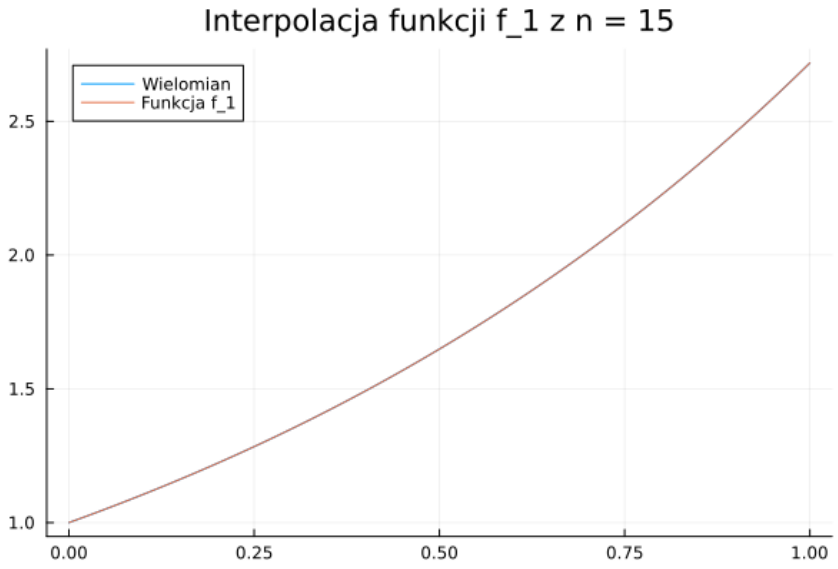
5.2 Wykresy



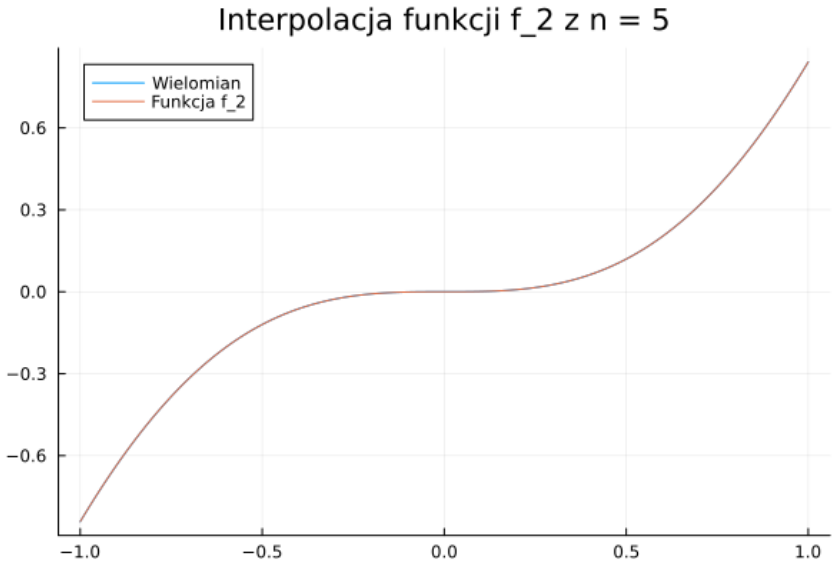
Rysunek 1: Wykres uzyskany przez funkcję `rysujNnfx(`e^x`, 0, 1, 5)`.



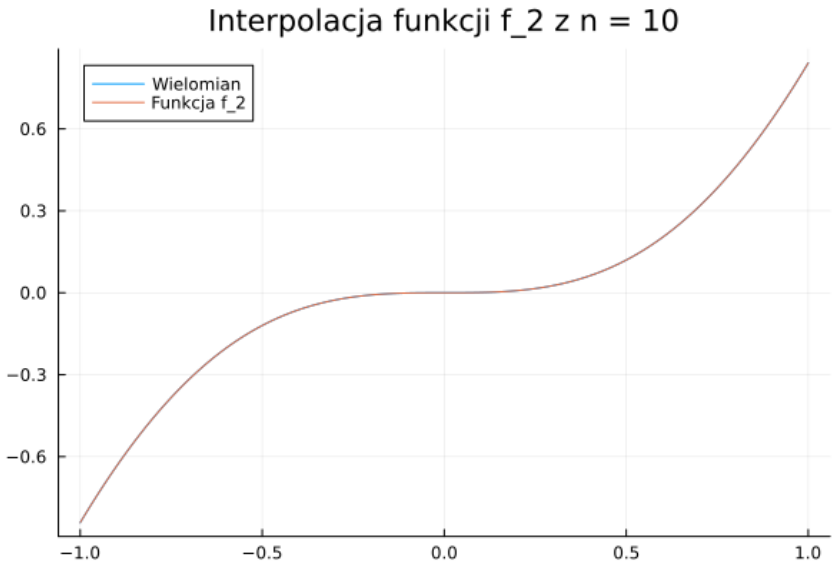
Rysunek 2: Wykres uzyskany przez funkcję `rysujNnfx(`e^x`, 0, 1, 10)`.



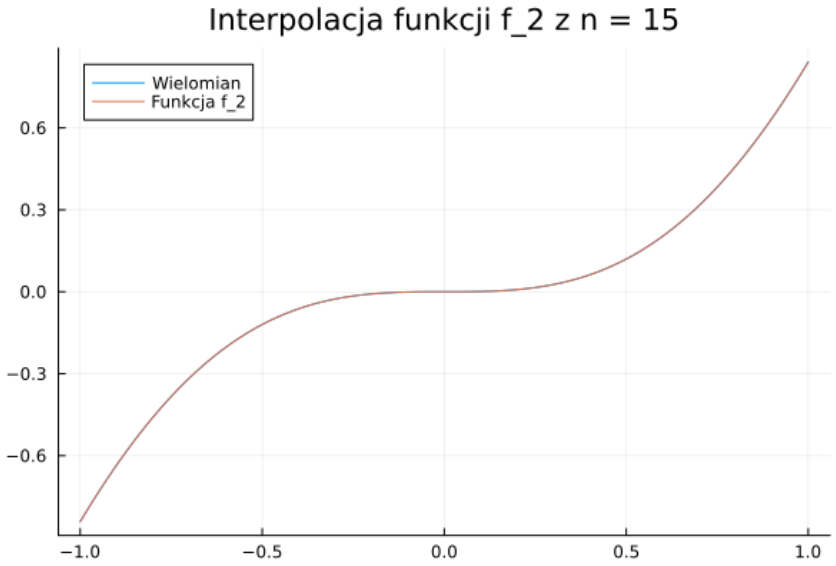
Rysunek 3: Wykres uzyskany przez funkcję `rysujNnfx(`e^x`, 0, 1, 15)`.



Rysunek 4: Wykres uzyskany przez funkcję `rysujNnfx(`$x^2 \sin x$`, -1, 1, 5)`.



Rysunek 5: Wykres uzyskany przez funkcję `rysujNnfx(`$x^2 \sin x$`, -1, 1, 10)`.



Rysunek 6: Wykres uzyskany przez funkcję `rysujNnfx(`$x^2 \sin x$`, -1, 1, 15)`.

5.3 Obserwacje i wnioski

Jak widać na wykresach, zarówno dla f_1, jak i f_2 interpolacja działa bardzo dobrze. Wartości wielomianów interpolacyjnch wszystkich sprawdzanych stopni pokrywają się z wartościami funkcji na całych przedziałach.

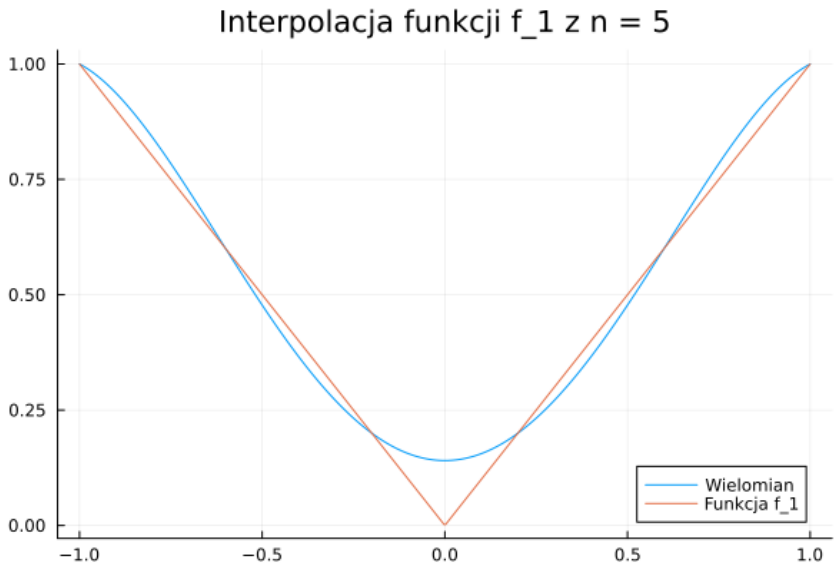
6 Zadanie 6

6.1 Opis problemu

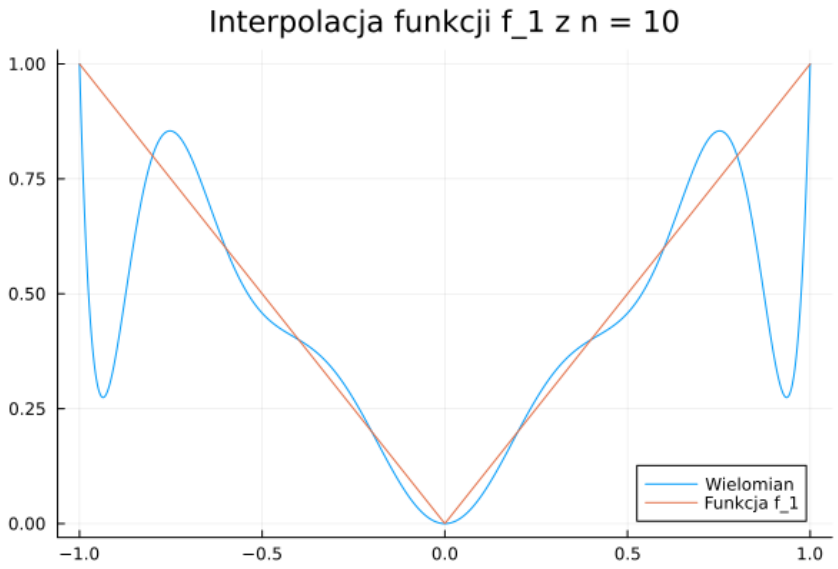
Przetestować funkcję `rysujNnfx(f,a,b,n)` na następujących przykładach (zjawisko rozbieżności):

- (a) $f_1(x) = |x|$, $[-1, 1]$, $n = 5, 10, 15$,
- (b) $f_2(x) = \frac{1}{1+x^2}$, $[-5, 5]$, $n = 5, 10, 15$ (zjawisko Runge'go).

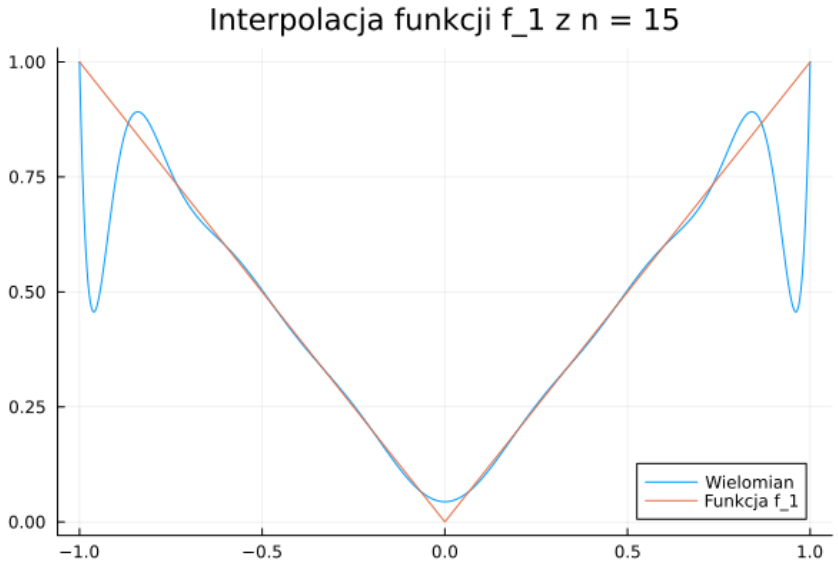
6.2 Wykresy



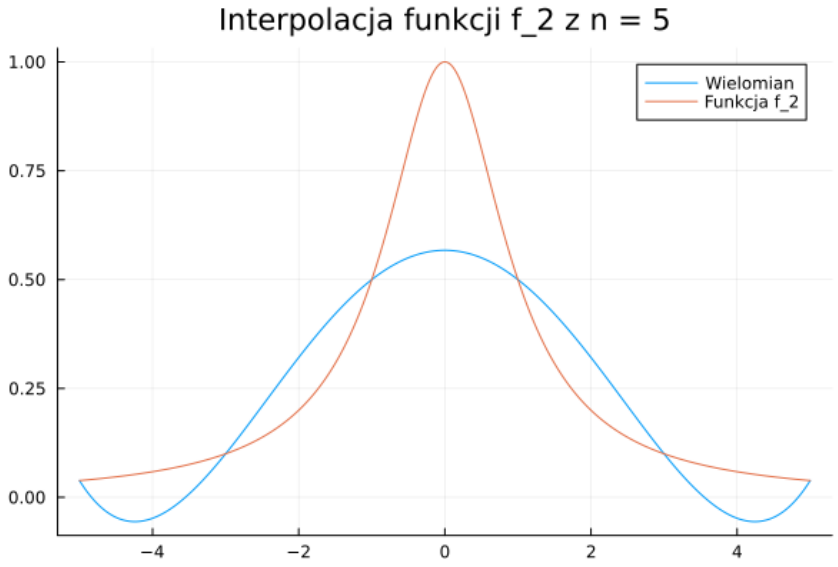
Rysunek 7: Wykres uzyskany przez funkcję `rysujNnfx(|x|, -1, 1, 5)`.



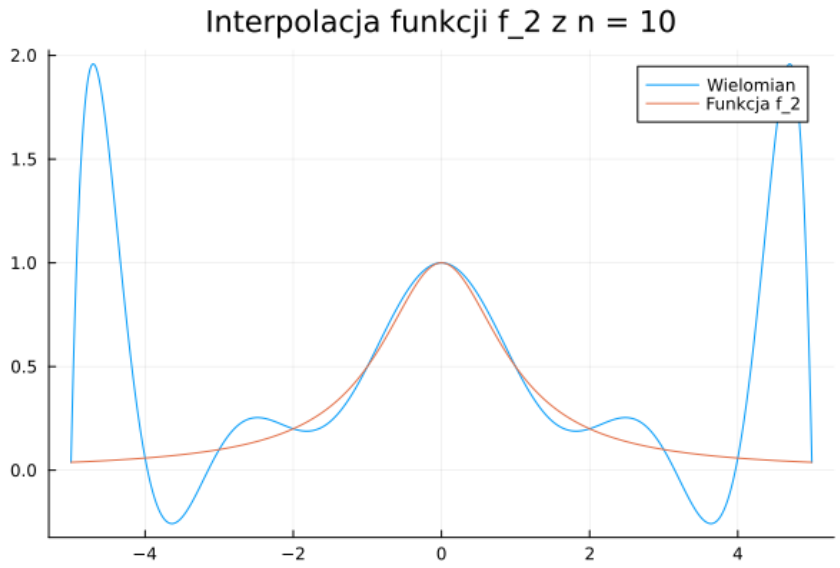
Rysunek 8: Wykres uzyskany przez funkcję `rysujNnfx(|x|, -1, 1, 10)`.



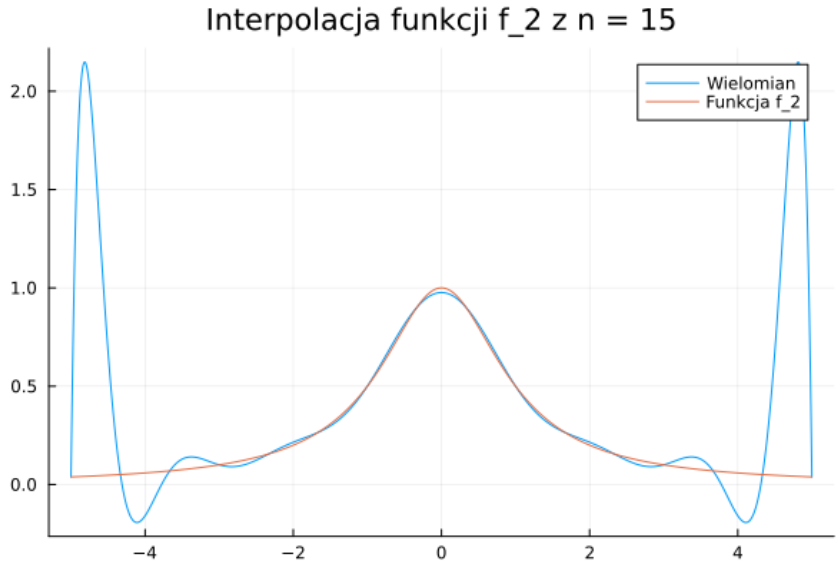
Rysunek 9: Wykres uzyskany przez funkcję `rysujNnfx(`$|x|$`, -1, 1, 15)`.



Rysunek 10: Wykres uzyskany przez funkcję `rysujNnfx(`$\frac{1}{1+x^2}$`, -5, 5, 5)`.



Rysunek 11: Wykres uzyskany przez funkcję `rysujNnfx(`$\frac{1}{1+x^2}$`, -5, 5, 10)`.



Rysunek 12: Wykres uzyskany przez funkcję `rysujNnfx(`$\frac{1}{1+x^2}$`, -5, 5, 15)`.

6.3 Obserwacje i wnioski

W tym przypadku, zarówno dla f_1, jak i f_2 interpolacja nie zadziałała tak dobrze. Zauważalne są różnice pomiędzy wartościami funkcji, a wielomianu interpolowanego.

Przy zwiększeniu stopnia wielomianu dokładność interpolacji maleje, co jest szczególnie widoczne na krańcach przedziału. Jest to całkiem nieoczywiste zjawisko, jako że dostajemy gorsze wyniki, mimo większej ilości węzłów. Intuicyjnie, większy stopień powinien prowadzić do lepszego przybliżenia.

Taką sytuację określa się mianem zjawiskia Runge'go. Błąd wielomianu interpolacyjnego wyliczonego za pomocą równoodległych węzłów staje się coraz większy wraz ze wzrostem jego stopnia.

Dobrym rozwiązaniem tego problemu są węzły Czebyszewa. Dzięki nim, występuje więcej węzłów w miejscach, które są trudne do przybliżenia, co zwiększa jego dokładność.