Sprawozdanie z listy czwartej

Karolina Bąk

Grudzień 2019

1 Zadanie pierwsze

Celem tego zadania było zaimplementowanie funkcji obliczającej ilorazy różnicowe dla podanych węzłów i odpowiadających im wartości, która przedstawi je w postaci wektora odpowiedniego dla uogólnionego algorytmu Hornera by uzyskać postać Newtona wielomianu interpolacyjnego w dalszej części listy.

1.1 Opis algorytmu:

Iloraz różnicowy k-tego rzędu można obliczyć z poniższej zależności rekurencyjnej:

$$\begin{aligned} f[x_0] &= f(x_0) \\ f[x_0, x_1] &= \frac{f(x_1) - f(x_1)}{x_1 - x_0} \\ f[x_0, x_1 \ldots, x_k] &= \frac{f[x_1, x_2 \ldots, x_k] - f[x_0, x_1 \ldots, x_{k-1}]}{x_k - x_0} \end{aligned}$$

Znając węzły x_k oraz wartości funkcji $f(x_k)$ można z pomocą wzoru utworzyć macierz ilorazów różnicowych wyższych rzędów. Zapisywanie całej macierzy jest nieopłacalne przez wysoką złożoność ($O(n^3)$), a do poprawnej reprezentacji wystarczy jednowymiarowa tablica fx. Początkowymi wartościami fx_j są odpowiadające im $f[fx_j]$, obliczane ze wzoru:

$$fx_j = \frac{fx_j - fx_{j-1}}{x_j - x_{j-i}},$$

gdzie i jest numerem kolumny w macierzy. Każda następna wartość jest aktualizowana kolumnami w kolejności z dołu do góry. Dzięki temu tablica fx w każdym momencie posiada wartości ilorazów potrzebne dla następnych kroków algorytmu.

1.2 Pseudokod:

Input: Wektor x długości n+1 węzłów oraz wektor f zawierający wartości funkcji interpolowanej w tych węzłach
Output: Wektor fx długości n+1 obliczonych ilorazów różnicowych

```
ln ← length(f) fx ← copy(f)
for i ← 2 to ln do
    for j ← ln to i do
        fx[j] = (fx[j] − fx[j − 1])/(x[j] − x[j − i + 1])
    end
end

return fx
```

2 Zadanie drugie

Celem tego zadania było napisanie funkcji obliczającej wartość wielomianu interpolacyjnego stopnia n w postaci Newtona $N_n(x)$ w punkcie $x = t$ za pomocą uogólnionego algorytmu Hornera, działającą w czasie $O(n)$.

2.1 Opis algorytmu:

Wielomian interpolacyjny Newtona jest zadany wzorem:

$$N_n(x) = \sum_{i=0}^{k} c_i \prod_{j=0}^{i-1} (x - x_j),$$

gdzie c_i jest ilorazem różnicowym stopnia i, a x_j węzłem interpolacji.
Aby obliczyć wartość tego wielomianu w danym punkcie można użyć uogólnionego algorytmu Hornera. Jego działanie można przedstawić następującymi zależnościami:

$$w_n(x) = f[x_0, x_1 \ldots, x_n]$$
$$w_k(x) = w_{k+1}(x - x_k) + f[x_0, x_1 \ldots, x_k]$$
$$N_n(x) = w_0(x)$$

Jeśli ilorazy są już obliczone, to zastosowanie powyższych wzorów pozwala znaleźć wartość wielomianu w czasie liniowym.

2.2 Pseudokod:

Input: Wektory x i fx zawierające odpowiednio węzły i ilorazy różnicowe długości n+1 oraz t jako punkt, w którym należy obliczyć wartość wielomianu
Output: Wartość wielomianu w punkcie nt

$ln \leftarrow length(x)$ $nt \leftarrow fx[ln]$
for $i \leftarrow ln - 1$ ***to*** 1 **do**
| $nt \leftarrow fx[i] + (t - x[i]) * nt$
end

return nt

3 Zadanie trzecie

Celem zadania było napisać funkcję obliczającą w czasie $O(n^2)$ współczynniki postaci naturalnej dla podanych współczynników wielomianu interpolacyjnego Newtona oraz węzłów $x_0, \ldots, x_n$.

3.1 Opis algorytmu:

Postać naturalną wielomianu stopnia n zmiennej x można przedstawić wyrażeniem:

$$\sum_{i=0}^{n} a_i x^i$$

Aby znaleźć współczynniki wielomianu interpolacyjnego w postaci naturalnej należy jako punkt wyjścia obrać uogólniony algorytm Hornera z poprzedniego zadania. Współczynnik a_n w szukanym wielomianie n-tego stopnia (przy najwyższej potędze x) jest równy c_n, z czego wynika, że w_n z algorytmu Hornera jest także równe a_n.

Posiadając początkowe warunki dla algorytmu w następnych krokach tworzone są wartości a_i opierające się na współczynnikach a_{i+1}. Aby znaleźć zależności między następnymi a_i algorytm przechodzi po wszystkich w_i od i równego n do 0, zmieniając tworzone współczynniki, tak aby dla każdego w_i doprowadzić w danej chwili do postaci naturalnej.

3.2 Pseudokod:

Input: Wektory x i fx długości n+1 zawierające węzły $x_0 \dots x_n$ oraz ilorazy różnicowe
Output: Wektor a długości n+1 zawierający obliczone współczynniki postaci naturalnej

$ln \leftarrow length(x)$ $a \leftarrow zeros(ln)$ $a[ln] \leftarrow fx[ln]$
for $i \leftarrow ln - 1$ ***to*** 1 **do**
  $a[i] \leftarrow fx[i] - a[i+1] * x[i]$ **for** $j \leftarrow i + 1$ ***to*** $ln - 1$ **do**
    $a[j] \leftarrow a[j] - a[j+1] * x[i]$
  end
end

return a

4 Zadanie czwarte

Celem zadania było napisać funkcję, która interpoluje zadaną funkcję $f(x)$ w przedziale $[a, b]$ za pomocą wielomianu interpolacyjnego n-tego stopnia w postaci Newtona, a następnie narysuje interpolowany wielomian oraz interpolowaną funkcję.

4.1 Opis algorytmu:

Do reprezentacji graficznej, aby uzyskać dobre efekty w interpolacji użyłam węzłów równo od siebie oddalonych w podanym przedziale. Wyznaczam je w następujący sposób:

$$x_k = a + kh \quad k = 0, 1 \dots n$$
$$h = \frac{b-a}{n}$$

Wielomian interpolacyjny nie był jawnie obliczany, gdyż wystarczyło tylko użyć funkcji z pierwszych dwóch zadań. Najpierw iteracyjnie wyznaczyłam wektor węzłów i wartości funkcji im odpowiadające. Z nich uzyskałam przy użyciu funkcji *ilorazyRoznicowe* wektor ilorazów pod algorytm Hornera. Następnie obliczyłam wartości wielomianu dla punktów na wykresie, korzystając z funkcji *warNewton*.

Ostatnim krokiem algorytmu było narysowanie wykresów funkcji oraz interpolowanego wielomianu. Do tej części użyłam pakietu PyPlot (pakiet z języka Python importowany do Julii). Liczba punktów narysowanych na wykresie jest dwadzieścia razy większa niż ilość węzłów użytych do wyznaczenia wielomianu w celach estetycznych. Nie ma to wpływu na dokładność interpolacji, gdyż liczba węzłów i ilorazów różnicowych nie jest zwiększana.

4.2 Pseudokod:

Input: Funkcja f, końce przedziału a, b na którym będzie interpolowana funkcja, ilość węzłów n
Output: Wykres funkcji oraz wielomianu na danym przedziale

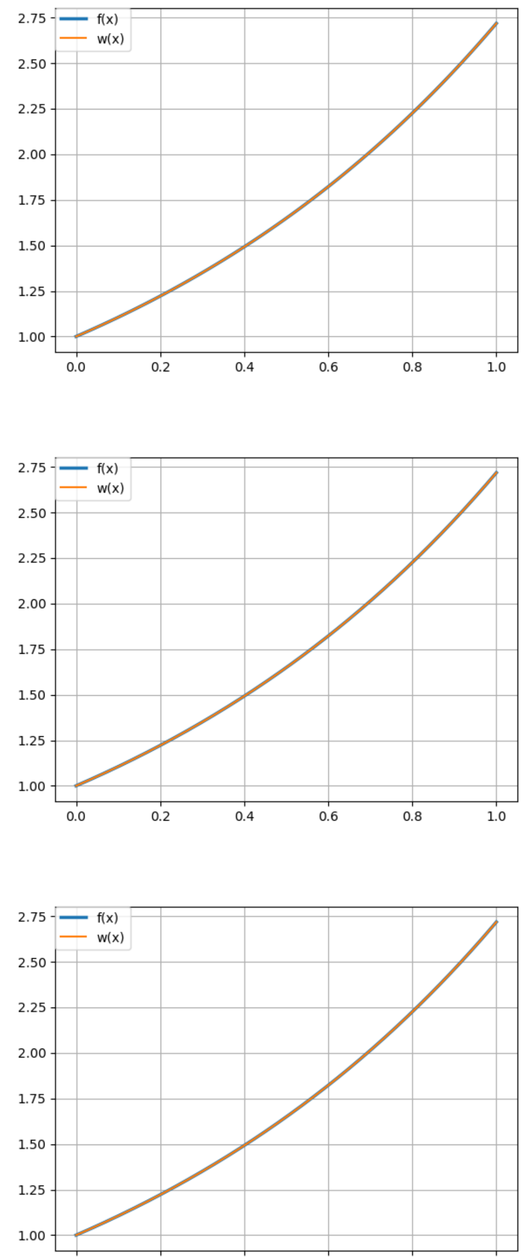
$maxnodes \leftarrow n + 1$
$x[maxnodes]$ $y[maxnodes]$ $fx[maxnodes]$
$plotargs[20 * maxnodes]$ $plotval[20 * maxnodes]$ $plotip[20 * maxnodes]$
$kh \leftarrow 0$ $h \leftarrow (b - a)/n$
for $i \leftarrow 1$ ***to*** $maxnodes$ **do**
  $x[i] \leftarrow a + kh$
  $y[i] \leftarrow f(x[i])$
  $kh \leftarrow kh + h$
end
$fx \leftarrow ilorazyRoznicowe(x, y)$
$kh \leftarrow 0$ $maxnodes \leftarrow maxnodes * 20$ $h \leftarrow (b - a)/(maxnodes - 1)$
for $i \leftarrow 1$ ***to*** $maxnodes$ **do**
  $plotargs[i] \leftarrow a + kh$
  $plotip[i] \leftarrow warNewton(x, fx, plotargs[i])$
  $plotval[i] \leftarrow f(plotargs[i])$
  $kh \leftarrow kh + h$
end
$clf()$
$plot(plotargs, plotval, "f(x)")$
$plot(plotargs, plotip, "w(x)")$
$savefig("plot.png")$

5 Zadanie piąte

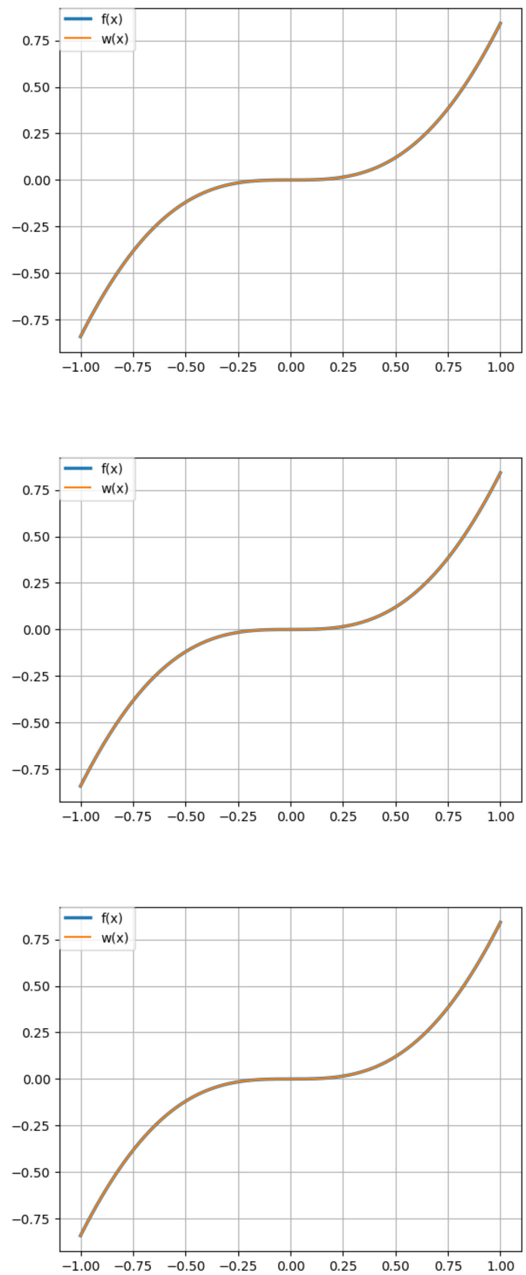
Celem zadania było przetestowanie funkcji z poprzedniego zadania $rysujNnfx(f, a, b, n)$ na następujących przykładach:

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

Poniżej przedstawiam uzyskane wyniki. Jak można zaobserwować, już dla 5 węzłów interpolowany wielomian pokrywa się z funkcją.



Rysunek 1: Wykresy funkcji e^x oraz wielomianu interpolującego na przedziale $[0, 1]$ dla $n = 5, 10, 15$



Rysunek 2: Wykresy funkcji $x^2 \sin x$ oraz wielomianu interpolującego na przedziale $[-1, 1]$ dla $n = 5, 10, 15$

6 Zadanie szóste

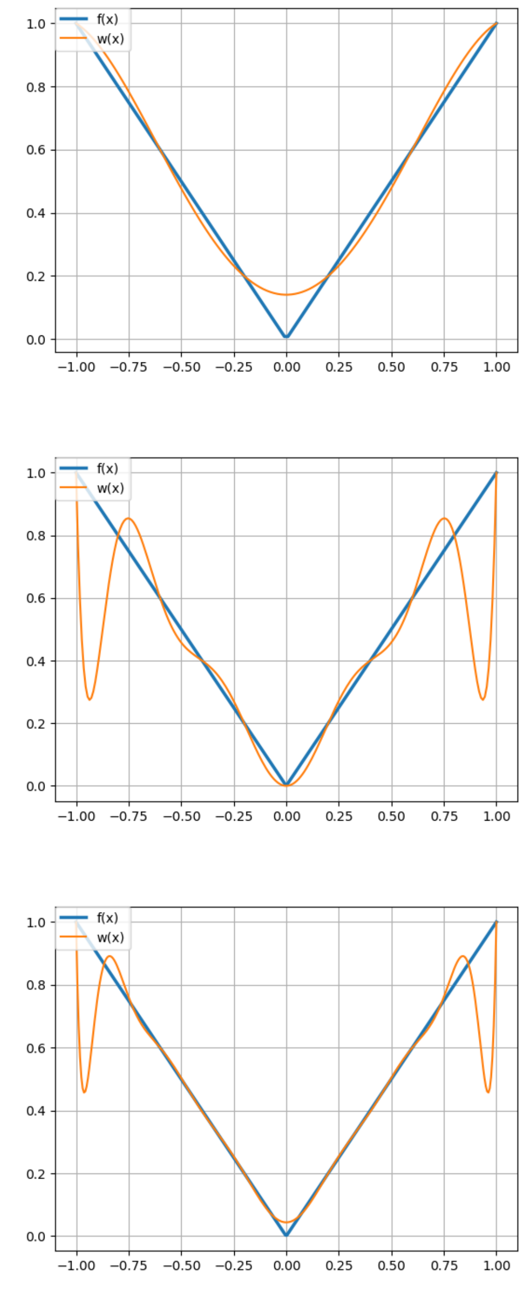
Celem zadania było przetestowanie funkcji $rysujNnfx(f, a, b, n)$ na ciekawszych przykładach:

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

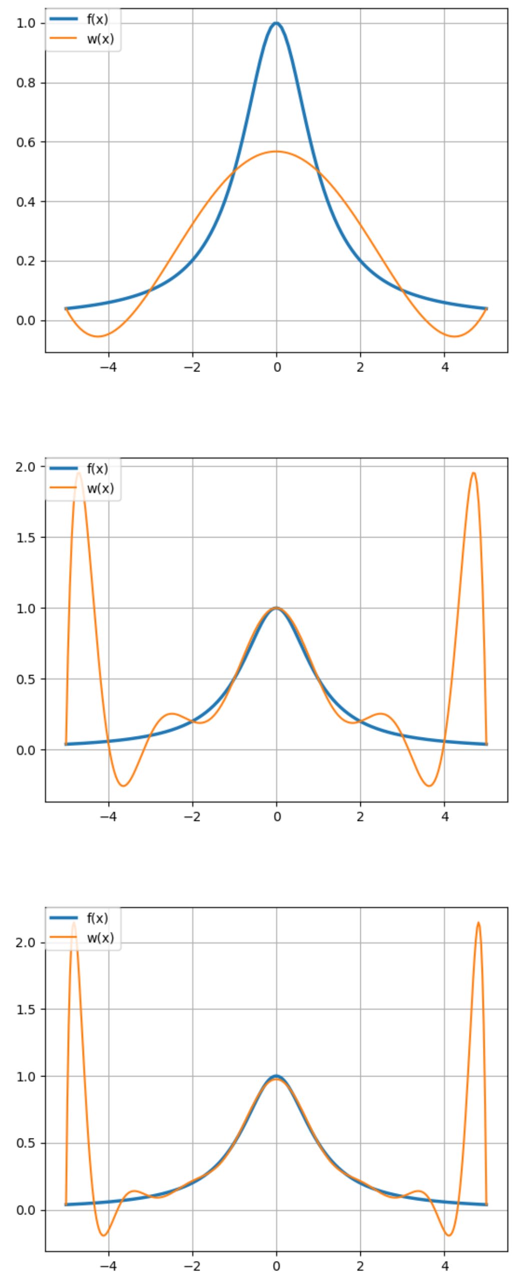
Otrzymane wykresy znajdują się na kolejnych stronach. Można na nich zauważyć, że interpolacja nawet dobrze zaimplementowana jest podatna na błędy. Widoczne jest to zwłaszcza przy funkcjach, które ciężko przedstawić w formie wielomianu - na przykład $|x|$.

Aby dokładność wzrosła można zwiększyć ilość węzłów, jednak nie zawsze uzyskuje się lepszą dokładność - $|x|$ dla $n = 10$. Na ogół jednak, jeśli zwiększymy stopień wielomianu to dostaniemy lepszą dokładność w środku przedziału. Niestety na krańcach przedziału pojawiają się wtedy duże wahania - widoczne szczególnie w $\frac{1}{1+x^2}$ dla $n = 10, 15$. Te zaburzenia są spowodowane charakterystyką badanych funkcji i nie można ich uniknąć przy węzłach równoodległych.

Z przeprowadzonych testów można wywnioskować, że interpolacja jest dobrym narzędziem do przybliżania funkcji. Jednak trzeba uważać przy korzystaniu z niej, gdyż są funkcje, które wymagają odpowiedniego rozstawienia węzłów do ich poprawnego przedstawienia, a jeśli nie wiemy jakiej funkcji się spodziewać możemy wysunąć błędne wnioski lub uzyskać błędne wyniki.



Rysunek 3: Wykresy funkcji $|x|$ oraz wielomianu interpolującego na przedziale $[-1, 1]$ dla $n = 5, 10, 15$



Rysunek 4: Wykresy funkcji $\frac{1}{1+x^2}$ oraz wielomianu interpolującego na przedziale $[-5, 5]$ dla $n = 5, 10, 15$