Mateusz Bartnicki, grupa nr 3 środa 16:45-18:15

Metody obliczeniowe w nauce i technice, ćwiczenie 1 - 28.02.2024 r.

**1. Treść zadania (nr 12)**

Dany jest wielomian Wilkinsona W(x) = (x−1)(x−2)...(x−20). Przekształcić wielomian do postaci W(x) = x20 + a19x19 + ... + a1x + a0. Podać wyznaczone wartości współczynników ai.  
Dla x = 1 oraz x = 20 dokładna wartość wielomianu wynosi 0. Obliczyć wartość wielomianu dla   
x = 1 oraz x = 20. Do wyznaczenia wartości wielomianu wykorzystać schemat Hornera oraz dowolny schemat sumacyjny. Obliczenia wykonać dla zmiennych typu float, double, long double.

**2. Dane techniczne**

Program został napisany przy użyciu języka Python (3.10.12) z wykorzystaniem biblioteki numpy. Ćwiczenie zostało wykonane na WSL (Windows Subsystem for Linux) - Ubuntu 22.04.3 LTS na procesorze Intel Core i5-11400H 2.70GHz.

**3. Przebieg ćwiczenia**

a) Wyznaczenie współczynników ai wielomianu Wilkinsona. Użyłem w tym celu funkcji poly z biblioteki numpy.

b) Obliczenie wartości wielomianu dla x = 1 oraz x = 20 z wykorzystaniem schematu Hornera, czyli obliczenie p(x) ze wzoru:

Obraz zawierający Czcionka, pismo odręczne, kaligrafia, tekst

Opis wygenerowany automatycznie

c) Obliczenie wartości wielomianu dla x = 1 oraz x = 20 z wykorzystaniem metody dziel i rządź

**4. Analiza otrzymanych wyników**

Na początku warto przedstawić otrzymane współczynniki. Prezentują się one następująco:

Tabela 1 – współczynniki wielomianu Wilkinsona, gdzie a0 to wyraz wolny, natomiast a20 to współczynnik przy x20

|  |  |
| --- | --- |
| a20 = 1 | a10 = 1307535010540395 |
| a19 = -210 | a9 = -1,014229986551145e+16 |
| a18 = 20615 | a8 = 6,30308120992949e+16 |
| a17 = -1256850 | a7 = -3,1133364316139066e+17 |
| a16 = 53327946 | a6 = 1,2066478037803732e+18 |
| a15 = -1672280820 | a5 = -3,599979517947607e+18 |
| a14 = 40171771630 | a4 = 8,037811822645052e+18 |
| a13 = -756111184500 | a3 = -1,2870931245150988e+19 |
| a12 = 11310276995381 | a2 = 1,3803759753640704e+19 |
| a11 = -135585182899530 | a1 = -8,7529480367616e+18 |
| a0 = 2,43290200817664e+18 | |

Jak widać, największa wartość jest liczbą rzędu 1019, więc dalsze obliczenia będą wykonywane na liczbach o podobnych rzędach wielkości.

Następnym krokiem było obliczenie wartości wielomianu w punkcie x = 1 oraz x = 20. Są to miejsca zerowe naszego wielomianu, a więc otrzymane wartości powinny wynosić 0. Wyniki są jednakże zupełnie inne.

Tabela 2 – Wartości uzyskane przy wykorzystaniu schematu Hornera dla różnych użytych typów liczbowych

|  |  |  |
| --- | --- | --- |
| Wartości uzyskane przy wykorzystaniu schematu Hornera | | |
| Typ liczbowy | Przybliżona otrzymana  wartość wielomianu dla x = 1 | Przybliżona otrzymana  wartość wielomianu  dla x = 20 |
| float | -5,20149438464 \* 1011 | -2,3361908901674756 \* 1021 |
| double | 1024 | -2,27029504 \* 1010 |
| long double | 1184 | -2,27029504 \* 1010 |

Tabela 3 – Wartości uzyskane przy wykorzystaniu metody dziel i rządź dla różnych użytych typów liczbowych

|  |  |  |
| --- | --- | --- |
| Wartości uzyskane przy wykorzystaniu metody dziel i rządź | | |
| Typ liczbowy | Przybliżona otrzymana  wartość wielomianu dla x = 1 | Przybliżona otrzymana  wartość wielomianu  dla x = 20 |
| float | -5,20149438612 \* 1011 | 1,744499869825665 \* 1027 |
| double | 1247 | 1,7444998473213508 \* 1027 |
| long double | 1184 | 1,7444998473213508509 \* 1027 |

Obliczone wartości znacząco różnią się od wartości oczekiwanej wynoszącej 0. Największą różnicę widać w wartości obliczonej w punkcie x = 20, gdzie przy wykorzystaniu metody dziel i rządź różnica jest liczbą rzędu 1027. Dostrzec można również, że żadna uzyskana wartość nie jest nawet bliska 0. Dzieje się tak, ponieważ otrzymane wcześniej współczynniki są bardzo dużymi liczbami, więc podczas dodawania dwóch liczb, które są normalizowane, istnieje ryzyko „zgubienia” pewnych cyfr wychodzących poza zakres mantysy. Gdy powielamy takie operacje, błąd staje się coraz większy, co skutkuje tak dużymi różnicami w obliczeniach.

Pokusiłem się również, aby wykorzystać funkcję roots z biblioteki numpy, która służy do znajdywania miejsc zerowych wielomianu. W ten sposób dla liczb typu double miejsce zerowe w punkcie x = 20 zamieniło się w miejsce zerowe w punkcie x = 19,999, a miejsce zerowe w punkcie x = 19 zamieniło się w miejsce zerowe w punkcie x = 19,001. Prawie na każdej pozycji wystąpiły niewielkie przesunięcia. Zmianie nie uległy tylko punkty ze zbioru {1, 2, 3}, co tłumaczy mniejsze rozbieżności dla x = 1, a znacząco większe dla x = 20.