Sprawozdanie

Obliczenia naukowe 2019/2020 – lista 4
Tomasz Janik

1. Zadanie 1.

1.1. Opis problemu

Zadanie to polegało na napisaniu funkcji, która obliczy ilorazy różnicowe na podstawie podanych węzłów oraz wartości funkcji w tych węzłach.

1.2. Opis algorytmu

Ilorazy różnicowe można obliczyć korzystając z następującego wzoru rekurencyjnego:

$$f[x_i, x_{i+1}, \dots, x_{i+j}] = \frac{f[x_{i+1}, x_{i+2}, \dots, x_{i+j}] - f[x_i, x_{i+1}, \dots, x_{i+j-1}]}{x_{i+j} - x_i},$$

$gdzie\ f[x_i] = f(x_i)$

$$oraz\ f[x_i, x_j] = \frac{f(x_j) - f(x_i)}{x_j - x_i}$$

Na podstawie tego wzoru można łatwo wyprowadzić algorytm bazujący na tablicy dwuwymiarowej, gdzie zapamiętujemy wszystkie wartości pośrednie i na ich podstawie wyliczamy następne ilorazy. Jednak nie potrzebujemy pamiętać wszystkich wartości pośrednich, więc możemy użyć tablicy jednowymiarowej oraz nadpisywać komórki po wyliczeniu nowego ilorazu, a po każdorazowym przejściu tablicy, będziemy mieli ilorazy rzędu o jeden większego niż w poprzedniej iteracji. Algorytm ten prezentuje się następująco:

```
function ILORAZYRÓŻNICOWE(x, f)
    size ← LENGTH(f)
    fx ← f
    for i ← 2 downto size do
        fx[j] ← (fx[j] − fx[j−1]) / (x[j] − x[j−i+1])
    return fx
```

- x – wektor węzłów,
- f – wektor wartości funkcji w węzłach,
- fx – wektor obliczonych ilorazów różnicowych ($fx[n] = f[x_0, \dots, x_{n-1}]$).

2. Zadanie 2.

2.1. Opis problemu

W tym zadaniu należało napisać funkcję obliczającą wartość wielomianu interpolacyjnego stopnia n w postaci Newtona $N_n(x)$ w punkcie $x = t$ za pomocą algorytmu Hornera, w czasie $O(n)$.

2.2. Opis algorytmu

Wielomian w postaci Newtona przedstawiony z pomocą ilorazów różnicowych wygląda następująco:

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

Powyższą zależność można łatwo zapisać za pomocą uogólnionych wzorów Hornera:

$$\begin{aligned}
&w_n = f[x_0, \dots, x_n] \\
&w_i = \ f[x_0, \dots, x_i] + (x - x_k) w_{k+1}(x), \qquad gdzie\ k = n-1, \dots, 0 \\
&N_n(x) = w_0(x).
\end{aligned}$$

Otrzymujemy z tej zależności następujący algorytm:

```
function warNewton(x, fx, t)
    size ← LENGTH(x)
    nt ← fx[size]
    for i ← size − 1 downto 1 do
        nt ← fx[i] + (t − x[i]) · nt
    return nt
```

- x – wektor węzłów,
- fx – wektor ilorazów różnicowych,
- t – punkt, w którym szukana jest wartość
- nt – wartość wielomianu interpolacyjnego w t

3. Zadanie 3

3.1. Opis problemu

W tym zadaniu należało napisać funkcję, która na podstawie współczynników wielomianu interpolacyjnego w postaci Newtona $c_0 = f[x_0], c_1 = f[x_0, x_1], \dots, c_n = f[x_0, \dots, x_n]$ oraz węzłów $x_0, x_1, \dots, x_n$ wyliczy współczynniki jego postaci naturalnej $a_0, \dots a_n$.

3.2. Opis algorytmu

Aby znaleźć współczynniki postaci naturalnej wielomianu interpolacyjnego Newtona możemy skorzystać z faktu, że $a_n = c_n$. Następnie, podobnie jak w poprzednim zadaniu, liczymy kolejne współczynniki zaczynając od tego przy najwyższej potędze. Przy kolejnych przejściach uaktualniamy wartości otrzymanych już współczynników, żeby utrzymać je w postaci naturalnej. Kiedy dojdziemy do ostatniego współczynnika, otrzymamy wszystkie współczynniki postaci naturalnej naszego wielomianu.

Algorytm realizujący tą metodę wygląda następująco:

```
function NATURALNA(x, fx)
    size ← LENGTH(x)
    a[size] ← fx[size]
    for i ← size − 1 downto 1 do
        a[i] = fx[i] − a[i + 1] · x[i]
        for j ← i + 1 to size − 1 do
            a[j] ← a[j] − a[j + 1] · x[i]
    return a
```

- x – wektor węzłów,
- fx – wektor obliczonych ilorazów różnicowych ($fx[n] = f[x_0, \dots, x_{n-1}]$),
- a – wektor obliczonych współczynników postaci naturalnej

4. Zadanie 4

4.1 Opis problemu

W tym zadaniu należało napisać funkcję, która zinterpoluje zadaną funkcję $f(x)$ w przedziale $[a, b]$ za pomocą wielomianu interpolacyjnego stopnia n w postaci Newtona. Następnie funkcja narysuje wielomian interpolacyjny oraz interpolowaną funkcję.

4.2 Opis algorytmu

Zadana funkcja ma być interpolowana za pomocą wielomianu stopnia n. Potrzebujemy zatem $n + 1$ równoodległych węzłów, tzn. takich x_k, że $x_k = a + kh, h = \frac{(b-a)}{n}, k = 0, 1, \dots, n$. Wyznaczamy więc te węzły oraz wartości zadanej funkcji w nich. Następnie obliczamy ilorazy różnicowe, za pomocą wcześniej napisanej funkcji *ilorazyRóżnicowe,* podając jako argumenty wyliczone węzły i wartości funkcji.

Następnie, na potrzeby wykresu obliczamy większą ilość równoodległych wartości x na przedziale $[a, b]$ (Tutaj 100 punktów, na każdy odcinek długości 1). Dla każdego x, wyliczamy rzeczywistą wartość funkcji oraz wartość wielomianu interpolacyjnego za pomocą wcześniej napisanej funkcji *warNewton,* podając jako argumenty początkowo wyliczone węzły, wektor ilorazów różnicowych oraz obecnie rozpatrywany x.

Na koniec na podstawie otrzymanych dwóch serii danych, rysujemy wykres i zapisujemy go (tutaj przy użyciu pakietu Plots).

Algorytm prezentuje się następująco:

```
function rysujNnfx(f, a, b, n)
    h ← (b−a)/n
    kh ← 0
    for i ← 1 to n+1 do
        xs[i] ← a + kh                              ▷ obliczanie kolejnych wezłów
        fs[i] ← f(xs[i])                 ▷ obliczanie wartości w kolejnych wezłach
        kh ← kh + h
    fx ← ilorazyRóżnicowe(xs, fs)
    density ← 100
    pointsNo ← ⌊(b − a) ∗ density)⌋ + 1
    h ← (b−a)/(pointsNo−1)
    kh ← 0
    for i ← 1 to pointsNo do
        plotx[i] ← a + kh
        realVals[i] ← f(plotx[i])
        interVals[i] ← warNewton(x, fx, plotx[i])
        kh ← kh + h
    plot(plotx, realVals, interVals)
```

- f – zadana funkcja do zinterpolowania,
- a, b – granice przedziału interpolacji,
- n – stopień wielomianu interpolacyjnego

5. Zadanie 5

5.1 Opis problemu

W tym zadaniu należało przetestować metodę z zadania 4. dla następujących funkcji:

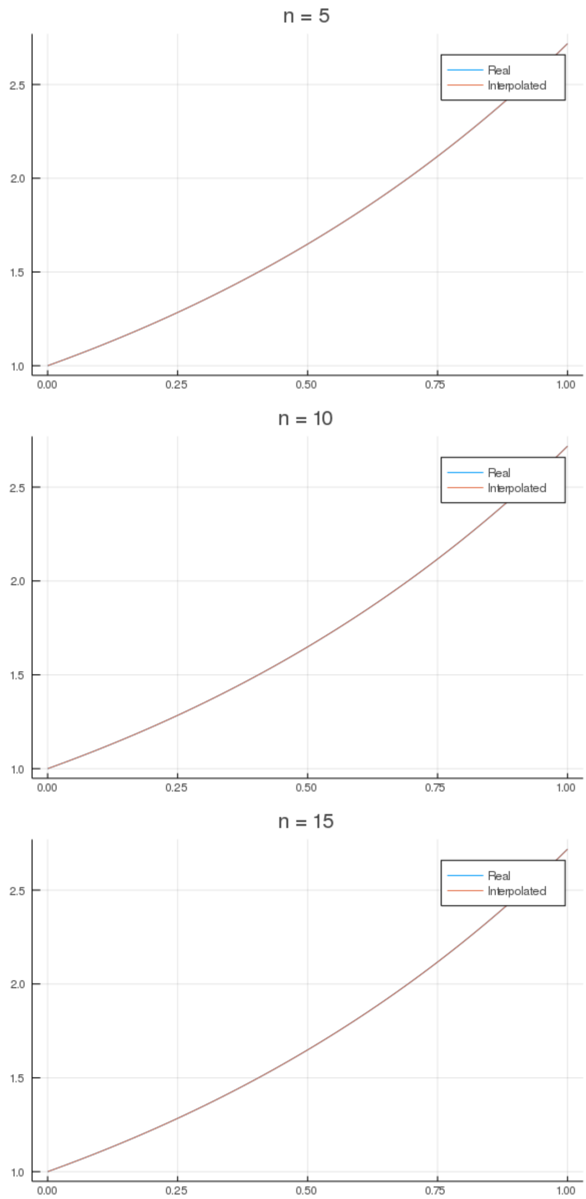
- e^x w przedziale $[0, 1]$, $n = 5, 10, 15$,
- $x^2 \sin x$ w przedziale $[-1, 1]$, $n = 5, 10, 15$.

5.2. Wyniki i wnioski

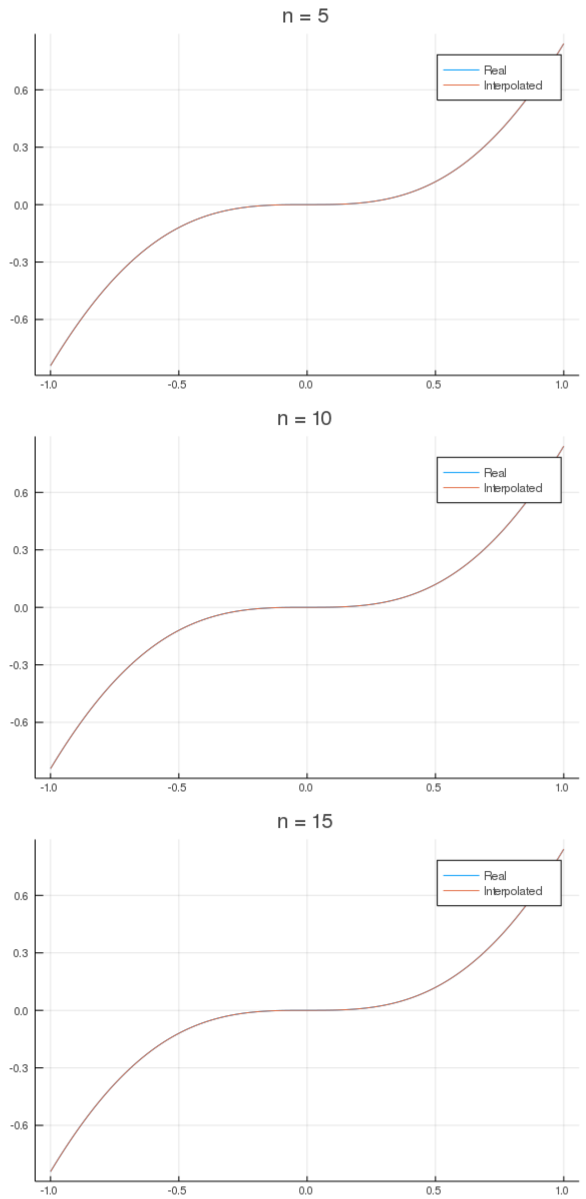
Na następnej stronie przedstawiono wykresy uzyskane dla naszych danych.

Jak widać wykresy funkcji rzeczywistej oraz interpolowanej praktycznie się pokrywają. Można więc wywnioskować, że dla zadanych funkcji, których pochodne są ograniczone przez stosunkowo niewielkie wartości, wielomian interpolacyjny daje bardzo dobre przybliżenie, nawet dla małych wartości n.

Wykresy dla funkcji e^x .



Wykresy dla funkcji $x^2 \sin x$.



5. Zadanie 6

6.1 Opis problemu

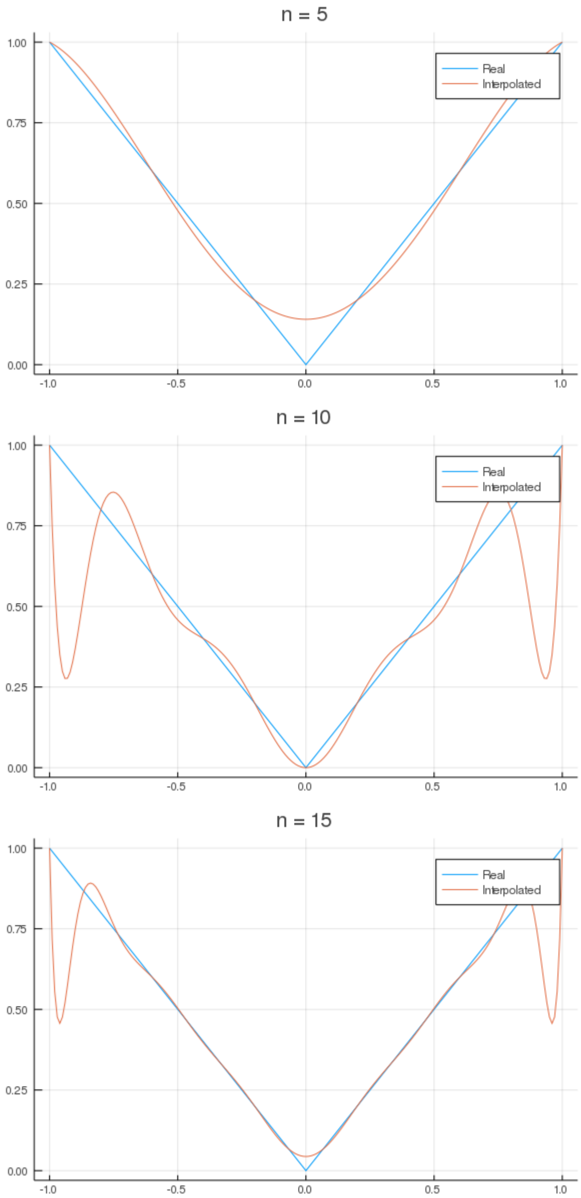
W tym zadaniu należało przetestować funkcję zaimplementowaną w zadaniu 4. na następujących przykładach i zbadać zjawisko rozbieżności:

- $|x|$ w przedziale $[-1, 1]$, $n = 5, 10, 15$,
- $\frac{1}{1+x^2}$ w przedziale $[-5, 5]$, $n = 5, 10, 15$ (zbadać zjawisko Runge'go)

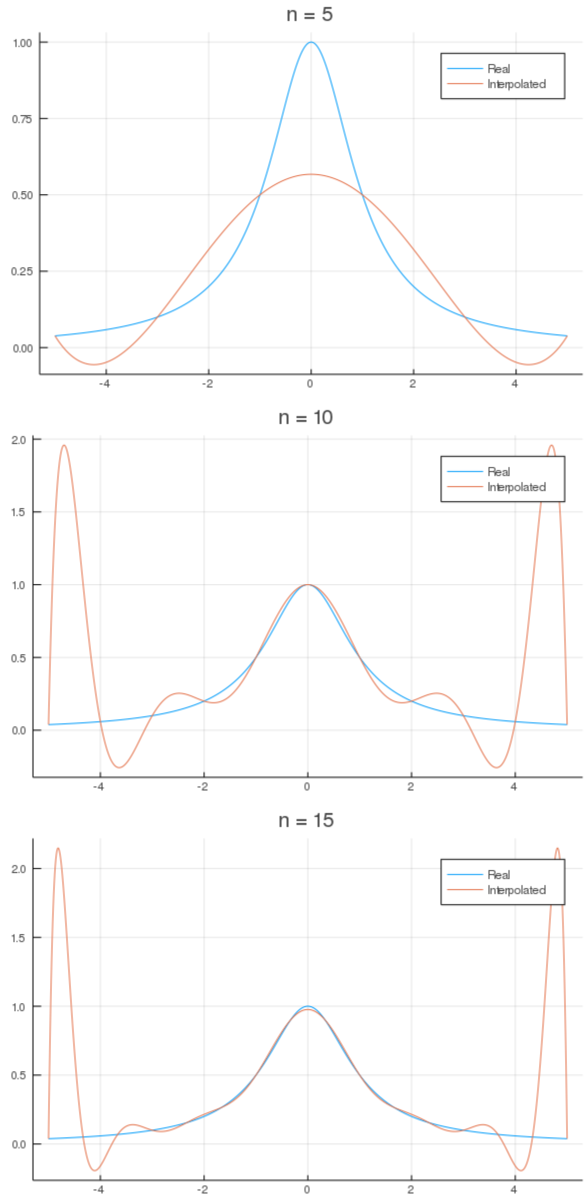
6.2. Wyniki i wnioski

Poniżej przedstawiono otrzymane wykresy:

Wykresy dla funkcji $|x|$.



Wykresy dla funkcji $\frac{1}{1+x^2}$.



Możemy zauważyć, że dla obu funkcji występują znaczne rozbieżności. Dla pierwszej funkcji wraz ze wzrostem wartości n wykresy co raz bardziej się pokrywają, natomiast dla drugiej zwiększają się rozbieżności przy końcach przedziału.

W przypadku pierwszej funkcji rozbieżność uzyskiwanych wartości wynika z faktu, że $|x|$ nie jest funkcją różniczkowalną.

W przypadku funkcji drugiej dochodzi do zjawiska Runge'go. Zachodzi ono wtedy, kiedy wraz ze wzrostem stopnia wielomianu interpolacyjnego zmniejsza się dokładność przybliżenia interpolowanej funkcji. Jego występowanie wynika z faktu, że w naszej funkcji używamy węzłów równoodległych, oraz wielomiany mają co raz wyższy stopień, co doprowadza do znacznych rozbieżności na końcach przedziałów. Może ono też występować, kiedy interpolowana funkcja jest nieciągła lub odbiega od funkcji gładkiej. Aby zapobiec temu zjawisku stosuje się interpolację, w której węzły są coraz gęściej rozmieszczone na krańcach przedziału interpolacji. Możemy przyjmować za węzły na przykład miejsca zerowe wielomianu Czybyszewa n-tego stopnia, których rozmieszczenie będzie gęstsze.