

Zadanie 7 Lista 4

Piotr Popis, 245162

December 7, 2019

1 Zadanie 1

1.1 Opis problemu

Celem zadania jest implementacja funkcji obliczającej ilorazy różnicowe. Danymi wejściowymi są:

- x- wektor długości $n+1$ zawierający węzły x_0, \dots, x_n $x[1] = x_0, \dots, x[n+1] = x_n$
- f- wektor długości $n+1$ zawierający wartości interpolowanej funkcji w węzłach $f(x_0), \dots, f(x_n)$

Wyniki:

- fx- wektor długości $n+1$ zawierający obliczone ilorazy różnicowe
$$fx[1] = f[x_0],$$
$$fx[2] = f[x_0, x_1], \dots, fx[n] = f[x_0, \dots, x_{n-1}], fx[n+1] = f[x_0, \dots, x_n].$$

Addytywnym utrudnieniem jest restrykcja użycia tablicy dwuwymiarowej, czyli macierzy.

1.2 Rozwiązanie

Ilorazem różnicowym n - tego rzędu funkcji $f : X \longrightarrow Y$ w punktach $x_0, \dots, x_n \in X$ nazywamy funkcję:

$$f(x_0, \dots, x_n) := \frac{\sum_i^n f(x_i)}{\prod_j^n (x_i - x_j)}$$

W celu realizacji zadania, czyli uniknięcia wykorzystania macierzy skorzystamy z zależności rekurencyjnej:

1. $i = 0$

$$f[x_0] = f(x_0)$$

2. $i = 1$

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

3. $i = n$

$$f[x_0, \dots, x_n] = \frac{f(x_1, \dots, x_n) - f(x_0, \dots, x_{n-1})}{x_n - x_0}$$

Znając węzły x_n i wartości funkcji $f(x_n)$, można utworzyć dwuwymiarową tablicę ilorazów różnicowych. Jednak algorytm można zoptymalizować, ponieważ wystarczy użyć tablicy jednowymiarowej w do zapamiętywania dwóch poprzednich wartości (tablicę aktualizujemy od dołu do góry i od lewej do prawej). Pozostałe wartości tylko i wyłącznie spowalniają nasz algorytm. Początkowymi wartościami są w_i są odpowiadające im $f[x_i]$. W kolejnych krokach aktualizowane jest jedno miejsce mniej.

2 Zadanie 2

2.1 Opis problemu

Napisać funkcję obliczającą wartość wielomianu interpolacyjnego stopnia n w postaci Newtona $N_n(x)$ w punkcie $x=t$ za pomocą algorytmu Hornera w czasie $\theta(n)$. Dane wejściowe:

$$\begin{aligned}x &- \text{wektor długości } n+1 \text{ zawierający węzły } x_0, \dots, x_n \\x[1] &= x_0, \dots, x[n+1] = x_n \\fx &- \text{wektor długości } n+1 \text{ zawierający ilorazy różnicowe} \\fx[1] &= f[x_0], \\fx[2] &= f[x_0, x_1], \dots, fx[n] = f[x_0, \dots, x_{n-1}], fx[n+1] = f[x_0, \dots, x_n]\end{aligned}$$

Wyniki:

$$\begin{aligned}a &- \text{wektor długości } n+1 \text{ zawierający obliczone współczynniki postaci naturalnej} \\a[1] &= a_0, \\a[2] &= a_1, \dots, a[n] = a_{n-1}, a[n+1] = a_n.\end{aligned}$$

2.2 Rozwiązanie

W celu wyznaczenia wartości wielomianu interpolacyjnego stopnia n w postaci Newton'a $N_n(x)$ w punkcie $x = t$ zaimplementowano uogólniony schemat Hornera.

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

Wartość wielomianu przyjmowana w danym punkcie t obliczamy, kosztując z:

$$w(x) = (x - t) \cdot q(z) \cdot w(t)$$

Można też śmiało stwierdzić, iż metoda Newton'a jest trudniejsza w implementacji niż np metoda Lagrange'a i wymaga większej ilości oddzielnych kroków. Jest jednak znacznie lepiej uwarunkowana numerycznie. W metodzie najpierw wyznaczane są odpowiednie ilorazy różnicowe, a dopiero później z ich użyciem interpolowana funkcja. Algorytm zaczynamy od przypisania do zmiennej nt konkretnego wektora ilorazów różnicowych. W kolejnych krokach od $n-1$ zwiększamy wartość wektora pomnożoną przez różnicę wartości węzła i wielomianu t . Otrzymujemy:

$$nt = f_x(i) + nt \cdot (t - \omega[i]), \text{ gdzie } f_x \text{ jest wektorem ilorazów różnicowych.}$$

Algorytm działa w czasie liniowym.

3 Zadanie 3

3.1 Opis Problemu

Celem zadania jest kreacja funkcji obliczającej współczynniki postaci naturalnej wielomianu interpolacyjnego znając współczynniki wielomianu interpolacyjnego w postaci Newton'a $c_0 = f[x_0]$, $c_1 = f[x_0, x_1]$, $c_2 = f[x_0, x_1, x_2]$, ..., $c_n = f[x_0, \dots, x_n]$ (ilorazy różnicowe) oraz węzły x_0, x_1, \dots, x_n działającą w czasie $\theta(n^2)$. Dane wejściowe:

x- wektor długości $n+1$ zawierający węzły x_0, \dots, x_n
fx- wektor długości $n+1$ zawierający ilorazy różnicowe
 $fx[1] = f[x_0]$, $fx[2] = f[x_0, x_1]$, ..., $fx[n] = f[x_0, \dots, x_{n-1}]$, $fx[n+1] = f[x_0, \dots, x_n]$

Wyniki:

a- wektor długości $n+1$ zawierający obliczone współczynniki postaci naturalnej
 $a[1] = a_0$,
 $a[2] = a_1, \dots, a[n] = a_{n-1}, a[n+1] = a_n$

3.2 Rozwiązanie

Korzystając z faktu, iż współczynnik sąsiadujący z x^k to $f[x_0, x_1, \dots, x_k]$, czyli c_n przy najwyższej potęgze wielomianu w postaci Newton'a. Ponownie wykorzystujemy uogólnienie schematu Hornera, a następnie z wzoru na postać Newton'a podanego na wykładzie:

$$p(x) = \sum_{k=0}^n f[x_0, \dots, x_k] \prod_{j=0}^{k-1} (x - x_j)$$

Wielomian posiada współczynniki przy odpowiadających im potęgach a_0, a_1, \dots, a_k , gdzie współczynnik a_k leży przy największej potęgze. Dokonujemy mnożenia danego wielomianu począwszy od jego ostatnich potęg. W każdym kroku mnożymy nasz wielomian przez dwumian $(x - x_{n-1})$. Ostateczny wielomian wynikowy będzie zaprezentowany jako prosty wektor współczynników analogiczny do wektora wejściowego. W wyniku wykonanych mnożeń otrzymujemy postać wynikową wielomianu, którego współczynniki wynoszą kolejno a_k, a_{k-1}, \dots, a_0 . Aby wynik był postaci $a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ musimy dokonać również przestawienia. Całkowita złożoność obliczeniowa wynosi $\theta(n^2)$, ponieważ złożoność obliczeniowa schematu Hornera wynosi $\theta(n)$ oraz mnożenie wielomianów również $\theta(n)$

4 Zadanie 4

4.1 Opis Problemu

Celem zadania jest utworzenie funkcji, która zinterpoluje zadaną funkcję $f(x)$ w przedziale $[a,b]$ za pomocą wielomianu interpolacyjnego stopnia n w postaci Newton'a, a następnie narysuje wielomian interpolacyjny i funkcję interpolowaną. Wykorzystać Plots lub PyPlot lub Gadfly do narysowania grafów.

W interpolacji użyć węzłów równoodległych,

$$x_k = a + k \cdot h,$$

$h = \frac{b-a}{n}, k = 0, 1, \dots, n$ W zadaniu należy nie wyznaczać jawnej postaci wielomianu interpolacyjnego. Należy skorzystać z funkcji zaimplementowanych w zadaniu 1 oraz zadaniu 2 odpowiednio `iloryRzonicowe` i `warNewton`. Dane wejściowe:

f – funkcja $f(x)$ zadana jako anonimowa funkcja,
 a, b – przedział interpolacji,
 n – stopień wielomianu interpolacyjnego

Wyniki:

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

4.2 Rozwiązanie