Sprawozdanie z Laboratorium
Obliczenia Naukowe - Lista 4

Wojciech Typer

3 grudnia 2025

Zadanie 1

Cel zadania

Celem zadania jest napisanie funkcji obliczającej ilorazy różnicowe. Funkcja ma przyjmować dwa argumenty:

- x → wektor długości n + 1 zawierający węzły: x_0, ..., x_n
- f → wektor długości n + 1 zawierający wartości interpolowanej funkcji w węzłach $f(x_0$, ..., $f(x_n))$

Funkcja ma zwracać fx → wektor długości n + 1, zawierający obliczone ilorazy róznicowe:

$$fx = f[x_0], \dots, fx[n+1] = f[x_0, ..., x_n]$$

Idea metody

Obliczymy to na podstawie wzoru rekurencyjnego: iloraz rzędu k możemy policzyć na podstawie ilorazu rzędku k - 1:

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

Implementacja

Algorithm 1 Obliczanie ilorazów różnicowych (Newton)

Require: x: wektor węzłów $(x_0, \dots, x_n)$
Require: f: wektor wartości funkcji $(f(x_0), \dots, f(x_n))$
Ensure: fx: wektor ilorazów różnicowych

function IlorazyRoznicowe(x, f)
    $n \leftarrow \text{length}(f) - 1$
    $fx \leftarrow \text{copy}(f)$

    for $j \leftarrow 1$ **to** n **do**
        for $i \leftarrow n+1$ **downto** $j+1$ **do**
            $licznik \leftarrow fx[i] - fx[i-1]$
            $mianownik \leftarrow x[i] - x[i-j]$
            $fx[i] \leftarrow \frac{licznik}{mianownik}$
        end for
    end for
    return fx
end function

Zadanie 2

Cel zadania

Celem zadania jest napisanie funkcji obliczającej wartości wielomianu interpolacyjnego stopnia n w postaci Newtona za pomocą uogólnionego schematu Hornera w czasie O(n). Funkcja ma przyjmować trzy argumenty:

- x → wektor długości n + 1 zawierający węzły: x_0, ..., x_n
- fx → wektor długości n + 1, zawierający ilorazy różnicowe: $fx = f[x_0]$, ..., $fx[n+1] = f[x_0, ..., x_n]$
- t → punkt, w którym liczymy wartość wielomianu

Funkcja ma zwracać wartość wielomianu interpolacyjnego w punkcie t.

Idea metody

Wielomian Newtona możemy zapisać w postaci:

$$N_n(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \cdots + c_n(x - x_0)(x - x_1)\cdots(x - x_{n-1})$$

Gdybyśmy chcieli policzyć każdy składnik osobno, to złożoność obliczeniowa wyniosłaby O(n^2). Możemy jednak skorzystać z uogólnionego schematu Hornera, który pozwala na obliczenie wartości wielomianu w czasie O(n):

$$N_n(x) = c_0 + (t - x_0)\,(c_1 + (t - x_1)\,(c_2 + \cdots + (t - x_{n-1})c_n))$$

Słownie mówiąc, zaczynamy od ostatniego współczynnika c_n i iteracyjnie dodajemy kolejne składniki, mnożąc je przez odpowiednie czynniki $(t - x_i)$ i dodając c_i.

Implementacja

Algorithm 2 Obliczanie wartości wielomianu Newtona (Schemat Hornera)

Require: x: wektor węzłów $(x_0, \ldots, x_n)$
Require: fx: wektor ilorazów różnicowych $(f[x_0], \ldots, f[x_0, \ldots, x_n])$
Require: t: punkt, w którym liczymy wartość
Ensure: nt: wartość wielomianu w punkcie t

function WarNewton(x, fx, t)
  $n \leftarrow \mathrm{length}(fx) - 1$ ▷ Stopień wielomianu
  $nt \leftarrow fx[n+1]$ ▷ Inicjalizacja ostatnim współczynnikiem

  for $i \leftarrow n$ **downto** 1 **do** ▷ Iteracja od przedostatniego wyrazu
    $nt \leftarrow fx[i] + (t - x[i]) \cdot nt$
  end for
  return nt
end function

Zadanie 3

Cel zadania

Celem zadanie jest napisanie funkcji obliczającej w czasie $O(n^2)$ współczynniki wielomianu zapisanego w postaci naturalnej. Funkcja ma przyjmować dwa argumenty:

- x → wektor długości n + 1 zawierający węzły: x_0, ..., x_n
- fx → wektor długości n + 1, zawierający ilorazy różnicowe: $fx = f[x_0]$, ..., $fx[n+1] = f[x_0, ..., x_n]$

Funkcja ma zwracać wektor a długości n + 1, zawierający współczynniki wielomianu w postaci naturalnej:

$$W_n(x) = a_0 + a_1x + a_2x^2 + ... + a_nx^n$$

Idea metody

Wielomian w postaci Newtona zadany jest wzorem:

$$N_n(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + ... + c_n(x - x_0)(x - x_1)...(x - x_{n-1})$$

Aby uzyskać postać naturalną, musimy wymnożyć wszystkie czynniki $(x - x_i)$ i uporządkować wyrazy według potęg x. Możemy to zrobić iteracyjnie, korzystając ze schematu Hornera do mnożenia wielomianów. Algorytm polega na aktualizacji tablicy współczynników, "wciągając"kolejne czynniki $(x - x_k)$ do obliczonych już współczynników postaci naturalnej. Złożoność obliczeniowa wynosi $O(n^2)$ ze względu na zagnieżdżone pętle (dla każdego z n czynników wykonujemy do n operacji aktualizacji).

Implementacja

Algorithm 3 Konwersja postaci Newtona do postaci naturalnej

Require: x: wektor węzłów $(x_0, \ldots, x_n)$
Require: fx: wektor ilorazów różnicowych $(c_0, \ldots, c_n)$
Ensure: a: wektor współczynników postaci naturalnej $(a_0, \ldots, a_n)$

```
function NATURALNA(x, fx)
    n ← length(fx) − 1
    a ← copy(fx)

    for k ← n downto 1 do                          ▷ Iteracja po węzłach x_{n−1} do x_0
        for j ← k to n do                          ▷ Aktualizacja współczynników
            a[j] ← a[j] − a[j + 1] · x[k]
        end for
    end for
    return a
end function
```

Zadanie 4

Cel zadania

Celem zadania jest napisanie funkcji, która będzie interpolować zadaną funkcję f w przedziale [a, b] za pomocą wielomianu interpolacyjnego stopnia n w postaci Newtona. Następnie funkcja ma za zadanie rysować wykres wielomianu interpolacyjnego i interpolowanej funkcji. Funkcja ma przyjmować pięć argumentów:

- f → funkcja anonimowa
- a → początek przedziału
- b → koniec przedziału
- n → stopień wielomianu interpolacyjnego
- wezly → typ węzłów: 'rownoodlegle' lub 'czebyszew'

W interpolacji należy użyć węzłów równoodległych, czyli:

$$x_k = a + kh, \text{ gdzie } h = \frac{b-a}{n} \text{ o}$$

lub węzłów będących zerami n + 1 wielomianu Czebyszewa.

Idea metody