Obliczenia Naukowe Lista 1

Paweł Prusisz

25.10.2021

1 Zadanie 1

W zadaniu 1 mamy do omówienia kilka zagadnień:

1.1 Epsilon maszynowy

1.1.1 Opis problemu

Epsilonem maszynowym (macheps) nazywamy najmniejszą liczbę w arytmetyce fl większą od 0 będącą odległością między 1 a następną liczbą. Naszym zadaniem jest jej wyznaczenie.

1.1.2 Rozwiązanie

Wartości macheps dla każdego z typów Float16, Float32 i Float64 były wyznaczane metodą iteracyjną. Zaczynając od $eps = 1.0$ sprawdzamy czy $1.0 + eps > 1.0$, jeśli tak to $eps = \frac{eps}{2}$

	Float16	Float32	Float64
eps iteracyjnie	0.000977	1.1920929e-7	2.220446049250313e-16
eps(Float)	0.000977	1.1920929e-7	2.220446049250313e-16
float.h		1.1920929e-7	2.220446049250313e-16

Wyniki Obliczeń epsilon metodą iteracyjną zgadzają się z tymi podanymi w dokumentacji.

1.2 Eta

1.2.1 Opis problemu

Liczbę maszynową eta nazywamy najmniejszą liczbę w arytmetyce fl niebędącą zerem. Naszym zadaniem jest jej wyznaczenie.

1.2.2 Rozwiązanie

Tak jak przy wyznaczaniu macheps użyjemy metody iteracyjnej, tym razem jednak warunkiem końca będzie $\frac{eta}{2} > 0.0$

	Float16	Float32	Float64
eta	6.0e-8	6.0e-8	5.0e-324
nextFloat	6.0e-8	6.0e-8	5.0e-324

Wyniki obliczen zgadają się z wynikami wbudowanej funkcji nextFloat(0.0)

1.3 Precyzja arytmetyki

1.3.1 Opis problemu

Jaki związek ma liczba macheps z precyzją arytmetyki (oznaczaną na wykładzie przez ϵ)?

1.3.2 Rozwiązanie

Obie liczby mają taką samą wartość ale opisują różne rzeczy. Precyzja arytmetyki opisuje największy błąd względy w przypadku zaokrąglenia. A macheps to błąd wokół 1.0

1.4 Związek eta z $\mathbf{Min_{sub}}$

1.4.1 Opis problemu

Jaki związek ma liczba eta z liczbą $\mathrm{Min}_{\mathrm{sub}}$?

1.4.2 Rozwiązanie

Obie liczby opisują to samo - najmniejsza liczba w danej arytmetyce fl, która jest większa od 0 (bez normalizacji tej liczby).

1.5 FloatMin() a Min_{nor}

1.5.1 Opis problemu

Co zwracają funkcje floatmin(Float32) i floatmin(Float64) i jaki jest związek zwraca- nych wartości z liczbą MIN_{nor}?

1.5.2 Rozwiązanie

Floatmin zwraca najmniejszą liczbę której wartość jest większa od 0 ale w przeciwieństwie do Min_{sub} wynik jest znormalizowany.

1.6 Max

1.6.1 Opis problemu

Wyznaczyć iteracyjnie liczbę Max dla wszytkich typów zmiennoprzecinkowych Float16, Float32, Float64.

1.6.2 Rozwiązanie

Podobnie jak w przypadku eps i eta użyjemy metody iteracyjnej. Nasze obliczenia będą odbywać się w 2 etapach. Na początku zaczniemy z $max = 1.0$ dopóki $max * 2 \neq inf \Rightarrow max = max * 2$
Po wyjściu z pętli zapamiętujemy wartość $h = \frac{max}{2}$, a następnie przechodzimy dokolejnej pętli dopóki $max + h \neq inf \Rightarrow max = max + h$, $h = \frac{h}{2}$

	Float16	Float32	Float64
Max	6.55e4	3.4028235e38	1.7976931348623157e308
FloatMax	6.55e4	3.4028235e38	1.7976931348623157e308
float.h		3.4028235e38	1.7976931348623157e308

Wyniki obliczeń tą metodą są identyczne do tego co można znaleźć w dokumentacji języka C oraz wyników zwracanych przez FloatMax.

2 Zadanie 2

2.1 Opis problemu

Kahan stwierdził, że epsilon maszynowy (macheps) można otrzymać obliczając wyrażenie $3(\frac{4}{3} - 1) - 1$ w arytmetyce zmiennopozycyjnej. Moim zadaniem jest sprawdzić czy Kahan miał rację.

2.2 Rozwiązanie

Wyniki wyrażenia $3(\frac{4}{3}-1)-1$ prezentują się następująco:

	Float16	Float32	Float64
eps	0.000977	1.1920929e-7	2.220446049250313e-16
Kahan	-0.000977	1.1920929e-7	-2.220446049250313e-16

Co zgadza się do wartości bezwzględnej z wartościamy z dokumentacji.

3 Zadanie 3

3.1 Opis problemu

W zadaniu 3 naszym problemem jest sprawdzenie jak rozmieszczone są liczby zmiennoprzecinkowe w przedziałach $[\frac{1}{2}, 1]$ oraz $[2, 4]$ oraz zailustrować to przy użyciu funkcji bitstring.

3.2 Rozwiązanie

Tak jak podane jest to w treści zadania, w przedziale $[1, 2]$ wszytkie liczby, które mają dokładną reprezentację można zapisać następująco: $x = 1 + k\delta$ gdzie $k = 1, 2, ..., 2^{52}$ i $\delta = 2^{\text{-}52}$

$[1, 2]$
0011111111110000000000000000000000000000000000000000000000000000
0011111111110000000000000000000000000000000000000000000000000001
0011111111110000000000000000000000000000000000000000000000000010
0011111111110000000000000000000000000000000000000000000000000011
0011111111110000000000000000000000000000000000000000000000000100

Jak widzimy idąc po $k = 1, 2, 3, ...$ mamy do czynienia z odliczaniem kolejnych bitów, co podobnie jak w przypadku liczb całkowitych w komputerze pozwala przeiterować się po wszytkich możliwych wartościach w danym typie.

Podobnie prezentuje się sytuacja w przedziale $[0.5, 1]$ ale w tym przypadku $x = \frac{1}{2} + k\delta$ gdzie $k = 1, 2, ..., 2^{52}$ oraz $\delta = 2^{\text{-}53}$

$[0.5, 1]$
0011111111100000000000000000000000000000000000000000000000000000
0011111111100000000000000000000000000000000000000000000000000001
0011111111100000000000000000000000000000000000000000000000000010
0011111111100000000000000000000000000000000000000000000000000011
0011111111100000000000000000000000000000000000000000000000000100

W przedziale $[2, 4]$ mamy $x = 2 + k\delta$ gdzie $k = 1, 2, ..., 2^{52}$ a $\delta = 2^{-51}$

$[2, 4]$
0100000000000000000000000000000000000000000000000000000000000000
0100000000000000000000000000000000000000000000000000000000000001
0100000000000000000000000000000000000000000000000000000000000010
0100000000000000000000000000000000000000000000000000000000000011
0100000000000000000000000000000000000000000000000000000000000100

Z obserwacji wynika, że w każdym kolejnym przedziale $[4, 8]$, $[8, 16]$... możemy zapisywać liczby co kolejno $\delta = 2^{-50}$, $\delta = 2^{-49}$...

4 Zadanie 4

4.1 Opis problemu

W tym zadaniu należy znaleźć najmniejszą liczbę w arytmetyce Float64 w przedziale $1 < x < 2$ taką, że $x * \frac{1}{x} \neq x$

4.2 Rozwiązanie

W tym zadaniu także zastosowałem podejście iteracyjne, zaczynając od $x = 1.0$ sprawdamy kolejne x, które możemy zapisać w arytmetyce Float64 i oblicamy wartość wyrażenia $\frac{1}{x} * x$. Najmniejsza taka liczba, dla której wartość wyrażenia $\frac{1}{x} * x \neq x$ to $x = 1.000000057228997$, dla którego wartość $\frac{1}{x} * x = 0.9999999999999999$

5 Zadanie 5

5.1 Opis problemu

W tym zadaniu należy obliczyć iloczyn skalarny dwóch vektorów na 4 różne sposoby i porównać wyniki z wartością dokładną.

5.2 Rozwiązanie

Vektory prezentują się następująco:

$x = [2.718281828, -3.141592654, 1.414213562, 0.5772156649, 0.3010299957]$
$y = [1486.2497, 878366.9879, -22.37492, 4773714.647, 0.000185049]$

Obliczenie powinno zostać przeprowadzone 4 różnymi metodami:
Metoda 1 w przód,
Metoda 2 w tył,
Metoda 3 od największego do najmniejszego,
Metoda 4 od najmniejszego do największego.

	Metoda 1	Metoda 2	Metoda 3	Metoda 4
Float32	-0.2499443	-0.2043457	-0.25	-0.25
Float64	1.0251881368296672e-10	-1.5643308870494366e-10	0.0	0.0

Wartość dokładna iloczynu to $1.00657107000000 * 10^{-11}$. W zależości od metody uzyskiwaliśmy różne wyniki, pomimo że wszytkie metody są z matematycznego punktu widzenia identyczne, co więcej żadna z nich nie dała nam poprawnego wyniku.

6 Zadanie 6

6.1 Opis problemu

Policzyć wartości funkcji:

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

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

Dla $x = 8^{-1}, 8^{-2}, 8^{-3}...$

6.2 Rozwiązanie

Wyniki obliczeń prezentują się następująco:

	x	f(x)	g(x)
1	0.125	0.0077822185373186414	0.0077822185373187065
2	0.015625	0.00012206286282867573	0.00012206286282875901
3	0.001953125	1.9073468138230965e-6	1.907346813826566e-6
4	0.000244140625	2.9802321943606103e-8	2.9802321943606116e-8
5	3.0517578125e-5	4.656612873077393e-10	4.6566128719931904e-10
6	3.814697265625e-6	7.275957614183426e-12	7.275957614156956e-12
7	4.76837158203125e-7	1.1368683772161603e-13	1.1368683772160957e-13
8	5.960464477539063e-8	1.7763568394002505e-15	1.7763568394002489e-15
9	7.450580596923828e-9	0.0	2.7755575615628914e-17
10	9.313225746154785e-10	0.0	4.336808689942018e-19
11	1.1641532182693481e-10	0.0	6.776263578034403e-21
12	1.4551915228366852e-11	0.0	1.0587911840678754e-22
13	1.8189894035458565e-12	0.0	1.6543612251060553e-24
14	2.2737367544323206e-13	0.0	2.5849394142282115e-26

Dla dużych wartości x, funkcja f oraz g zwraca bardzo bliskie sobie wartości, jednak od $x = 8^{-9}$ funkcja f zaczyna zwracać wartości 0.0, co w oczywisty sposób nie jest możliwe dla $x \neq 0$

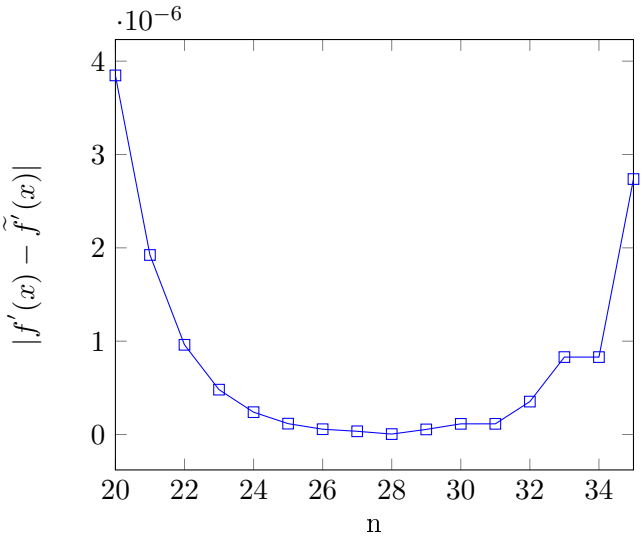
7 Zadanie 7

7.1 Opis problemu

W naszym ostatnim zadaniu musimy policzyć wartość pochodnej funkcji $f(x) = \sin x + \cos 3x$ w punkcie $x = 1.0$ przy pomocy wzoru
$f'(x) = \widetilde{f}'(x) = \frac{f(x_0+h)-f(x_0)}{h}$

7.2 Rozwiązanie

Wartości pochodnej liczonej przez granice $\frac{f(x_0+h)-f(x_0)}{h}$ dla $h = 2^{-n}, n = 0, 1, 2, 3, ..., 54$ porównywane z dokładnymi wartościami $f'(x) = \cos x - 3 \sin 3x$ prezentują się następująco:



Na wykresie dla zachowania odpowiedniej skali mamy tylko wartości $n = 20, 21, 22, ..., 33, 34, 35$. Najbliżej rzeczywistej wartości pochodnej jesteśmy dla $n = 28$, wtedy $h = 3.725290298461914 * 10^{-9}$, a wartość $|f'(x) - \widetilde{f'}(x)| = 4.802855890773117 * 10^{-9}$. Zmniejszając wartość h od tego momentu dostajemy coraz gorsze przybliżenia wartości pochodnej.