Sprawozdanie

Kajetan Bilski 244942

2 grudnia 2019

1 Zadanie 1.

W tym zadaniu trzeba zaimplementować funkcję liczącą ilorazy różnicowe funkcji na podstawie jej wartośći w wybranych punktach według podanej specyfikacji. Algorytm liczący nie może wykorzystywać macierzy (tablicy dwuwymiarowej).
Kody do zadań 1 - 4 są w pliku zad1234.jl, a testy do nich w testy.jl.
Pseudokod funkcji:

```
n ← length(x)
for i ← 1 to n do
    r[i] ← f[i]
    for k ← 1 to i − 1 do
        r[i − k] ← (r[i−k+1] − r[i−k]) / (x[i] − x[i−k])
    end
    fx[i] ← r[1]
end
return fx
```

W trakcie wykonywania się tablica r ”przesuwa się” po kolejnych przekątnych macierzy trójkątnej normalnie używanej do liczenia ilorazów różnicowych. W ten sposób rekurencyjna metoda liczenia jest zachowana, ale zmniejszamy złożoność pamięciową używając tablicy o maksymalnym rozmiarze n zamiast macierzy.

2 Zadanie 2.

W tym zadaniu trzeba napisać według specyfikacji funkcję liczącą wartości wielomianu interpolacyjnego, która za argumenty przyjmuje punkty x, ilorazy różnicowe dla tych punktów i punkt w którym szukamy wartości. Funkcja ma być implementacją algorytmu Hornera z zadania 8. z listy 4. na ćwiczenia i mieć złożoność czasową $O(n)$.
Kody do zadań 1 - 4 są w pliku zad1234.jl, a testy do nich w testy.jl.

Pseudokod funkcji:

```
n ← length(x)
nt ← fx[n]
for i ← 1 to n − 1 do
|  nt ← (t − x[n − i]) ∗ nt + fx[n − i]
end
return nt
```

Jak widać mamy tylko jedną pętlę wykonującą się $n-1$ razy, czyli złożoność czasowa to $O(n)$. Funkcja jest bezpośrednią implementacją algorytmu Hornera z listy na ćwiczenia.

3 Zadanie 3.

W tym zadaniu trzeba zaimplementować funkcję wyliczającą współczynniki postaci naturalnej wielomianu interpolacyjnego w czasie $O(n^2)$, mając punkty x i jego ilorazy różnicowe dla tych punktów.
Kody do zadań 1 - 4 są w pliku zad1234.jl, a testy do nich w testy.jl.
Pseudokod funkcji:

```
n ← length(x)
a jest tablicą n zer
a[n] ← fx[n]
for i ← 1 to n − 1 do
|  a[n − i] ← fx[n − i]
|  for k ← n − i to n − 1 do
|  |  a[k] ← a[k] − x[n − i] ∗ a[k + 1]
|  end
end
return a
```

Jak widać mamy dwie pętle, których łączny czas wykonania jest $O(n^2)$. Funkcja zaczyna od współczynnika a_n który jest równy $f[x_0, ..., x_n]$ i jest n-tą pochodną szukanego wielomianu. Następnie w każdej iteracji pierwszej pętli używa x i ilorazu różnicowego o niższym indeksie do znalezienia pochodnej wielomianu o 1 stopień niższej, aż dojdzie do stopnia 0, czyli szukanego wielomianu.

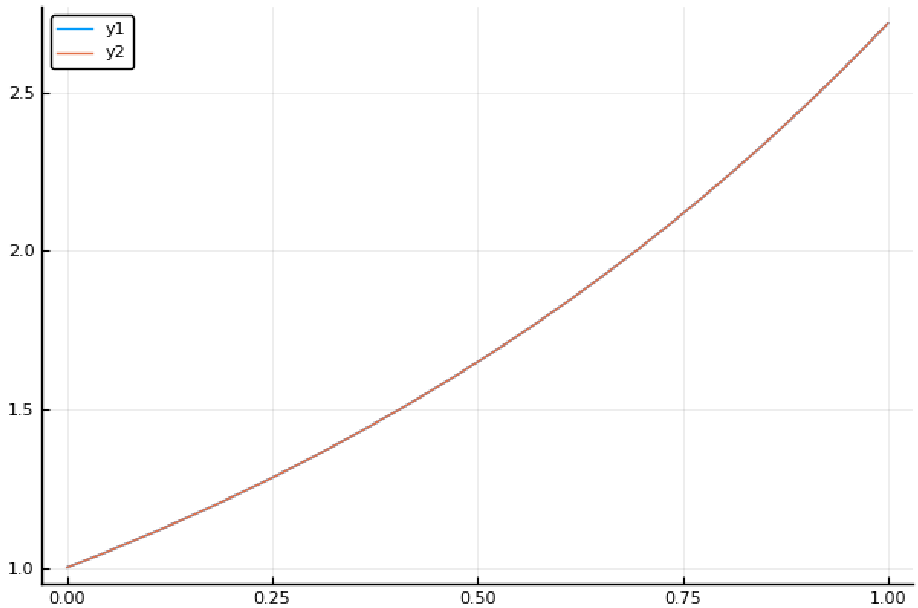
4 Zadanie 4.

W tym zadaniiu trzeba zaimplementować funkcję, która dla zadanej funkcji stworzy przybliżający ją wielomian interpolacyjny dla $n+1$ punktów w przedziale $[a, b]$, a następnie narysuje je obok siebie na wykresie używając funkcji z zadań 1 i 3.
Kody do zadań 1 - 4 są w pliku zad1234.jl, a testy do nich w testy.jl.
Pseudokod funkcji:

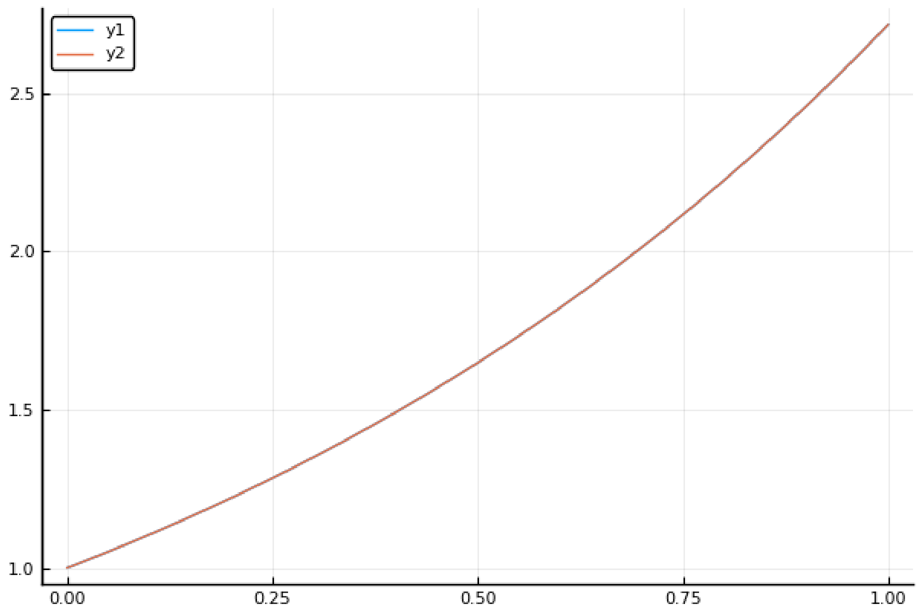
$h \leftarrow (b-a)/n$
for $i \leftarrow 0$ **to** n **do**
  $x_i \leftarrow a + i * h$
  $y_i \leftarrow f(x_i)$
end
$fx \leftarrow$ `ilorazyRoznicowe(`x,y`)`
Następnie funkcja nakłada na wykres na przedziale $z \in [a,b]$ *funkcje* $f(z)$ *i* $warNewton(x, fx, z)$.

5 Zadanie 5.

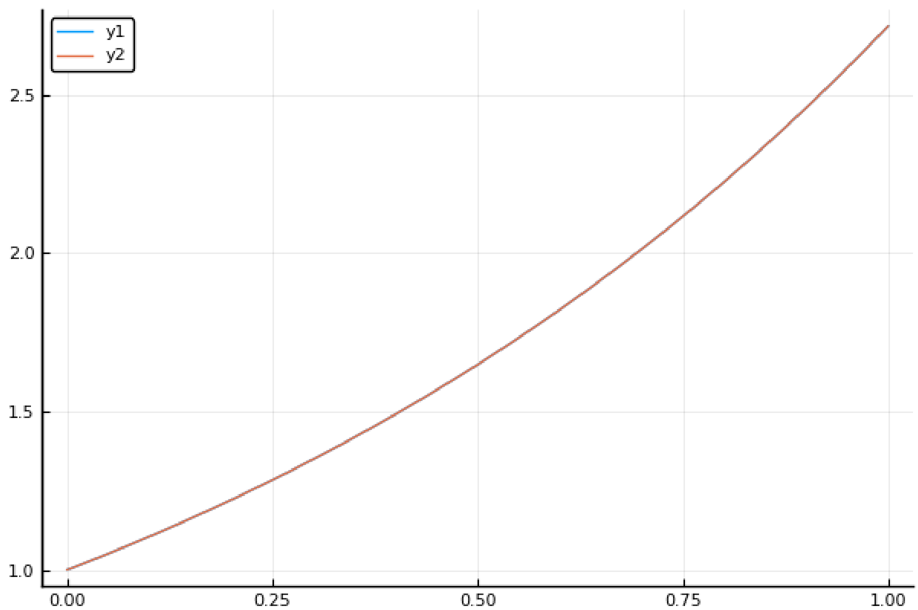
W tym zadaniu trzeba użyć funkcji `rysujNnfx` żeby narysować wykresy z podanymi danymi. Kod w pliku zad5.jl.



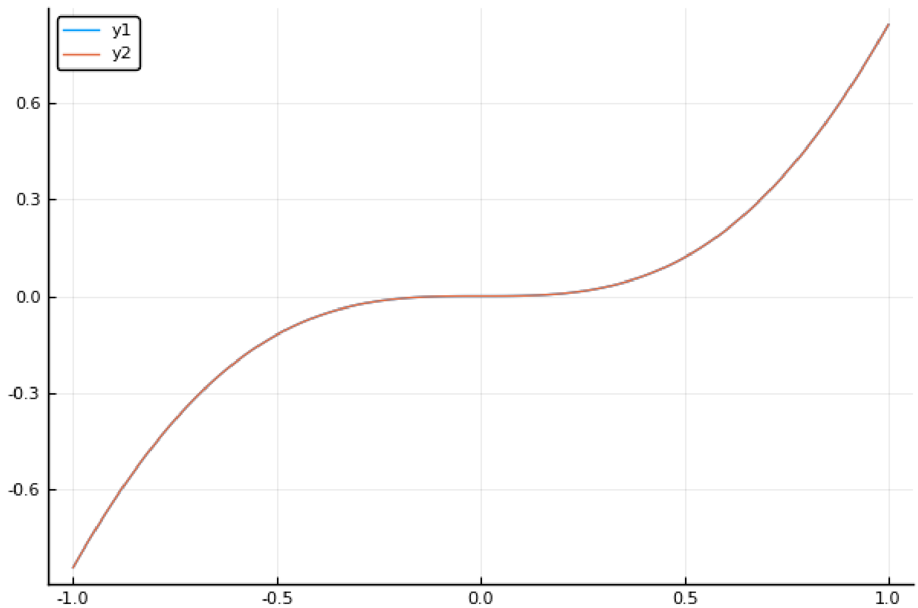
Rysunek 1: $f(x) = e^x$, $[a,b] = [0,1]$, $n = 5$



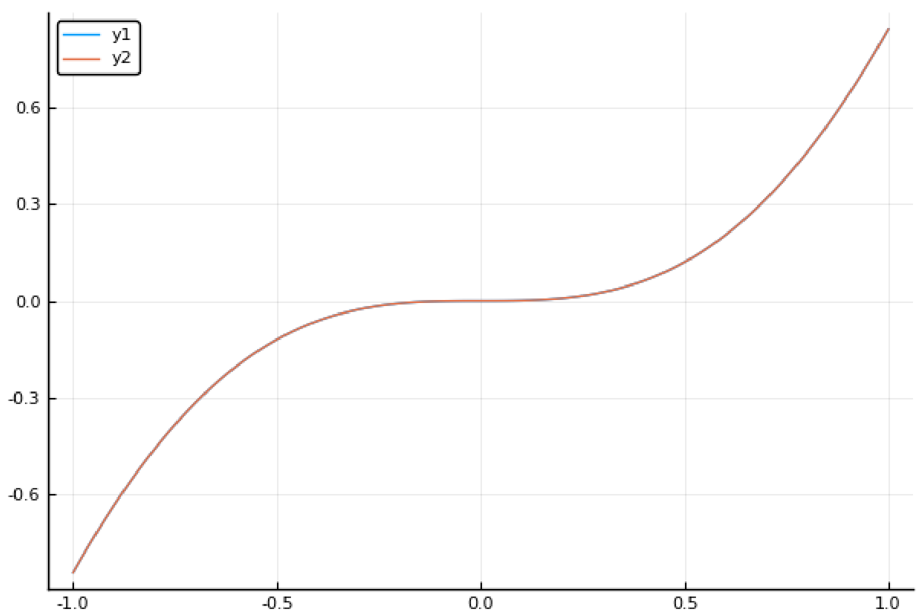
Rysunek 2: $f(x) = e^x$, $[a, b] = [0, 1]$, $n = 10$



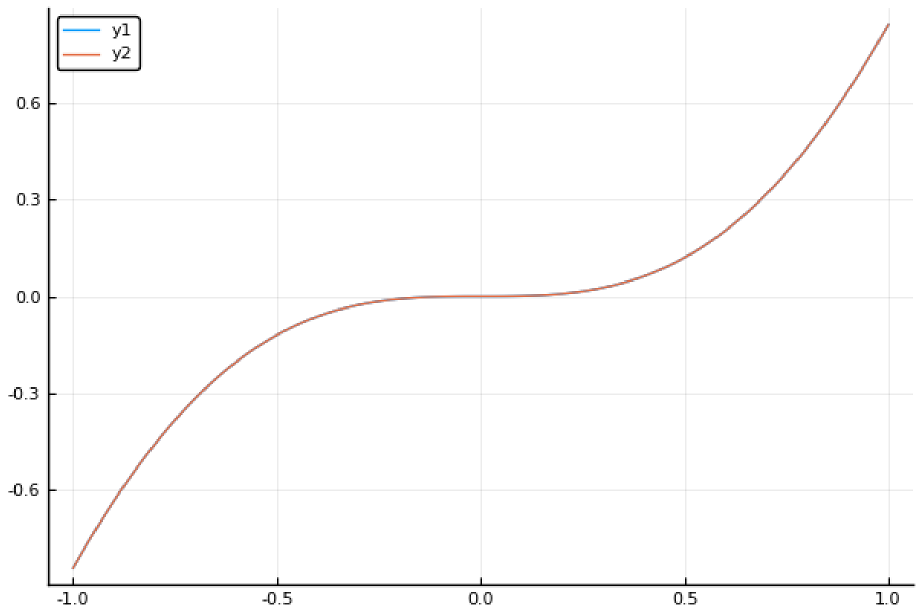
Rysunek 3: $f(x) = e^x$, $[a, b] = [0, 1]$, $n = 15$



Rysunek 4: $f(x) = x^2 sinx$, $[a, b] = [-1, 1]$, $n = 5$



Rysunek 5: $f(x) = x^2 sinx$, $[a, b] = [-1, 1]$, $n = 10$

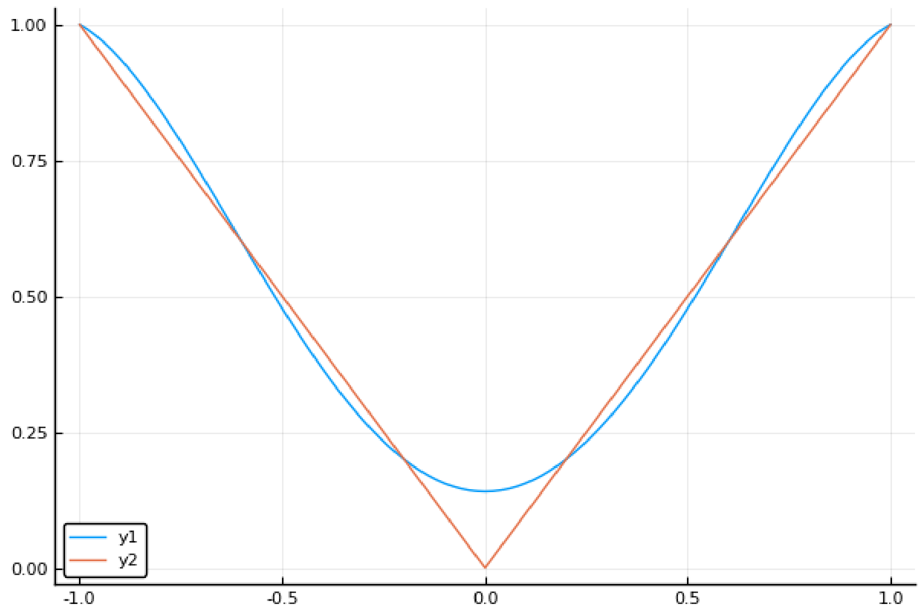


Rysunek 6: $f(x) = x^2 sinx$, $[a, b] = [-1, 1]$, $n = 15$

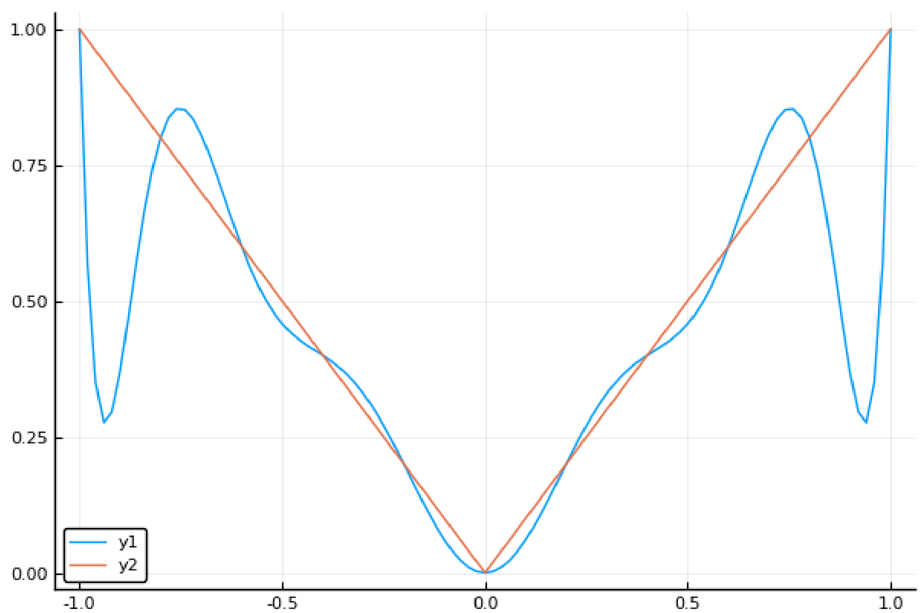
Jak widać wielomiany interpolacyjne bardzo dobrze przybliżają dane funkcje na odpowiednich przedziałach. Nie ma widocznych różnic.

6 Zadanie 6.

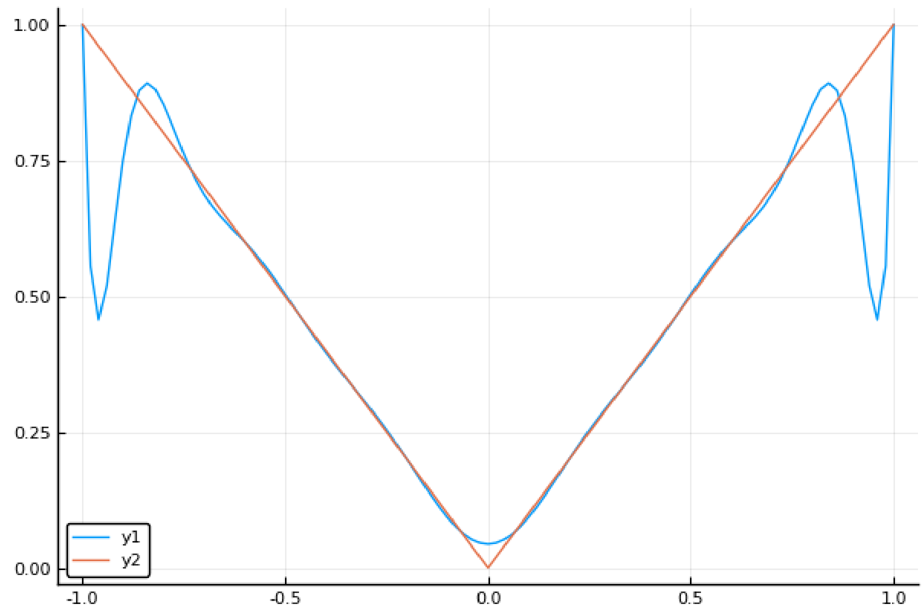
W tym zadaniu trzeba zrobić to samo co w poprzednim, tylko z innymi danymi. Kod w pliku zad6.jl.



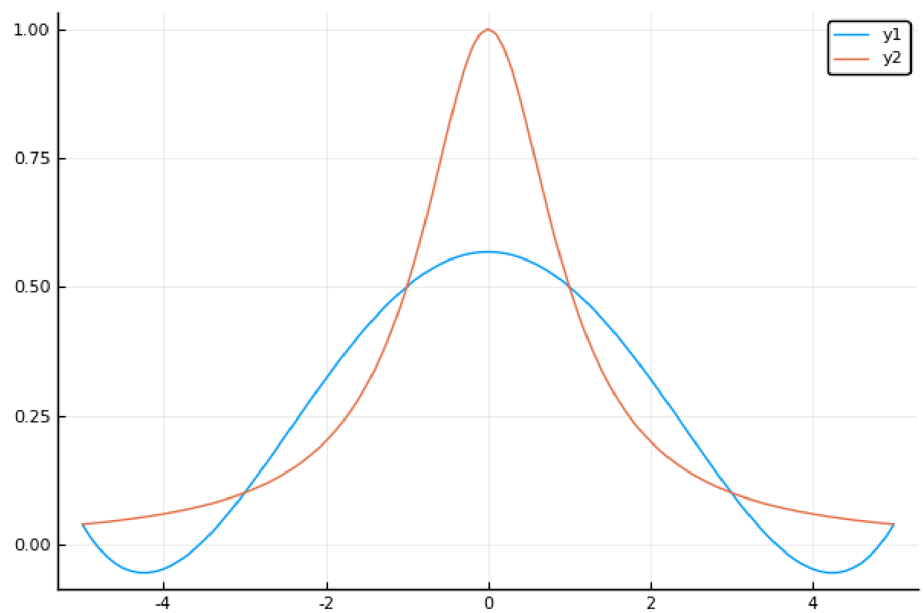
Rysunek 7: $|x|$, $[a, b] = [-1, 1]$, $n = 5$



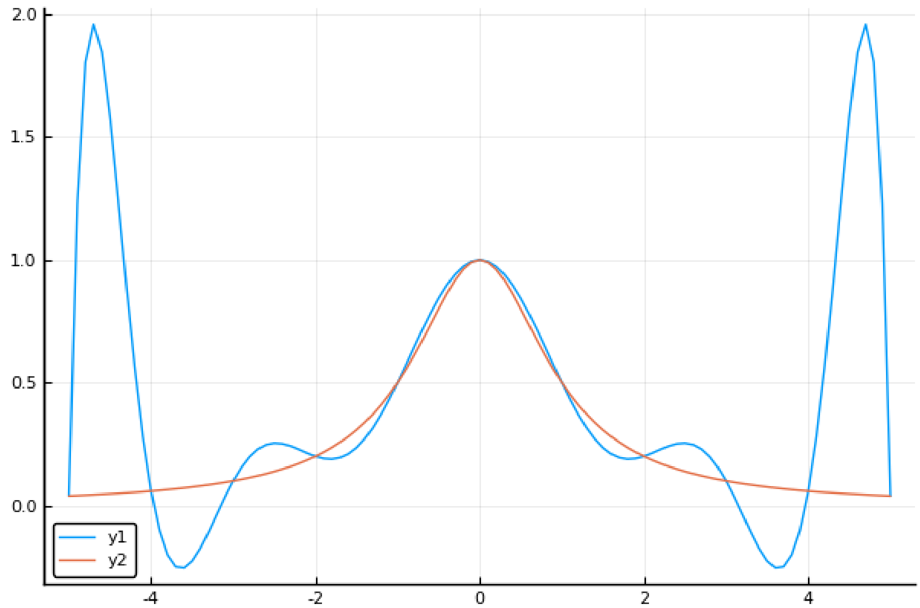
Rysunek 8: $|x|$, $[a, b] = [-1, 1]$, $n = 10$



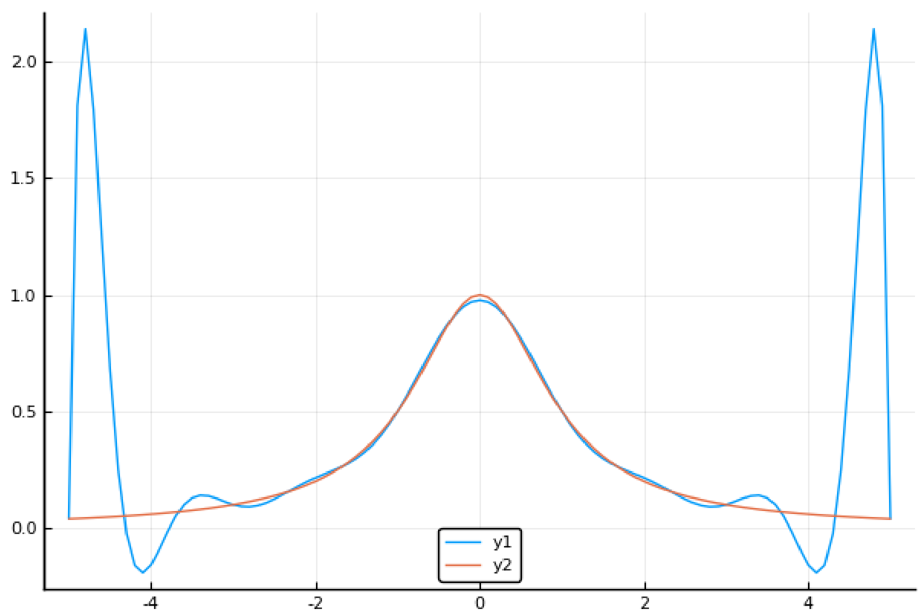
Rysunek 9: $|x|$, $[a, b] = [-1, 1]$, $n = 15$



Rysunek 10: $\frac{1}{1+x^2}$, $[a, b] = [-5, 5]$, $n = 5$



Rysunek 11: $\frac{1}{1+x^2}$, $[a, b] = [-5, 5]$, $n = 10$



Rysunek 12: $\frac{1}{1+x^2}$, $[a, b] = [-5, 5]$, $n = 15$

W tym zadaniu mocno widać pojawienie się efektu Rungego, czyli pogorszenia się jakości interpolacji wielomianowej pomimo zwiększenia liczby jej węzłów. Jest to szczególnie widoczne na końcach przedziału.