Obliczenia naukowe
Lista 4

Szymon Janiak

December 2, 2023

Zadanie 1

Opis problemu

Napisać funkcję obliczającą ilorazy różnicowe dla podanych węzłów oraz wartości danej funkcji w tych węzłach.

Dane wejściowe

- `x` — wektor długości $n + 1$ zawierający węzły $x_0, \ldots, x_n$
- `f` — wektor długości $n+1$ zawierający wartości interpolowanej funkcji w węzłach $f(x_0), \ldots, f(x_n)$

Dane wyjściowe

- `fx` — wektor długości $n + 1$ zawierający obliczone ilorazy różnicowe

Rozwiązanie

Algorithm 1 Obliczanie ilorazów różnicowych

```
function DifferenceQuotients(x, f)
    fx ← []
    f_copy ← copy(f)
    len ← length(x)
    for i ← 1 to len do
        for k ← (i − 1) downto 1 do
            a ← f_copy[k + 1] − f_copy[k]
            b ← (x[i] − x[k])
            f_copy[k] ← a/b
        end for
        push!(fx, f_copy[1])
    end for
    return fx
end function
```

Opis użytego algorytmu

Do działania naszego algorytmu tworzymy kopię wartości fx, w której będziemy obliczać kolejne iteracje naszego algorytmu. Dla oszczędności wykorzystywanej pamięci będziemy zapamiętywać jedynie ostatnio obliczone wartości, gdyż tylko takie będą nam potrzebne. Zewnętrzna pętla iteruje po kolejnych węzłach x_i, natomiat wewnętrzna oblicza ilorazy różnicowe dla danej iteracji według wzoru:

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

Po zakończeniu obliczeń dla danego i do naszej tablicy wyników dodajemy jedynie pierwszy iloraz różnicowy. Algorytm ten bazuje na obliczaniu ilorazów różnicowych dla coraz to mniejszych zbiorów węzłów.

Zadanie 2

Opis problemu

Napisać funkcję obliczającą wartość wielomianu interpolacyjnego stopnia n w postaci Newtona $N_n(x)$ w punkcie $x = t$ za pomocą algorytmu uogólnionego Hornera w czasie $O(n)$.

Dane wejściowe

- `x` — wektor długości $n + 1$ zawierający węzły $x_0, \ldots, x_n$
- `fx` — wektor długości $n + 1$ zawierający ilorazy różnicowe $f[x_0], \ldots, f[x_0, \ldots, x_n]$
- `t` — punkt, w którym należy obliczyć wartość wielomianu

Dane wyjściowe

- `nt` — wartość wielomianu w punkcie `t`

Rozwiązanie

Algorithm 2 Wartość wielomianu Newtona

function NewtonValue(x, fx, t)
    len $\leftarrow$ length(x)
    $n\_value \leftarrow fx$[len]
    for $k \leftarrow$ (len $- 1$) **downto** 1 **do**
        $n\_value \leftarrow n\_value \cdot (t - x[k]) + fx[k]$
    end for
    return n_value
end function

Opis użytego algorytmu

Algorytm bazuje na uogólnionym sposobie Hornera w czasie liniowym. Początkowo ustawiamy naszą wartość na wartość współczynnika o najwyższym stopniu, czyli $f[len]$, gdzie len to liczba węzłów. W każdej iteracji pętli aktualizowana jest wartość według uogólnionego wzoru Hornera:

$$n(x) = f[x_1] + (x - x_1)\,[f[x_2] + (x - x_2)\,[f[x_3] + \ldots + (x - x_{n-1})\,[f[x_n]]\ldots]]$$

Po zakończeniu pętli algorytm zwraca obliczoną wartość, która reprezentuje wartość wielomianu Newtona w punkcie t.

Zadanie 3

Opis problemu

Napisać funkcję obliczającą współczynniki postaci naturalnej wielomianu interpolacyjnego stopnia n w postaci Newtona $N_n(x)$.

Dane wejściowe

- `x` — wektor długości $n+1$ zawierający węzły $x_0, \ldots, x_n$
- `fx` — wektor długości $n+1$ zawierający ilorazy różnicowe $f[x_0], \ldots, f[x_0, \ldots, x_n]$

Dane wyjściowe

- `a` — wektor długości $n+1$ zawierający obliczone współczynniki postaci naturalnej $(a_n x^n + a_{n-1}x^{n-1} + \cdots a_1 x + a_0)$

Rozwiązanie

Algorithm 3 Obliczanie współczynników naturalnego kształtu wielomianu

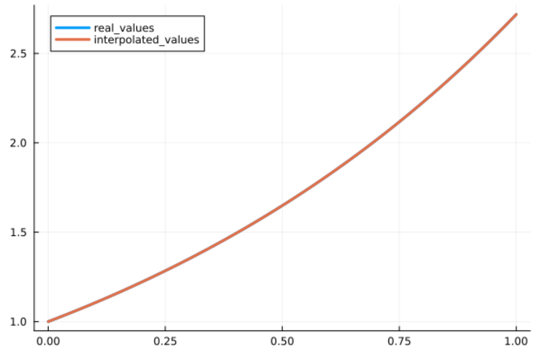
function NATURAL(x, fx)
    len $\leftarrow$ length(x)
    $f\_copy \leftarrow$ copy(fx)
    for $i \leftarrow$ (len $-$ 1) **downto** 1 **do**
        $f\_copy[i] \leftarrow fx[i] - f\_copy[i+1] \cdot x[i]$
        for $j \leftarrow (i+1)$ **to** (len $-$ 1) **do**
            $f\_copy[j] \leftarrow f\_copy[j] - f\_copy[j+1] \cdot x[i]$
        end for
    end for
    return f_copy
end function

Opis użytego algorytmu

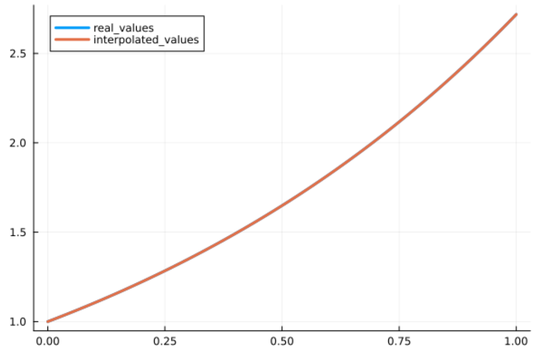
Zadanie 5

Przetestować metodę `draw_Nnfx` na kilku zadanych poniżej funkcjach.

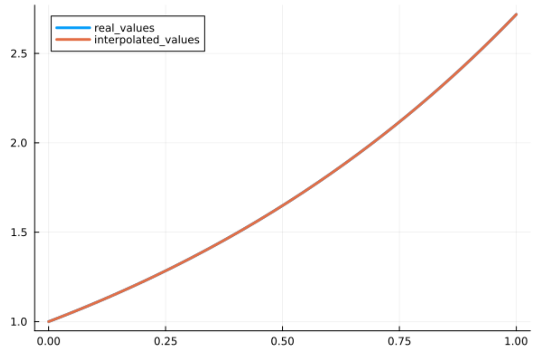
Wyniki



(a) $n = 5$

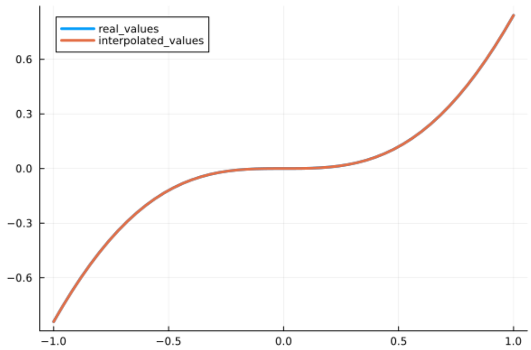


(b) $n = 10$

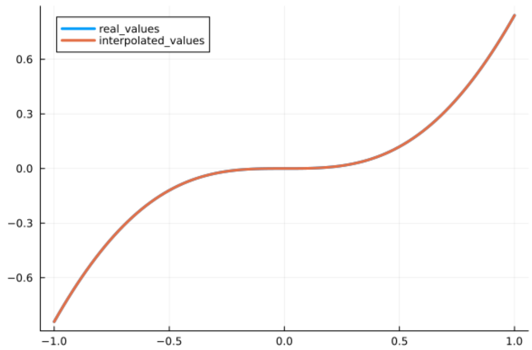


(c) $n = 15$

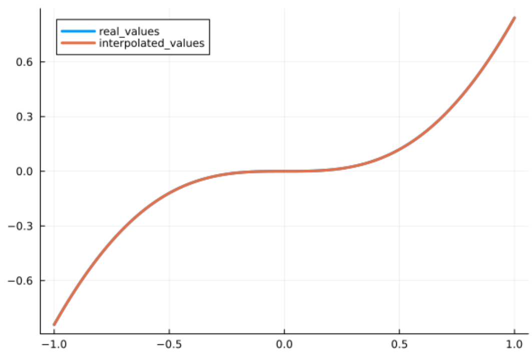
$f(x) = e^x$ w przedziale $[0; 1]$ dla $n = 5, 10, 15$



(a) $n = 5$



(b) $n = 10$



(c) $n = 15$

$f(x) = x^2 \cdot sinx$ w przedziale $[-1; 1]$ dla $n = 5, 10, 15$

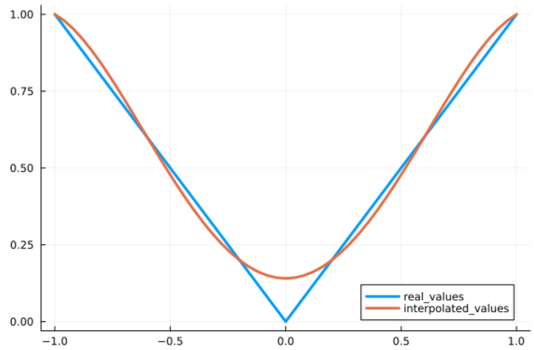
Wnioski

Na wykresach możemy zauważyć, że wykres interpolowanej przez nas funkcji pokrywa się praktycznie idealnie z rzeczywistymi wartościami dla stopni $n = 5, 10, 15$

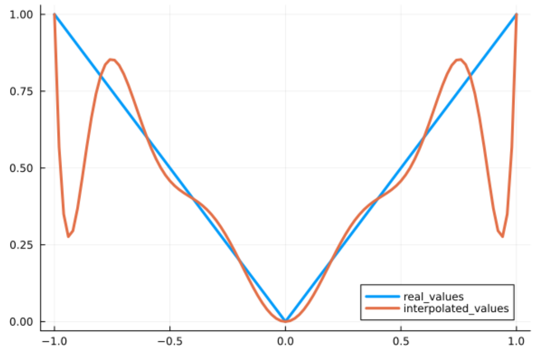
Zadanie 6

Przetestować metodę `draw_Nnfx` na kilku zadanych poniżej funkcjach.

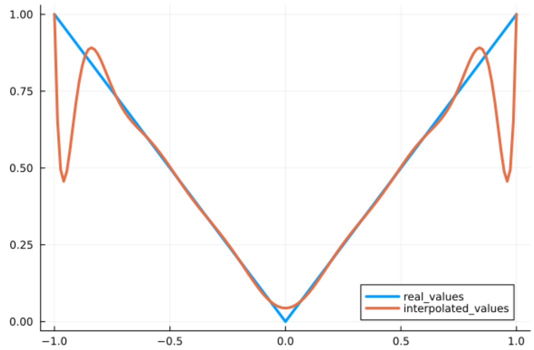
Wyniki



(a) $n = 5$

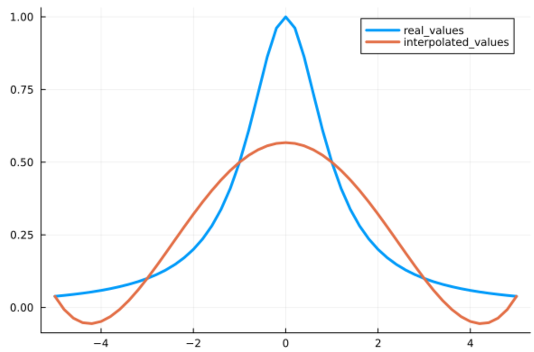


(b) $n = 10$

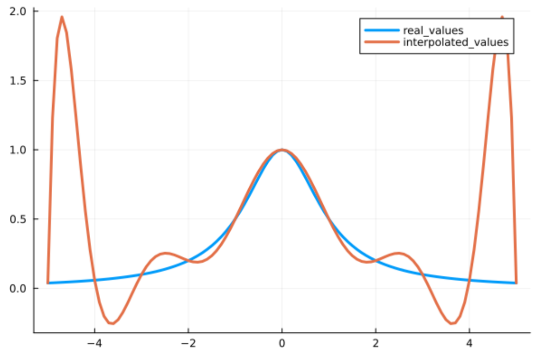


(c) $n = 15$

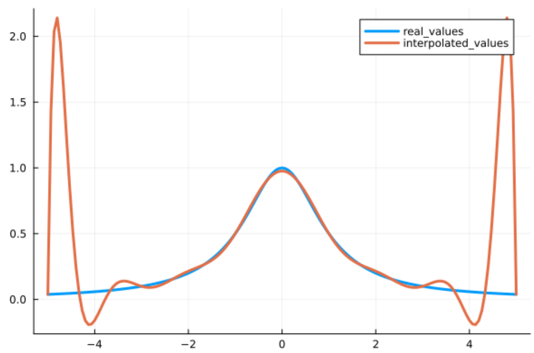
$f(x) = |x|$ w przedziale $[-1; 1]$ dla $n = 5, 10, 15$



(a) $n = 5$



(b) $n = 10$



(c) $n = 15$

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

Wnioski

Widzimy, że w tym przypadku funkcje nie interpolują się za dobrze. Nie widać również poprawy przy zwiększaniu stopnia n co daje różne rezultaty. Pokazuje to dobrze ograniczenia interpolacji wielomianowej. Lepsze efekty mogłaby przynieść interpolacja wielomianu poprzez podzielenie go na mniejsze przedziały i lepsze dobranie wartości węzłów interpolacji.