Sprawozdanie laboratorium lista 4
Obliczenia naukowe

Zofia Tarchalska, indeks: 279699

Wstęp do zadań 1-3

W pierwszych trzech zadaniach należy zaimplementować algortymy dotyczące interpolacji wielomianowej.
Interpolacja wielomianowa jest jednym z podstawowych narzędzi analizy numerycznej, pozwalającym na przybliżanie funkcji za pomocą wielomianów przechodzących przez zadane punkty. Szczególnie użyteczna jest postać Newtona, w której współczynniki wyznaczane są za pomocą ilorazów różnicowych. Ta reprezentacja jest wygodna obliczeniowo i umożliwia łatwe dodawanie nowych węzłów bez konieczności ponownego wyznaczania całego wielomianu.
W sprawozdaniu rozważymy trzy powiązane ze sobą algorytmy:

1. Obliczanie ilorazów różnicowych na podstawie wartości funkcji w węzłach interpolacji.
2. Wyznaczanie wartości wielomianu Newtona w punkcie t za pomocą uogólnionego algorytmu Hornera, działającego w czasie liniowym $O(n)$.
3. Przekształcanie wielomianu Newtona do postaci naturalnej $a_0+a_1x+\cdots+a_nx^n$, co wymaga czasu $O(n^2)$.

Zadanie 1

Algorytm obliczania ilorazów różnicowych

Wejście

- Wektor węzłów $x = [x_0, x_1, \ldots, x_n]$ długości $n + 1$.
- Wektor wartości funkcji $f = [f(x_0), f(x_1), \ldots, f(x_n)]$ długości $n + 1$.

Wyjście

- Wektor $fx = [f[x_0], f[x_0, x_1], \ldots, f[x_0, \ldots, x_n]]$ długości $n+1$, zawierający kolejne ilorazy różnicowe.

Idea działania

Algorytm oblicza współczynniki dla wielomianu interpolacyjnego Newtona. Zamiast tworzyć tablicę dwuwymiarową, wszystkie obliczenia wykonuje w jednym wektorze, nadpisując jego wartości w kolejnych krokach. Pierwszy element wektora wynikowego jest wartością funkcji w pierwszym węźle, a kolejne elementy powstają poprzez iteracyjne obliczanie ilorazów różnicowych wyższych rzędów.

Opis kroków

1. Utwórz wektor fx długości $n+1$ i skopiuj do niego wartości funkcji $f(x_i)$.
2. Ustal $fx[1] = f(x_0)$.
3. Dla każdego rzędu $j = 2, 3, \ldots, n+1$:
   - Przejdź po wektorze od końca do początku.
   - Zastąp element $fx[i]$ wartością:

   $$fx[i] \leftarrow \frac{fx[i] - fx[i-1]}{x[i] - x[i-j+1]}$$

   - Pierwszy element z danego rzędu odpowiada współczynnikowi Newtona i pozostaje w $fx[j]$.
4. Po zakończeniu pętli wektor fx zawiera wszystkie ilorazy różnicowe.

Złożoność

- Czasowa: $O(n^2)$, ponieważ algorytm wykonuje podwójną pętlę.
- Pamięciowa: $O(n)$, ponieważ używany jest tylko jeden wektor długości $n+1$.

Zadanie 2

Algorytm obliczania wartości wielomianu Newtona

Wejście

- Wektor węzłów $x = [x_0, x_1, \ldots, x_n]$ długości $n+1$.
- Wektor ilorazów różnicowych $fx = [f[x_0], f[x_0, x_1], \ldots, f[x_0, \ldots, x_n]]$ długości $n+1$.
- Punkt $t \in \mathbb{R}$, w którym należy obliczyć wartość wielomianu interpolacyjnego Newtona.

Wyjście

- Wartość wielomianu Newtona $N_n(t)$ w punkcie t.

Idea działania

Ponieważ wielomian Newtona jest postaci:

$$N_n(t) = f[x_0] + f[x_0, x_1](t - x_0) + f[x_0, x_1, x_2](t - x_0)(t - x_1) \\ + \cdots + f[x_0, \ldots, x_n](t - x_0)(t - x_1) \ldots (t - x_{n-1}) \quad (1)$$

Bezpośrednie obliczanie wymagałoby mnożenia coraz dłuższych iloczynów i dodawania kolejnych wyrazów, co prowadzi do kosztu rzędu $O(n^2)$. Uogólniony algorytm Hornera pozwala uniknąć tego problemu, przekształcając obliczenia w prostą procedurę rekurencyjnego „zwijania” wielomianu.
Zaczynamy od najwyższego współczynnika (ilorazu różnicowego rzędu n) i cofamy się w dół, w każdej iteracji doklejając czynnik $(t - x_i)$ oraz dodając współczynnik niższego rzędu. Dzięki temu wielomian jest budowany krok po kroku, bez potrzeby jawnego rozwijania wszystkich iloczynów.

Opis kroków

1. Ustaw zmienną pomocniczą $w \leftarrow fx[n+1]$, czyli najwyższy współczynnik.
2. Dla $i = n, n-1, \ldots, 1$ wykonaj:

$$w \leftarrow w \cdot (t - x[i]) + fx[i].$$

3. Po zakończeniu pętli zmienna w zawiera wartość wielomianu Newtona w punkcie t.

Złożoność

- Czasowa: $O(n)$, ponieważ wykonujemy dokładnie n kroków.
- Pamięciowa: $O(1)$, ponieważ używana jest tylko jedna zmienna pomocnicza w.

Zadanie 3

Algorytm przekształcania wielomianu Newtona do postaci naturalnej

Wejście

- Wektor węzłów $x = [x_0, x_1, \ldots, x_n]$ długości $n + 1$.
- Wektor ilorazów różnicowych $fx = [f[x_0], f[x_0, x_1], \ldots, f[x_0, \ldots, x_n]]$ długości $n + 1$.

Wyjście

- Wektor współczynników postaci naturalnej $a = [a_0, a_1, \dots, a_n]$, gdzie

$$N_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n.$$

Idea działania

Wielomian Newtona zapisany jest w postaci iloczynowej z ilorazami różnicowymi jako współczynnikami. Aby uzyskać jego postać naturalną, należy kolejno rozwijać czynniki $(x - x_i)$ i aktualizować współczynniki wielomianu. Algorytm działa iteracyjnie: zaczyna od najwyższego współczynnika $fx[n+1]$, a następnie w każdej iteracji „dokleja" kolejny czynnik $(x - x_k)$, przesuwając współczynniki w dół i dodając odpowiedni iloraz różnicowy. Dzięki temu po n krokach powstaje pełny wektor współczynników postaci naturalnej.

Opis kroków

1. Utwórz wektor a długości $n + 1$ i wypełnij zerami.
2. Ustaw $a[n + 1] \leftarrow fx[n + 1]$ (najwyższy współczynnik Newtona).
3. Dla $k = n, n - 1, \dots, 1$:
   (a) Dla $j = k, k - 1, \dots, 1$ wykonaj:
   $$a[j] \leftarrow a[j] - x[k] \cdot a[j + 1].$$
   (b) Następnie zaktualizuj:
   $$a[k] \leftarrow a[k] + fx[k].$$
4. Po zakończeniu pętli wektor a zawiera współczynniki postaci naturalnej.

Złożoność

- Czasowa: $O(n^2)$, ponieważ każda z n iteracji wymaga aktualizacji do n współczynników.
- Pamięciowa: $O(n)$, ponieważ przechowywany jest tylko wektor wynikowy a.