Obliczenia Naukowe

Laboratorium 3

Bartosz Grelewski

Grudzień 2022

1 Zadanie 1

1.1 Opis Problemu

Napisać funkcję obliczającą ilorazy różnicowe.

function ilorazyRoznicowe (x::VectorFloat64, f::VectorFloat64)

Dane:

$$x - \text{wektor długości } n+1 \text{ zawierający węzły } x_0, ..., x_n \; x[1] = x_0, ..., x[n+1] = x_n$$
$$f - \text{wektor długości } n+1 \text{ zawierający wartości interpolowanej funkcji w węzłach } f(x_0), ..., f(x_n)$$

Wyniki:

$$fx - \text{wektor długości } n+1 \text{ zawierający obliczone ilorazy różnicowe } fx[1] = f[x_0], fx[2] = f[x_0, x_1], ..., fx[n] = f[x_0, ..., x_{n-1}], fx[n+1] = f[x_0, ..., x_n].$$

W celu wyznaczenia postaci Newtona wielomianu interpolacyjnego należy obliczyć ilorazy różnicowe.

Iloraz różnicowy k-tego rzędu zadany jest poniższą zależnością rekurencyjną:

$$f[x_0, x_1, \dots, x_k] = \frac{f[x_1, x_2, \dots, x_k] - f[x_0, x_1, \dots, x_{k-1}]}{x_k - x_0}$$
$$f[x_0] = f(x_0)$$
$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

1.2 Rozwiązanie

Celem listy jest zbadanie dokładności i efektywności przybliżeń funkcji przy pomocy interpolacji wielomianowej. Wykorzystanie tej metody pozwala na oszacowanie wartości funkcji, gdy znamy jej wartości tylko w wybranych punktach.

Znając węzły x_k i wartości funkcji $f(x_k)$ (czyli również ilorazy $f[x_k]$ zerowego rzędu) można za pomocą tego wzoru utworzyć macierz ilorazów różnicowych wyższych rzędów. Zapisywanie całej macierzy w pamięci jest jednak nieoptymalne. Do reprezentacji macierzy ilorazów różnicowych wystarczy użyć jednowymiarowej tablicy t. Początkowymi wartościami zmiennych t_i są odpowiadające im $f[t_i]$, czyli każde t_i obliczane jest ze wzoru:

$$t_i = \frac{t_i - t_{i-1}}{x_i - x_{i-j}},$$

gdzie j oznacza numer kolumny w macierzy. Każda kolejna wartość uaktualniana jest kolumnami, zaś wewnątrz każdej kolumny - z dołu do góry. Dzięki takiej kolejności obliczeń tablica t zawiera w każdej chwili ilorazy, które będą potrzebne w kolejnych krokach algorytmu. Funkcja znajduje się w pliku "zad1.jl".

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 $O(n)$, (implementacja algorytmu z zadania 8 lista nr 4 – ćwiczenia).
Dane:

$$x - \text{wektor długości } n+1 \text{ zawierający węzły } x_0, ..., x_n \; x[1] = x_0, ..., x[n+1] = x_n$$
$$fx - \text{wektor długości } n+1 \text{ zawierający ilorazy różnicowe } fx[1] = f[x_0],$$
$$fx[2] = f[x_0, x_1], ..., fx[n] = f[x_0, ..., x_{n-1}], fx[n+1] = f[x_0, ..., x_n]$$
$$t - \text{punkt, w którym należy obliczyć wartość wielomianu.}$$

Wyniki:

$$nt - \text{wartość wielomianu w punkcie } t$$

2.2 Rozwiązanie

W tym wypadku musimy wyznaczyć wartość naszego wielomianu w postaci Newtona w danym punkcie. Jednak aby wykonać to w złożoności O(n) musimy zacząć obliczać nasz wielomian od środka od najbardziej zagnieżdżonych elementów. Czyli zastosować schemat Hornera. Obliczenie ilorazów różnicowych pozwala nam wyznaczyć wielomian interpolacyjny Newtona zadany wzorem:

$$N_n(x) = \sum_{i=0}^{k} c_i \prod_{j=0}^{i-1} (x - x_j)$$

gdzie c_i jest ilorazem różnicowym stopnia i, zaś x_j - węzłem interpolacji. Korzystamy z postaci Newtona, która pomimo bardziej skomplikowanej implementacji jest lepiej uwarunkowana numerycznie niż postać LaGrange'a.
Stosujemy uogólniony schemat Hornera.

$$\begin{aligned}
&w_n(x) := f[x_0, x_1, \ldots, x_n] \\
&w_k(x) := w_{k+1}(x - x_k) + f[x_0, x_1, \ldots, x_k] \quad (k = n-1, n-2, \ldots, 0) \\
&N_n(x) = w_0(x)
\end{aligned}$$

Skorzystałem ze wzorów z listy 4 z zadania 8. Algorytm znajduje się w pliku "zad2.jl".

3 Zadanie 3

3.1 Opis Problemu

Znając współczynniki wielomianu interpolacyjnego w postaci Newtona $c_0 = f[x_0], c_1 = f[x_0, x_1], c_2 = f[x_0, x_1, x_2], ..., c_n = f[x_0, ..., x_n]$ (ilorazy różnicowe) oraz węzły $x_0, x_2, ..., x_n$ napisać funkcję obliczającą, w czasie $O(n^2)$, współczynniki jego postaci naturalnej $a_0, ..., a_n$ tzn. $a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + ... + a_1 x + a_0$ (implementacja algorytmu z zadania9 lista nr 4−ćwiczenia).

function naturalna (x::VectorFloat64, fx::VectorFloat64)

Dane:

x − wektor długości $n + 1$ zawierający węzły $x_0, ..., x_n$ $x[1] = x_0, ..., x[n + 1] = x_n$
fx − wektor długości $n + 1$ zawierający ilorazy różnicowe $fx[1] = f[x_0]$,
$fx[2] = f[x_0, x_1], ..., fx[n] = f[x_0, ..., x_{n-1}], fx[n + 1] = f[x_0, ..., x_n]$
t − punkt, w którym należy obliczyć wartość wielomianu.

Wyniki:

a − wektor długości $n + 1$ zawierający obliczone współczynniki postaci naturalnej $a[0] = a_0$,
$a[2] = a_1, ..., a[n] = a_{n-1}, a[n + 1] = a_n$.

3.2 Rozwiązanie

Aby napisać funkcję obliczającą współczynniki postaci naturalnej wielomianu interpolacyjnego Newtona w czasie $O(n^2)$ musimy zastanowić się nad tym co już wiemy o Schemacie Hornera. Wykorzystam uogólnione wzory Hornera z poprzedniego zadania. Na samym początku możemy skorzystać z zależności, że $a_n = c_n$. Następnie wyliczamy kolejne wartości częściowe dla wielomianu interpolacyjnego ale przy wykorzystaniu faktu, że przy każdej naszej iteracji każdy składowy wielomian jest doprowadzany do postaci naturalnej. Czyli w ten sposób uzyskujemy kolejne i kolejne współczynniki dla postaci naturalnej wielomianu interpolacyjnego.

4 Zadanie 4

4.1 Opis Problemu

Napisać funkcję, 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. Do rysowania zainstaluj np. pakiet Plots, PyPlot lub Gadfly.

W interpolacji użyć węzłów równoodległych, tj. $x_k = a + kh, h = (b - a)/n, k = 0, 1, ..., n$.

Nie wyznaczać wielomianu interpolacyjnego w jawnej postaci. Należy skorzystać z funkcji

ilorazyRoznicowe i warNewton.

function rysujNnfx(f,a::Float64,b::Float64,n::Int)

4.2 Rozwiązanie

Na starcie musimy utworzyć wektor n+1 równoległych węzłów orad drugi wektor wartości funkcji w tych węzłach. Korzystając z nich jesteśmy w stanie wyliczyć ilorazy różnicowe korzystając z uprzednio napisanych przez nas metod. Później generujemy dwa wektory wartości funkcji oraz wartości wielomianu interpolacyjnego (war Newton()). na podstawie tych dwóch wektorów rysujemy wykres.

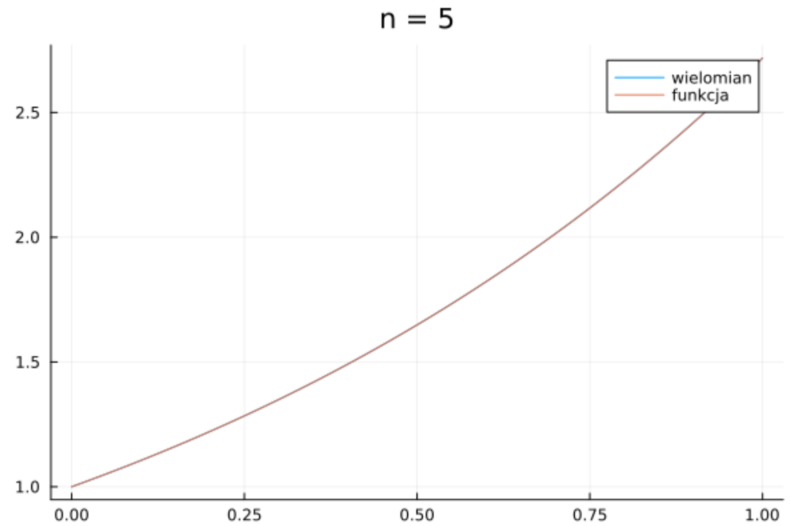
5 Zadanie 5

5.1 Opis Problemu

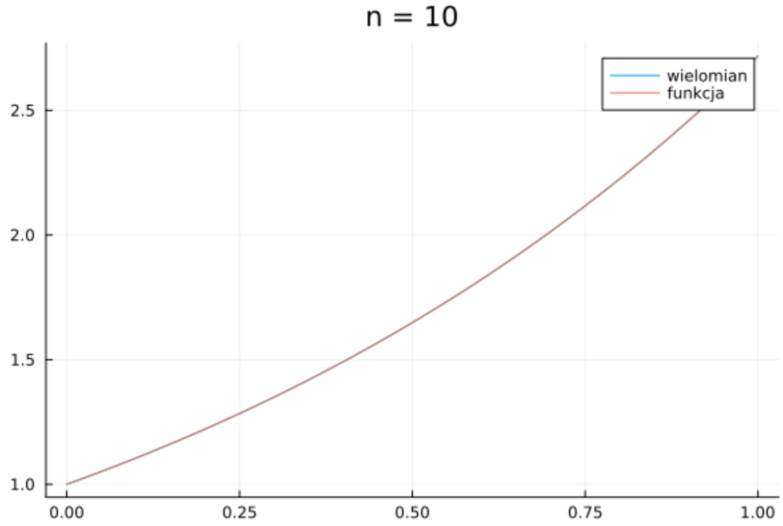
Mamy przetestować rysowanie wykresów dla następujących przykładów:

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

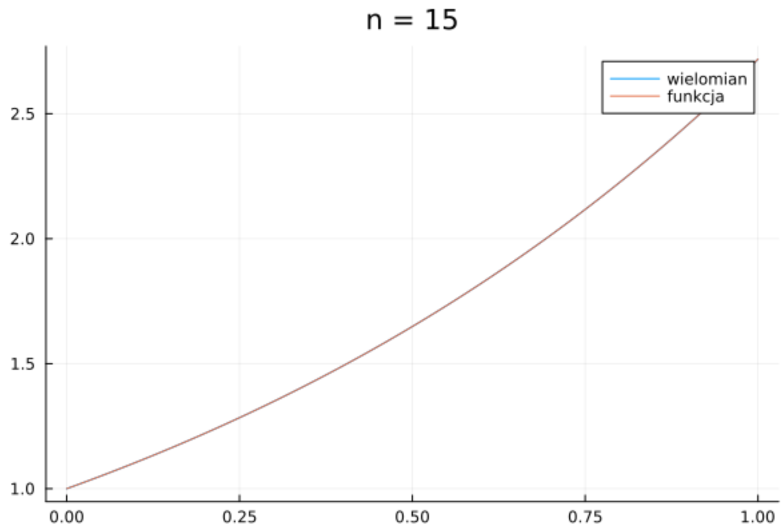
5.2 Rozwiązanie



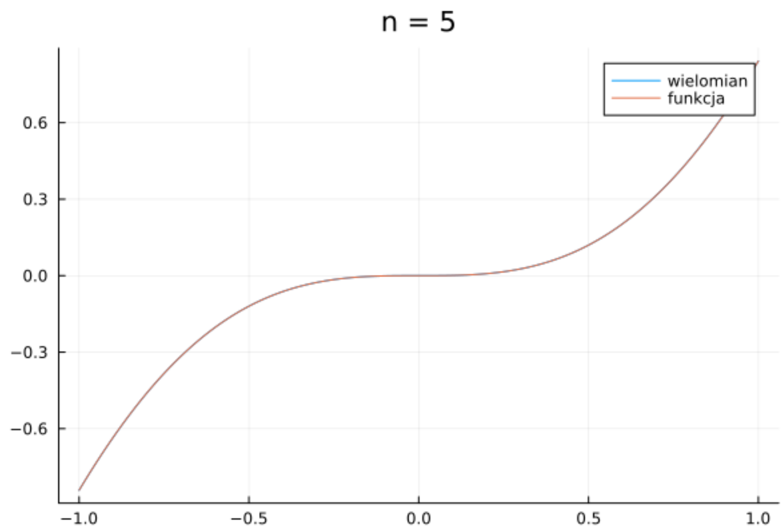
e^x, $[-1, 1]$


n = 10
wielomian
funkcja
2.5
2.0
1.5
1.0
0.00
0.25
0.50
0.75
1.00

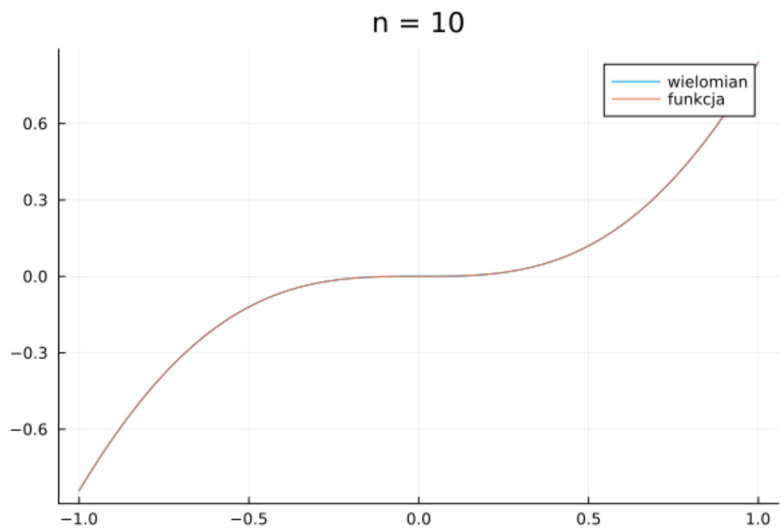
e^x, $[-1, 1]$


n = 15
wielomian
funkcja
2.5
2.0
1.5
1.0
0.00
0.25
0.50
0.75
1.00

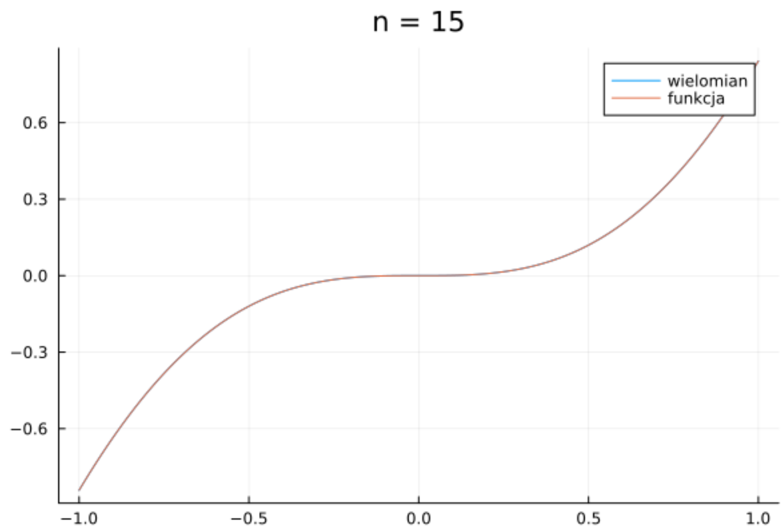
e^x, $[-1, 1]$



$x^2 sin(x)$, $[-1, 1]$



$x^2 sin(x)$, $[-1, 1]$



$x^2 sin(x)$, $[-1, 1]$

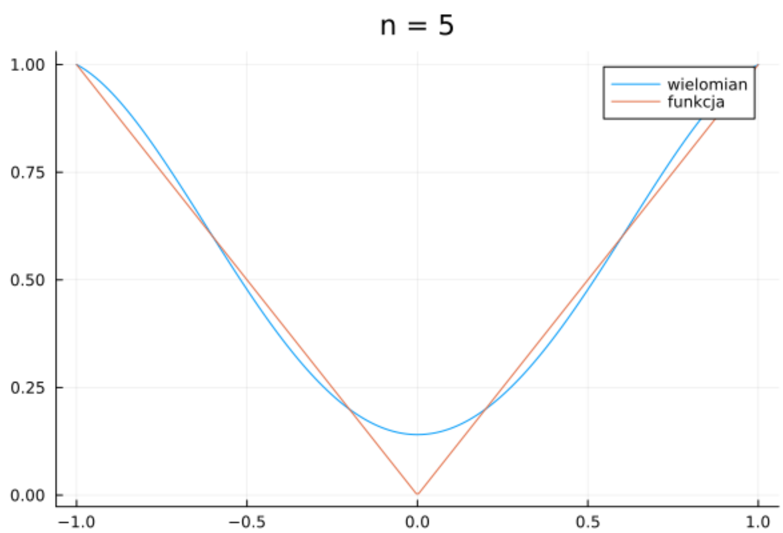
6 Zadanie 6

6.1 Opis Problemu

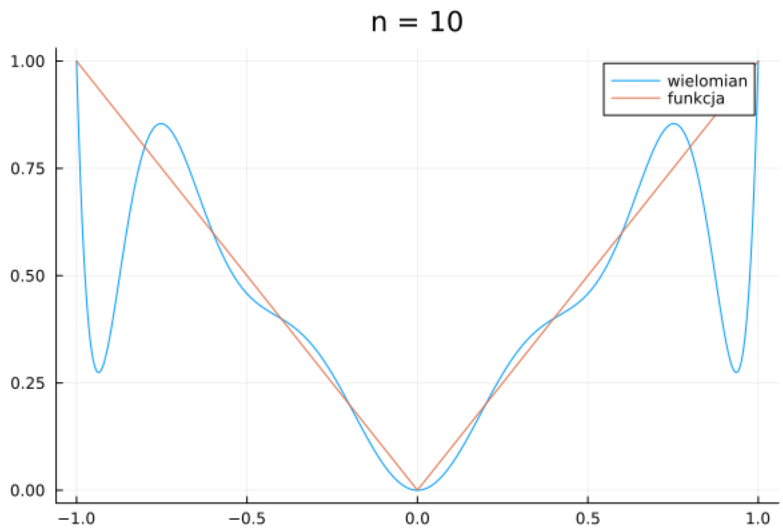
Mamy przetestować rysowanie wykresów dla następujących przykładów:

(a) $|x|$, $[-1, 1]$, n = 5,10,15 (b) $1/(1 + x^2)$, $[-5, 5]$, n = 5,10,15.

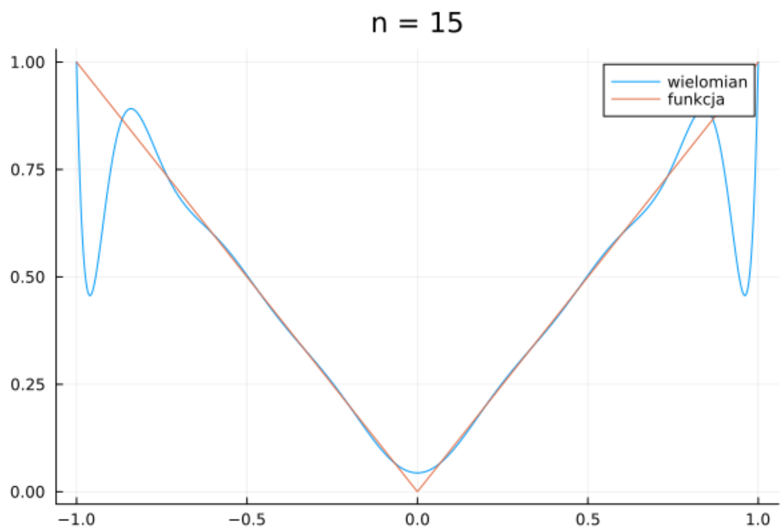
6.2 Rozwiązanie



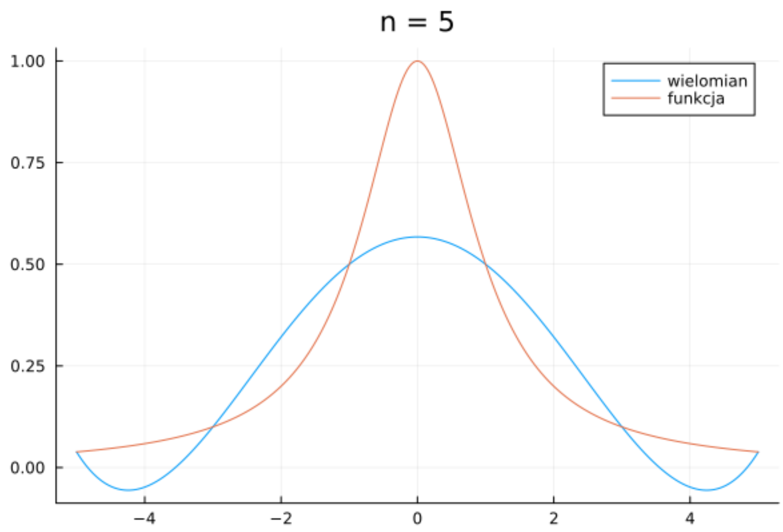
$|x|$, $[-1, 1]$


n = 10
wielomian
funkcja
1.00
0.75
0.50
0.25
0.00
−1.0
−0.5
0.0
0.5
1.0

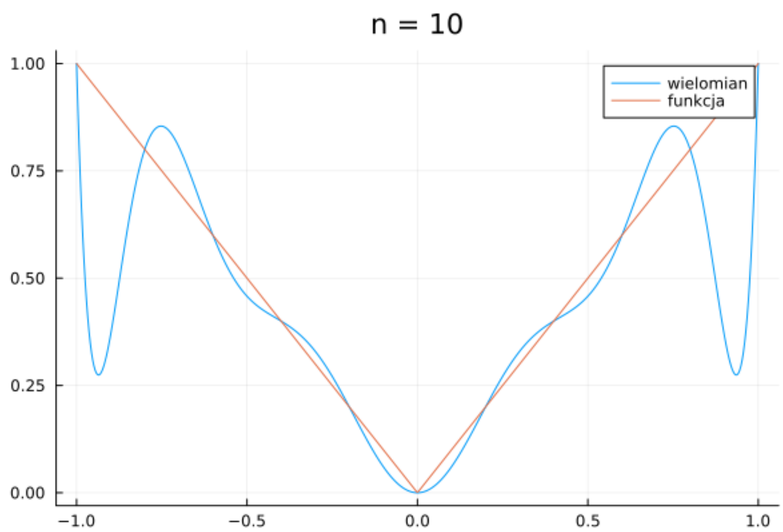
$|x|$, $[-1, 1]$


n = 15
wielomian
funkcja
1.00
0.75
0.50
0.25
0.00
−1.0
−0.5
0.0
0.5
1.0

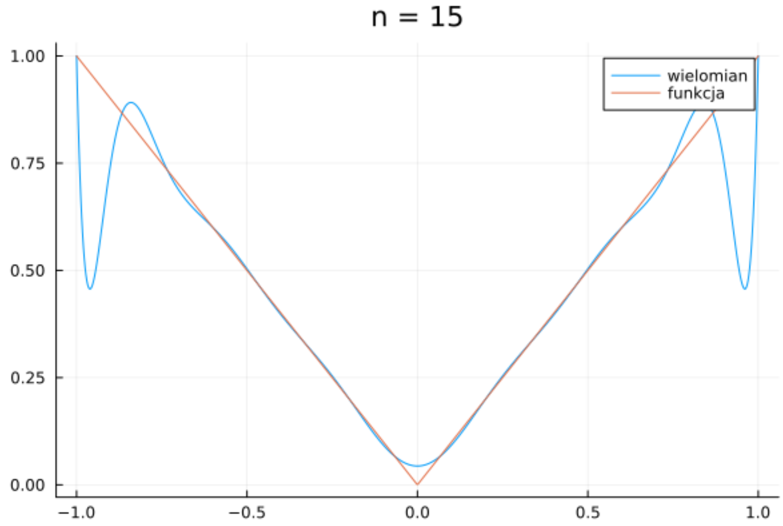
$|x|$, $[-1, 1]$



$1/(1+x^2)$, $[-5,5]$



$1/(1+x^2)$, $[-5,5]$



$1/(1+x^2)$, $[-5,5]$