Laboratorium 2

Arytmetyka komputerowa (cd.)

Bartłomiej Szubiak

12.03.2024

Zad 1

Napisać algorytm do obliczenia funkcji wykładniczej e^x przy pomocy nieskończonych szeregów e^x= 1+ x + x^2/2! + x^3/3! + ...

(1a) Wykonując sumowanie w naturalnej kolejności, jakie kryterium zakończenia obliczeń przyjmiesz ?

(1b) Proszę przetestować algorytm dla: x= +-1, +-5, +-10 i porównać wyniki z wynikami wykonania standardowej funkcji exp(x)

(1c) Czy można posłużyć się szeregami w tej postaci do uzyskania dokładnych wyników dla x < 0 ?

(1d) Czy możesz zmienić wygląd szeregu lub w jakiś sposób przegrupować składowe żeby uzyskać dokładniejsze wyniki dla x < 0 ?

```
import math

def exp_series(x, epsilon=1e-10):
   result = 0
   term = 1
   n = 0
   while abs(term) > epsilon:
      result += term
      n += 1
      term = x**n / math.factorial(n)
   return result
```

(1a) Kryterium zakończenia przyjmuje jako moment gdzie wartość kolejnego składnika szeregu jest mniejsza niż pewna wartość **epsilon**, gdzie epsilon jest małą liczbą reprezentującą błąd tolerancji.

(1b) Wyniki testów:

x = 1:
moja implementacja: 2.7182818284467594
math.exp(x): 2.718281828459045
roznica: 1.2285727990501982e-11

x = -1:
moja implementacja: 0.36787944116069127
math.exp(x): 0.36787944117144233
roznica: 1.0751066703562628e-11

x = 5:
moja implementacja: 148.41315910255133
math.exp(x): 148.4131591025766
roznica: 2.5266899683629163e-11

x = -5:
moja implementacja: 0.006737947017131784
math.exp(x): 0.006737946999085467
roznica: 1.804631270807544e-11

x = 10:
moja implementacja: 22026.465794806667
math.exp(x): 22026.465794806718
roznica: 5.093170329928398e-11

x = -10:
moja implementacja: 4.5399898677314684e-05
math.exp(x): 4.5399929762484854e-05
roznica: 3.108517017061255e-11

Wyniki nie znacznie się różnia od implementacji bibliotecznej, każdy o wartość mniejszą niż epsilon

(1c) Można, lecz spowoduje to powstanie **większego błędu** niż w przypadku kiedy x jest dodatni. Spowodowane jest to przez to, że ujemna liczba podniesiona do nieparzystej potęgi jest dalej ujemna, tak więc będziemy odejmować liczby. W przypadku odejmowania bliskich liczb może powstać zjawisko (ang) „catastrophic cancellation" wpływające na powstanie większego błędu

(1d) Tak wystarczy, że funkcje będę rozwijał w szereg Taylora blisko punktu x , tzn. $x_0 \approx x$, tak aby wartość $(x - x_0)^n$ była > 0 , oraz na tyle daleko od x aby nie powstawało zjawisko (ang) „catastrophic cancellation"

szereg będzie postaci:

$$e^x = \sum_{n=0}^{\infty} \frac{(x - x_0)^n}{n!} \cdot e^{x_0} = e^{x_0} \cdot \sum_{n=0}^{\infty} \frac{(x - x_0)^n}{n!}$$

Lub prościej:
Jeśli $x < 0$ niech $a = -x$, więc $a > 0$:

$$e^x = \frac{1}{e^a} = \frac{1}{\sum_{n=0}^{\infty} \frac{a^n}{n!}}$$

Wnioski:
Aby zachować stabilność numeryczną i zapobiec zjawisku (ang) „catastrophic cancellation" wystarczy użyć prostych przekształceń algebraicznych bądź zmodyfikować wzór.

Zad2

Które z dwóch matematycznie ekwiwalentnych wyrażeń x**2 - y**2 oraz (x - y)*(x + y) może być obliczone dokładniej w arytmetyce zmienno-przecinkowej ? Dlaczego ?

Dokładniejsze do obliczeń jest wyrażenie $(x - y)(x + y)$, ponieważ ma ono tylko jedno mnożenie w przeciwieństwie do drugiego wyrażenia. Operacja mnożenia jest mniej dokładna niż operacja dodawania czy odejmowania.

Jeśli fl jest wynikiem algorytmu zmiennoprzecinkowego to:

(1) Dla wyrażenia $(x - y)(x + y)$:

$$fl\big((x - y)(x + y)\big) = fl\big((x - y + \zeta_1)(x + y + \zeta_2)\big) =$$

$$x^2 - y^2 + (x^2 + y^2)\xi + \zeta_1 + \zeta_2$$

(2) Dla wyrażenia $x^2 - y^2$:

$$fl(x^2 - y^2) = fl(x^2(1 + \xi_1) - y^2(1 + \xi_2)) =$$

$$x^2 - y^2 + x^2\xi_1 + y^2\xi_2 + (x^2 - y^2)\zeta$$

Gdzie:
ζ *to błąd operacji dodawania/odejmowania*
ξ *to błąd operacji mnożenia*

Podsumowanie:
Jak można zauważyć w przypadku (1) pojawiają się 2 błędy dodawania i 1 mnożenia, a w przypadku (2) 2 mnożenia i 1 dodawania. Biorąc pod uwagę fakt, że operacje mnożenia są mniej dokładne mniejszy błąd otrzymamy licząc wyrażenie (1)

Zad3

Dla jakich wartości x i y, względem siebie, istnieje wyraźna różnica w dokładności dwóch wyrażeń ?

Widoczną różnice można dostrzec kiedy $x \gg y$ np. niech $x = 100$, $y = 1$ oraz weźmy system zmiennoprzecinkowy o podstawie β = 10 i długości mantysy 5, wtedy:

(1) $(x - y)(x + y) = 0.99990 \cdot 10^4$

(2) $x^2 - y^2 = 0.10000 \cdot 10^5$:

Gdzie prawdziwa wartość: 9999 co jest równe (1) przypadkowi.

Zad4

Zakładamy że rozwiązujemy równanie kwadratowe ax**2 + bx + c = 0,
z a = 1.22, b = 3.34 i c = 2.28, wykorzystując znormalizowany system zmienno-przecinkowy z podstawa beta = 10 i dokładnością p = 3.
(a) ile wyniesie obliczona wartość b**2 – 4ac ?
(b) jaka jest dokładna wartość wyróżnika w rzeczywistej (dokładnej) arytmetyce ?
(c) jaki jest względny błąd w obliczonej wartości wyróżnika ?

(a) Wstawiając wartości i uwzględniając reprezentację:

$$\hat{\Delta} = (0.334 \cdot 10^1) \cdot (0.334 \cdot 10^1) - (0.4 \cdot 10^1) \cdot (0.122 \cdot 10^1) \cdot (0.228 \cdot 10^1) =$$
$$= 0.112 \cdot 10^2 - 0.488 \cdot 10^1 \cdot 0.228 \cdot 10^1 = 0.112 \cdot 10^2 - 0.111 \cdot 10^2 = 0.001 \cdot 10^2 = \mathbf{0.1}$$

(b) Dokładna wartość: $\Delta = 0.0292$

(c) Błąd względny: $b_w = \frac{\hat{\Delta} - \Delta}{\Delta} = \frac{0.1 - 0.0292}{0.0292} = 2.42$

Wnioski:
Można na tym przykładzie zobaczyć jak operacje zmienna przecinkowe zaburzają prawdziwą wartość

Bibliografia:

prezentacja podana na pierwszych zajęciach
wykład dr inż. Katarzyna Rycerz
https://en.wikipedia.org/wiki/Catastrophic_cancellation
https://en.wikipedia.org/wiki/Floating-point_arithmetic