Sprawozdanie z listy 4. - Obliczenia naukowe

Jakub Bachanek

6 grudnia 2020

1 Zadanie 1.

1.1 Opis problemu

Celem zadania jest napisanie funkcji w języku `Julia`, która oblicza ilorazy różnicowe dla danych węzłów $x_0, ..., x_n$ oraz wartości interpolowanej funkcji w tych węzłach $f(x_0), ..., f(x_n)$.

1.2 Rozwiązanie

Będziemy korzystać z tego, że ilorazy różnicowe spełniają zależność:

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

Znając węzły x_i i wartości funkcji $f(x_i)$ można za pomocą tego wzoru tworzyć trójkątną tablicę ilorazów różnicowych wyższych rzędów. Ogólniej przyjmując, że $c_{ij} = f[x_i, x_{i+1}, ..., x_{i+j}]$, tablicę wyrażamy tak:

$$
\begin{array}{ll|lllll}
x_0 & c_{00} & c_{01} & c_{02} & \cdots & c_{0,n-1} & c_{0n} \\
x_1 & c_{10} & c_{11} & c_{12} & \cdots & c_{1,n-1} & \\
\cdots & \cdots & \cdots & \cdots & & & \\
x_{n-1} & c_{n-1,0} & c_{n-1,1} & & & & \\
x_n & c_{n0} & & & & &
\end{array}
$$

W algorytmie będziemy liczyć "w miejscu" korzystając tylko z jednej tablicy jednowymiarowej fx. Początkową wartością $fx[i]$ jest $c_{i0} = f(x_i)$, następnymi wartościami – $c_{i-1,1}, ..., c_{1,i-1}, c_{01}$, przy czym ostatnia z tych wartości znajduje się w pierwszym wierszu tablicy trójkątnej. Tablicę tworzymy kolumnami, obliczając z dołu do góry. Dzięki tej kolejności fx zawiera w każdej chwili ilorazy, które będą poźniej potrzebne.

Dane: x, f
Wynik: fx
for $i = 0$ **to** n **do**
| $fx[i] \leftarrow f(x_i)$
end
for $j = 1$ **to** n **do**
| **for** $i = n$ **to** j **do**
| | $fx[i] \leftarrow (fx[i] - fx[i-1])/(x_i - x_{i-j})$
| **end**
end
return fx

2 Zadanie 2.

2.1 Opis problemu

Celem zadania jest napisanie funkcji w języku `Julia`, która oblicza 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 Rozwiązanie

Będziemy korzystać z tego, że $N_n(x)$ można obliczać za pomocą wzorów:
$w_n(x) = f[x_0, x_1, ..., x_n]$
$w_k(x) = f[x_0, x_1, ..., x_k] + (x - x_k)w_{k+1}(x)$ $(k = n - 1, ..., 0)$
$N_n(x) = w_0(x)$

Aby uzyskać czas $O(n)$ wynik obliczymy w jednej pętli, w której iterujemy od n do 1.

Dane: x, fx, t
Wynik: nt
$nt \leftarrow fx[n]$
for $i = n - 1$ **to** 1 **do**
| $nt \leftarrow f[i] + (t - x[i]) * nt$
end
return nt

3 Zadanie 3.

3.1 Opis problemu

Celem zadania jest napisanie funkcji w języku `Julia`, która mając współczynniki wielomianu interpolacyjnego w postaci Newtona c_i (ilorazy różnicowe) oraz węzły $x_0, ..., x_n$, obliczy, w czasie $O(n^2)$, współczynniki jego postaci naturalnej $a_0, ..., a_n$.

3.2 Rozwiązanie

Skorzystamy z własności z poprzedniego zadania, czyli uogólnionego algorytmu Hornera. Zauważmy, że $a_n = c_n$, ponieważ $w_n(x) = f[x_0, x_1, ..., x_n]$. Teraz mając a_n możemy wyliczać kolejne współczynniki iteracyjnie doprowadzając je do końcowej postaci. Zaczynamy od obliczenia a_i bazując na a_{i+1}, następnie w pętli wewnętrznej aktualizujemy współczynniki idąc od $i+1$ do $n-1$. W ten sposób w trakcie działania algorytmu będziemy operowali na częściowych współczynnikach, a po jego zakończeniu uzyskamy poprawne końcowe wyniki.

Dane: x, fx
Wynik: a
$a[n] \leftarrow fx[n]$
for $i = n-1$ **to** 0 **do**
  $a[i] \leftarrow fx[i] - a[i+1] * x[i]$
  for $j = i+1$ **to** $n-1$ **do**
    $a[j] \leftarrow a[j] - a[j+1] * x[i]$
  end
end
return a

3.3 Testy do funkcji

Testy dla kilku przykładów znajdują się w pliku `tests.jl`. Wszystkie funkcje zwracały poprawne wyniki dla zadanych danych, zatem są dobrze zaimplementowane.

4 Zadanie 4.

4.1 Opis problemu

Celem zadania jest napisanie funkcji w języku `Julia`, która zinterpoluje zadaną funkcję f na przedziale $[a, b]$ za pomocą wielomianu interpolacyjnego stopnia n w postaci Newtona. Następnie należy narysować wielomian interpolacyjny i interpolowaną funkcję.

4.2 Rozwiązanie

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

Najpierw obliczamy węzły oraz wartości funkcji w tych węzłach. Następnie przekazujemy te wartości do funkcji `ilorazyRoznicowe`, której wynik znów przekazujemy do funckji `warNewton`, która oblicza wartości wielomianu interpolacyjnego stopnia n w punktach, przy czym używamy większej liczby równoodległych węzłów, aby lepiej zobrazować wynik.

Ostatecznie wykresy rysujemy za pomocą `PyPlot`.

5 Zadanie 5.

5.1 Opis problemu

Celem zadania jest przetestowanie funkcji `rysujNnfx` na przykładach:

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

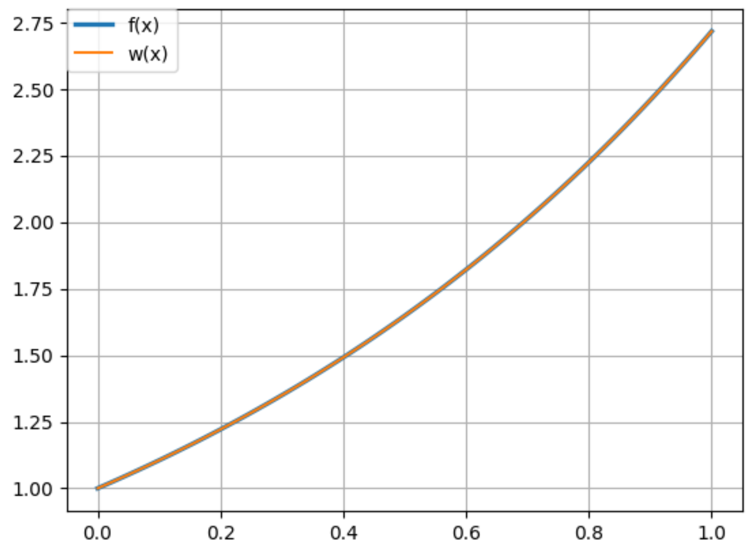
5.2 Rozwiązanie

Używamy naszego modułu.

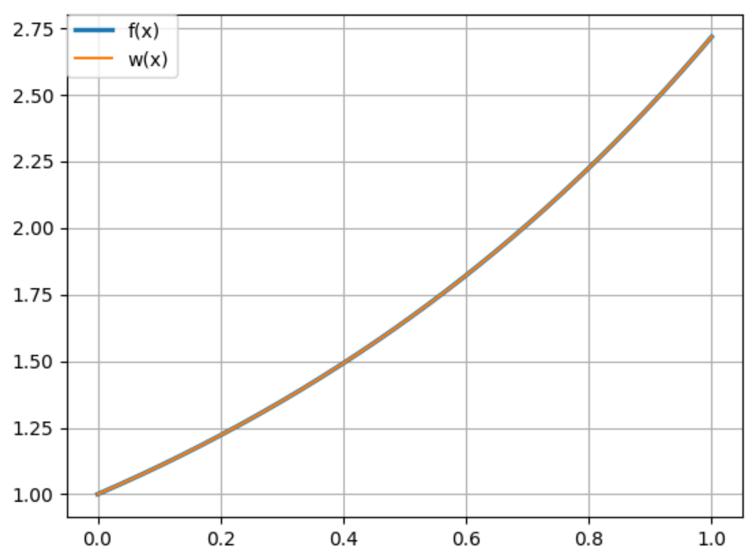
```
include("../module/interpolation_module.jl")
using .interpolation_module

f(x) = exp(x)
g(x) = (x^2)*sin(x)

for i in 1:3
    rysujNnfx(f, 0.0, 1.0, 5*i)
    rysujNnfx(g, -1.0, 1.0, 5*i)
end
```

5.3 Wyniki i interpretacja

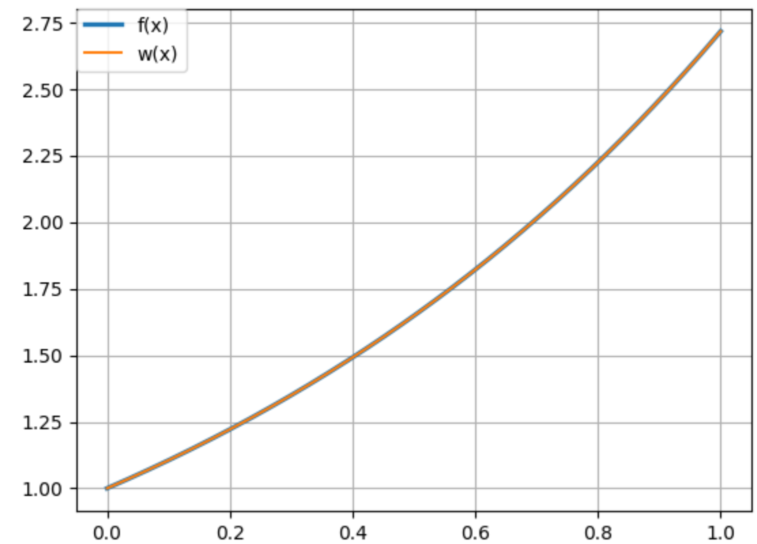
Poniższe wykresy pokazują wyniki kolejno dla a) oraz b).

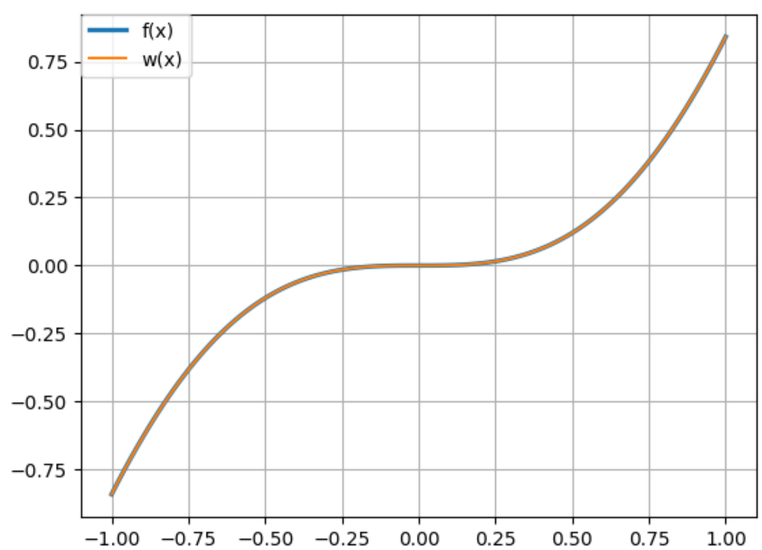


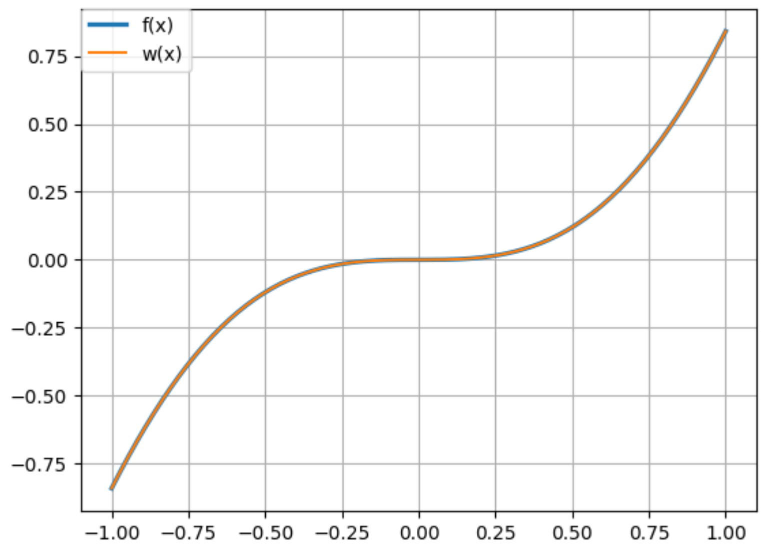
f(x)
w(x)
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
0.0
0.2
0.4
0.6
0.8
1.0

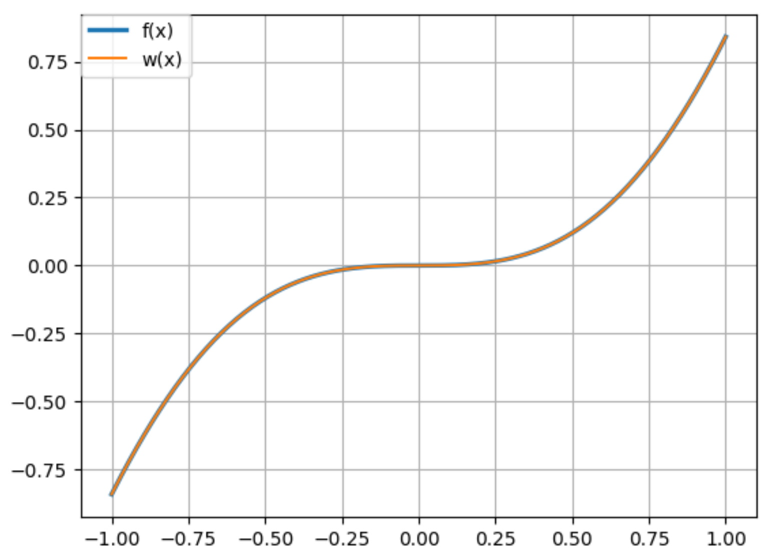


f(x)
w(x)
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
0.0
0.2
0.4
0.6
0.8
1.0


f(x)
w(x)
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
0.0
0.2
0.4
0.6
0.8
1.0


f(x)
w(x)
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00





Widzimy, że poprawne wyniki na tych przedziałach uzyskaliśmy dla wszystkich przypadków, również, gdy n było stosunkowo małe.

5.4 Wnioski

Dla niektórych funkcji i niewielkich przedziałów dokładna interpolacja zachodzi nawet dla małych n.

6 Zadanie 6.

6.1 Opis problemu

Celem zadania jest przetestowanie funkcji `rysujNnfx` na przykładach:

- $|x|, [-1, 1], n = 5, 10, 15$
- $\frac{1}{1+x^2}, [-5, 5], n = 5, 10, 15$

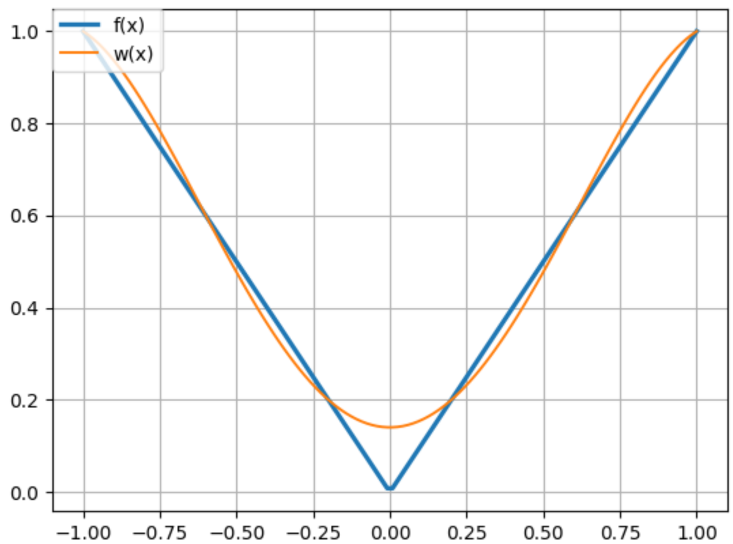
6.2 Rozwiązanie

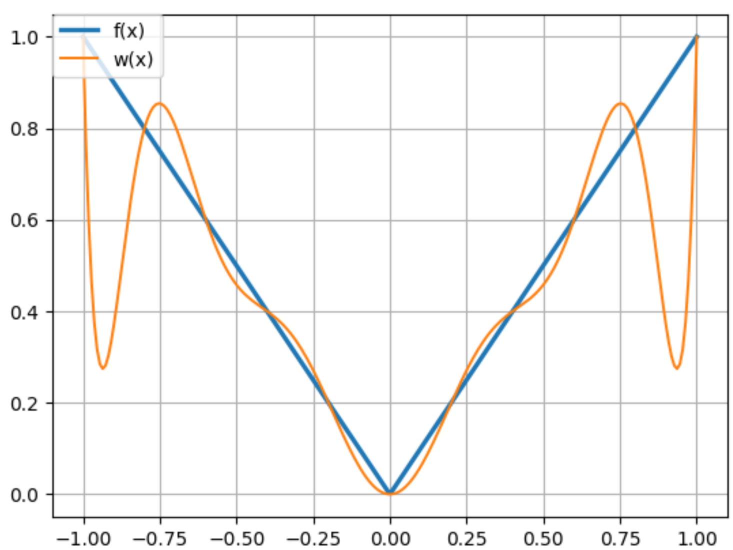
Używamy naszego modułu.

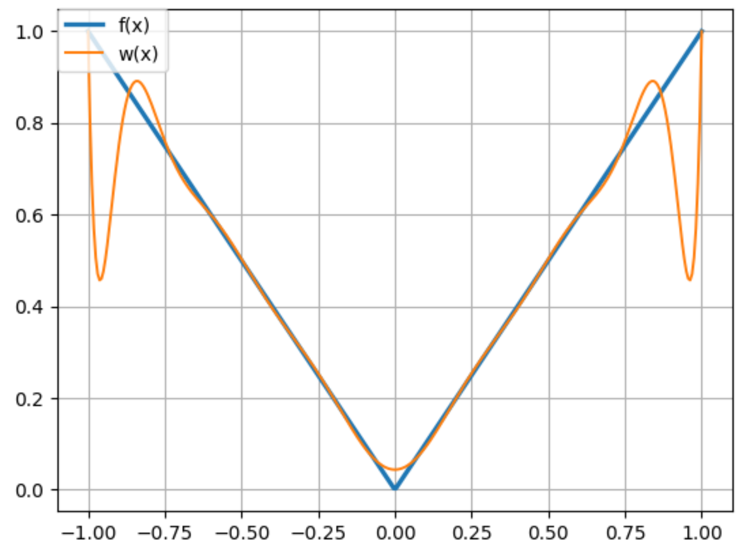
```
include("../module/interpolation_module.jl")
using .interpolation_module

h(x) = abs(x)
u(x) = 1/(1 + x^2)

for i in 1:3
    rysujNnfx(h, -1.0, 1.0, 5*i)
    rysujNnfx(u, -5.0, 5.0, 5*i)
end
```

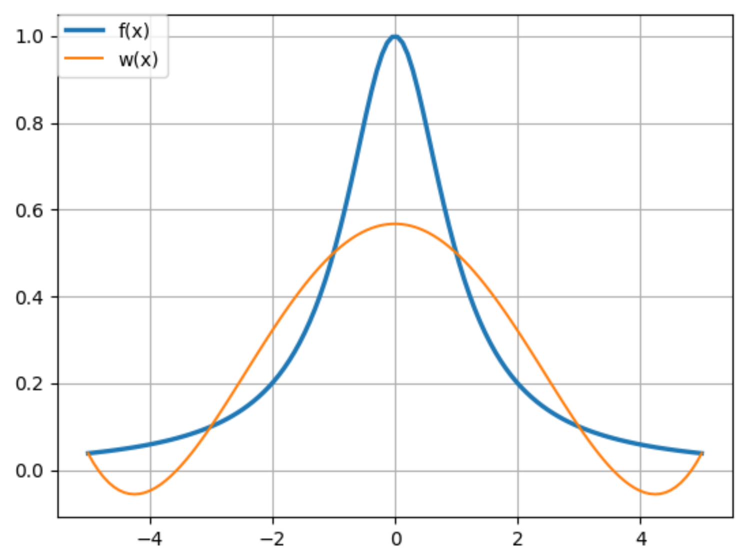
6.3 Wyniki i interpretacja

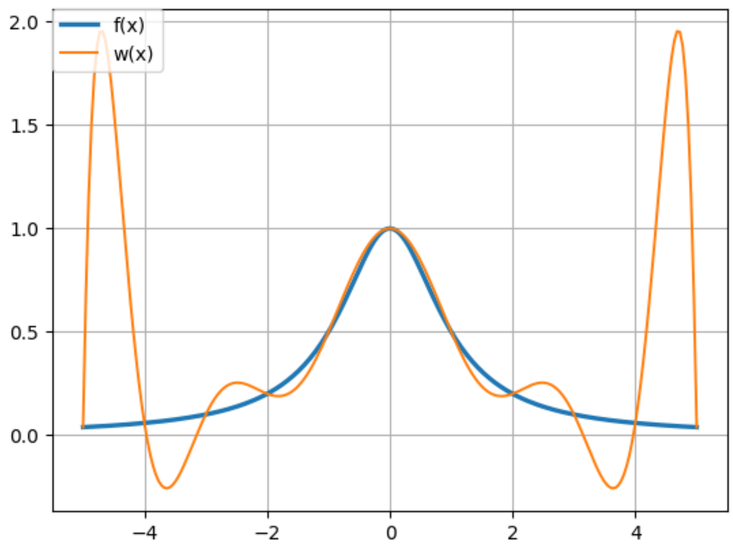
Poniższe wykresy pokazują wyniki kolejno dla a) oraz b).

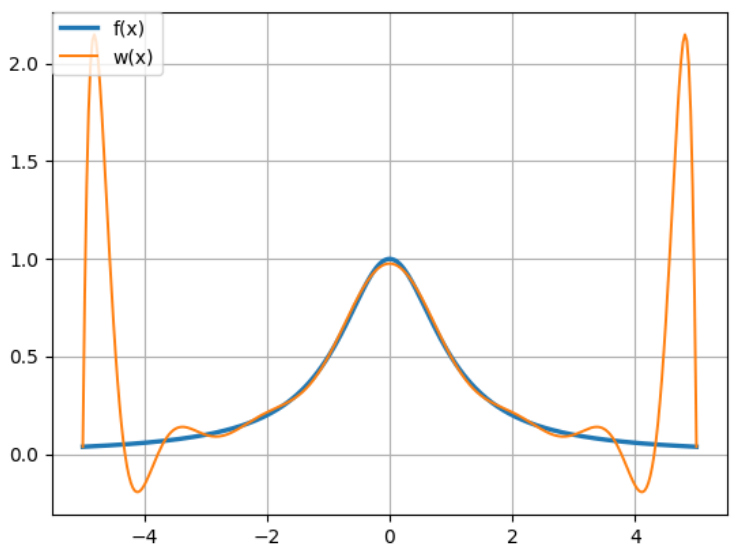

f(x)
w(x)
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


f(x)
w(x)
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


f(x)
w(x)
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


f(x)
w(x)
1.0
0.8
0.6
0.4
0.2
0.0
−4
−2
0
2
4





Zauważamy, że w obu przypadkach otrzymaliśmy niepoprawną interpolację, blisko końców przedziałów wielomian oscyluje z bardzo dużymi odchyleniami od prawidłowej wartości. Zwiększenie wartości n wcale nie rozwiązuje tego problemu, ponieważ zachodzi zjawisko Runge'go, czyli jakość interpolacji się zmniejsza, mimo wzrostu liczby węzłów. Jest to spowodowane przez stałą odległość między węzłami, a także występuje, gdy interpolowana funkcja jest nieciągła lub znacznie odbiega od funkcji gładkiej.

6.4 Wnioski

Niektóre interpolowane funkcje generują znaczne błędy w zaimplementowanych metodach, gdy używamy równoodległych węzłów. Aby uniknąć takiego zachowania należy inaczej rozlokować węzły, na przykład używając wielomianów Czebyszewa.