Obliczenia naukowe - Lista 4

Wojciech Pakulski (250350)

1 Obliczenie ilorazów różnicowych funkcji

Zadanie pierwsze polegało na zaimplementowaniu algorytmu, który oblicza wektor zawierający ilorazy różnicowe funkcji, dla danych wektorów węzłów i wartości funkcji interpolowanej, w tych węzłach.

Kod źródłowy programu znajduje się w pliku lista4.jl, jako funkcja *ilorazyRoznicowe(x, f).*

Pseudokod:

Dane: x – wektor długości *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 f_x – wektor długości *n+1* zawierający obliczone ilorazy różnicowe

N ← *length(x)*
for i ← 1 **to** N **do**
  $f_x[i] \leftarrow f[i]$
end for
for i ← 2 **to** N **do**
  for j ← N **to** i **step** -1 **do**
    $f_x[j] = \frac{f_x[j] - f_x[j-1]}{x[j] - x[j-i+1]}$
  end for
end for
return f_x

Zasada działania:

Algorytm opiera się na własności (2) ilorazu różnicowego, który można wykorzystać rekurencyjnie i dzięki temu kolejny iloraz różnicowy, można wyliczyć mając poprzedni.

(1) $f[x_0] = f(x_0)$
(2) $f[x_k, x_{k+1}, \dots, x_{k+m}] = \frac{f[x_{k+1}, x_{k+2}, \dots, x_{k+m-1}]}{x_{k+m} - x_k}$ $(0 \leq k < k+m \leq N)$

Korzystając z tej zależności rekurencyjnej nie musimy używać tablicy kwadratowej.

Na początku wypełniamy tablicę wartościami interpolowanej funkcji w węzłach, a przy kolejnych iteracjach aktualizujemy, obliczając je od tyłu (ze wzoru (2)), żeby uwzględnić wcześniej wyliczone ilorazy różnicowe.

Dzięki wykonywaniu obliczeń „od tyłu", nasza tablica przetrzymuje tylko ilorazy, które będą potrzebne później.

Złożoność algorytmu wynosi $O(n^2)$, ponieważ mamy dwie zagnieżdżone pętle.

2 Obliczanie wartości wielomianu interpolacyjnego w postaci Newtona w punkcie metodą Hornera

W zadaniu drugim należało napisać funkcję obliczającą wartość wielomianu interpolacyjnego stopnia n w postaci Newtona w punkcie x = t za pomocą uogólnionego algorytmu Hornera.

Wartość takiej funkcji w punkcie x:

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

gdzie q_i jest ilorazem różnicowym, a x_j węzłem interpolacji.

Kod źródłowy programu znajduje się w pliku lista4.jl, jako funkcja *warNewton(x, f_x, t).*

Pseudokod:

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

n ← *length(x)*
$nt \leftarrow f_x[n]$
for i ← $n - 1$ **to** 1 **step** -1 **do**
  $nt \leftarrow f_x[i] + (t - x[i]) \cdot nt$
end for
return nt

Zasada działania:

Ten algorytm działa korzystając z uogólnienia algorytmu Hornera. Zakładamy, że najwyższa potęga znajduje się na końcu, więc pierwsza wartość znajduje się na końcu ($w_n(x) = f[x_0, \ldots, x_n]$). Potem liczymy kolejne korzystając z zależności rekurencyjnej:

$$w_i(x) = f[x_0, x_1 \ldots, x_i] + (x - x_i) w_{k+1}(x)$$

Ostatnia wyliczona wartość to nasza szukana wartość funkcji w punkcie.

Złożoność obliczeniowa wynosi $O(n)$, ponieważ wykonujemy jedną pętlę, która wykonuje n-obrotów.

3 Obliczanie współczynników postaci naturalnej wielomianu interpolacyjnego w postaci Newtona

W tym zadaniu mieliśmy zaimplementować algorytm, który będzie wyliczał wartości współczynników postaci naturalnej wielomianu interpolacyjnego w postaci Newtona.

Kod źródłowy programu znajduje się w pliku lista4.jl, jako funkcja *naturalna(x, f_x).*

Pseudokod:

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

n ← *length(x)*
$a[n] \leftarrow f_x[n]$
for i ← $n - 1$ **to** 1 **step** -1 **do**
    $a[i] \leftarrow f_x[i]$
    for j ← i **to** $n - 1$ **do**
        $a[j] \leftarrow a[j] - x[i] \cdot a[j+1]$
    end for
end for
return a

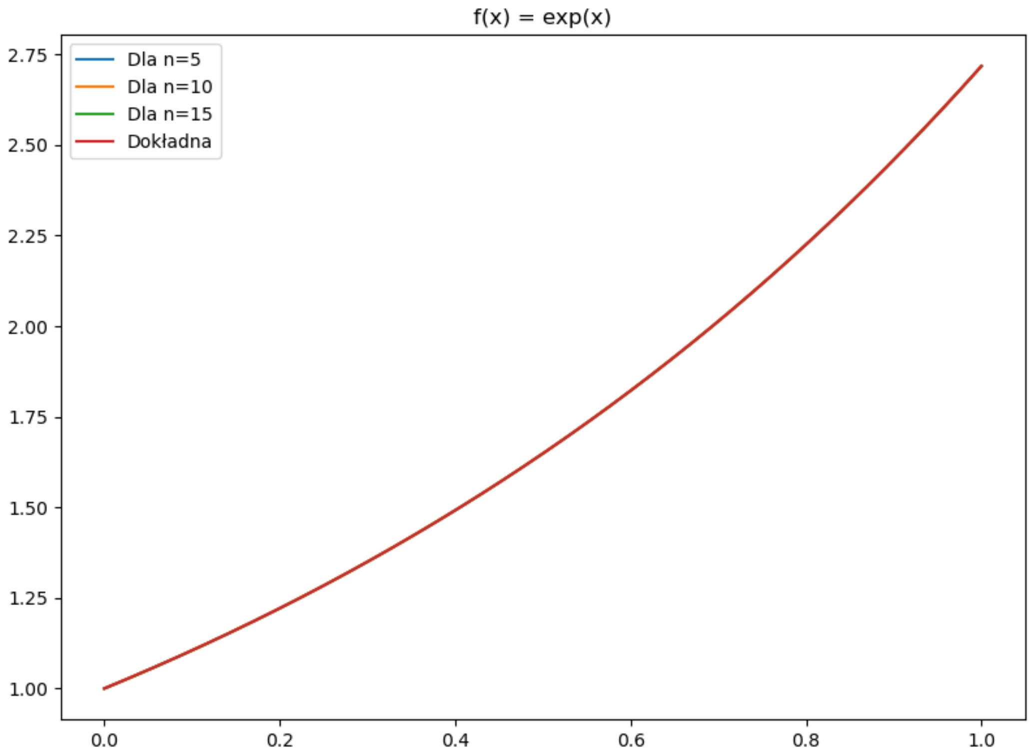
Zasada działania:

Współczynnik wielomianu interpolacyjnego w postaci Newtona ($N_n(x) = \sum_{i=0}^{n} f(x_0, x_1, \ldots, x_i] \prod_{j=0}^{i-1}(x - x_1)$) można zapisać jako $c_n = f[x_0, x_1 \ldots, x_n]$, jest on równy współczynnikowi a_n stojącego przy najwyższej potędze wielomianu w postaci naturalnej. Dlatego na początku do ostatniej komórki tablicy a przypisujemy wartość ilorazu różnicowego węzła na pozycji n. Dalej algorytm wylicza kolejne wartości ze wzoru $a[j] = a[j] - x[i] \cdot a[j+1]$. Korzystaliśmy tu z zależności wyprowadzonej z uogólnionego algorytmu Hornera ($w_i(x) = f[x_0, x_1 \ldots, x_i] + (x - x_i)w_{k+1}(x), (k = n-1, \ldots, 0)$) i wykonuje kolejne kroki tworząc a_i w oparciu o wcześniej policzone współczynniki stojące przy najwyższych potęgach.
Złożoność obliczeniowa wynosi $O(n^2)$, ponieważ mamy dwie zagnieżdżone pętli wykonywane maksymalnie n razy każda.

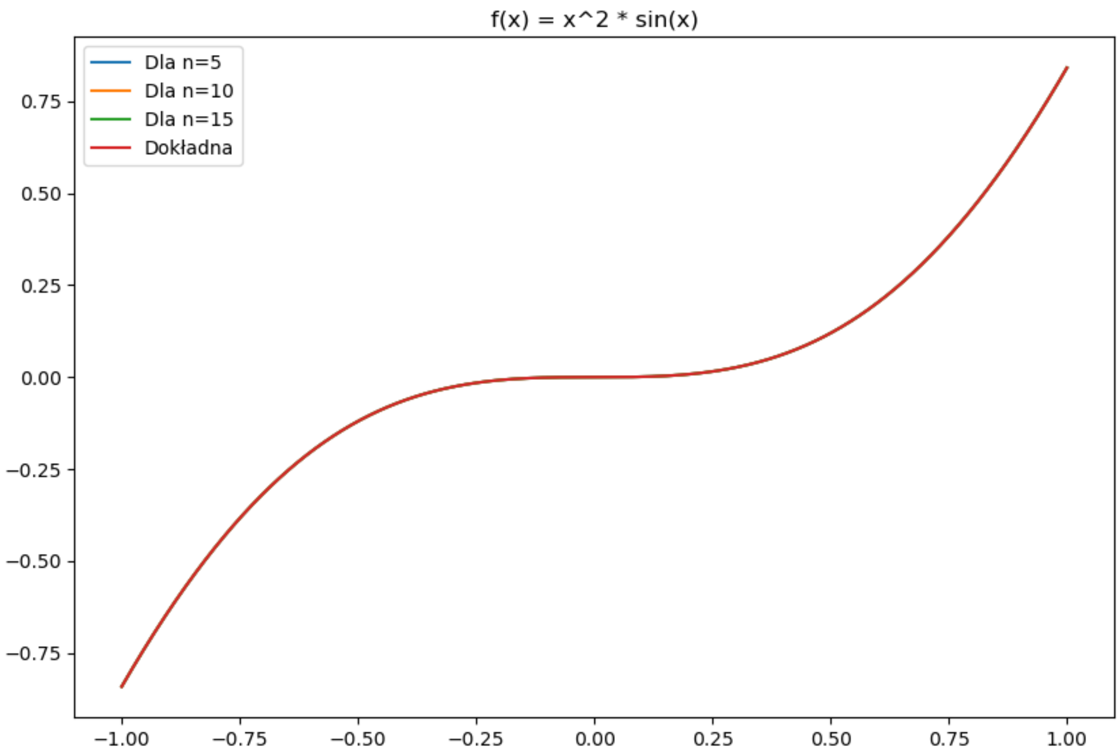
4 Rysowanie prostych wykresów funkcji

To zadanie polegało na narysowaniu funkcji f(x) za pomocą wielomianu interpolacyjnego i interpolowanej funkcji.

(a) $f(x) = e^x$ na przedziale [0, 1] dla stopni wielomianu interpolowanego n = 5, 10, 15. W tym wypadku wszystkie trzy wygenerowane wykresy pokrywają się z właściwym wykresem funkcji.

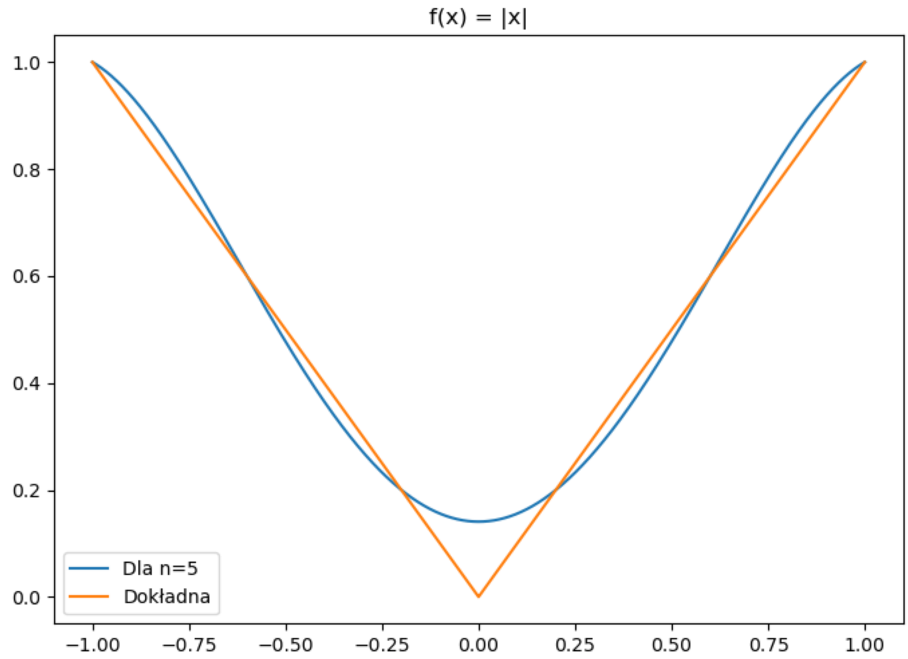


(b) $f(x) = x^2 \, sin \, x$ na przedziale [0, 1] dla stopni wielomianu interpolowanego n = 5, 10, 15. W tym wypadku wszystkie trzy wygenerowane wykresy również pokrywają się z właściwym wykresem funkcji.

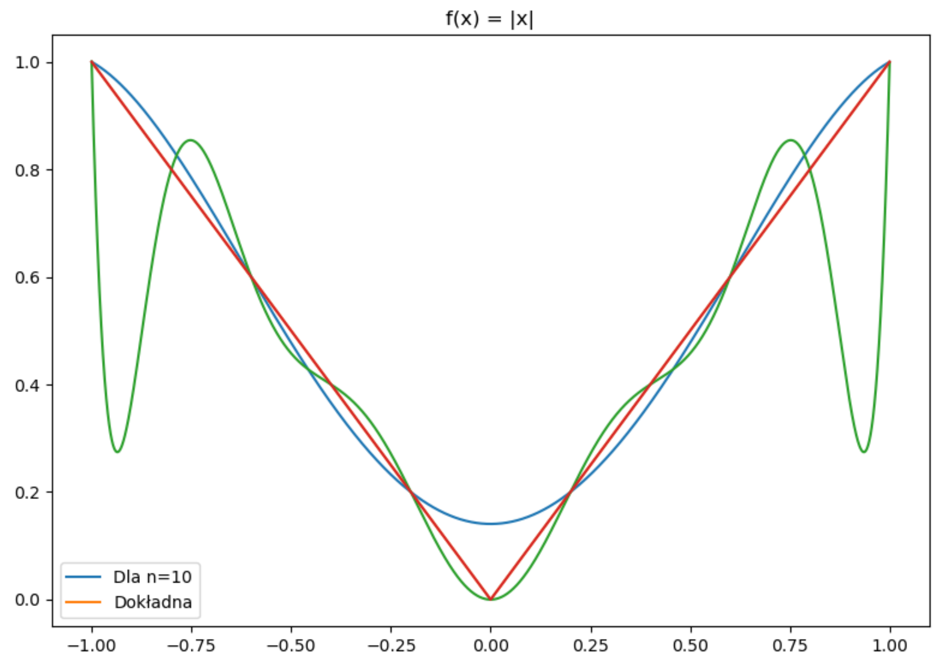


5 Rysowanie wykresów wielomianów interpolacyjnych, które nie pokrywają się z oryginalną funkcją

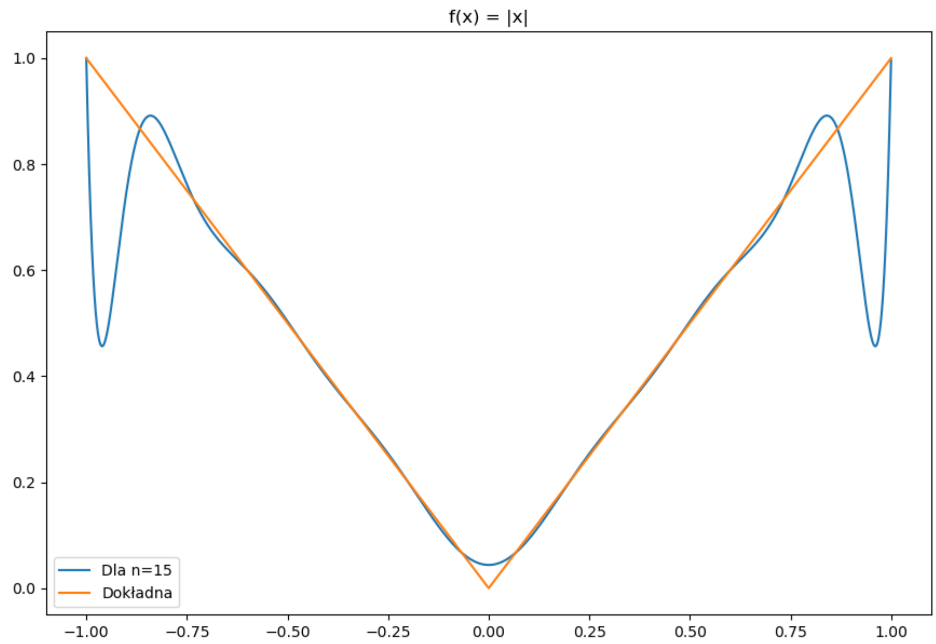
Pierwszy przykład jest dla funkcji $f(x) = |x|$, dla przedziału [-1, 1]. Oto wygenerowane wykresy n = 5, 10, 15.



Dla n = 5

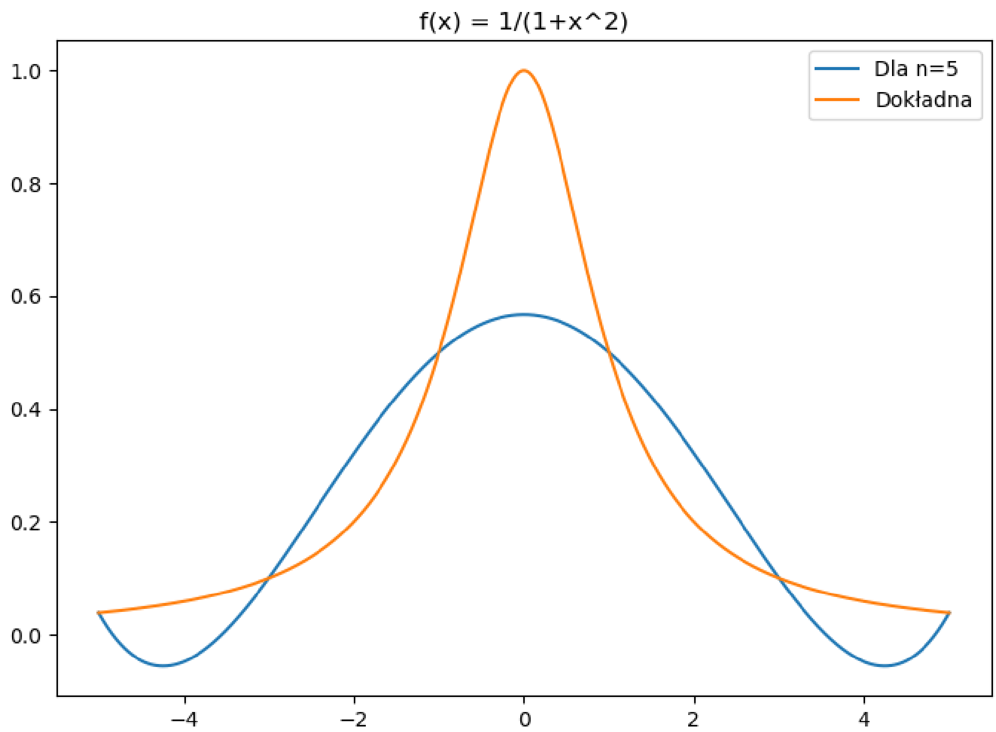


Dla n = 10

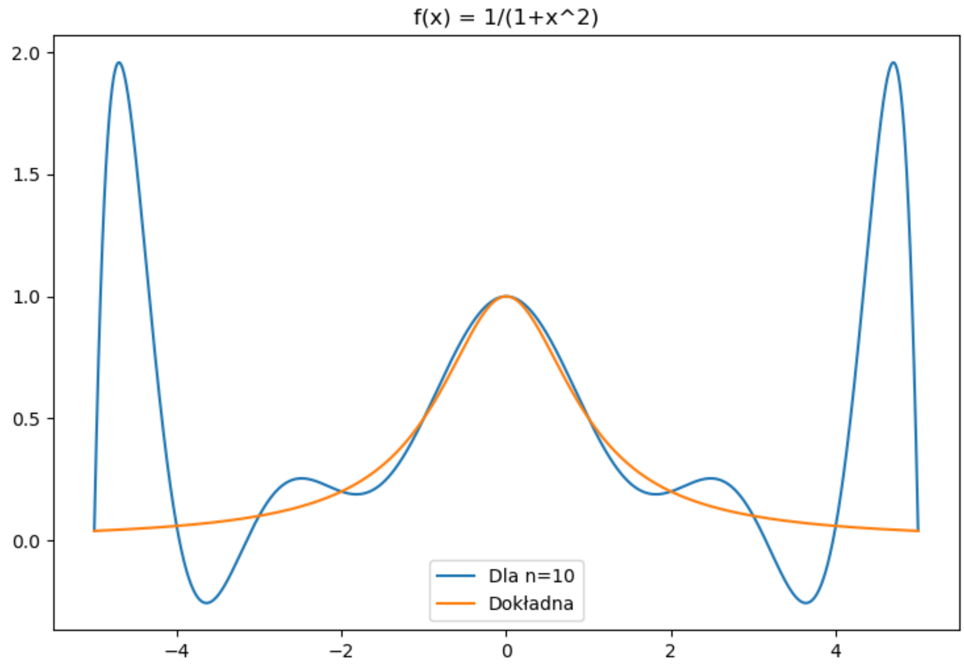


Dla n = 15

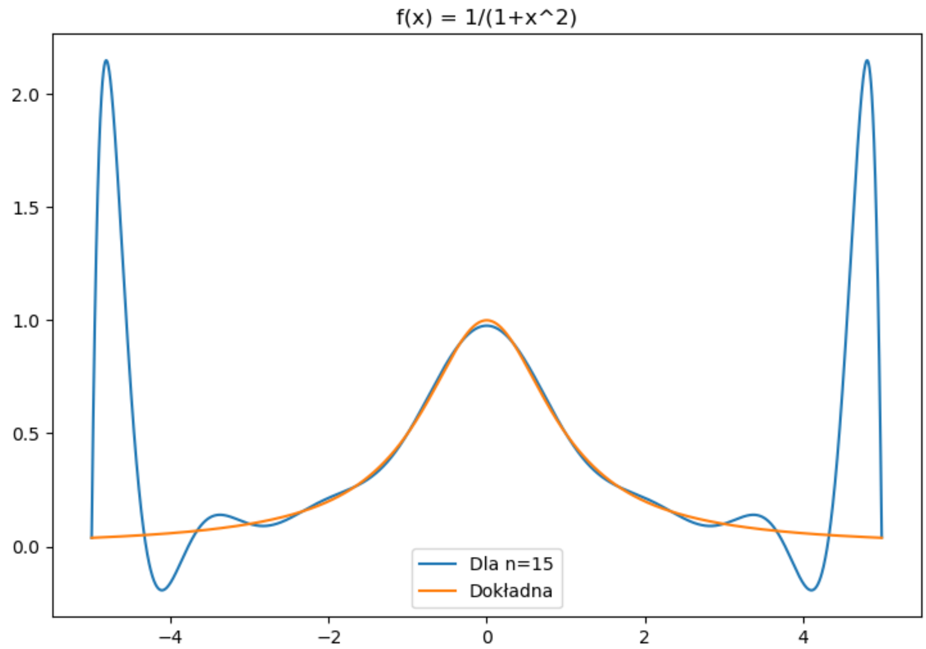
Drugi przykład jest dla funkcji $g(x) = \frac{1}{1+x^2}$, dla przedziału [-5, 5], dla n = 5, 10, 15. Oto wyniki.



n = 5



Dla n = 10



Dla n = 15

Możemy zaobserwować, że wartości funkcji dla danych argumentów nie pokrywają się z wartościami wielomianów interpolacyjnych.

Drugą obserwacją jest wystąpienie efektu Runge'go, czyli pogarszanie się dokładności interpolacji wielomianowej mimo zwiększania liczby węzłów. Efekt ten jest szczególnie widoczny na końcach przedziału. Zjawisko występuje przede wszystkim, gdy interpolacja jest przeprowadzona dla równooddalonych węzłów, dlatego żeby zmniejszyć jego wpływ należy zwiększyć gęstość węzłów na krańcach przedziału interpolacji. Zjawisko w ostatnim przykładzie, a nie wcześniejszych, ponieważ w tym przypadku wielkość pochodnych n-tego rzędu tej funkcji rośnie szybko, gdy n wzrasta.