Sprawozdanie 1

Aleksander Głowacki

20.10.2022

Spis treści

1 Zadanie 1. **2**
- 1.1 Opis problemu . . . 2
- 1.2 Sposób rozwiązania . . . 2
- 1.3 Wyniki . . . 4
- 1.4 Wnioski . . . 4

2 Zadanie 2. **5**
- 2.1 Opis problemu . . . 5
- 2.2 Sposób rozwiązania . . . 5
- 2.3 Wyniki . . . 5
- 2.4 Wnioski . . . 5

3 Zadanie 3. **5**
- 3.1 Opis problemu . . . 5
- 3.2 Sposób rozwiązania . . . 6
- 3.3 Wyniki . . . 6
- 3.4 Wnioski . . . 6

4 Zadanie 4. **7**
- 4.1 Opis problemu . . . 7
- 4.2 Sposób rozwiązania . . . 7
- 4.3 Wyniki . . . 7
- 4.4 Wnioski . . . 7

5 Zadanie 5. **7**
5.1 Opis problemu 7
5.2 Sposób rozwiązania 7
5.3 Wyniki 8
5.4 Wnioski 8

6 Zadanie 6. **8**
6.1 Opis problemu 8
6.2 Sposób rozwiązania 9
6.3 Wyniki 9
6.4 Wnioski 10

7 Zadanie 7. **10**
7.1 Opis problemu 10
7.2 Sposób rozwiązania 10
7.3 Wyniki 11
7.4 Wnioski 12

1 Zadanie 1.

1.1 Opis problemu

Zadanie dotyczy wyznaczania szczególnych wartości liczbowych:

1. epsilon maszynowy - najmniejsza liczbę macheps > 0 taka, że: $fl(1.0+\text{macheps}) > 1.0$ i $fl(1.0+\text{macheps}) = 1+\text{macheps}$
2. eta - liczba maszynowa eta to najmniejsza wartość większa od 0.0
3. MAX - największa wartość liczbową jaką możemy określić w danej arytmetyce

1.2 Sposób rozwiązania

Sprawdzam w pętli czy zachodzi określony warunek z szukaną liczbą względem liczby $one(typ)$ lub $zero(typ)$. (te funkcje zwracają mi wartość jedynki i zera
w danym sytemie.)
Gdy warunek zostanie złamany to znajduję szukaną wartość.

1. ε - x
   $cond : one(type) + x/2 \neq one(type)$
   zaczynam z $x = one(type)$ a w pętli dzielę $x/2$

2. η - x
   $cond : x/2 \neq zero(type)$
   zaczynam z $x = one(type)$ a w pętli dzielę $x/2$

3. MAX - x
   w pierwszej pętli zaczynamy od $x = one(type)$. W środku pętli mnożymy x razy 2 i zatrzymujemy się, gdy
   $x * 2 = inf$
   Następnie do naszego x dodajemy uzupełnienie o postaci $dodatek = x/2$.
   Zatrzymujemy się gdy $x + dodatek/2 = inf$

1.3 Wyniki

Tabela 1: macheps

typ danych	moja funkcja	funkcja biblioteczna	float.h
Float16	0.000977	0.000977	brak
Float32	1.1920929e-7	1.1920929e-7	1.192093e-7
Float64	2.220446049250313e-16	2.220446049250313e-16	2.220446e-16

Tabela 2: eta

typ danych	moja funkcja	funkcja biblioteczna
Float16	6.0e-8	6.0e-8
Float32	1.0e-45	1.0e-45
Float64	5.0e-324	5.0e-324

Tabela 3: MAX

typ danych	moja funkcja	funkcja biblioteczna
Float16	6.55e4	6.55e4
Float32	3.4028235e38	3.4028235e38
Float64	1.7976931348623157e308	1.7976931348623157e308

1.4 Wnioski

1. Nie można zaprezentować wszystkich liczb rzeczywistych na komputerze
   W poszczególnych typach danych mamy określoną gęstość liczb oraz zakres.

2. Porównując eta z MINsub i floatmin(type) z MINnor zgadza się tylko rząd wielkości

3. $macheps = 2 * \text{precyzja arytmetyki}$

2 Zadanie 2.

2.1 Opis problemu

Wyznaczenie epsilona maszynowego używając formuły:

$$3(4/3 - 1) - 1$$

2.2 Sposób rozwiązania

Przepisanie formuły pamiętając o właściwym podstawieniu wartości 1.0

2.3 Wyniki

Tabela 4: macheps

typ danych	**moja funkcja**	**funkcja biblioteczna**
Float16	-0.000977	0.000977
Float32	1.1920929e-7	1.1920929e-7
Float64	-2.220446049250313e-16	2.220446049250313e-16

2.4 Wnioski

1. Otrzymane przeze mnie wartości mają zgodny moduł z wartościami z funkcji $eps(type)$

3 Zadanie 3.

3.1 Opis problemu

Zbadać gęstość przedziałów liczbowych:

1. $[\frac{1}{2}, 1)$
2. $[1, 2)$
3. $[2, 4)$

3.2 Sposób rozwiązania

1. Iterujemy się po przedziale $[1, 2)$ z krokiem $\delta = 2^{-52}$
   i sprawdzamy czy otrzymaliśmy $nextfloat(liczba)$

2. Nie znamy kroku więc jest trudniej.
   Wiemy, że przesunięcie w kodowaniu eksponenty wynosi 1023_{10}
   Znamy własności systemu dwójkowego. Eksponenta zmienia się nam w kolejnych potęgach liczby 2. Zatem przedziały o takiej samej eksponencie przy dążeniu do ∞ będą sie nam rozciągać.
   Jednak ilość bitów w eksponencie i mantysie
   jest stała.
   Co skłania nas do przypuszczenia, że ilość liczb w danym przedzialale pomiędzy potęgami 2 jest stała. Funkcją *bitsring* sprawdzamy czy eksponenta krańcowych liczb z przedziału jest identyczna.
   Zwracamy wartość $2^{e-1023-52}$, gdzie e jest eksponentą, 1023 to przesunięcie, a 52 to liczba bitów mantysy

3.3 Wyniki

Tabela 5: macheps

$[\frac{1}{2}, 1)$	$[1, 2)$	$[2, 4)$
1.1102230246251565e-16	2.220446049250313e-16	4.440892098500626e-16

3.4 Wnioski

1. Im bliżej zera, tym gęstsze mamy liczby w systemie IEE754 podwójnej precyzji

2. kroki pomiędzy liczbami rzeczywistymi wynoszą kolejno:

   (a) 2^{-53}

   (b) 2^{-52}

   (c) 2^{-51}

3. Przeskoki w gęstościach pokrywają się z przeskokami w potęgach dwójki

4 Zadanie 4.

4.1 Opis problemu

Znaleźć liczbę $x \in (1, 2)$ taką, że $x * \frac{1}{x} \neq x$

4.2 Sposób rozwiązania

sprawdzać ten warunek w pętli od $nextfloat(one(Float64))$ aktualizując $x := nextfloat(x)$

4.3 Wyniki

$x = 1.000000057228997$

4.4 Wnioski

W systemie Float64 dzielenie przez liczbę nie jest tożsame z mnożeniem przez odwrotność.

5 Zadanie 5.

5.1 Opis problemu

Różne sposoby obliczenia iloczynu skalarnego zadanych wektorów.

5.2 Sposób rozwiązania

Przepisanie algorytmu z treści zadania do Julii.

5.3 Wyniki

Wartość dokładna: $-1.00657107000000 * 10^{-11}$

Tabela 6: wyniki we Float32

nr algorytmu	**wynik**
1	-0.3472038161853561
2	-0.3472038162872195
3	-0.5
4	-0.5

Tabela 7: wyniki we Float64

nr algorytmu	**wynik**
1	1.0251881368296672e-10
2	-1.5643308870494366e-10
3	0.0
4	0.0

5.4 Wnioski

Kolejność wykonywania działań mocno zmienia końcowy wynik.
Wynika to ze względu na utratę precyzji na różnych etapach obliczeń.
Czasem sie straci więcej, czasem mniej.
Tak czy siak każdy wynik ma niewiele wspólnego z rzeczywistością.

6 Zadanie 6.

6.1 Opis problemu

Sprawdzić, czy dwie funkcje inaczej zapisane, ale matematycznie równoważne zwracają te same wartości dla pewnych danych.

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

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

6.2 Sposób rozwiązania

Przepisać bezpośrednio powyższe funkcje
i przetestować je dla kolejnych wartości o postaci:
$x = 8^{-k}$

6.3 Wyniki

Tabela 8: wartości funkcji

k	textbf$f(8^{-k})$	$g(8^{-k})$
0	0.41421356237309515	0.4142135623730951
1	0.0077822185373186414	0.0077822185373187065
2	0.00012206286282867573	0.00012206286282875901
3	1.9073468138230965e-6	1.907346813826566e-6
4	2.9802321943606103e-8	2.9802321943606116e-8
5	4.656612873077393e-10	4.6566128719931904e-10
6	7.275957614183426e-12	7.275957614156956e-12
7	1.1368683772161603e-13	1.1368683772160957e-13
8	1.7763568394002505e-15	1.7763568394002489e-15
9	0.0	2.7755575615628914e-17
10	0.0	4.336808689942018e-19
11	0.0	6.776263578034403e-21
12	0.0	1.0587911840678754e-22
13	0.0	1.6543612251060553e-24
14	0.0	2.5849394142282115e-26
15	0.0	4.0389678347315804e-28
16	0.0	6.310887241768095e-30
17	0.0	9.860761315262648e-32
18	0.0	1.5407439555097887e-33
19	0.0	2.407412430484045e-35
20	0.0	3.76158192263132e-37
21	0.0	5.877471754111438e-39
22	0.0	9.183549615799121e-41

6.4 Wnioski

1. Funkcja f i g dają różne wyniki, przy czym g jest dokładniejsza
2. Funkcja f w pewnym momencie traci całą precyzję i spada do zera
3. Funkcja f radzi sobie gorzej ze względu na odejmowanie 1 od pierwiastka gdzie traci precyzję
4. Należy tak dobierać kolejność działań, żeby liczba cyfr znaczących nie różniła się zbytnio

7 Zadanie 7.

7.1 Opis problemu

Przybliżenie wartości pochodnej funkcji $f(x)$ w punkcie x

7.2 Sposób rozwiązania

Podstawienie wszystkiego do zadanego wzoru, obliczenie pochodnej na papierze.

7.3 Wyniki

Wartość dokladna: $f'(x) = 0.11694228168853815$

Tabela 9: wartości funkcji

n	przybliżona pochodna	błąd	1+h
0	2.0179892252685967	1.9010469435800585	2.0
1	1.8704413979316472	1.753499116243109	1.5
2	1.1077870952342974	0.9908448135457593	1.25
3	0.6232412792975817	0.5062989976090435	1.125
4	0.3704000662035192	0.253457784514981	1.0625
5	0.24344307439754687	0.1265007927090087	1.03125
9	0.1248236929407085	0.007881411252170345	1.001953125
10	0.12088247681106168	0.0039401951225235265	1.0009765625
15	0.11706539714577957	0.00012311545724141837	1.000030517578125
16	0.11700383928837255	6.155759983439424e-5	1.0000152587890625
18	0.11695767106721178	1.5389378673624776e-5	1.0000038146972656
27	0.11694231629371643	3.460517827846843e-8	1.0000000074505806
28	0.11694228649139404	4.802855890773117e-9	1.0000000037252903
29	0.11694222688674927	5.480178888461751e-8	1.0000000018626451
30	0.11694216728210449	1.1440643366000813e-7	1.0000000009313226
35	0.11693954467773438	2.7370108037771956e-6	1.0000000000291038
37	0.1169281005859375	1.4181102600652196e-5	1.000000000007276
38	0.116943359375	1.0776864618478044e-6	1.000000000003638
39	0.11688232421875	5.9957469788152196e-5	1.000000000001819
41	0.116943359375	1.0776864618478044e-6	1.0000000000004547
43	0.1162109375	0.0007313441885381522	1.0000000000001137
50	0.0	0.11694228168853815	1.0000000000000009
51	0.0	0.11694228168853815	1.0000000000000004
52	-0.5	0.6169422816885382	1.0000000000000002
53	0.0	0.11694228168853815	1.0
54	0.0	0.11694228168853815	1.0

7.4 Wnioski

1. Największa precyzja jest dla n = 28

2. Błąd rośnie symetrycznie względem środka (n=28)

3. Badając pochodną musimy unikać wartości zbliżonych do 0 w trakcie obliczeń,
   ponieważ tracimy wtedy precyzję. Dla małego n mamy niedokładny wynik z powodu słabego parametru
   ale zwiększanie go do oporu nic nam nie da, bo tracimy precyzję gdy zbliża sie on do 0