

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

- \mathbf{x} — wektor długości $n + 1$ zawierający węzły x_0, \dots, x_n
- \mathbf{f} — wektor długości $n + 1$ zawierający wartości interpolowanej funkcji w węzłach $f(x_0), \dots, f(x_n)$

Dane wyjściowe

- \mathbf{fx} — wektor długości $n + 1$ zawierający obliczone ilorazy różnicowe

Rozwiązanie

Algorithm 1 Obliczanie ilorazów różnicowych

```
1: function DIFFERENCEQUOTIENTS( $x, f$ )
2:    $fx \leftarrow []$ 
3:    $f\_copy \leftarrow \text{copy}(f)$ 
4:    $len \leftarrow \text{length}(x)$ 
5:   for  $i \leftarrow 1$  to  $len$  do
6:     for  $k \leftarrow (i - 1)$  downto  $1$  do
7:        $a \leftarrow f\_copy[k + 1] - f\_copy[k]$ 
8:        $b \leftarrow (x[i] - x[k])$ 
9:        $f\_copy[k] \leftarrow \frac{a}{b}$ 
10:    end for
11:     $\text{push!}(fx, f\_copy[1])$ 
12:  end for
13:  return  $fx$ 
14: 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 , natomiast wewnętrzna oblicza ilorazy różnicowe dla danej iteracji według wzoru:

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

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

Dane wyjściowe

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

Rozwiązanie

Algorithm 2 Wartość wielomianu Newtona

```
1: function NEWTONVALUE( $x, fx, t$ )
2:    $len \leftarrow \text{length}(x)$ 
3:    $n\_value \leftarrow fx[len]$ 
4:   for  $k \leftarrow (len - 1)$  downto 1 do
5:      $n\_value \leftarrow n\_value \cdot (t - x[k]) + fx[k]$ 
6:   end for
7:   return  $n\_value$ 
8: 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:

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

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

- \mathbf{x} — wektor długości $n + 1$ zawierający węzły x_0, \dots, x_n
- \mathbf{fx} — wektor długości $n + 1$ zawierający ilorazy różnicowe $f[x_0], \dots, f[x_0, \dots, x_n]$

Dane wyjściowe

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

Rozwiązanie

Algorithm 3 Obliczanie współczynników naturalnego kształtu wielomianu

```
1: function NATURAL( $x, fx$ )
2:    $\text{len} \leftarrow \text{length}(x)$ 
3:    $f\_copy \leftarrow \text{copy}(fx)$ 
4:   for  $i \leftarrow (\text{len} - 1)$  downto 1 do
5:      $f\_copy[i] \leftarrow fx[i] - f\_copy[i + 1] \cdot x[i]$ 
6:     for  $j \leftarrow (i + 1)$  to  $(\text{len} - 1)$  do
7:        $f\_copy[j] \leftarrow f\_copy[j] - f\_copy[j + 1] \cdot x[i]$ 
8:     end for
9:   end for
10:  return  $f\_copy$ 
11: end function
```

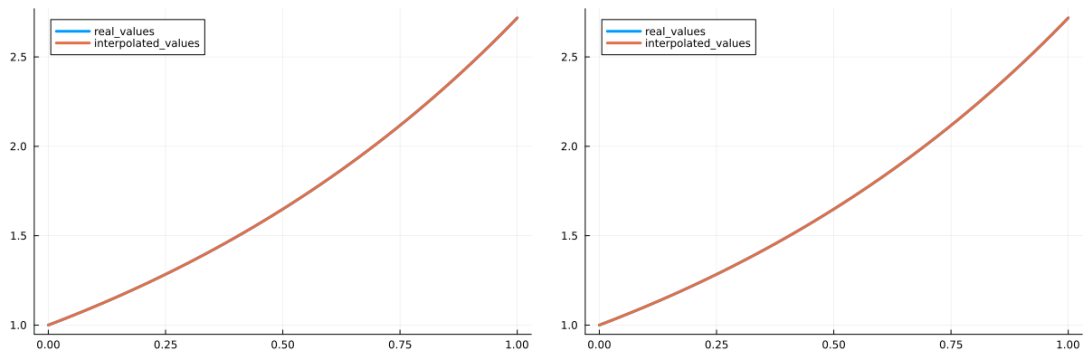
Opis użytego algorytmu

Algorytm ten korzysta z obliczonych ilorazów różnicowych dla wielomianu interpolacyjnego. Ilorazy różnicowe to $c_0, c_1, c_2 \dots c_n$. Współczynnikiem a_n stojącym przy najwyższej potędze, x^n jest c_n

Zadanie 5

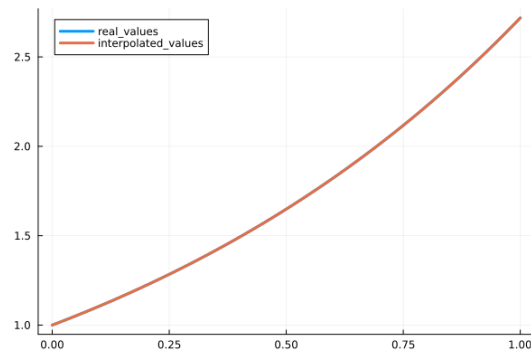
Przetestować metodę `draw_Nnfx` na kilku zadanych poniżej funkcjach.

Wyniki



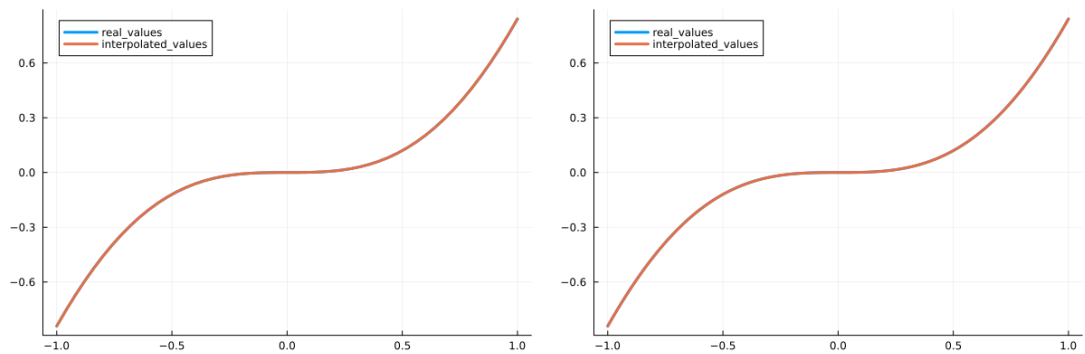
(a) $n = 5$

(b) $n = 10$



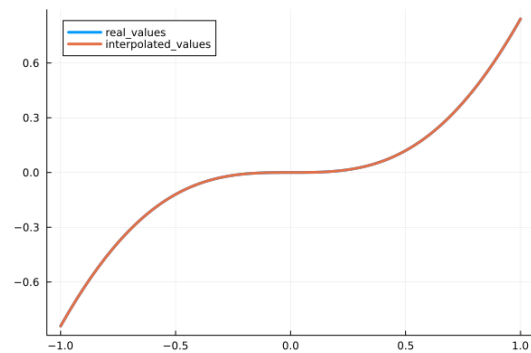
(c) $n = 15$

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



(a) $n = 5$

(b) $n = 10$



(c) $n = 15$

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

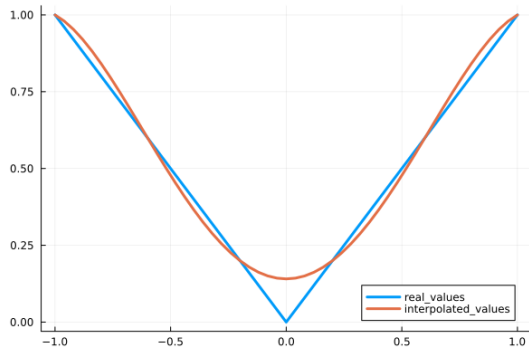
Wnioski

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

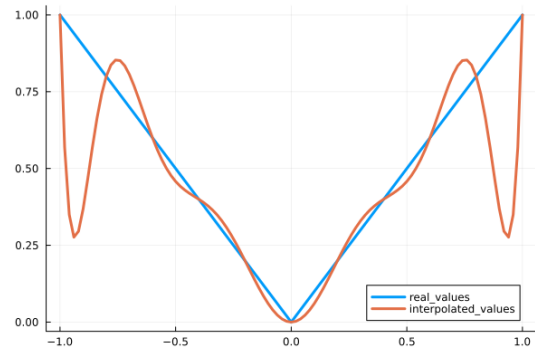
Zadanie 6

Przetestować metodę `draw_Nnfx` na kilku zadanych poniżej funkcjach.

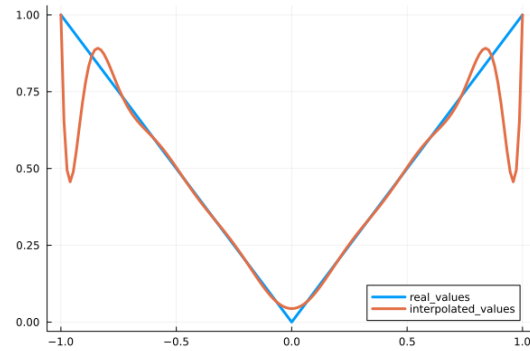
Wyniki



(a) $n = 5$

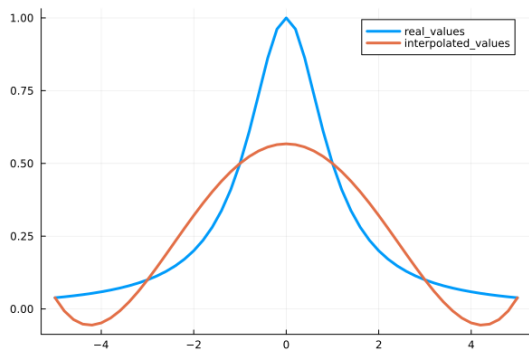


(b) $n = 10$

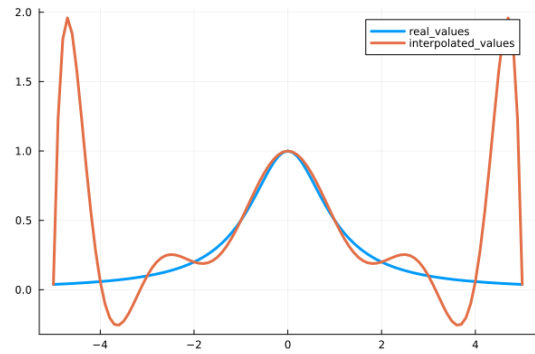


(c) $n = 15$

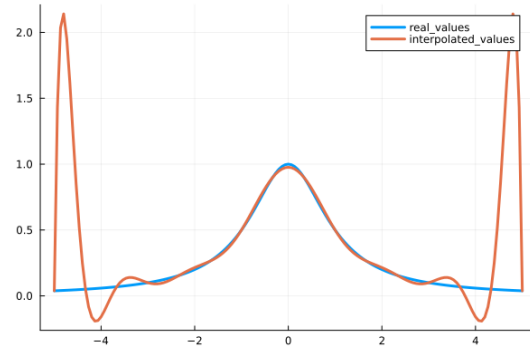
$f(x) = |x|$ w przedziale $[-1; 1]$ dla $n = 5, 10, 15$



(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.