Obliczenia naukowe
Lista 4

Joanna Szołomicka

8 grudnia 2019

1 Zadanie 1

1.1 Opis problemu

Napisanie funkcji `function ilorazyRoznicowe (x::VectorFloat64, f::VectorFloat64)` obliczającej ilorazy różnicowe na podstawie podanych węzłów i odpowiadających im wartości funkcji bez użycia tablicy dwuwymiarowej.

Dane:

`x` – wektor długości $n+1$ zawierający węzły $x_0, \ldots, x_n$,

`f` – wektor długości $n+1$ zawierający wartości interpolowanej funkcji w węzłach $f(x_0), \ldots, f(x_n)$.

Wyniki:

`fx` – wektor długości $n+1$ zawierający obliczone ilorazy różnicowe.

1.2 Rozwiązanie

Iloraz różnicowy k-tego rzędu można obliczyć stosując następujący wzór rekurencyjny:

1. dla $k = 0$
$$f[x_i] = f(x_i),$$
2. dla $k = 1$
$$f[x_i, x_j] = \frac{f(x_j) - f(x_i)}{x_j - x_i},$$
3. dla $k > 1$
$$f[x_i, x_{i+1}, \ldots, x_{i+k}] = \frac{f[x_{i+1}, x_{i+2}, \ldots, x_{i+k}] - f[x_i, x_{i+1}, \ldots, x_{i+k-1}]}{x_k - x_i}.$$

Można zauważyć, że iloraz różnicowy nie zależy od kolejności węzłów x_i, co jest wykorzystywane w praktycznym wykorzystaniu ilorazów różnicowych. Ilorazy wyliczone za pomocą powyższego wzoru możemy reprezentować w tablicy trójkątnej przyjmując następujące oznaczenie $c_{ik} = f[x_i, x_{i+1}, \ldots, x_{i+k}]$:

$$\begin{array}{llllll} c_{0,0} & c_{0,1} & c_{0,2} & \cdots & c_{0,k-1} & c_{0,k} \\ c_{1,0} & c_{1,1} & c_{1,2} & \cdots & c_{1,k-1} & \\ \cdots & \cdots & \cdots & & & \\ c_{k-1,0} & c_{k-1,1} & & & & \\ c_{k,0} & & & & & \end{array} \tag{1}$$

Można zauważyć, że algorytm obliczania ilorazów różnicowych nie musi wykorzystywać powyższej tablicy dwuwymiarowej, gdyż celem zadania jest obliczenie jedynie pierwszego wiersza tablicy trójkątnej. Wystarczająca jest jednowymiarowa tablica fx, której początkowymi wartościami zmiennych fx_i są $c_{i,0} = f(x_i)$, a następnymi wartościami odpowiednio $c_{i-1,1}, \ldots, c_{1,i-1}, c_{0,i}$ dla każdej kolejnej zmiennej fx_i. W każdej iteracji algorytmu zmienne tablicy fx to kolejne kolumny tablicy trójkątnej. Każda kolumna aktualizowana jest "od dołu do góry", dzięki czemu tablica zawiera ilorazy różnicowe potrzebne do wyliczenia nowych wartości w każdej iteracji.

Algorithm 1: Obliczanie ilorazów różnicowych

```
function ilorazyRoznicowe(x,f)
    for i ← 1 to length(f) do
        fx[i] ← f[i]
    end
    for i ← 2 to length(f) do
        for j ← length(f) downto i do
            fx[j] ← (fx[j] − fx[j − 1]) / (x[j] − x[j − i])
        end
    end
    return fx
```

2 Zadanie 2

2.1 Opis problemu

Napisanie funkcji `function warNewton (x::VectorFloat64, fx::VectorFloat64, t::Float64)` obliczającej wartość wielomianu interpolacyjnego stopnia n w postaci Newtona $N_n(x)$ w punkcie $x = t$ za pomocą uogólnionego algorytmu Hornera działającej w czasie $O(n)$.

Dane:

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

Wyniki:

`nt`	–	wartość wielomianu w punkcie t

2.2 Rozwiązanie

Wzór wielomianu interpolacyjnego Newtona można przedstawić używając ilorazów różnicowych:

$$N_n(x) = \sum_{i=0}^{n} f[x_0, x_1, \ldots, x_i] \prod_{j=0}^{i-1} (x - x_j).$$

Można zauważyć zależność wielomianu interpolacyjnego N_n od funkcji f.
Powyższy wzór jest dobrze uwarunkowany pod względem numerycznym. Dodanie nowych punktów (x_i, y_i) nie narusza obliczonych wcześniej współczynników $c_k = f[x_0, x_1, \ldots, x_k]$. Wartość tak przedstawionego wielomianu można obliczyć za pomocą uogólnionego schematu Hornera:

$$\begin{aligned}
&w_n(x) := f[x_0, x_1, \ldots, x_n] \\
&w_k(x) := w_{k+1}(x - x_k) + f[x_0, x_1, \ldots, x_k], \quad dla \quad (k = n-1, n-2, \ldots, 0) \\
&N_n(x) = w_0(x)
\end{aligned} \tag{2}$$

Algorithm 2: Obliczanie wartości wielomianu interpolacyjnego w punkcie t.

```
function warNewton(x, fx, t)
    n ← length(fx)
    nt ← fx[n]
    for i ← n − 1 downto 1 do
        nt ← fx[i] + (t − x[i]) ∗ nt
    end
    return nt
```

Po n iteracjach pętli otrzymamy wartość wielomianu interpolacyjnego w punkcie t. Złożoność algorytmu to $O(n)$.

3 Zadanie 3

3.1 Opis problemu

Napisanie funkcji obliczającej współczynniki $a_0, \ldots, a_n$ postaci naturalnej wielomianu interpolacyjnego tzn. $a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$ dla zadanych współczynników $c_0 = f[x_0], c_1 = f[x_0, x_1], \ldots c_n = f[x_0, \ldots, x_n]$ tego wielomianu w postaci Newtona oraz węzłów $x_0, \ldots, x_n$. Złożoność czasowa funkcji `function naturalna (x::VectorFloat64, fx::VectorFloat64)`: $O(n^2)$.

Dane:

`x`	–	wektor długości $n+1$ zawierający węzły $x_0, \ldots, x_n$,
`fx`	–	wektor długości $n+1$ zawierający ilorazy różnicowe

Wyniki:

`a`	–	wektor długości $n+1$ zawierający obliczone współczynniki postaci naturalnej

3.2 Rozwiązanie

Żeby znaleźć współczynniki wielomianu interpolacyjnego w postaci naturalnej należy zastosować uogólniony algorytm Hornera przedstawiony w poprzednim zadaniu. W wielomianie interpolacyjnym n-tego stopnia współczynnik a_n przy najwyższej potędze x jest równy $c_n = f[x_0, \ldots, x_n]$, z czego wynika że $w_n = a_n$. Korzystając z tego faktu w kolejnych krokach algorytmu można wyliczyć wartości a_i bazujące na współczynnikach a_{i+1}. Podczas każdej iteracji każdy "składowy" wielomian sprowadzamy do postaci naturalnej, uzyskując w ten sposób szukane współczynniki postaci naturalnej wielomianu interpolacyjnego. Złożoność algorytmu wynosi $O(n^2)$ - pętla zewnętrzna wykona się n razy i pętla zewnętrzna w pesymistycznym przypadku również n razy.

Algorithm 3: Współczynniki naturalne wielomianu interpolacyjnego.

```
function naturalna(x, fx)
    n ← length(fx)
    a[n] ← fx[n]
    for i ← n − 1 downto 1 do
        a[i] ← fx[i] − a[i + 1] ∗ x[i]
        for j ← i + 1 downto n − 1 do
            a[j] ← a[j] − a[j + 1] ∗ x[i]
        end
    end
    return a
```

4 Zadanie 4

4.1 Opis problemu

Napisanie funkcji interpolującej zadaną funkcję $f(x)$ w przedziale $[a, b]$ za pomocą wielomianu interpolacyjnego stopnia n w postaci Newtona. Należy narysować wykresy f i wielomianu interpolującego. W interpolacji funkcji należy użyć węzłów równoodległych, tj. $x_k = a + kh, \quad h = (b - a)/n, k = 0, 1, \ldots, n$. Wykorzystać funkcje `ilorazyRoznicowe`, `warNewton`.

Funkcja `rysujNnfx`:
Dane:

`f`	–	zadana funkcja
`a, b`	–	przedział interpolacji
`n`	–	stopień wielomianu interpolacyjnego

Wyniki:

– funkcja rysuje wielomian interpolacyjny i interpolowaną funkcję w przedziale $[a, b]$

4.2 Rozwiązanie

Algorytm wykorzystując wcześniej zaimplementowane funkcje `ilorazyRoznicowe` oraz `warNewton` wyznacza ilorazy różnicowe i wartość wielomianu interpolacyjnego. Na początku wyznaczane są węzły interpolacji $(x_1, \ldots, x_{n+1})$, które znajdują się w odległości $\frac{b-a}{n}$ od siebie w przedziale $[a, b]$, a także wartości funkcji f w stworzonych węzłach: $(f(x_1), \ldots, f(x_{n+1}))$. Następnie przy pomocy funkcji `ilorazyRoznicowe` obliczone zostały ilorazy różnicowe dla wyznaczonych węzłów. Żeby uzyskać dokładniejsze wykresy wielomian interpolacyjny oraz funkcja f próbkowane są w $20*(n+1)$ równoodległych punktach, dla których wartości wielomianu obliczane są za pomocą funkcji `warNewton`. Na koniec rysowany jest odpowiedni wykres funkcji f i jej wielomianu interpolacyjnego.

Algorithm 4: Wykres funkcji interpolującej zadaną funkcję $f(x)$ oraz funkcji $f(x)$.

function `rysujNnfx`(f,a,b,n)
  $maxK \leftarrow n + 1$
  $kh \leftarrow 0$
  $multPlot \leftarrow 20$
  $h \leftarrow \frac{b-a}{n}$
  for $i \leftarrow 1$ **downto** $maxK$ **do**
    $x[i] \leftarrow a + kh$
    $y[i] \leftarrow f(x[i])$
    $kh \leftarrow kh + h$
  end
  $fx \leftarrow ilorazyRoznicowe(x, y)$
  $kh \leftarrow 0$
  $maxK \leftarrow maxK * multPlot$
  $h \leftarrow \frac{b-a}{maxK-1}$
  for $i \leftarrow 1$ **downto** $maxK$ **do**
    $img_x[i] = a + kh$
    $img_interp[i] = warNewton(x, fx, img_x[i])$
    $img_y[i] = f(img_x[i])$
    $kh \leftarrow kh + h$
  end
  $drawPlot(img_x, img_y, img_interp)$

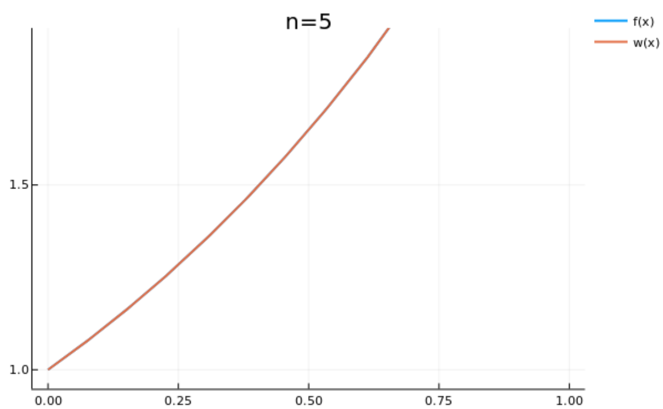
5 Zadanie 5

5.1 Opis problemu

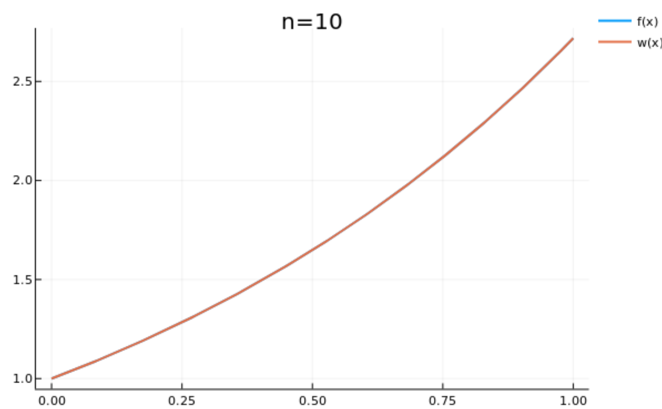
Przetestowanie funkcji `rysujNnfx(f,a,b,n)` z zadania 4 na następujących przykładach:

1. $f(x) = e^x$, $[a, b] = [0, 1]$, $n \in \{5, 10, 15\}$,
2. $f(x) = x^2 \sin x$, $[a, b] = [-1, 1]$, $n \in \{5, 10, 15\}$.

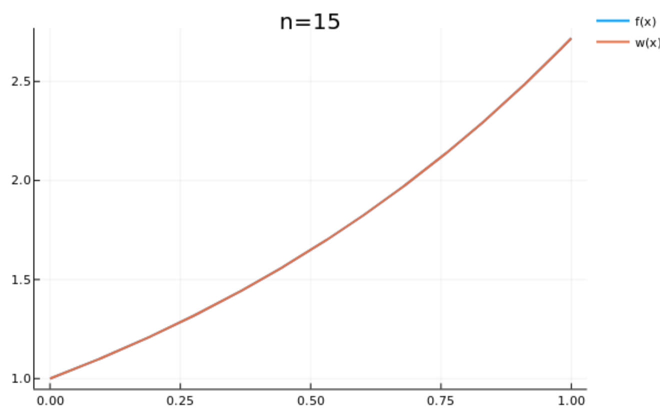
5.2 Wyniki



(a)



(b)

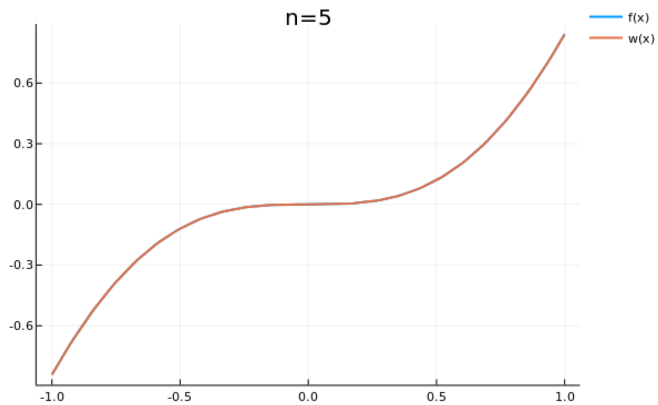


(c)

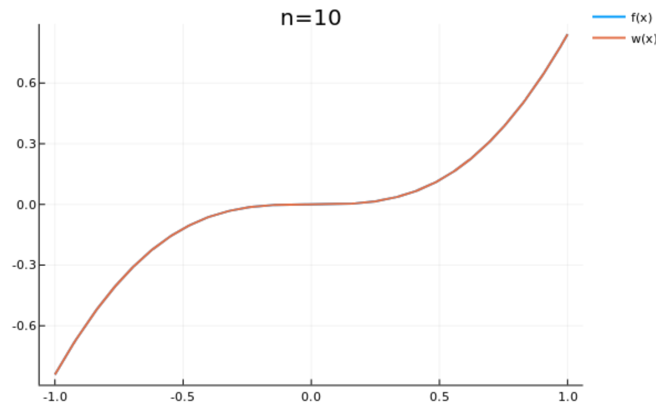
Rysunek 1: Wykresy e^x i jej wielomianu interpolacyjnego w zależności od stopnia n

5.3 Wnioski

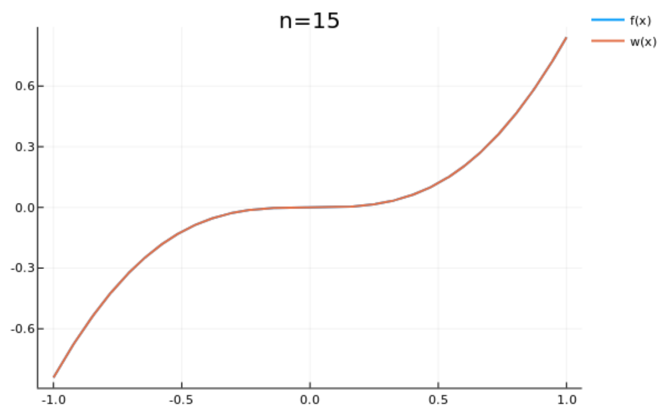
Na przedstawionych wykresach można zauważyć, że wartości dla zadanych funkcji i ich wielomianów interpolacyjnych są bardzo zbliżone dla wszystkich podanych $n \in \{5, 10, 15\}$. Wielomian interpolacyjny dobrze przybliżył funkcje, dzięki zastosowaniu równoodległych węzłów interpolacji.



(a)



(b)



(c)

Rysunek 2: Wykresy $x^2 \sin x$ i jej wielomianu interpolacyjnego w zależności stopnia n

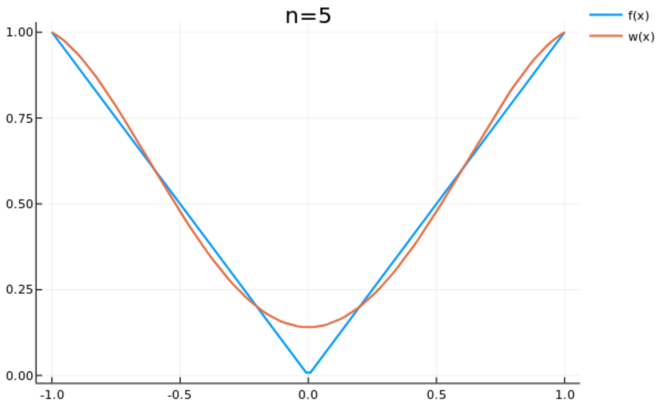
6 Zadanie 6

6.1 Opis problemu

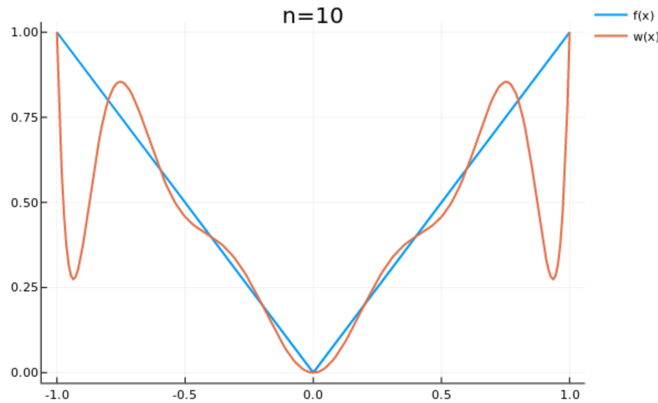
Przetestowanie funkcji `rysujNnfx(f,a,b,n)` z zadania 4 na następujących przykładach:

1. $f(x) = |x|$, $[a, b] = [-1, 1]$, $n \in \{5, 10, 15\}$,
2. $f(x) = \frac{1}{1+x^2}$, $[a, b] = [-5, 5]$, $n \in \{5, 10, 15\}$.

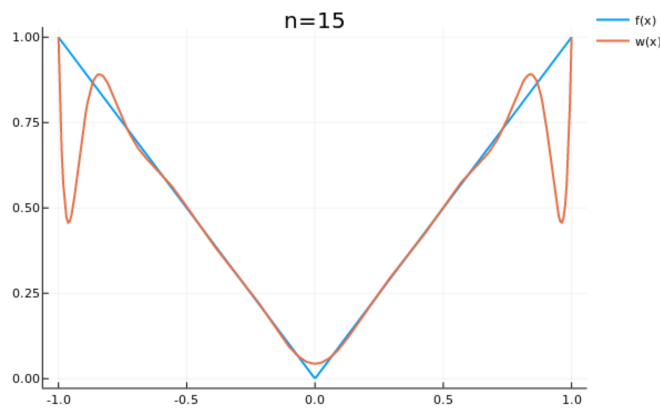
6.2 Wyniki



(a)



(b)

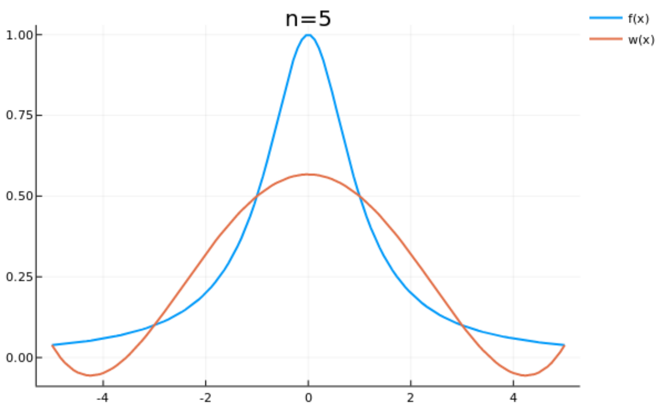


(c)

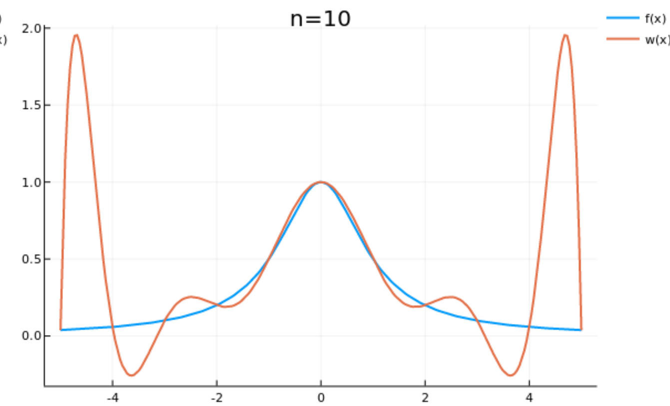
Rysunek 3: Wykresy $|x|$ i jej wielomianu interpolacyjnego w zależności od stopnia n

6.3 Wnioski

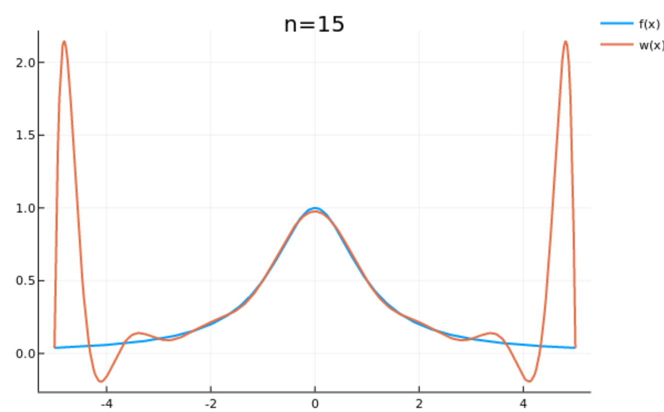
Na przedstawionych wykresach można zauważyć, że wartości dla obu funkcji i ich wielomianów interpolacyjnych są znacznie rozbieżne. W przypadku funkcji $f(x) = |x|$ rozbieżność ta wynika z faktu, że funkcja $|x|$ nie jest różniczkowalna - nie istnieje pochodna w punkcie $x_0 = 0$. Dla funkcji $f(x) = \frac{1}{1+x^2}$ można zaobserwować zjawisko Rungego. Im większa liczba węzłów n tym przybliżenie interpolacyjne jest gorsze. Zjawisko to pojawia się, gdy węzły wielomianu interpolacyjnego są równo od siebie oddalone i wielomian jest wysokiego stopnia. Powoduje to duże odchylenia na krańcach przedziału, na który funkcja jest interpolowana. Wielomiany wysokiego stopnia mają dużą liczbę zer i szybko rozbiegają do nieskończoności, a więc przyjmują skrajne wartości. Tam, gdzie interpolacja jest trudniejsza przypada niewielka liczba punktów, gdyż punkty są rownoodległe. Aby uniknąć takich błędów można zastosować wielomiany Czybyszewa n-tego stopnia, gdzie węzłami są ich miejsca zerowe. Miejsca zerowe w takich wielomianach są gęściej rozmieszczone na zadanym przedziale.



(a)



(b)



(c)

Rysunek 4: Wykresy $\frac{1}{1+x^2}$ i jej wielomianu interpolacyjnego w zależności stopnia n