Obliczenia Naukowe - Lista nr 2

Jakub Jaśków

December 1, 2023

1 zad

1.1 Opis

Napisz funkcję obliczającą ilorazy różnicowe na podstawie podanych węzłów i odpowiadających im wartości funkcji bez użycia matrycy (tabeli 2D).

Wzór rekurencyjny dla ilorazu różnicowego k-tego rzędu:
1. $k = 0$

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

2. $k = 1$

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

3. $k > 1$

$$f[x_i, x_{i+1}, \ldots, x_{i+k}] = \frac{f[x_{i+1}, x_{i+2}, \ldots, x_{i+k}] - f[x_i, x_{i+1}, \ldots, x_{i+k-1}]}{x_k - x_i}$$

Dzięki powyższemu wzorowi możemy wywnioskować, że iloraz różnicowy nie zależy od kolejności węzłów.

Rozwiązanie

Dzięki znajomości węzłów x_i, wartości funkcji $f(x_i)$ oraz powyższego wzoru, jesteśmy w stanie stworzyć tablicę ilorazów różnicowych wyższych rzędów. Przyjmijmy $c_{ik} = f[x_i, x_{i+1}, \ldots, x_{i+k}]$. Postać tablicy:

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

Algorytm funkcji którą mamy stworzyć mógł by wykorzystać powyższą tablicę (gdyby nie polecenie). Nie jest to jednak rozwiązanie optymalne. Wystarczy użyć jednowymiarowej tablicy fx. Początkowe wartości zmiennych bierzemy z tabeli powyżej - $c_{i,0} = f(x_i)$, a następnie zamieniamy je na $c_{i-1,1}, \ldots, c_{1,i-1}, c_{0,i}$ dla każdej kolejnej zmiennej fx_i.

Zmienne tablicy fx za każdym przejściem tworzą kolejne kolumny tablicy ilorazów różnicowych, które aktualizowane są od dołu do góry. Taka kolejność wykonywania poleceń zapewnia, że tablica fx zawierać będzie ilorazy potrzebne w następnych iteracjach.

```
function ilorazyRoznicowe(x,f)
    for i ← 1 to length(f) do
        fx[i] ← f[i]
    end
    for i ← 1 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
```

Algorithm 1: Obliczanie ilorazów różnicowych

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.

2 zad

Opis

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

Aby lepiej zrozumieć zależność wielomianu interpolacyjnego N_n od funkcji f możemy przedstawić go za pomocą ilorazów różnicowych:

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

Zaletą powyższego wzoru jest fakt, że dodanie kolejnych (x_i, y_i) nie narusza obliczonych wcześniej współczynników $c_k = f[x_0, x_1, \ldots, x_k]$. Wartość tak przedstawionego wielomianu można łatwo obliczyć posługując się uogólnionym algorytmem Hornera.

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

Rozwiązanie

function `warNewton`(x, fx, t)
- $n \leftarrow$ `length`(fx)
- $nt \leftarrow fx[n]$
- **for** $i \leftarrow n-1$ **downto** 1 **do**
  - $nt \leftarrow fx[i] + (t - x[i]) \times nt$
- **end**
- **return** nt

Algorithm 2: Obliczanie wartości wielomianu interpolacyjnego w punkcie t

Dane:

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

Wyniki:

`nt`	–	wartość wielomianu w punkcie t

3 zad

Opis

Znając współczynniki wielomianu interpolacyjnego w postaci Newtona $c_0 = f[x_0], c_1 = f[x_0, x_1], \dots, c_n = f[x_0, \dots, x_n]$ (ilorazy różnicowe) oraz węzły $x_0, x_1 \dots, x_n$ napisać funkcję obliczającą w czasie $O(n^2)$ współczynniki jego postaci naturalnej $a_0, \dots, a_n$.

Rozwiązanie

Aby znaleźć współczynniki wielomianu interpolacyjnego posłużymy się funkcją napisaną w poprzednim zadaniu, opartą o uogólniony algorytm Hornera.

Współczynnik a_n przy najwyższej potędze x jest równy c_n. Zatem w_n z algorytmu Hornera także jest równy a_n. Wiedząc, że $a_n = w_n$ tworzymy wartości $a_i = a_{i+1}$. W celu znalezienia zależności pomiędzy następnymi a_i przechodzimy po każdym w_i ($i \in [n, \dots, 0]$) zmieniając poszczególne a_i tak, że dla każdego w_i w pewnym momencie dochodzimy do postaci naturalnej.

```
function naturalna(x, fx)
    n ← length(fx)
    a[n] ← fx[n]
    for i ← n − 1 downto 0 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
```

Algorithm 3: Współczynniki naturalne wielomianu interpolacyjnego.

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

4 zad

Opis

Napisać funkcję, która zinterpoluje zadaną funkcję $f(x)$ w przedziale $[a.b]$ za pomocą wielomianu interpolacyjnego stopnia n w postaci Newtona. Następnie narysuje wielomian interpolacyjny i interpolowaną funkcję. Do rysowania zainstaluj np. pakiet **Plots**, **PyPlot** lub **Gadfly**.

4.1 Rozwiązanie

Wydzielamy n+1 węzłów o równych odstępach pomiędzy sobą a następnie wyliczamy ich wartości funkcji. Przy użyciu węzłów, ich wartości oraz funkcji z zadania 1 wyznaczamy ilorazy różnicowe. Dzięki wykonanym wcześniej krokom i funkcji z zadania 3 możemy wyznaczyć wartości wielomianu w punkcie. Dzielimy przedział [a,b] na ciąg punktów o równych odległościach pomiędzy sobą. Dla każdego wyznaczonego punktu obliczamy wartość funkcji i wielomianu, a następnie umieszczamy na wykresie i generujemy.

Funkcja `rysujNnfx`:

Dane:

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

Wyniki:

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

5 zad

Opis

Przetestowanie funkcji `rysujNnfx(f,a,b,n)` na przykładach:
$f(x) = e^x$, $[a, b] = [0, 1]$, $n \in \{5, 10, 15\}$,
$f(x) = x^2 \sin x$, $[a, b] = [-1, 1]$, $n \in \{5, 10, 15\}$

Rozwiązanie

Wywołanie funkcji `rysujNnfx(f,a,b,n)` dla danych podanych powyżej.

Wyniki

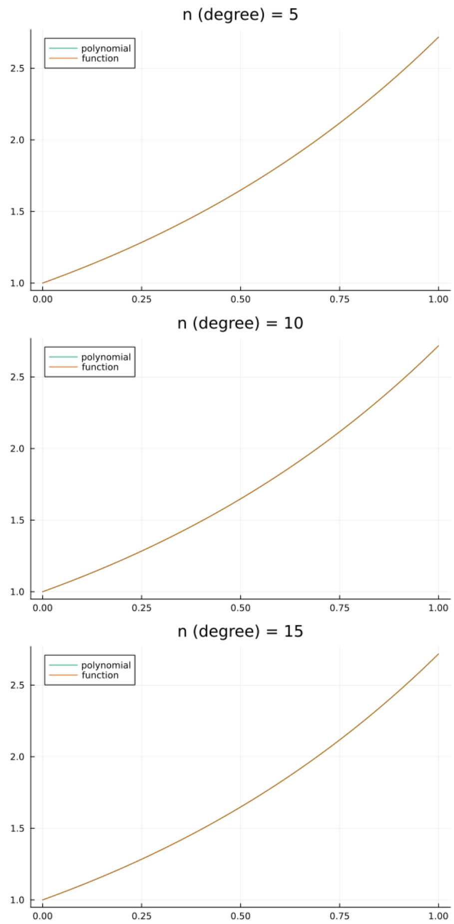

Figure 1: Wykres e^x i jej wielomianu interpolacyjnego dla danego stopnia n

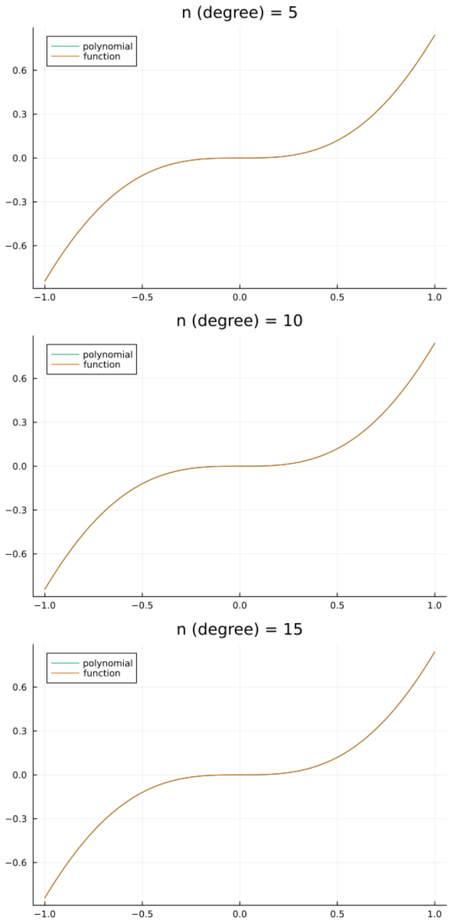

Figure 2: Wykres $x^2 \sin x$ i jej wielomianu interpolacyjnego dla danego stopnia n

Wnioski

Obie funkcje da się bardzo łatwo i dokładnie zinterpolować co potwierdzają nakładające się na siebie wykresy wielomianu oraz funkcji dla każdego stopnia interpolacji wielomianu.

6 zad

Opis problemu

Przetestowanie funkcji `rysujNnfx(f,a,b,n)` (zadanie 4) na następujących przykładach:
$f(x) = |x|$, $[a, b] = [-1, 1]$, $n \in \{5, 10, 15\}$,
$f(x) = \frac{1}{1+x^2}$, $[a, b] = [-5, 5]$, $n \in \{5, 10, 15\}$

Rozwiązanie

Wywołanie funkcji `rysujNnfx(f,a,b,n)` dla danych podanych powyżej.

Wyniki

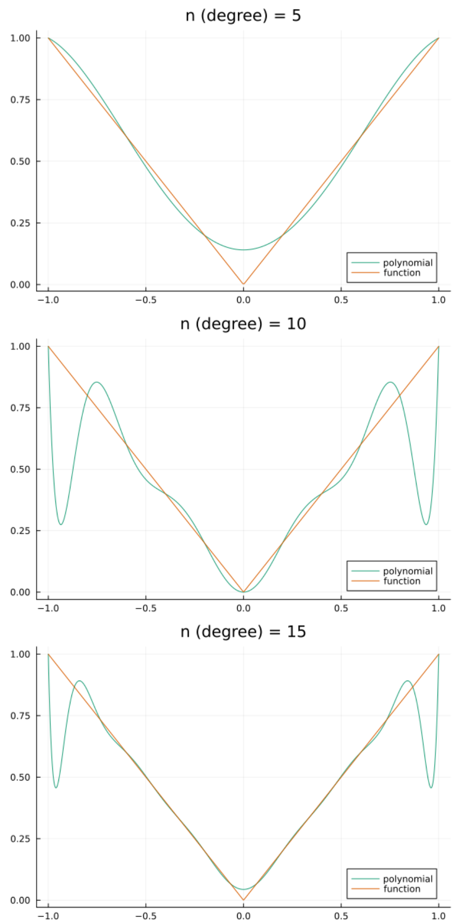

Figure 3: Wykres $|x|$ i jej wielomianu interpolacyjnego dla danego stopnia n

Jak możemy zaobserwować na wykresie w przeciwieństwie do poprzedniego zadania wykres funkcji oraz wielomianu stanowczo się różnią. Wzrost stopnia wielomianu nie poprawia dokładności interpolacji.
W sytuacji funkcji $|x|$ pojawia się kwestia nieróżniczkowalności. Można intuicyjnie stwierdzić, że wielomiany mają tendencję do bycia raczej okrągłymi, co sprawia, że znajdowanie wierzchołka wykresu staje się dla nich wyzwaniem.
Funkcja $f(x) = \frac{1}{1+x^2}$, $[a, b] = [-5, 5]$, $n \in \{5, 10, 15\}$ prezentuje nam kolejne wyzwanie w postaci zjawiska Rungego - jest to zwiększanie się rozbieżności na końcach przedziału wraz ze zwiększaniem stopnia wielomianu. Można je zaobserwować w przypadku, gdy węzły interpolacji ustawione są równolegle, tak jak ma to miejsce w naszym przypadku. Jedną z metod zapobiegania takiemu obrotowi spraw jest zagęszczenie węzłów na krańcach przedziału $[a, b]$.

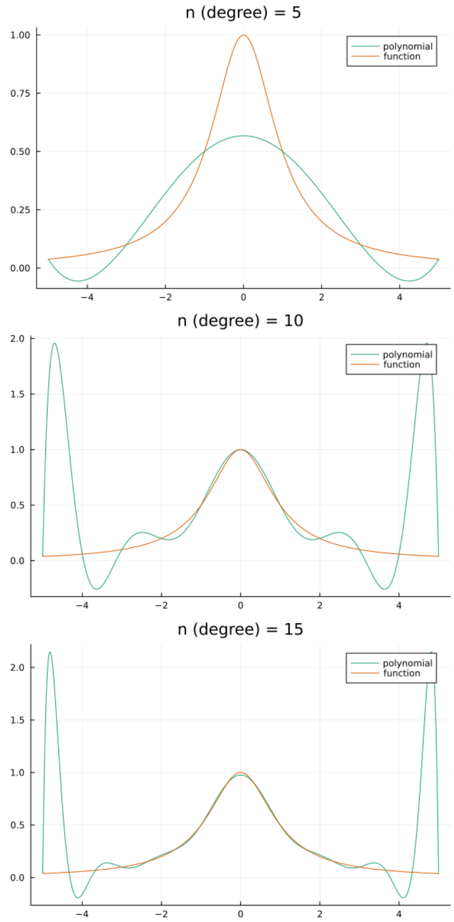

Figure 4: Wykres $\frac{1}{1+x^2}$ i jej wielomianu interpolacyjnego dla danego stopnia n

6.1 Wnioski

Kiedy znane nam są wartości funkcji tylko w niektórych punktach interpolacja wielomianowa zdaje się być dobrą metodą przybliżania funkcji. Bardzo dobrze radzi sobie z gładkimi, zaokrąglonymi funkcjami (zadanie 5). Niewłaściwe jest jednak ślepe przyjmowanie zwróconych przez interpolację wyników, ponieważ możemy się na przykład natknąć na funkcję $f(x) = |x|$ z zadanie 6, która na pierwszy rzut oka wydaje się być okej. Nie powinniśmy też na ślepo zwiększać stopnia wielomianu, ponieważ może to doprowadzić do wygenerowania jeszcze większych odstępstw od funkcji.
Narysowanie wykresu może dać nam wskazówki dotyczące tego jak prawidłowo powinniśmy rozmieścić węzły interpolacji.