Obliczenia Naukowe

Lista I

Kacper Pieniążek, 236606

Wtorek TP 15:15

21 października 2018

Spis treści

1 Zadanie 1

1.1 Cel zadania

Celem zadania jest wyznaczenie pewnych wartości dla typów zmiennopozycyjnych zgodnych z IEE 754 (half, single, double). Poszukiwane wartości to epsilon maszynowy, eta oraz największa liczba w dannej arytmetyce.

1.2 Rozwiązanie

Do rozwiązania zadania wykorzystano język Julia. Wszystkie wartości zostały wyznaczone iteracyjnie.

1.2.1 Znalezienie epsilonu maszynowego

Epsilon maszynowy *macheps* to najmniejsza liczba $macheps > 0$ taka, że $fl(1.0 + macheps) > 1.0$.

Algorithm 1 Wyznaczenie epsilona maszynowego

$macheps \leftarrow fl(1.0)$
while $fl(1.0 + fl(macheps/2.0)) > 1.0$ **do**
    $macheps \leftarrow fl(macheps/2.0)$
end while
return macheps

Algorytm 1. szuka w pętli epsilona maszynowego do momentu napotkania liczby $x <= 1.0$. Wtedy wiadomo, że poprzednia liczba jako ostatnia spełniała warunek pętli i jest zwracana.

1.2.2 Znalezienie liczby η

Eta η to najmniejsza liczba taka, że $\eta > 0.0$ dla danego typu zmiennopozycyjnego.

Algorithm 2 Wyznaczenie η

$eta \leftarrow fl(1.0)$
while $fl(eta/2.0)) > 0.0$ **do**
    $eta \leftarrow fl(eta/2.0)$
end while
return eta

Algorytm 2. działa analogicznie do poprzedniego algorytmu, zmienia się jedynie warunek pętli. Program kończy działanie, gdy $current <= 0.0$

1.2.3 Znalezienie największej wartości

Poszukiwana jest największa wartość *max* przedstawialna w danym typie zmiennopozycyjnym taka, że $max < Inf$

Algorithm 3 Wyznaczenie max. wartości

$val \leftarrow prevFl(1.0)$
while $!isInf(fl(val * 2.0))$ **do**
    $val \leftarrow fl(val/2.0)$
end while
return val

W porównaniu do poprzednich podproblemów, Algorytm 3. różni się jedynie warunkiem pętli. Program kończy działanie, gdy $current == Inf$

1.2.4 Wyniki

Typ	Macheps	Eta	Max
Float16	0.000977	6.0e-8	6.55e4
Float32	1.1920929e-7	1.0e-45	3.4028235e38
Float64	2.220446049250313e-16	5.0e-324	1.7976931348623157e308

1.3 Wnioski

Otrzymane wyniki pokrywają się z wartościami zwracanymi przez bibliotekę standardową języka Julia, oraz ze specyfikacją IEE 754.

2 Zadanie 2

2.1 Cel zadania

Celem zadania jest sprawdzenie, że wyrażenie $3(4/3 - 1) - 1$ wylicza *macheps* w danej arymetyce zmiennopozycyjnej, porównując otrzymane wyniki z prawdziwymi wartościami

2.2 Rozwiązanie

Rozwiązanie polega na obliczeniu wyrażenia dla każdego z typów half, single, double. Wyniki zostają porównane z oczekiwanymi wartościami, wyznaczonymi przez bibliotekę standardową języka Julia.

2.2.1 Wyniki

Type	Float16	Float32	Float64
Value	-0.000977	1.1920929e-7	-2.220446049250313e-16
Expected	0.000977	1.1920929e-7	2.220446049250313e-16

2.3 Wnioski

Ponieważ liczby maszynowe mają określoną długość mantysy, podczas obliczania $3(4/3-1)-1$ zachodzą zaokrąglenia, np. ze względu nieskończonego rozwinięcia $4/3$ w postaci binarnej, czy odejmowania bliskich sobie wartości.

3 Zadanie 3

3.1 Cel zadania

Celem zadania jest sprawdzenie, czy liczby zmiennoprzecinkowe podwójnej precyzji są równomiernie rozmieszczone na przedziale $[1, 2]$ z krokiem $\delta = 2^{-52}$. Należy również zbadać rozmieszczenie liczb na przedziałach $[0.5, 1]$ oraz $[2, 4]$.

3.2 Rozwiązanie

Rozwiązanie iteruje po kolejnych wartościach w określonym przedziale o ustalony krok. Zapis bitowy pokazuje, że liczby są swoimi następnikami w arytmetyce.

3.3 Wnioski

Liczby są rozmieszczone równomiernie na przedziałach kolejnych potęg dwójki. Wynika to z faktu, że wyższe potęgi liczby 2 zwiększają długość zapisu w bitach, zwiększając cechę liczby i zmniejszając dokładność zapisu. To znaczy, że na przedziale $[0.5, 1]$ liczby są rozmieszczone równomiernie z $\delta = 2^{-53}$, a na $[2, 4] \rightarrow 2^{-51}$

$\delta = 2^{-52}$ to również epsilon maszynowy dla typu double, więc jest to najmniejsza odległość, jaka może występować pomiędzy kolejnymi liczbami na przedziale zerowej potęgi.

4 Zadanie 4

4.1 Cel zadania

Celem jest znalezienie dowolnej liczby $1 < x < 2$ takiej, że $fl(x * fl(1/x)) \neq 1.0$ oraz najmniejszej liczby spełniającej drugi warunek.

4.2 Rozwiązanie

Rozwiązanie obu części zadania sprowadza się do iterowania po wartościach w ustalonym przedziale i sprawdzania, czy liczby spełniają warunek $fl(x * fl(1/x))$. Dla pierwszej części zadania jest przedział $[1, 2$, dla drugiej - $[-Inf, Inf]$

4.2.1 Wyniki

Najmniejsza liczba taka, że $x*(1/x)! = 0$ na przedziale $[1, 2]$ jest równa 1.000000057228997. Najmniejsza liczba spełniająca ten warunek na przedziale $[-Inf, Inf]$ to -1.7976931348623157e308.

4.3 Wnioski

Iterując po wszystkich wartościach od $-Inf$ do Inf można zauważyć, że przy wielkościach rzędu e308 warunek zadania jest spełniony bardzo często. Wynika to z utraty precyzji przy zaokrąglaniu liczb. Ograniczenie zapisu liczby do skończonej ilości bitów mantysu prowadzi do utraty wielu cyfr znaczących, szczególnie przy bardzo małych częściach ułamkowych.

5 Zadanie 5

5.1 Cel zadania

Celem jest eksperymentalne obliczenie iloczynu skalarnego danych wektorów według różnych algorytmów, a następnie porównanie wyników z prawidłową wartością.

5.2 Rozwiązanie

Algorithm 4 Sposób I

$S \leftarrow 0$
for $i = 1, n$ **do**
    $S \leftarrow S + x[i] * y[i]$
end for

Algorithm 5 Sposób II

$S \leftarrow 0$
for $i = n, 1$ **do**
    $S \leftarrow S + x[i] * y[i]$
end for

Algorithm 6 Sposoby III, IV

$S \leftarrow Float[n]$
for $i = 1, n$ **do**
    $S[i] \leftarrow x[i] * y[i]$
end for
$S \leftarrow sort(S)$
return $sum(calcPartialSums(S))$

5.2.1 Wyniki

Sposób	Float32	Float64
I	-0.4999443	1.0251881368296672e-10
II	-0.4543457	-1.5643308870494366e-10
III	-0.5	0.0
IV	-0.5	0.0

5.3 Wnioski

Porównując z prawidłowym wynikiem $-1.00657107000000e-11$ widać, że najdokładniejszy okazał się sposób II.

6 Zadanie 6

6.1 Cel zadania

Celem zadania jest porównanie wyników zwracanych przez zadane funkcje matematyczne dla argumentów $x = 8^{-1}, 8^{-2}, 8^{-3}, \cdots$ w arytmetyce Float64.

$$f(x) = \sqrt{x^2 + 1} - 1$$

$$g(x) = \frac{x^2}{\sqrt{x^2 + 1} + 1}$$

Istotną częścią zadania jest zrozumienie działania operacji matematycznych na liczbach maszynowych. Pomimo $f = g$, wyniki zwracanie przez funkcje różnią się od siebie.

6.2 Rozwiązanie

Rozwiązanie wykorzystuje dwie funkcje f, g do obliczania wartości dla kolejnych wartości w pętli dla potęg $[-1, -2, \cdots, -12]$. Do zaimplementowania algorytmu wykorzystaniu język programowania Julia.

6.2.1 Wyniki

x	f(x)	g(x)
8^{-1}	0.0077822185373186414	0.0077822185373187065
8^{-2}	0.00012206286282867573	0.00012206286282875901
8^{-9}	0.0	2.7755575615628914e-17
8^{-10}	0.0	4.336808689942018e-19
8^{-11}	0.0	6.776263578034403e-21

6.3 Wnioski

Chociaż $f = g$, obie funkcje zwracają różne wartości dla tych samych argumentów. Wynika to z odejmowania coraz bliższych sobie liczb w funkcji f, co wpływa negatywnie na dokładność i utratę cyfr znaczących. Funkcja g jest bardziej wiarygodna.

7 Zadanie 7

7.1 Cel zadania

Celem zadania jest obliczenie pochodnej funkcji $f(x) = sin(x) + cos(3x)$ przy wykorzystaniu wzoru na przybliżenie pochodnej funkcji w punkcie $x_0 = 1$ z krokiem $h = 2^{-1}, 2^{-2}, \cdots, 2^{-54}$.

7.2 Rozwiązanie

Rozwiązanie iteruje po kolejnych wartościach h, wyliczając $(f(x0 + h) - f(x0))/h$ dla $f(x) = sin(x) + cos(3x)$.

7.2.1 Wyniki

x	f.(x)	$\lvert f'(x) - f.(x)\rvert$
2^0	2.0179892252685967	1.9010469435885966
2^{-1}	1.8704413979316472	1.7534991162516471
2^{-2}	1.1077870952342974	0.9908448135542974
2^{-3}	0.6232412792975817	0.5062989976175817
8^{-26}	0.11694233864545822	5.6965458225533006e-8
8^{-27}	0.11694231629371643	3.461371643476152e-8
8^{-28}	0.11694228649139404	4.811394047066209e-9
8^{-29}	0.11694222688674927	5.4793250728324416e-8
8^{-30}	0.11694216728210449	1.1439789550371504e-7

7.3 Wnioski

Od potęgi -28 wartość błędu bezwzględnego rośnie. Wpływ na taki wynik może mieć kilka czynników. Funkcje *sin*, *cos* wpływają na utratę dokładności. Przykładowo, $sin(x)$ jest przybliżane do x dla bardzo małych x.

Brak poprawy przybliżenia pochodnej spowodowany jest utratą cyfr znaczących oraz niedokładnością działań w arytmetyce zmiennoprzecinkowej. $f(x_0 + h) - f(x_0)$ może powodować coraz większą utratę dokładności dla coraz mniejszych h.