Obliczenia Naukowe - Lista 4

Paweł Dychus (244941)

Grudzień 2019

Zadanie 1

Opis problemu

Napisać funkcję obliczającą ilorazy różnicowe na podstawie podanych węzłów i odpowiadających im wartości funkcji. Rozwiązać efektywnie bez użycia macierzy.

Iloraz różnicowy k-tego rzędu obliczymy stosując następujący wzór:

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

Rozwiązanie

Używając powyższego wzoru rekurencyjnego możemy stworzyć tablicę ilorazów różnicowych wyższych rzędów. Przyjmiemy, że $b_{ik} = f[x_i, x_{i+1}, \ldots, x_{i+k}]$, otrzymamy:

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

Dla tak określonego wzoru rekurencyjnego widać, że można zastosować tablicę dwuwymiarową, ale taki sposób nas nie satysfakcjonuje, ponieważ istnieje efektywniejszy algorytm. Weźmy tablicę jednowymiarową fx. Inicjalizujemy ją wartościami $fx[i] = b_{i,0}$. W następnej iteracji wyznaczmy: $fx[i] = b_{i,1}, i > 0$ oraz $fx[0] = b_{0,0}$, tzn. zauważamy, że dla kolejnych iteracji, ilość elementów w kolejnych kolumnach maleje, a indeksy $k < i$ możemy wykorzystać do trzymania wyznaczonych ilorazów różnicowych kolejnego stopnia. Po iteracji po wszystkich kolumnach otrzymamy: $fx[0] = b_{0,0}, fx[1] = b_{0,1}, \ldots, fx[k] = b_{0,k}$. Algorytm jest przedstawiony w poniższym pseudokodzie.

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

Algorithm 1 Obliczanie ilorazów różnicowych

```
function ILORAZYRÓŻNICOWE(x, f)
    for i ← 1 to length(f) do
        fx[i] ← f[i]                                   ▷ f[x_0], f[x_1], ..., f[x_n]
    for i ← 1 to length(f) do
        for j ← length(f) down to i do                 ▷ Na koniec pętli j: fx[i] = b_{0,i}
            fx[j] ← (fx[j] − fx[j−1]) / (x[j] − x[j−1])
return fx
```

Zadanie 2

Opis problemu

Napisać funkcję obliczającą wartość wielomianu interpolacyjnego stopnia n w postaci Newtona $N_n(x)$ w punkcie $x = t$ z użyciem uogólnionego algorytmu Hornera działającego w czasie $(O(n))$.

Wzór wielomianu interpolacyjnego Newtona z użyciem ilorazów różnicowych, przedstawiamy poniższym wzorem:

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

Cechą charakterystyczną tego przedstawienia jest, to że dodanie kolejnej pary punktów (x_i, y_i) nie narusza obliczonych wcześniej współczynników wielomianu $c_k = f[x_0, x_1, \ldots, x_k]$

Rozwiązanie

W celu rozwiązania zadania przedstawimy wzór z użyciem uogólnionego algorytmu Hornera. Dokładniej wyrzucimy kolejne czynniki $x - x_j$ przed nawias tak, że otrzymamy:

$$f[x_0] + (x - x_0)(f[x_0, x_1] + (x - x_1)(\ldots(x - x_{n-2}(f[x_0, x_1, \ldots, x_{n-1}] + (x - x_{n-1})f[x_0, x_1, \ldots, x_n])\ldots)$$

Po czym zwiniemy podstawiając:

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

Funkcja obliczającą wartość wielomianu interpolacyjnego Newtona $N_n(x)$ stopnia n w punkcie $x = t$ wykorzystującą zadany wyżej algorytm (2) przedstawia poniższy pseudokod.

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

Algorithm 2 Obliczanie wartości wielomianu interpolacyjnego w punkcie t

```
function WARNEWTON(x, fx, t)
    n ← length(x)
    nt ← fx[n]                                  ▷ w_n(x) := f[x_0, x_1, ..., x_n]
    for i ← n − 1 down to 1 do
        nt ← fx[i] + (t − x[i]) · nt            ▷ w_k(x) := f[x_0, x_1, ..., x_k] + (x − x_k)w_{k+1}
return nt                                       ▷ N_n(x) = w_0(x)
```

Zadanie 3

Opis problemu

Napisać funkcję obliczającą współczynniki $a_0, \ldots, a_n$ w postaci naturalnej wielomianu interpolacyjnego dla określonych współczynników $c_0 = f[x_0], c_1 = f[x_0, x_1], \ldots c_n = f[x_0, \ldots, x_n]$ (ilorazy różnicowe) tego wielomianu w postaci Newtona oraz węzłów $x_0, \ldots, x_n$. Algorytm ma się wykonywać w czasie: $O(n^2)$.

Rozwiązanie

W wielomianie interpolacyjnym, dla którego ilorazy różnicowe to $c_0, c_1, \ldots, c_n$, współczynnik a_n stojący przy najwyższej potędze x^n wynosi c_n, co więcej wynosi w_n z uogólnionego algorytmu Hornera. W następnych iteracjach (po $a_n = w_n$), algorytm tworzy wartości a_i bazujące na współczynnikach a_{i+1}. Żeby określić zależności pomiędzy kolejnymi a_i, algorytm przechodzi po w_i od i równego n do 1, tak aby były one w danym momencie, dla każdego w_i w postaci naturalnej.

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

Algorithm 3 Wyznaczanie współczynników naturalnych

```
function NATURALNA(x, fx)
    n ← length(x)
    a[n] ← fx[n]                                   ▷ Z twierdzenia, a_n = c_n
    for i ← n − 1 down to 1 do
        a[i] ← fx[i] − a[i + 1] · x[i]
        for j ← i + 1 to n − 1 do                  ▷ Wyznaczanie częściowej postaci naturalnej
            a[j] ← a[j + 1] · x[i]
return a
```

Zadanie 4

Opis problemu

Napisać funkcję interpolująca zadaną funkcję $f(x)$ w przedziale $[a, b]$ z wykorzystaniem wielomianu interpolacyjnego stopnia n w postaci Newtona. Funkcja ma rysować wielomian interpolacyjny i interpolowaną funkcję. Należy wykorzystać algorytm `ilorazyRoznicowe` i `warNewton`, a w interpolacji używać węzłów równoodległych.

Rozwiązanie

Najpierw należało wyznaczyć węzły interpolacji $(x_1, \ldots, x_{n+1})$, które są odległe od siebie daną długością $\frac{b-a}{n}$. Dla tak określonych węzłów wyznaczymy również wartości tj. $(f(x_1), \ldots, f(x_{n+1}))$. Następnie z wykorzystaniem funkcji z zadania 1, `ilorazyRoznicowe` wygenerujemy ilorazy różnicowe, dla wcześniej stworzonych węzłów interpolacji i towarzyszącym im wartością. Dodatkowo dla zwiększenia dokładności rysunku zwiększymy ilość iteracji mnożąc ją dziesięciokrotnie, tj. $10 \cdot (n + 1)$. Teraz w kolejnej pętli będziemy używać `warNewton`, żeby uzyskać wartość interpolowanej funkcji, która zapiszemy do tablicy. Na koniec korzystając z plots.jl wygenerujemy wykres dla wyznaczonych tablic.

Funkcja `rysujNnfx`:
Dane:

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

Wyniki:

– funkcja generuje wykres wielomianu interpolacyjnego i interpolowanej funkcji w przedziale $[a, b]$

Zadanie 5

Opis problemu

Przetestować funkcję `rysujNnfx(f,a,b,n)` na zadanych przykładach:

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

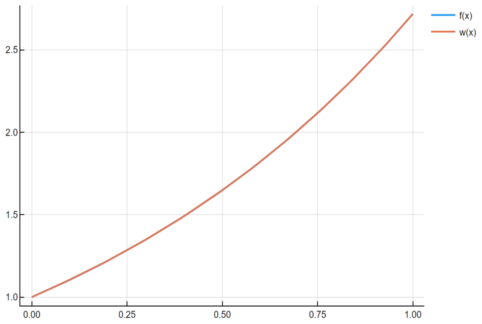
b) $f(x) = x^2 \sin x$, $[-1, 1]$, $n \in \{5, 10, 15\}$.

Rozwiązanie

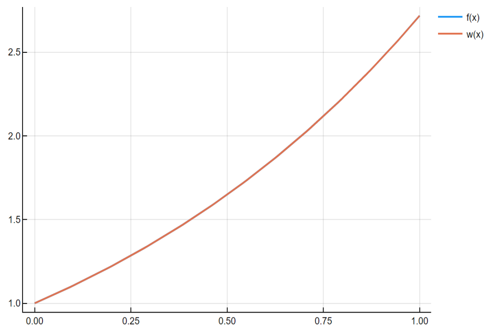
Rozwiązanie polega na wywołaniu funkcji `rysujNnfx` na zadanych przykładach.

Wyniki

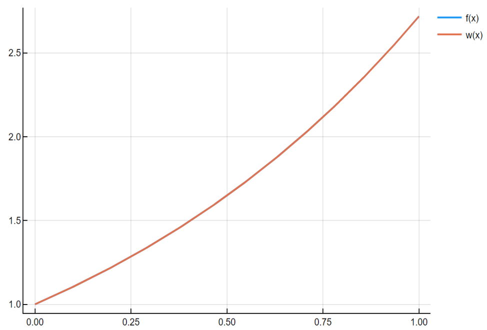
Wykresy otrzymane za pomocą metody `rysujNnfx(f,a,b,n)` prezentują wygenerowane obrazy: 1 i 2.



(a) $n = 5$

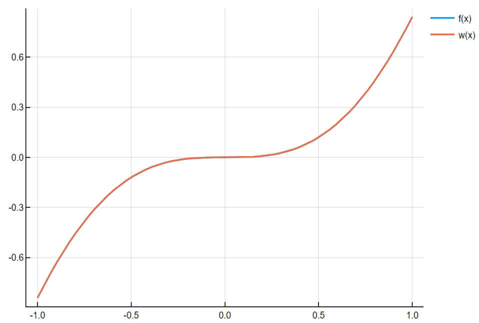


(b) $n = 10$

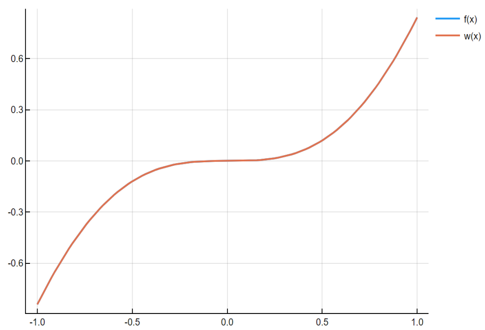


(c) $n = 15$

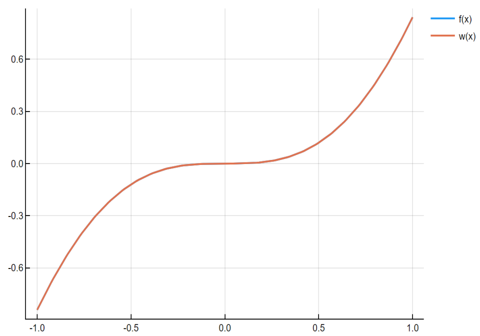
Rysunek 1 – Wykres e^x i jej wielomianu interpolacyjnego dla danego stopnia n



(a) $n = 5$



(b) $n = 10$



(c) $n = 15$

Rysunek 2 – Wykres $x^2 \sin x$ i jej wielomianu interpolacyjnego dla danego stopnia n

Otrzymaliśmy wykresy zbieżne, tj. dobrze przybliżone.

Wnioski

Dla tak określonych funkcji tj. dla e^x jak i $x^2 \sin x$ oraz ich parametrów, przy zastosowaniu równoodległych węzłów interpolacji otrzymujemy bardzo dobre przybliżenia interpolowanych funkcji. Powodem braku rozbieżności są niewielkie zmiany wartości interpolowanej funkcji na zadanym przedziale, co daje się łatwo interpolować.

Zadanie 6

Opis problemu

Przetestować funkcję `rysujNnfx(f,a,b,n)` na następujących przykładach:

a) $f(x) = |x|$, $[-1, 1]$, $n \in \{5, 10, 15\}$,

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

Rozwiązanie

Rozwiązanie polega na wywołaniu funkcji `rysujNnfx` na zadanych przykładach.

Wyniki

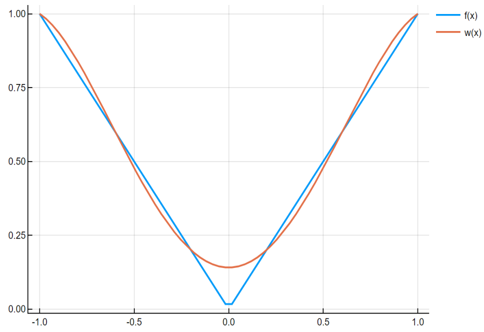
Wykresy otrzymane za pomocą metody `rysujNnfx(f,a,b,n)` prezentują wygenerowane obrazy: 3 i 4.

Otrzymaliśmy wykresy rozbieżne, tj. niedobrze przybliżone.

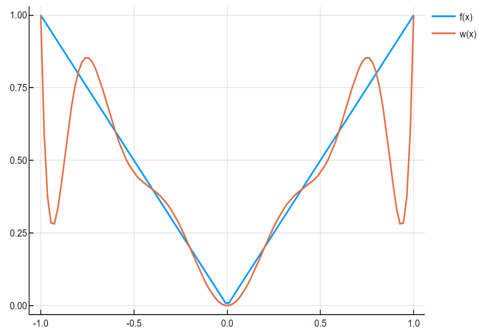
Wnioski

W przypadku $|x|$, mamy funkcję nieróżniczkowalną, dlatego tak ciężko jest ją interpolować wielomianami. Widoczne zwłaszcza dla większych n, skoki wartości w okolicy końca przedziałów, spowodowane są wysokim stopniem wielomianu i stałymi przedziałami(funkcja będzie zbiegała szybko do nieskończoności). Można by temu zaradzić, wystarczy podzielić funkcję na dwie funkcje różniczkowalne i ich interpolacje, co też nie jest idealne.

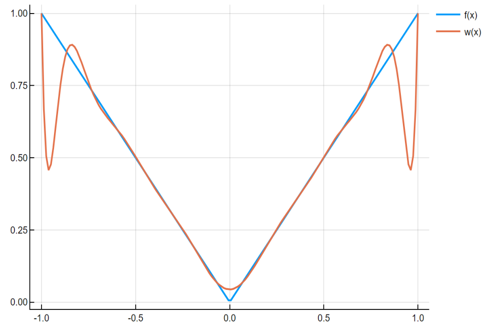
Dla przypadku $\frac{1}{1+x^2}$ mamy do czynienia ze zjawiskiem Runge'go. Wbrew logice, funkcja dla większej ilości węzłów interpolacji, zamiast zyskiwać na dokładności, staje się coraz bardziej rozbieżna, co jest widoczne zwłaszcza na końcach przedziału. Sam efekt skoków wartości dla większych n, na końcach przedziału jest spowodowany, podobnie jak w poprzednim przykładzie wysokim stopniem wielomianu interpolacyjnego, którego miejsca zerowe są ustawione równomiernie. Można by temu zaradzić, jeżeli zwiększylibyśmy ilość węzłów w okolicy miejsc problematycznych, czyli w okolicy końca przedziałów. Na pomoc przychodzą nam węzły Czebyszewa, które mniej oscylują ponieważ są zagęszczone na końcach przedziału.



(a) $n = 5$

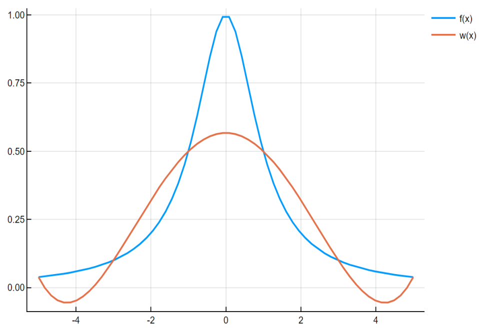


(b) $n = 10$

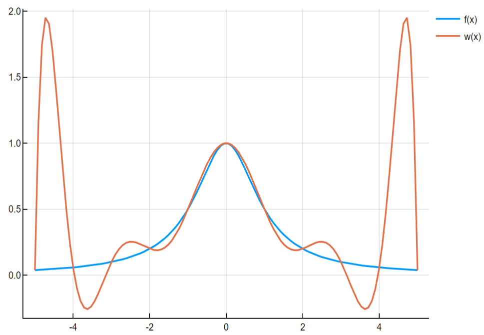


(c) $n = 15$

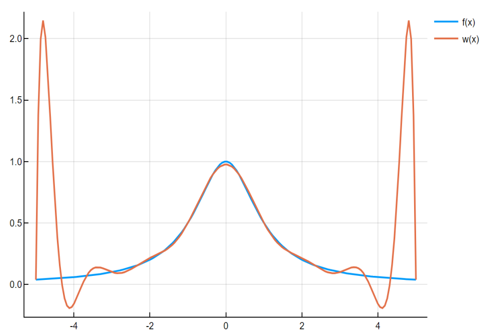
Rysunek 3 – Wykres $|x|$ i jej wielomianu interpolacyjnego dla danego stopnia n



(a) $n = 5$



(b) $n = 10$



(c) $n = 15$

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