

Obliczenia Naukowe, lista 2

Yauheni Tsyrkunovich, 250542

November 2020

Programy na wyjściu mają dużo linijek, dlatego wszystkie wyniki i zrzuty są w folderze /out (Zrzuty źle widać w sprawozdaniu)

1 Zadanie 1.

```
1 Float16
2 properValue:  -1.00657107000000004e-11
3 properValue2: -0.004296343192495245
4
5 ~~~(x1, y1)~~~
6 In    order sum: -Inf
7 Difference: Inf
8 Back order sum: Inf
9 Difference: Inf
10 From big values to small: NaN
11 Difference: NaN
12 From small values to big: NaN
13 Difference: NaN
14
15 ~~~(x2, y2)~~~
16 In    order sum: -Inf
17 Difference: Inf
18 Back order sum: Inf
19 Difference: Inf
20 From big values to small: NaN
21 Difference: NaN
22 From small values to big: NaN
23 Difference: NaN
24
25 Float32
26 properValue:  -1.00657107000000004e-11
27 properValue2: -0.004296343192495245
28
29 ~~~(x1, y1)~~~
30 In    order sum: -0.12593332
31 Difference: 0.12593331932014976
32 Back order sum: -0.23105995
33 Difference: 0.2310599535603002
34 From big values to small: -0.25
35 Difference: 0.2499999999899343
36 From small values to big: -0.25
```

```

37 Difference: 0.2499999999899343
38
39 ~~~(x2, y2)~~~
40 In order sum: -0.13022967
41 Difference: 0.12593332363661394
42 Back order sum: -0.23105995
43 Difference: 0.22676361037787066
44 From big values to small: -0.25
45 Difference: 0.24570365680750475
46 From small values to big: -0.25
47 Difference: 0.24570365680750475
48
49 Float64
50 properValue: -1.0065710700000004e-11
51 properValue2: -0.004296343192495245
52
53 ~~~(x1, y1)~~~
54 In order sum: 1.0251881368296672e-10
55 Difference: 1.1258452438296672e-10
56 Back order sum: -1.5643308870494366e-10
57 Difference: 1.4636737800494365e-10
58 From big values to small: 0.0
59 Difference: 1.0065710700000004e-11
60 From small values to big: 0.0
61 Difference: 1.0065710700000004e-11
62
63 ~~~(x2, y2)~~~
64 In order sum: -0.004296342739891585
65 Difference: 4.526036594121319e-10
66 Back order sum: -0.004296342998713953
67 Difference: 1.9378129136049527e-10
68 From big values to small: -0.004296342842280865
69 Difference: 3.502143800654389e-10
70 From small values to big: -0.004296342842280865
71 Difference: 3.502143800654389e-10

```

Widać, że po pierwsze wynik bardzo się zmienił (się zwiększył), a po drugie to wpływa na wyniki w drugim przypadku. Też jednak widać że n.p. wynik sumowania od dużych liczb w Float32 się nie zmienił chociaż różnica pomiędzy prawidłową odpowiedzią jest istotna

2 Zadanie 2.

W tym zadaniu trzeba przyjrzeć się funkcji $e^x * \ln(1 + e^{-x})$. Na wykresach "z2Geogebra.png" i "z2Gnuplot.png" widać że wykresy dobrze się zachowują do pewnego momentu. Natomiast przy $n \approx 30$ widać, że funkcja się zachowuje dziwnie. Podejrzewam, że ta różnica jest od tego że przy dużych x wartość $e^{-x} + 1 = 1$, stąd mamy $f(x) = 0$. Za pomocą Reguły de l'Hospitala łatwo pokazać, że $\lim_{x \rightarrow \infty} f(x) = 1$

3 Zadanie 3.

W tym zadaniu trzeba rozwiązać równanie $Ax = b$ dla różnych macierzy. W pliku out/output3.txt jest wynik działania programu. Widać, że macierz losowa ma bardzo dobre wyniki, natomiast macierz Hilberta można nazwać "przypadkiem skrajnym", bo różnica pomiędzy odpowiedzią poprawną i wynikiem programu jest bardzo duża.

4 Zadanie 4.

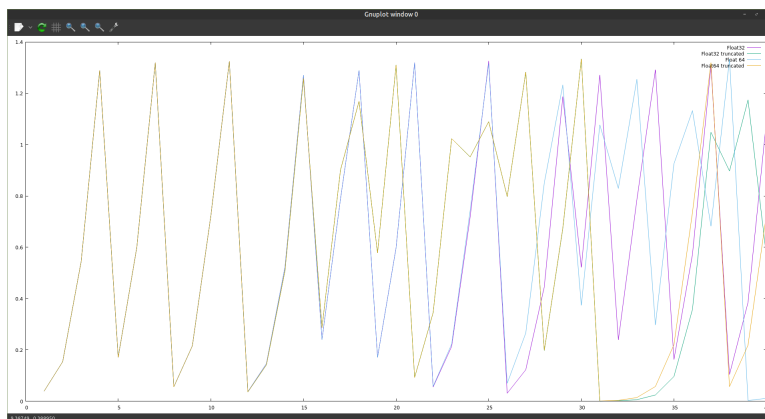
W tym zadaniu odpowiedź tak się różni od rzeczywistości, że ciężko powiedzieć czy mam błąd w obliczeniach.(output4a.txt, output4b.txt)

5 Zadanie 5.

Mamy równanie rekurencyjne, trzeba zobaczyć jak ono wygląda dla typów Float64 i Float32, też trzeba zobaczyć co się stanie, jeśli w momencie 10. iteracji obciąć wynik.

To są wyniki równania(40. iteracja):

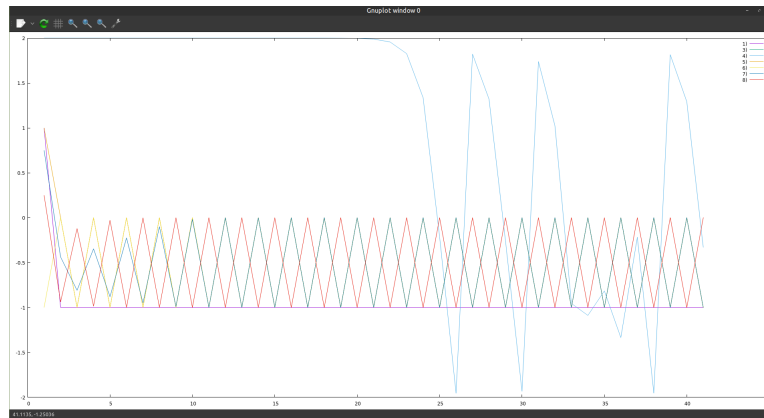
```
1 Float32: 1.0952892585098744
2 Float32 truncated: 0.5614353369949385
3 Float64: 0.011611238029748606
4 Float64 truncated: 0.7305550338104317
```



Rysunek 1: wykresy
wykresy bardzo dobrze pokazują, że małe zmiany powodują dużą różnicę tych ciągów.

6 Zadanie 6.

W tym zadaniu trzeba prześledzić kilka ciągów, a mianowicie jak się zachowują przy zwiększającym się n . Oto wykresy tych ciągów:



Rysunek 2: wykresy

Widać że od pewnego momentu $c_4 = c_5 = c_6 = -c_7$, natomiast c_1 i c_2 się nie zmieniają już od $n=2$. Jedyne ciągi, które się dziwnie zachowują to c_3