

# **MOwNiT - prezentacja laboratorium 1**

Seweryn Tasior

# Treść zadania

Dla  $x_0 > 1$  ciąg

$$x_{k+1} = 2^{k+1} \left( \sqrt{1 + 2^{-k} x_k} - 1 \right)$$

jest zbieżny do  $\log(x_0 + 1)$ . Przekształcić wzór tak, by uniknąć utraty dokładności. Porównać dwa sposoby liczenia wyrazów ciągu. Obliczenia wykonać dla zmiennych typu *float*, *double*, *long double*.

# Dane techniczne sprzętu

Obliczenia wykonano na laptopie Lenovo IdeaPad Gaming 3 15ACH6 o parametrach sprzętowych:

- Procesor: AMD Ryzen 5 5600, 6 rdzeni, 12 wątków, taktowanie do 4,4 GHz
- Pamięć RAM: 32 GB (2x16 GB), taktowanie 3200 MHz
- System operacyjny: Ubuntu 24.04.1 LTS

# Narzędzia

- Język wykorzystywany do obliczeń : Python (wersja 3.12.3).
- Do obliczenia wartości ciągów z wysoką precyzją: Numpy (float, double), mpmath (long double).
- Rysowanie wykresów: matplotlib
- Tworzenie tabel danych: Pandas

# Obliczenia

Przekształcamy wzór do prostszej postaci, aby uzyskać większą dokładność wyniku:

$$x_{k+1} = 2^{k+1} \left( \sqrt{1 + 2^{-k} x_k} - 1 \right) = 2^{k+1} \left( \sqrt{1 + 2^{-k} x_k} - 1 \right) \frac{\sqrt{1 + 2^{-k} x_k} + 1}{\sqrt{1 + 2^{-k} x_k} + 1} = \\ 2^{k+1} \frac{2^{-k} x_k}{\sqrt{1 + 2^{-k} x_k} + 1} = \frac{2x_k}{\sqrt{1 + 2^{-k} x_k} + 1}$$

W dalszej części prezentacji do obliczeń przyjmuje oznaczenia:

$$a\_x\_k = a_{x_k} = 2^{k+1} \left( \sqrt{1 + 2^{-k} x_k} - 1 \right)$$

$$b\_x\_k = b_{x_k} = \frac{2x_k}{\sqrt{1 + 2^{-k} x_k} + 1}$$

$$\text{rel\_err\_a} = \left| \frac{a_{x_k} - \log(x_0 + 1)}{\log(x_0 + 1)} \right| - \text{błąd względny } x_k \text{ w postaci } a_{x_k}$$

$$\text{rel\_err\_b} = \left| \frac{b_{x_k} - \log(x_0 + 1)}{\log(x_0 + 1)} \right| - \text{błąd względny } x_k \text{ w postaci } b_{x_k}$$

# Obliczenia

Dla typów danych zostały przyjęte warunki

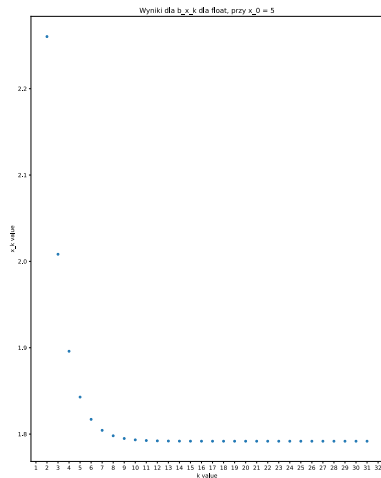
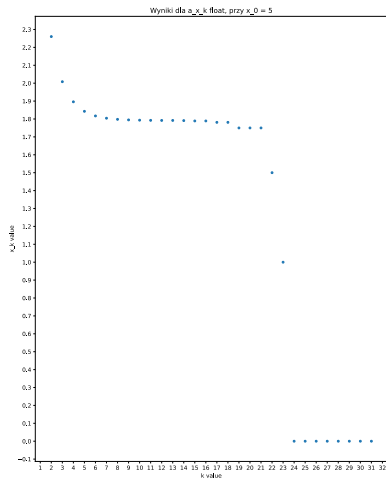
- Float
  - Rozmiar: 32 bity
  - Precyzja: 7 cyfr znaczących
- Double
  - Rozmiar: 64 bity
  - Precyzja: 17 cyfr znaczących
- Long double
  - Rozmiar: 128 bity
  - Precyzja: 33 cyfry znaczące
- W obliczeniach wartości  $a_{x_k}$  i  $b_{x_k}$  porównywałem z docelową wartością zbieżną ciągu, dzięki czemu uzyskiwałem błąd względny.
- Przyjąłem dwie wartości  $x_0$ : 5 i 10000, aby zobaczyć zachowanie się ciągu dla mniejszych i większych wartości

# Wyniki obliczeń dla float - $x_0 = 5$

Wartość dla  $\log(x_0 + 1) = 1.7917595$  dla dokładności 7 cyfr znaczących

	k	القيمة	relError	القيمة	relError
1	2	2.2603383	0.2615188	2.2603383	0.2615188
2	3	2.0082674	0.1208354	2.0082674	0.1208354
3	4	1.8959370	0.0581426	1.8959370	0.0581426
4	5	1.8428726	0.0285268	1.8428717	0.0285262
5	6	1.8170776	0.0141303	1.8170766	0.0141297
6	7	1.8043671	0.0070364	1.8043587	0.0070318
7	8	1.7988652	0.0035193	1.7988443	0.0035076
8	9	1.7949219	0.0017650	1.7948982	0.0017517
9	10	1.7933350	0.0008793	1.7933277	0.0008752
10	11	1.7924805	0.0004604	1.7925433	0.0004374
11	12	1.7919922	0.0002399	1.7921512	0.0002186
12	13	1.7919922	0.0002399	1.7919551	0.0001092
13	14	1.7910156	0.0004152	1.7918572	0.0000546
14	15	1.7890625	0.0015052	1.7918084	0.0000273
15	16	1.7890625	0.0015052	1.7917840	0.0000137
16	17	1.7812500	0.0058655	1.7917719	0.0000069
17	18	1.7812500	0.0058655	1.7917659	0.0000036
18	19	1.7500000	0.0233064	1.7917629	0.0000019
19	20	1.7500000	0.0233064	1.7917614	0.0000011
20	21	1.7500000	0.0233064	1.7917606	0.0000006
21	22	1.5000000	0.1628241	1.7917601	0.0000003
22	23	1.0000000	0.4418894	1.7917598	0.0000002
23	24	0.0000000	1.0000000	1.7917598	0.0000002
24	25	0.0000000	1.0000000	1.7917598	0.0000002
25	26	0.0000000	1.0000000	1.7917598	0.0000002
26	27	0.0000000	1.0000000	1.7917598	0.0000002
27	28	0.0000000	1.0000000	1.7917598	0.0000002
28	29	0.0000000	1.0000000	1.7917598	0.0000002
29	30	0.0000000	1.0000000	1.7917598	0.0000002
30	31	0.0000000	1.0000000	1.7917598	0.0000002

# Wyniki obliczeń dla float - $x_0 = 5$



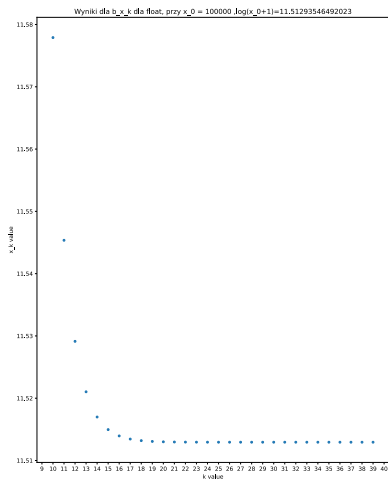
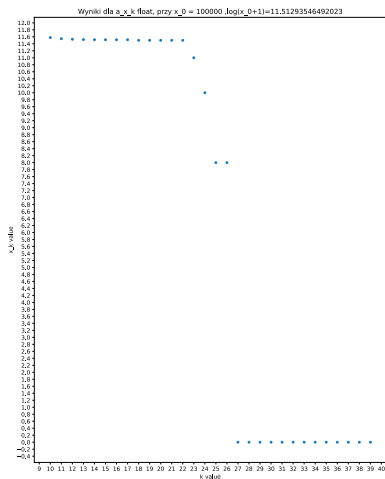


# Wyniki obliczeń dla float - $x_0 = 10^5$

Wartość dla  $\log(x_0 + 1) = 11.5129356$  dla dokładności 7 cyfr znaczących

	$k$	$\alpha_{k+1}$	$\alpha_{k+1}^2$	$\alpha_{k+1}$	$\alpha_{k+1}^2$
1	10	11.5778809	0.0056411	11.5779009	0.0056428
2	11	11.5454102	0.0029207	11.5453577	0.0029161
3	12	11.5292969	0.0014211	11.5291219	0.0014066
4	13	11.5214844	0.0007425	11.5216384	0.0007031
5	14	11.5175781	0.0004032	11.5169930	0.0003516
6	15	11.5156250	0.0002336	11.5149593	0.0001756
7	16	11.5156250	0.0002336	11.5139475	0.0000879
8	17	11.5156250	0.0002336	11.5134420	0.0000440
9	18	11.5000000	0.0011236	11.5131893	0.0000220
10	19	11.5000000	0.0011236	11.5130634	0.0000111
11	20	11.5000000	0.0011236	11.5130005	0.0000056
12	21	11.5000000	0.0011236	11.5129690	0.0000029
13	22	11.5000000	0.0011236	11.5129528	0.0000015
14	23	11.0000000	0.0445530	11.5128442	0.0000007
15	24	10.0000000	0.1314118	11.5128404	0.0000004
16	25	8.0000000	0.3051294	11.5129375	0.0000002
17	26	8.0000000	0.3051294	11.5129375	0.0000002
18	27	0.0000000	1.0000000	11.5129375	0.0000002
19	28	0.0000000	1.0000000	11.5129375	0.0000002
20	29	0.0000000	1.0000000	11.5129375	0.0000002
21	30	0.0000000	1.0000000	11.5129375	0.0000002
22	31	0.0000000	1.0000000	11.5129375	0.0000002
23	32	0.0000000	1.0000000	11.5129375	0.0000002
24	33	0.0000000	1.0000000	11.5129375	0.0000002
25	34	0.0000000	1.0000000	11.5129375	0.0000002
26	35	0.0000000	1.0000000	11.5129375	0.0000002
27	36	0.0000000	1.0000000	11.5129375	0.0000002
28	37	0.0000000	1.0000000	11.5129375	0.0000002
29	38	0.0000000	1.0000000	11.5129375	0.0000002
30	39	0.0000000	1.0000000	11.5129375	0.0000002

# Wyniki obliczeń dla float - $x_0 = 10^5$

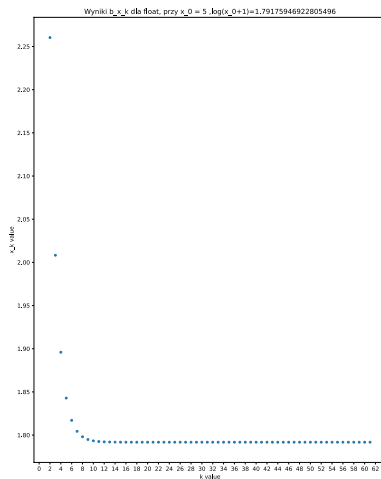
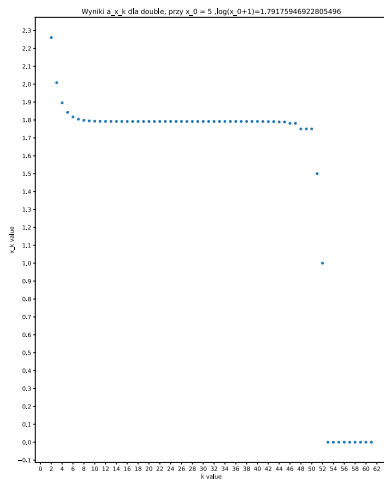


## Wyniki obliczeń dla double - $x_0 = 5$

Wartość dla  $\log(x_0 + 1) = 1.79175946922805496$  dla dokładności 17 cyfr znaczących

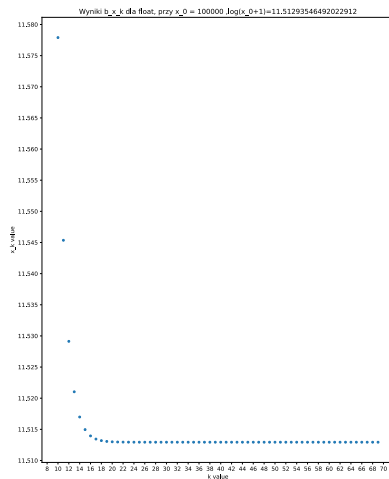
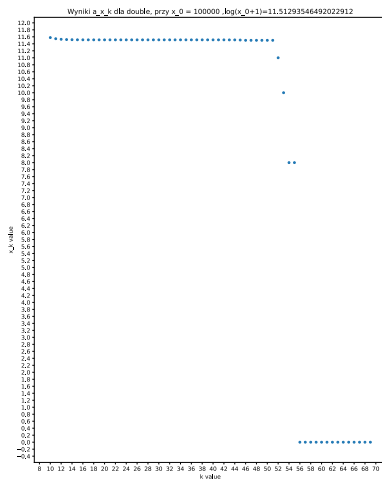
[illegible]

# Wyniki obliczeń dla double - $x_0 = 5$



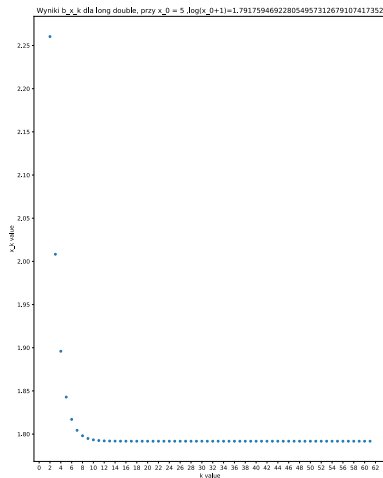
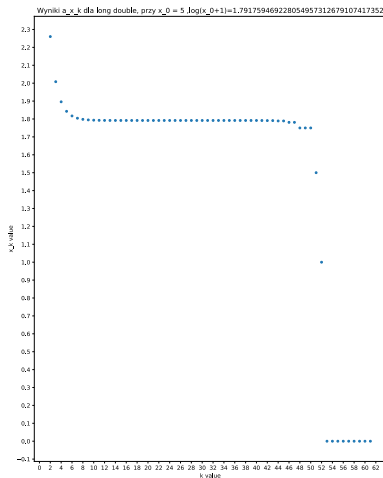


# Wyniki obliczeń dla double - $x_0 = 10^5$





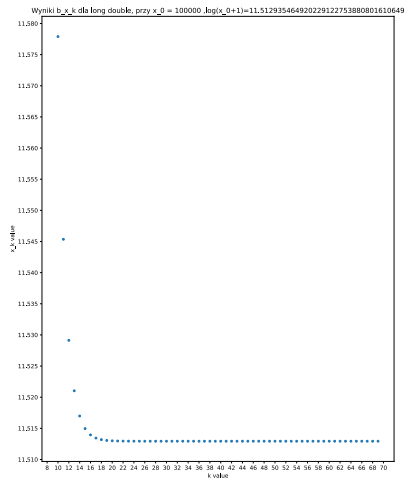
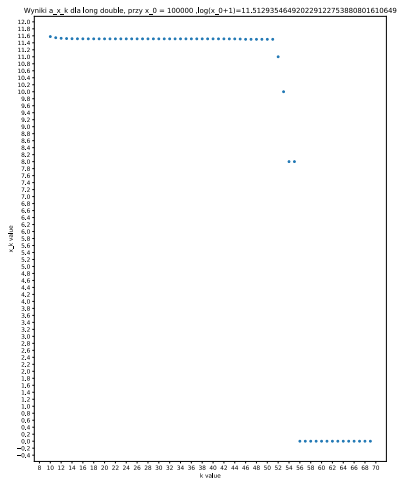
# Wyniki obliczeń dla long double - $x_0 = 5$







# Wyniki obliczeń dla long double - $x_0 = 10^5$



# Podsumowanie

Wyniki obliczeń dają następujące wnioski:

- Przekształcenie wzoru rekurencyjnego znacząco, poprawiało dokładność działań arytmetycznych, przez co umożliwia uzyskanie wyniku dokładniejszego wraz ze wzrostem  $k$
- Wzór w postaci  $a_{x_k}$  bardzo szybko zaczyna odbiegać od docelowej wartości. Z powodu nakładającej się niedokładności, od pewnego wyrazu ciągu, wartości są zerowe.