Sprawozdanie z Laboratorium
Obliczenia Naukowe - Lista 4

Wojciech Typer

6 grudnia 2025

Zadanie 1

Cel zadania

Celem zadania jest napisanie funkcji obliczającej ilorazy różnicowe. Funkcja ma przyjmować dwa argumenty:

- x $\rightarrow$ wektor długości n + 1 zawierający węzły: x_0, ..., x_n
- f $\rightarrow$ wektor długości n + 1 zawierający wartości interpolowanej funkcji w węzłach $f(x_0$, ..., $f(x_n))$

Funkcja ma zwracać $fx \rightarrow$ wektor długości n + 1, zawierający obliczone ilorazy róznicowe:

$$fx = f[x_0], ..., fx[n+1] = f[x_0, ..., x_n]$$

Idea metody

Obliczymy to na podstawie wzoru rekurencyjnego: iloraz rzędu k możemy policzyć na podstawie ilorazu rzędku k - 1:

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

Implementacja

Algorithm 1 Obliczanie ilorazów różnicowych (Newton)

Require: x: wektor węzłów $(x_0, \ldots, x_n)$
Require: f: wektor wartości funkcji $(f(x_0), \ldots, f(x_n))$
Ensure: fx: wektor ilorazów różnicowych

```
function IlorazyRoznicowe(x, f)
    n ← length(f) − 1
    fx ← copy(f)

    for j ← 1 to n do
        for i ← n + 1 downto j + 1 do
            licznik ← fx[i] − fx[i − 1]
            mianownik ← x[i] − x[i − j]
            fx[i] ← licznik / mianownik
        end for
    end for
    return fx
end function
```

Zadanie 2

Cel zadania

Celem zadania jest napisanie funkcji obliczającej wartości wielomianu interpolacyjnego stopnia n w postaci Newtona za pomocą uogólnionego schematu Hornera w czasie O(n). Funkcja ma przyjmować trzy argumenty:

- x → wektor długości n + 1 zawierający węzły: x_0, ..., x_n
- fx → wektor długości n + 1, zawierający ilorazy różnicowe: $fx = f[x_0]$, ..., $fx[n+1] = f[x_0, ..., x_n]$
- t → punkt, w którym liczymy wartość wielomianu

Funkcja ma zwracać wartość wielomianu interpolacyjnego w punkcie t.

Idea metody

Wielomian Newtona możemy zapisać w postaci:

$$N_n(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \cdots + c_n(x - x_0)(x - x_1)\cdots(x - x_{n-1})$$

Gdybyśmy chcieli policzyć każdy składnik osobno, to złożoność obliczeniowa wyniosłaby O(n^2). Możemy jednak skorzystać z uogólnionego schematu Hornera, który pozwala na obliczenie wartości wielomianu w czasie O(n):

$$N_n(x) = c_0 + (t - x_0)\,(c_1 + (t - x_1)\,(c_2 + \cdots + (t - x_{n-1})c_n))$$

Słownie mówiąc, zaczynamy od ostatniego współczynnika c_n i iteracyjnie dodajemy kolejne składniki, mnożąc je przez odpowiednie czynniki $(t - x_i)$ i dodając c_i.

Implementacja

Algorithm 2 Obliczanie wartości wielomianu Newtona (Schemat Hornera)

Require: x: wektor węzłów $(x_0, \ldots, x_n)$
Require: fx: wektor ilorazów różnicowych $(f[x_0], \ldots, f[x_0, \ldots, x_n])$
Require: t: punkt, w którym liczymy wartość
Ensure: nt: wartość wielomianu w punkcie t

function WarNewton(x, fx, t)
  $n \leftarrow \text{length}(fx) - 1$ ▷ Stopień wielomianu
  $nt \leftarrow fx[n+1]$ ▷ Inicjalizacja ostatnim współczynnikiem

  for $i \leftarrow n$ **downto** 1 **do** ▷ Iteracja od przedostatniego wyrazu
    $nt \leftarrow fx[i] + (t - x[i]) \cdot nt$
  end for
  return nt
end function

Zadanie 3

Cel zadania

Celem zadanie jest napisanie funkcji obliczającej w czasie $O(n^2)$ współczynniki wielomianu zapisanego w postaci naturalnej. Funkcja ma przyjmować dwa argumenty:

- x → wektor długości n + 1 zawierający węzły: x_0, ..., x_n
- fx → wektor długości n + 1, zawierający ilorazy różnicowe: $fx = f[x_0]$, ..., $fx[n+1] = f[x_0, ..., x_n]$

Funkcja ma zwracać wektor a długości n + 1, zawierający współczynniki wielomianu w postaci naturalnej:

$$W_n(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n$$

Idea metody

Wielomian w postaci Newtona zadany jest wzorem:

$$N_n(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + ... + c_n(x - x_0)(x - x_1)...(x - x_{n-1})$$

Aby uzyskać postać naturalną, musimy wymnożyć wszystkie czynniki $(x - x_i)$ i uporządkować wyrazy według potęg x. Możemy to zrobić iteracyjnie, korzystając ze schematu Hornera do mnożenia wielomianów. Algorytm polega na aktualizacji tablicy współczynników, "wciągając"kolejne czynniki $(x - x_k)$ do obliczonych już współczynników postaci naturalnej. Złożoność obliczeniowa wynosi $O(n^2)$ ze względu na zagnieżdżone pętle (dla każdego z n czynników wykonujemy do n operacji aktualizacji).

Implementacja

Algorithm 3 Konwersja postaci Newtona do postaci naturalnej

Require: x: wektor węzłów $(x_0, \dots, x_n)$
Require: fx: wektor ilorazów różnicowych $(c_0, \dots, c_n)$
Ensure: a: wektor współczynników postaci naturalnej $(a_0, \dots, a_n)$

function NATURALNA(x, fx)
  $n \leftarrow \text{length}(fx) - 1$
  $a \leftarrow \text{copy}(fx)$

  for $k \leftarrow n$ **downto** 1 **do** ▷ Iteracja po węzłach x_{n-1} do x_0
    for $j \leftarrow k$ **to** n **do** ▷ Aktualizacja współczynników
      $a[j] \leftarrow a[j] - a[j+1] \cdot x[k]$
    end for
  end for
  return a
end function

Zadanie 4

Cel zadania

Celem zadania jest napisanie funkcji, która będzie interpolować zadaną funkcję f w przedziale [a, b] za pomocą wielomianu interpolacyjnego stopnia n w postaci Newtona. Następnie funkcja ma za zadanie rysować wykres wielomianu interpolacyjnego i interpolowanej funkcji. Funkcja ma przyjmować pięć argumentów:

- f → funkcja anonimowa
- a → początek przedziału
- b → koniec przedziału
- n → stopień wielomianu interpolacyjnego
- wezly → typ węzłów: 'rownoodlegle' lub 'czebyszew'

W interpolacji należy użyć węzłów równoodległych, czyli:

$$x_k = a + kh, \text{ gdzie } h = \frac{b-a}{n}$$

lub węzłów będących zerami n + 1 wielomianu Czebyszewa, czyli:

$$x_k = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{(2k+1)\pi}{2(n+1)}\right)$$

Implementacja

Algorithm 4 Pseudokod funkcji `rysujNnfx`

Utwórz tablicę x_nodes o długości $n+1$
if wezly = rownoodlegle **then**
    $h \leftarrow (b-a)/n$
    for $k = 0$ to n **do**
        $x\_nodes[k] \leftarrow a + k \cdot h$
    end for
else if wezly = czebyszew **then**
    for $k = 1$ to $n+1$ **do**
        $x\_raw \leftarrow \cos\left(\frac{(2k-1)\pi}{2(n+1)}\right)$
        $x\_nodes[k-1] \leftarrow \frac{a+b}{2} + \frac{b-a}{2} \cdot x\_raw$
    end for
else
    Zgłoś błąd: "Nieznany typ węzłów"
end if
$y\_nodes \leftarrow f(x\_nodes)$
$coeffs \leftarrow$ ilorazyRoznicowe(x_nodes, y_nodes)
Utwórz siatkę x_plot (1000 punktów na $[a, b]$)
$y\_true \leftarrow f(x\_plot)$
$y\_interp \leftarrow$ wartości warNewton(x_nodes, $coeffs$, t) dla każdego t z x_plot
Narysuj wykres $f(x)$
Narysuj wykres $N_n(x)$ przerywaną linią
Narysuj punkty (x_nodes, y_nodes) jako węzły
Ustaw etykiety osi i tytuł
Wyświetl wykres

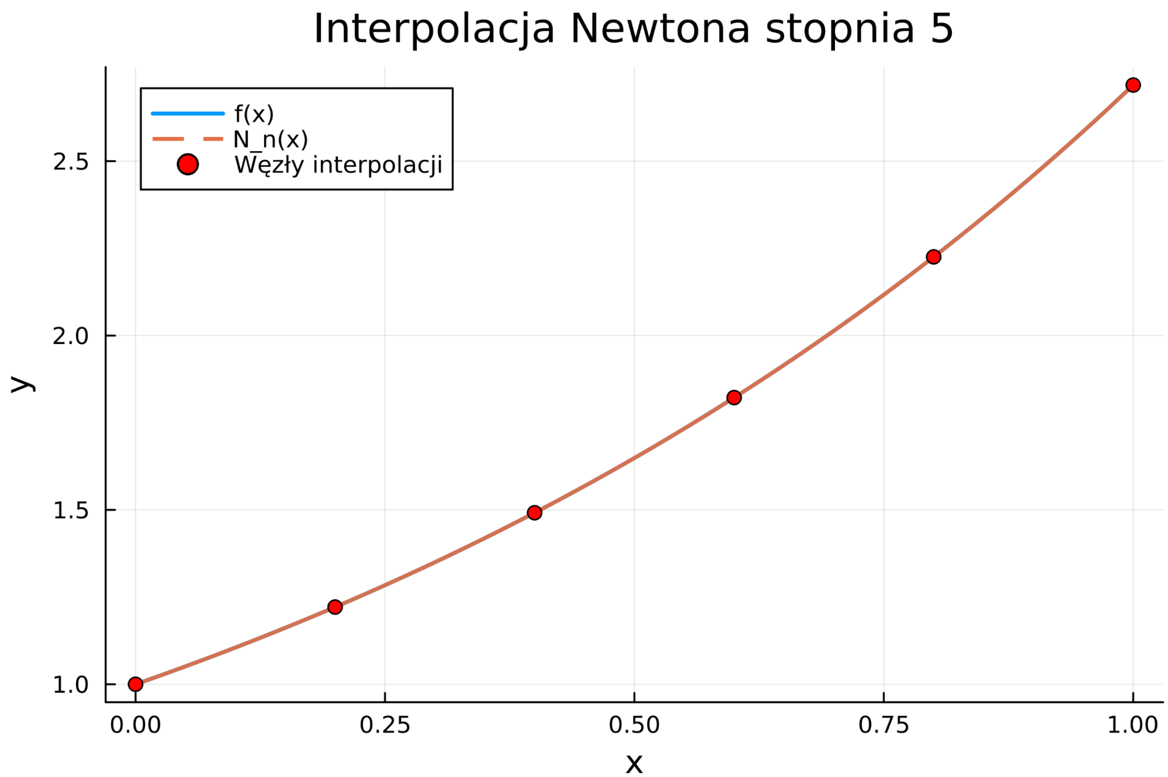
Zadanie 5

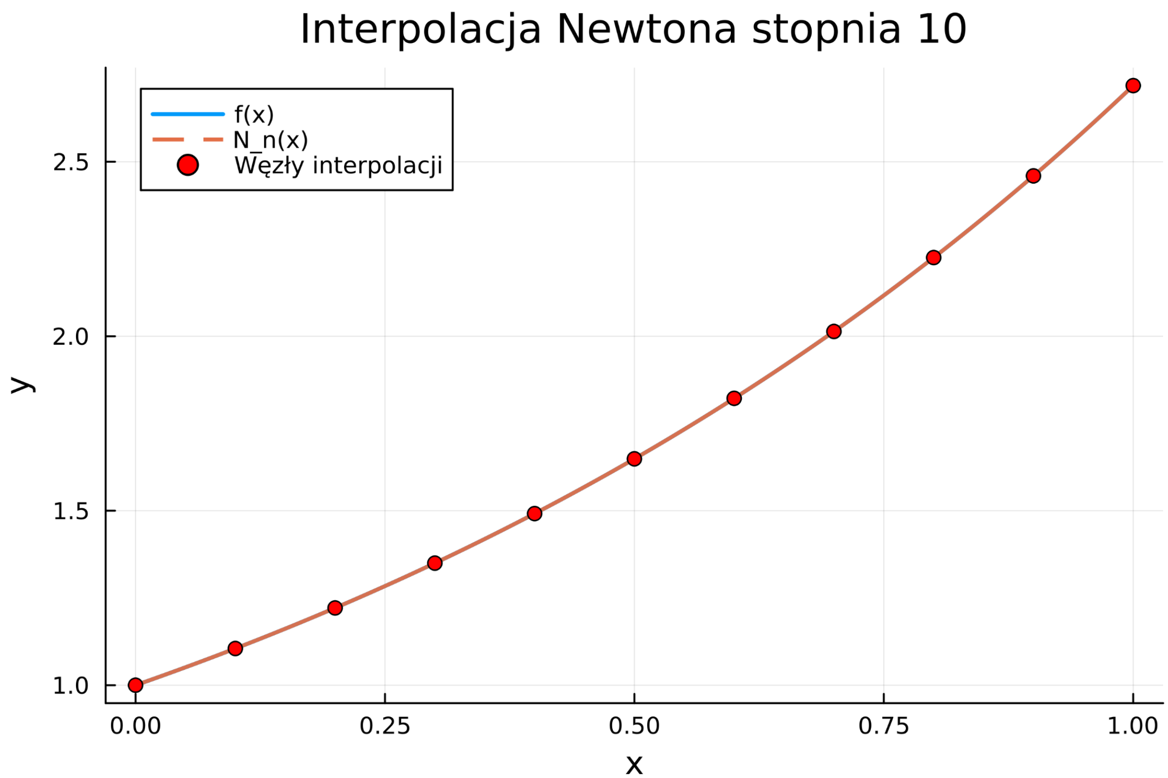
Celem zadania jest przetestowanie funkcji z poprzedniego zadania, na węzłach równoodległych na następujących przykładach:

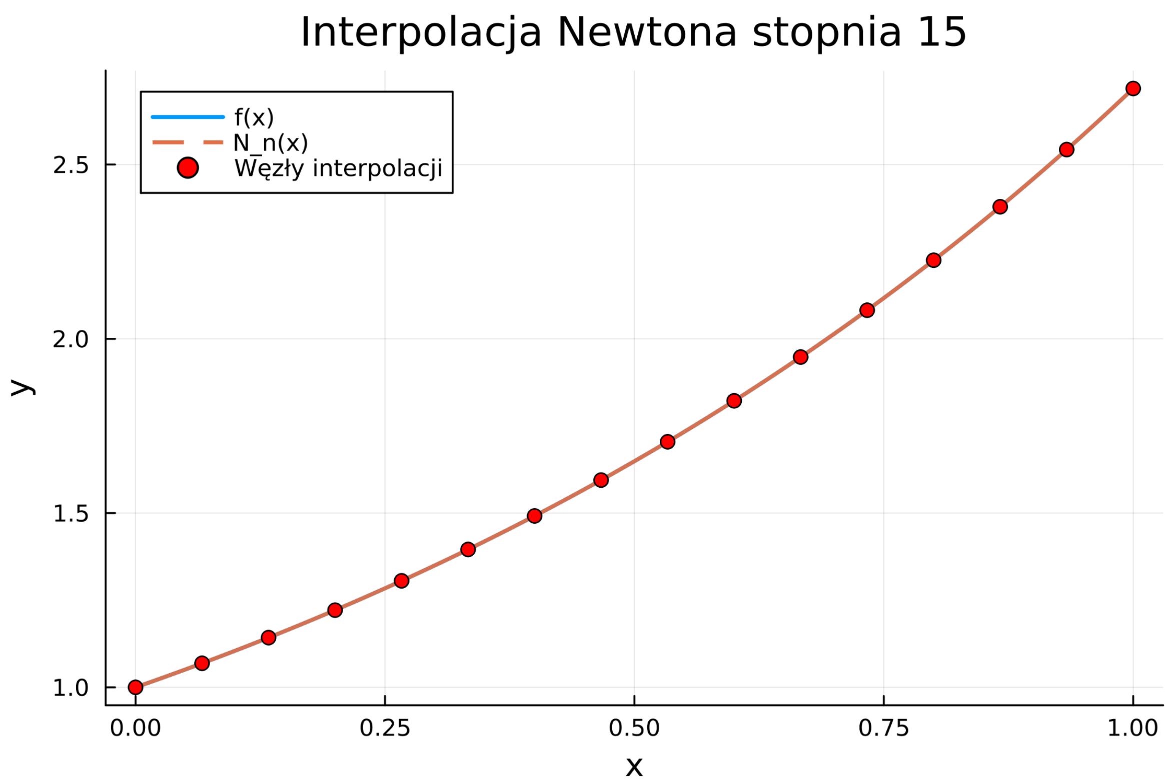
- $f(x) = e^x$ na przedziale [0, 1] dla n = 5, 10, 15
- $f(x) = x^2 \cdot \sin(x)$ na przedziale [-1, 1] dla n = 5, 10, 15

Wyniki

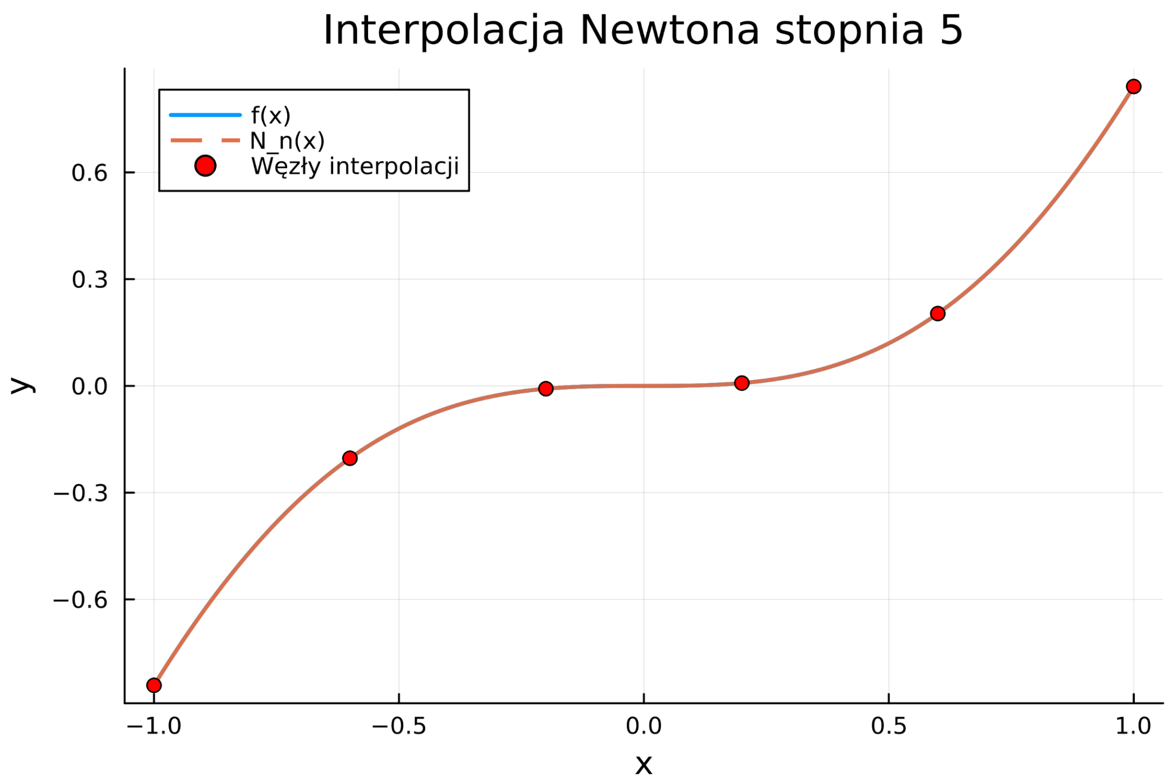
Wyniki dla $f(x) = e^x$ na przedziale [0, 1]:

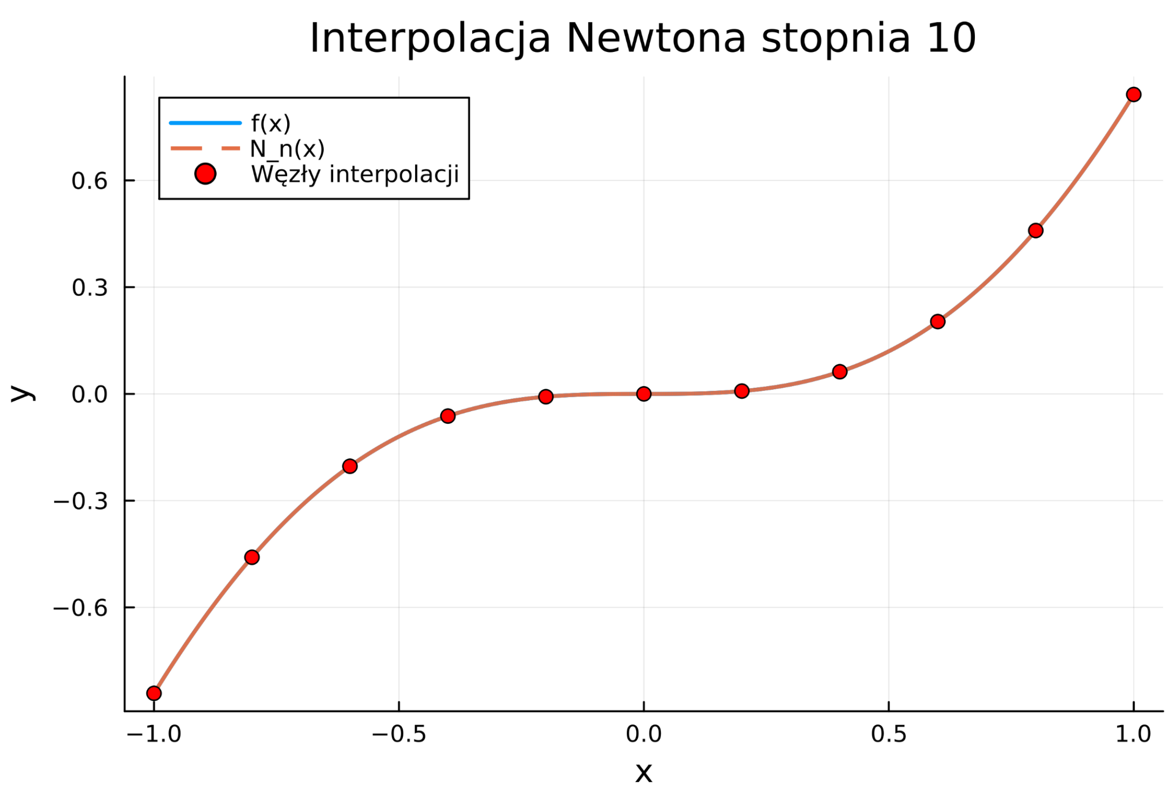


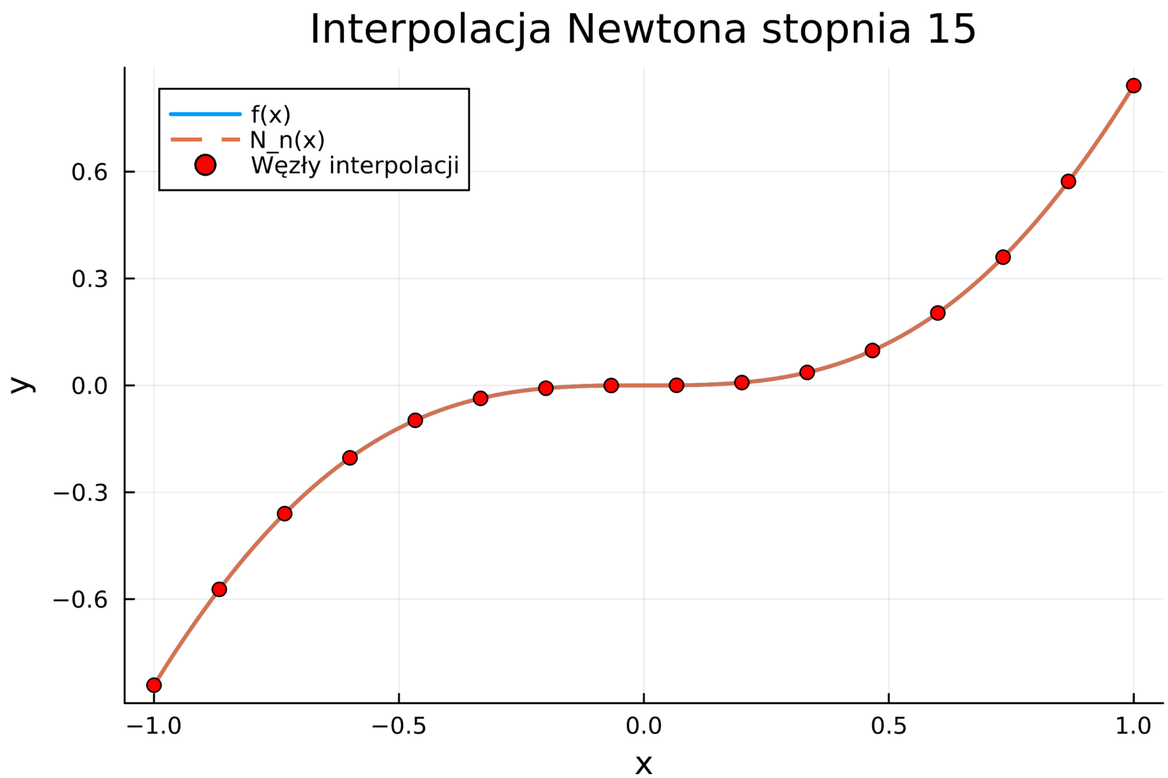

Interpolacja Newtona stopnia 10
f(x)
N_n(x)
Węzły interpolacji
2.5
2.0
1.5
1.0
y
0.00
0.25
0.50
0.75
1.00
x


Interpolacja Newtona stopnia 15
f(x)
N_n(x)
Węzły interpolacji
2.5
2.0
1.5
1.0
y
0.00
0.25
0.50
0.75
1.00
x

Wyniki dla $f(x) = x^2 \cdot \sin(x)$ na przedziale [-1, 1]:







Wnioski

Na powyższych wykresach widać, że zarówno dla $f(x) = e^x$ jak i dla $f(x) = x^2 \cdot \sin(x)$, już dla ilości węzłow n = 5, wielomian interpolacyjny bardzo dobrze pokrywa się z wykresem funkcji. Zwiększanie ilości węzłów do n = 10 i n = 15 powoduje jedynie minimalne poprawienie dokładności interpolacji. Możemy na tej podstawie wysnuć wnioski, że dla tych funkcji w przedziałach [0, 1] oraz [-1, 1], węzły równoodległe są wystarczające do uzyskania dobrej jakości interpolacji już przy niewielkiej liczbie węzłów.

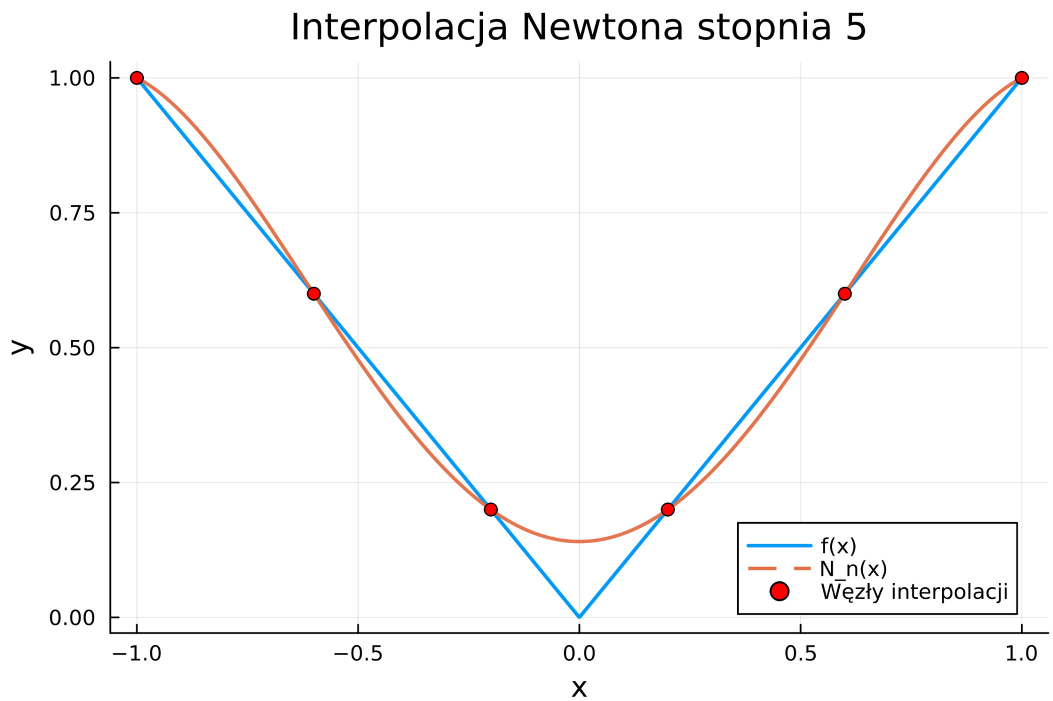
Zadanie 6

Celem zadanie jest przetestowanie funkcji z zadania 4 na węzłach równoodległych i węzłach Czebyszewa na następujących przykładach:

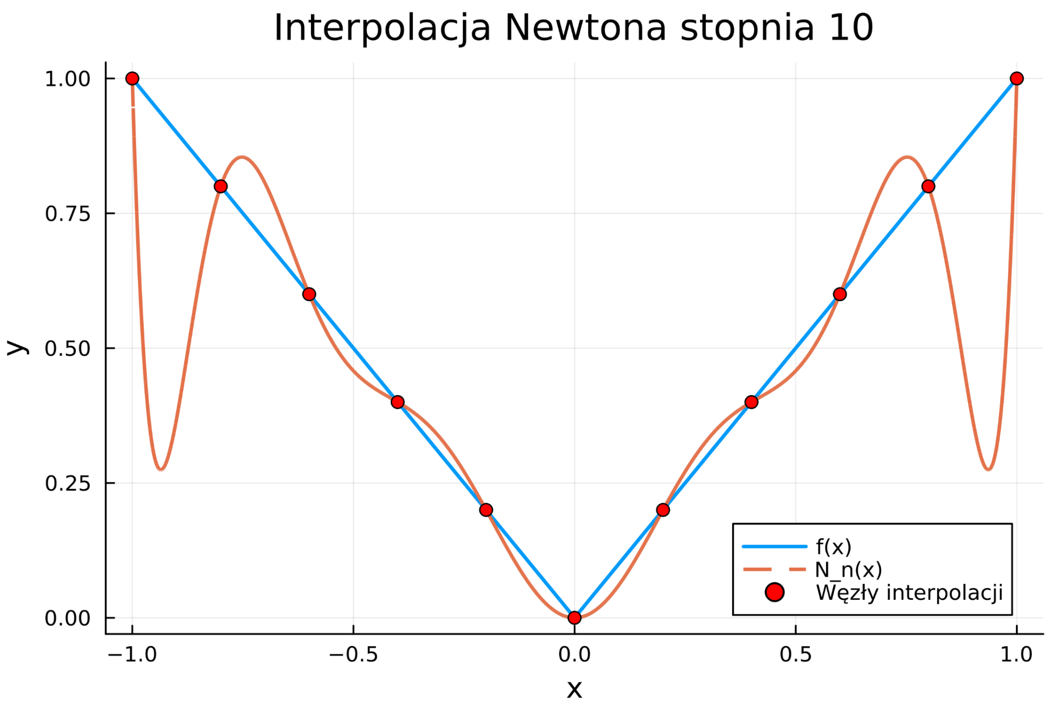
- $f(x) = |x|$ na przedziale [-1, 1] dla n = 5, 10, 15
- $f(x) = \frac{1}{1+x^2}$ na przedziale [-1, 1] dla n = 5, 10, 15

Wyniki

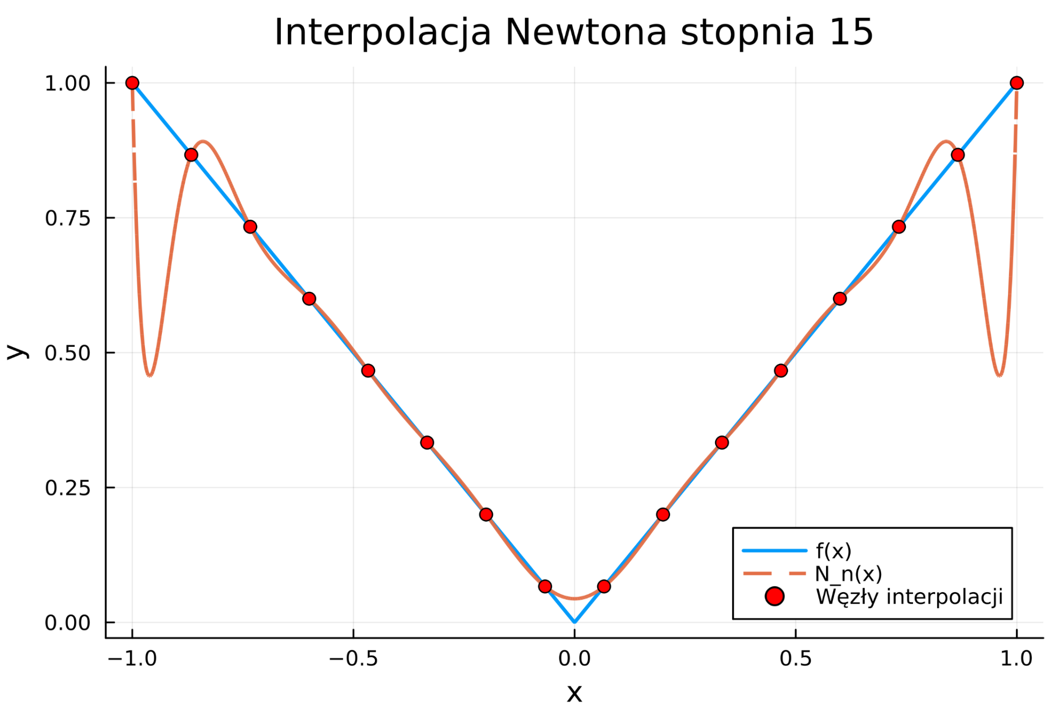
Wyniki dla $f(x) = |x|$ na przedziale [-1, 1]:



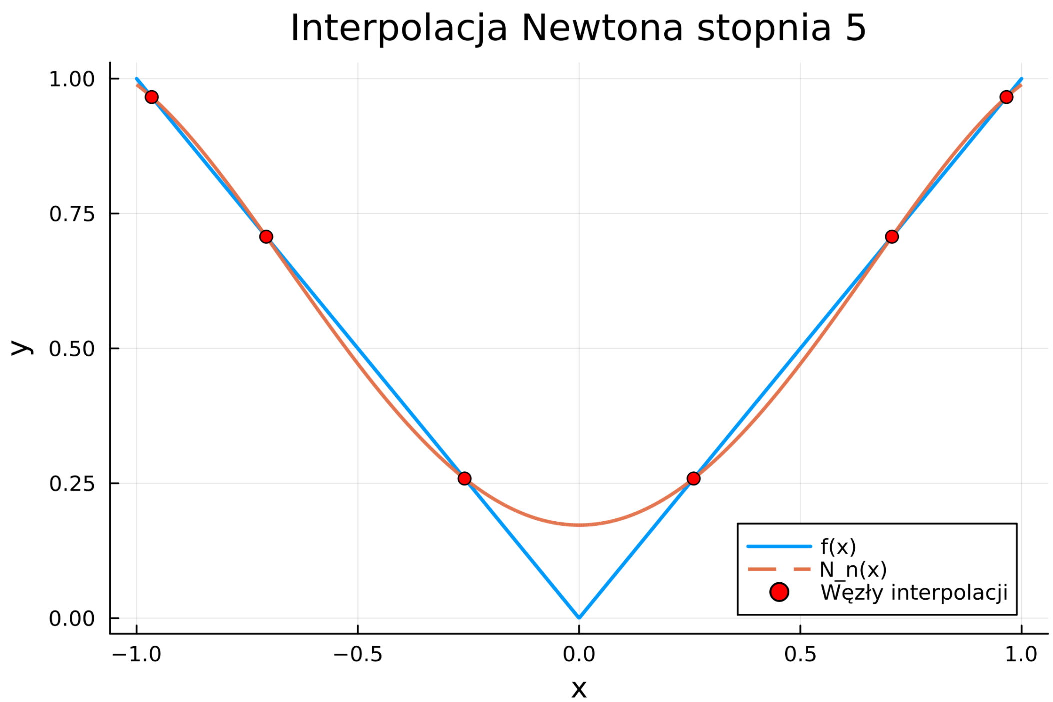
Rysunek 1: Stopień wielomianu: 5, węzły równoodległe, $f(x) = |x|$



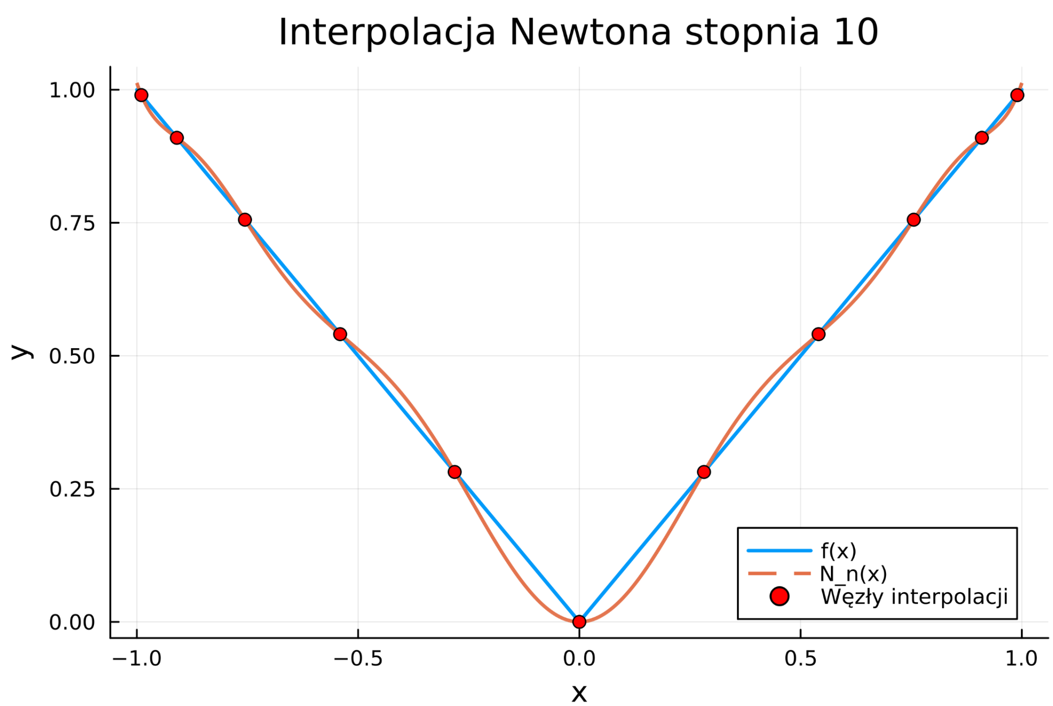
Rysunek 2: Stopień wielomianu: 10, węzły równoodległe, $f(x) = |x|$



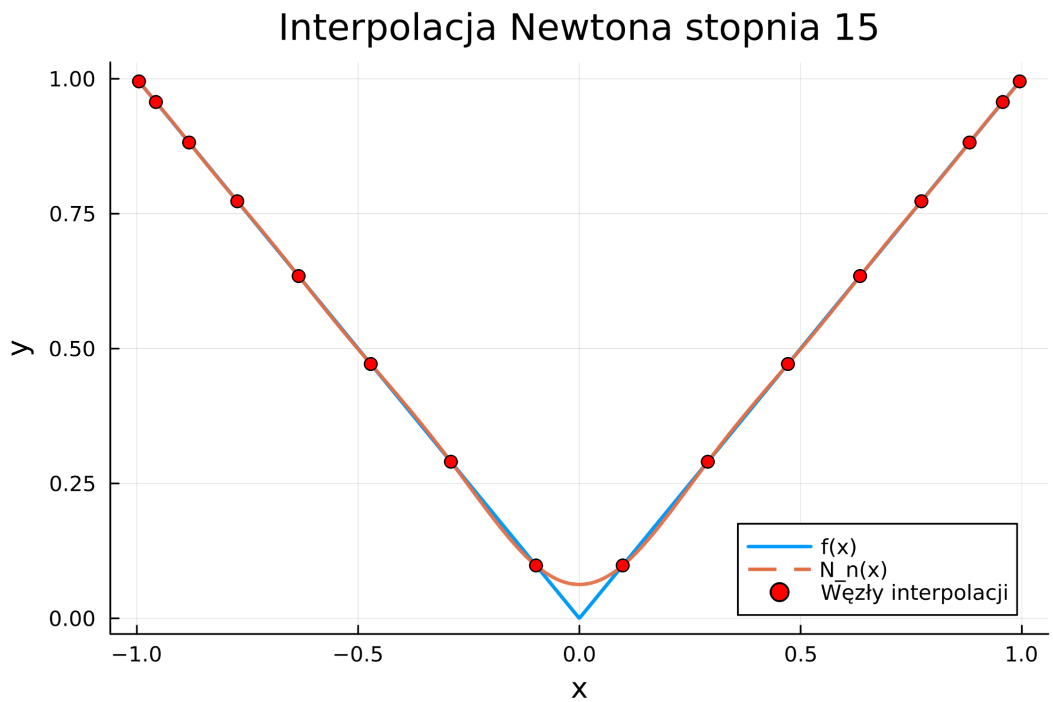
Rysunek 3: Stopień wielomianu: 15, węzły równoodległe, $f(x) = |x|$



Rysunek 4: Stopień wielomianu: 5, węzły Czebyszewa, $f(x) = |x|$

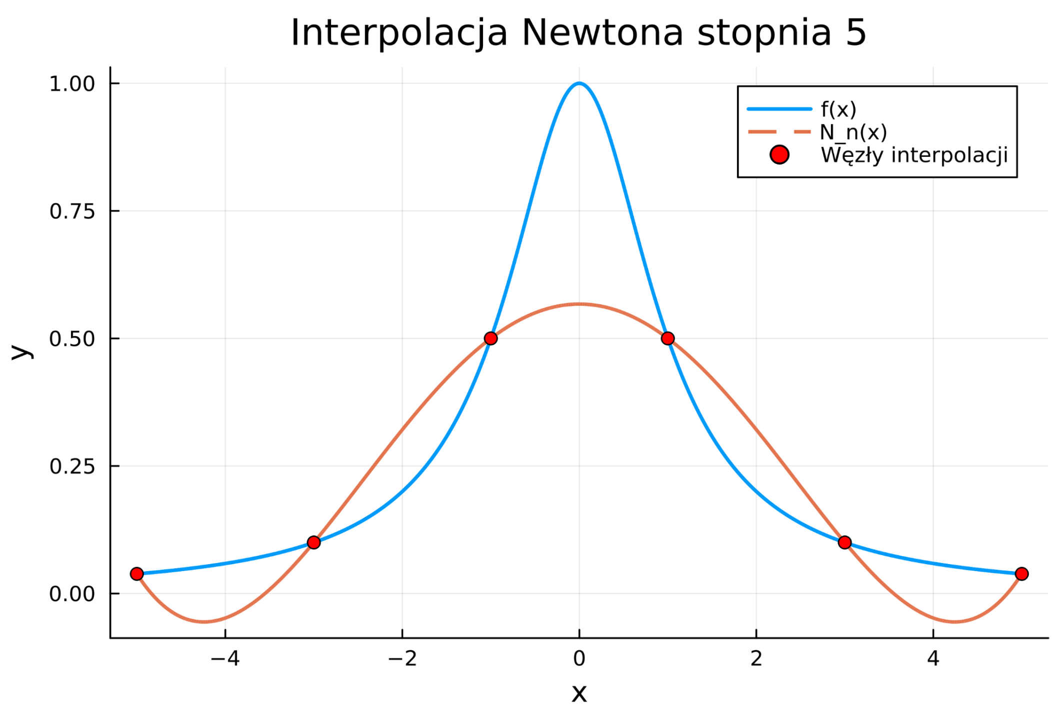


Rysunek 5: Stopień wielomianu: 10, węzły Czebyszewa, $f(x) = |x|$

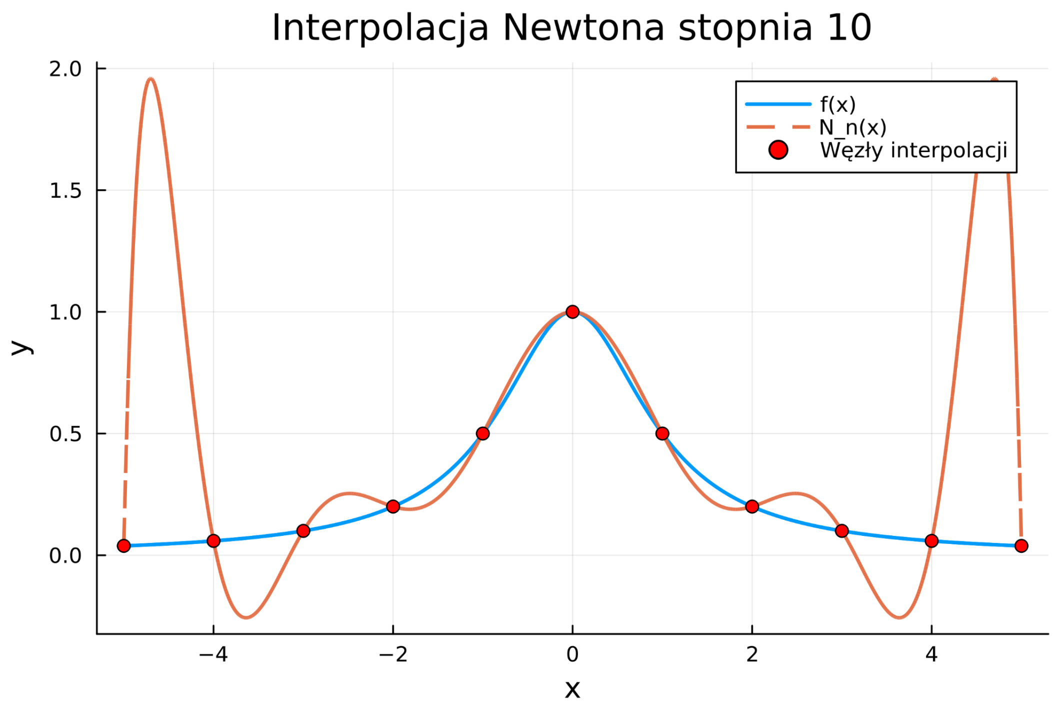


Rysunek 6: Stopień wielomianu: 15, węzły Czebyszewa, $f(x) = |x|$

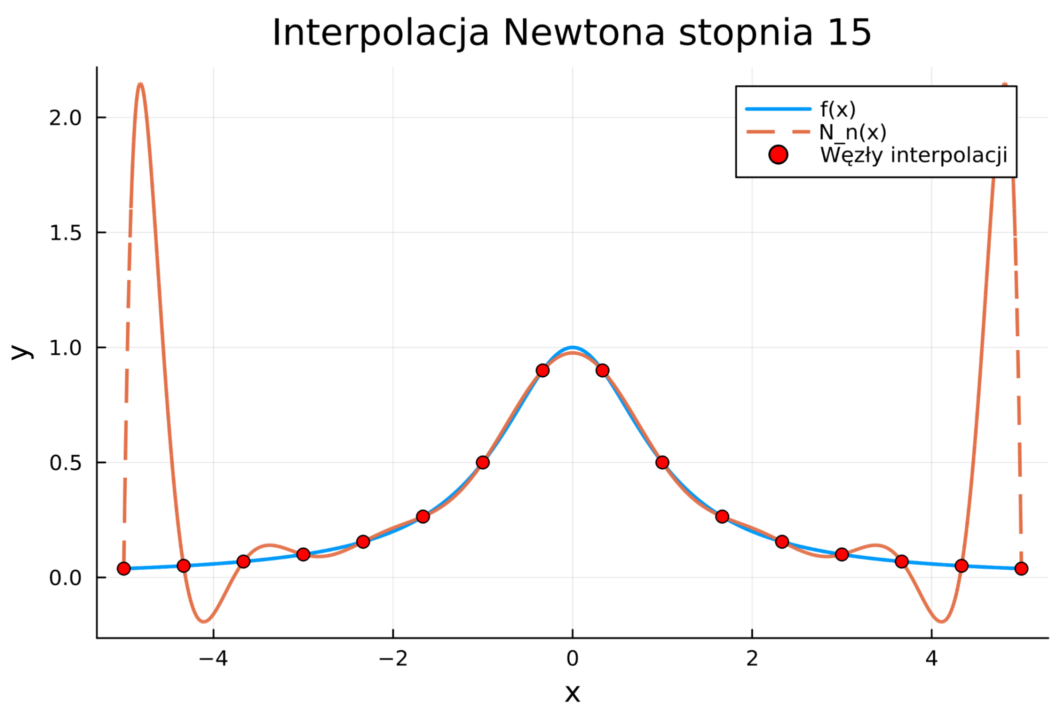
Wyniki dla $f(x) = \frac{1}{1+x^2}$ na przedziale [-5, 5]:



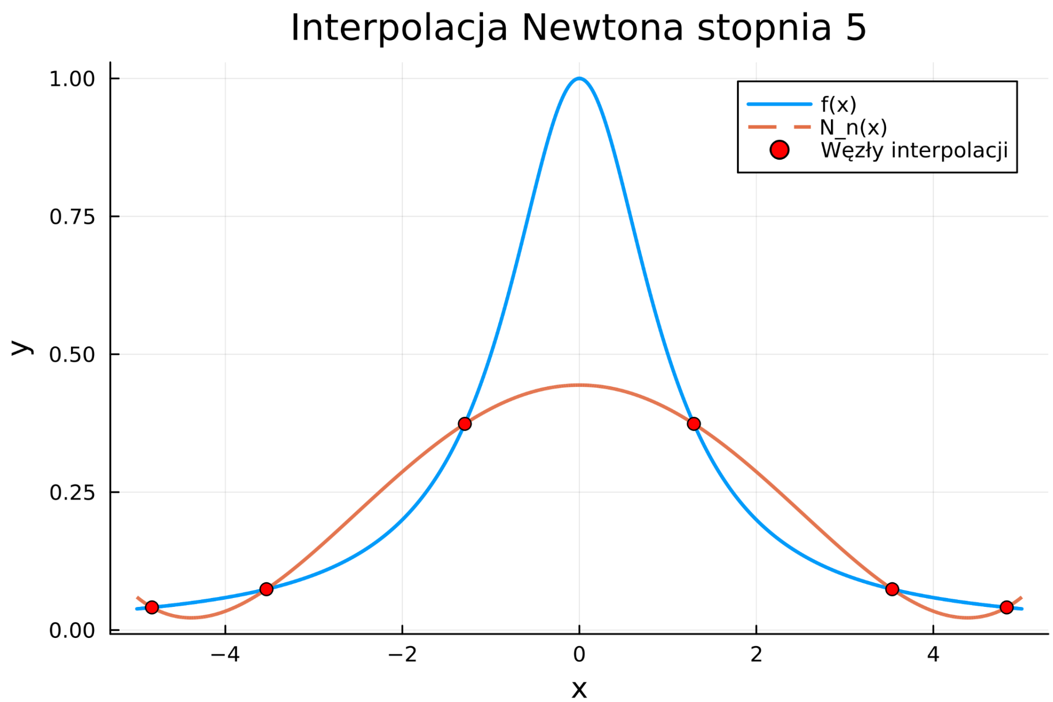
Rysunek 7: Stopień wielomianu: 5, węzły równoodległe, $f(x) = \frac{1}{1+x^2}$



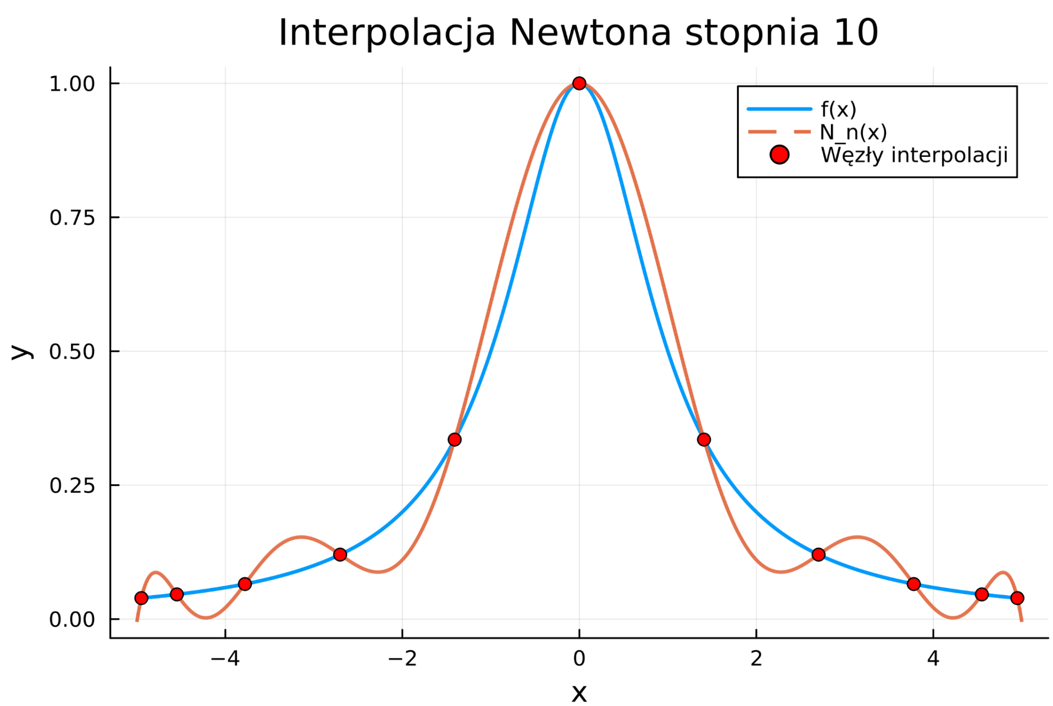
Rysunek 8: Stopień wielomianu: 10, węzły równoodległe, $f(x) = \frac{1}{1+x^2}$



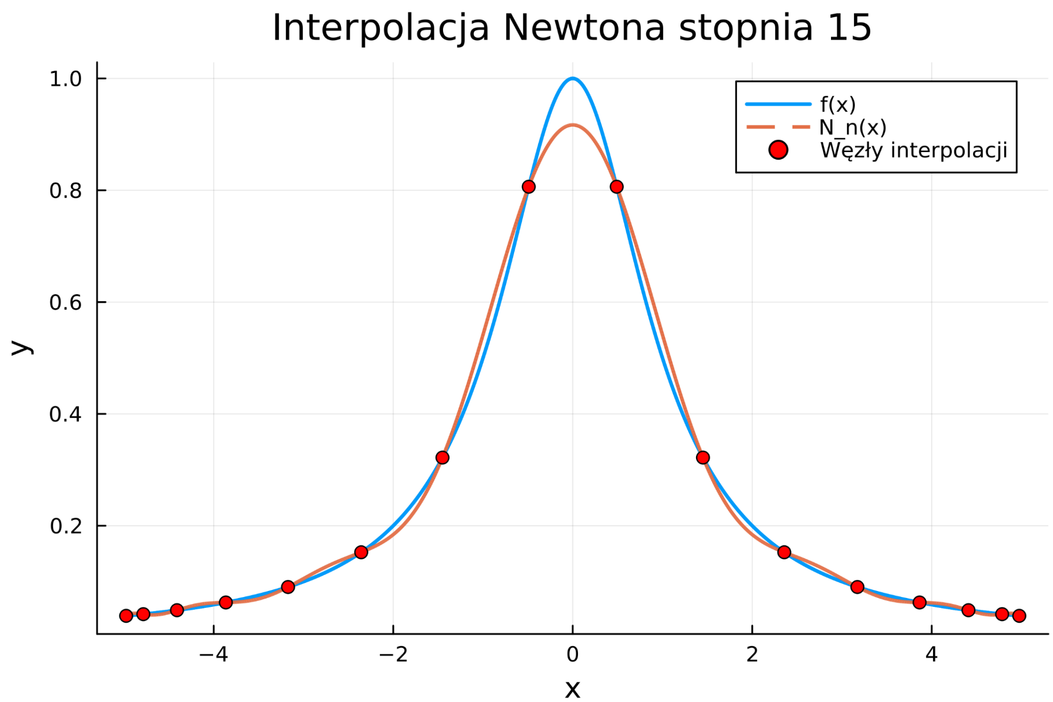
Rysunek 9: Stopień wielomianu: 15, węzły równoodległe, $f(x) = \frac{1}{1+x^2}$



Rysunek 10: Stopień wielomianu: 5, węzły Czebyszewa, $f(x) = \frac{1}{1+x^2}$



Rysunek 11: Stopień wielomianu: 10, węzły Czebyszewa, $f(x) = \frac{1}{1+x^2}$



Rysunek 12: Stopień wielomianu: 15, węzły Czebyszewa, $f(x) = \frac{1}{1+x^2}$

Wnioski

Powyższe wykresy pokazują, że sytuacja w porównaniu z tą z zadania 5 jest zupełnie inna. Widzimy tu znaczną różnicę pomiędzy wielomianami interpolacyjnymi a funkcjami, chociaż zwiększenie ilości węzłów poprawia jakość interpolacji. Dla $f(x) = |x|$ i dla $f(x) = \frac{1}{1+x^2}$, węzły Czebyszewa dają zauważalnie lepsze wyniki niż węzły równoodległe, szczególnie przy wyższych stopniach wielomianu.

Efekt Rungego

Efekt Rungego to zjawisko, które występuje podczas interpolacji wielomianowej funkcji na równoodległych węzłach. Polega ono na tym, że wraz ze wzrostem stopnia wielomainu interpolacyjnego, zwłaszcza w pobliżu krawędzi przedziału interpolacji. W przypadku funkcji $f(x) = \frac{1}{1+x^2}$, efekt Rungego jest właśnie szczególnie widoczny przy krawędziach przedziału - zauważalne są tam duże odchylenia wielomianu interpolacyjnego od rzeczywistej funkcji, zwłaszcza przy wyższych stopniach wielomianu. Użycie węzłów Czebyszewa skutecznie pomaga zminimalizować efekt Rungego, ponieważ te węzły są gęściej rozmieszczone na krawędziach przedziału, co prowadzi do lepszej jakości interpolacji i zmniejszenia oscylacji wielomianu.