

# Metody obliczeniowe w nauce i technice - laboratorium 0 - ćwiczenie 6

Jakub Radek

## Treść zadania

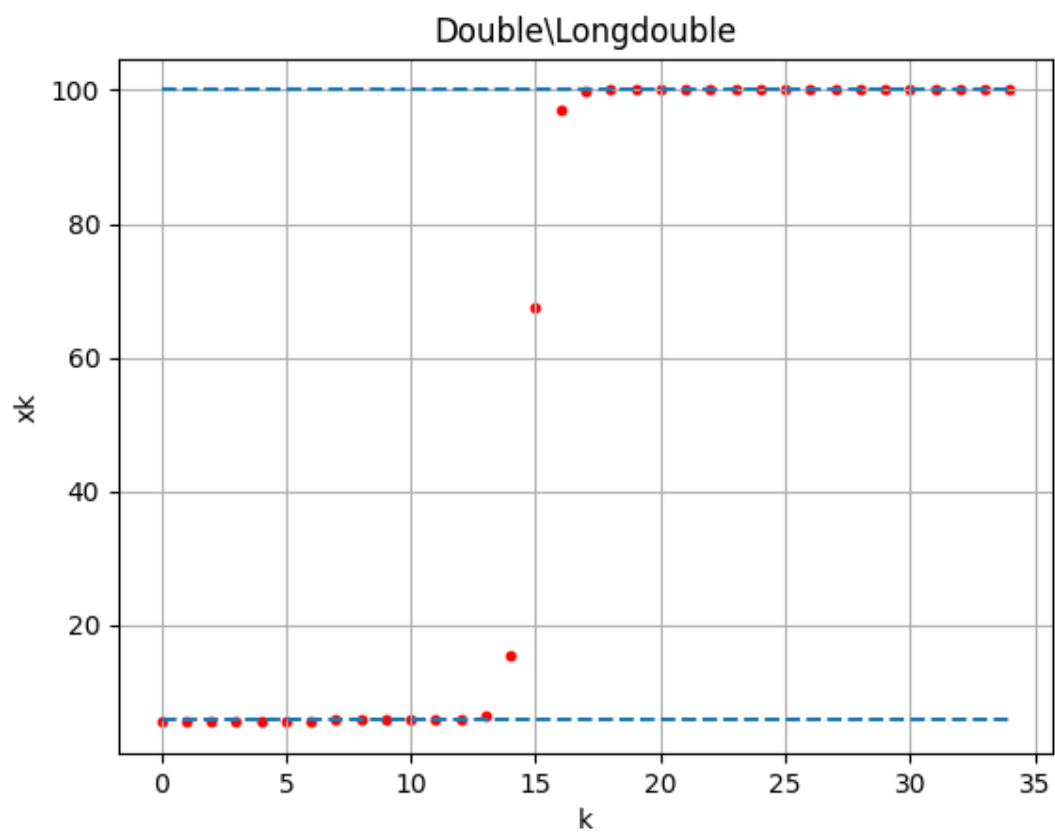
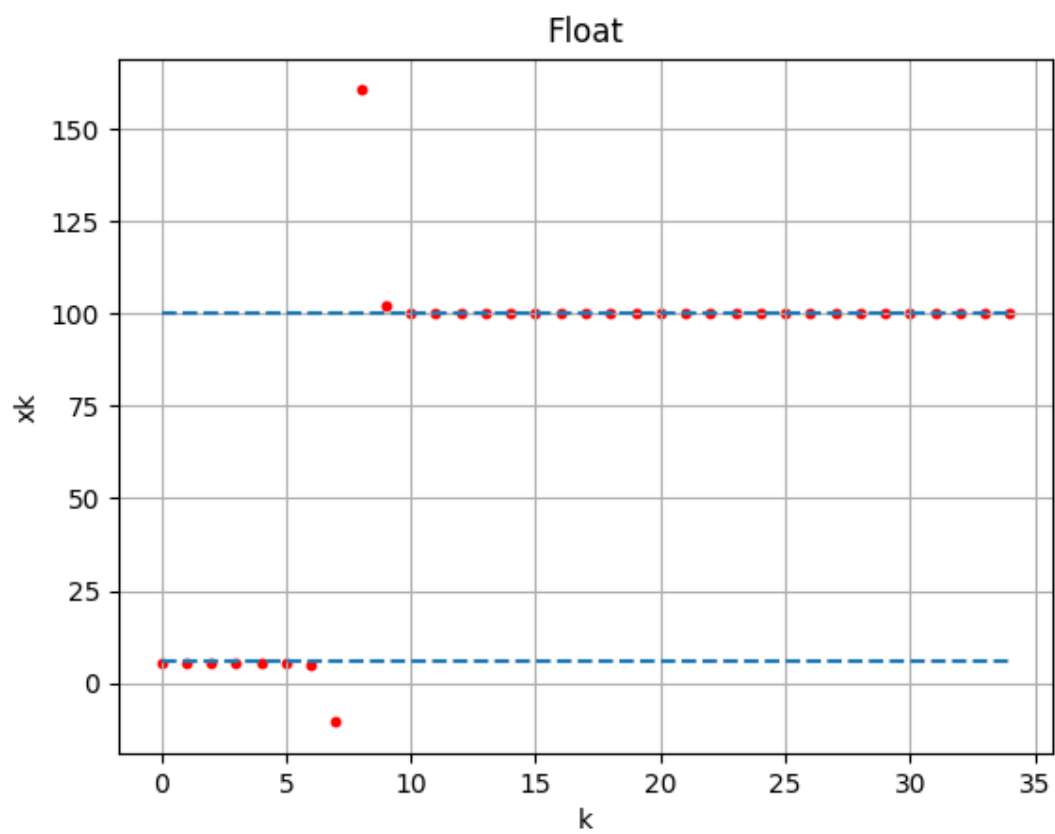
W dokładnej arytmetyce ciąg  $x_{k+1} + 1 = 111(1130 - 3000/x_{k-1})x_k$ ,  $x_0 = 11/2$ ,  $x_1 = 61/11$  jest rosnący i zbieżny do 6. Obliczyć na swoim komputerze  $x_{34}$  (dla zmiennych typu float, double, long double) i spróbować wyjaśnić uzyskane wyniki. Dokładna wartość (zaokrąglona do 4 cyfr znaczących) jest równa  $x_{34} = 5.998$ .

## Wynik

Do uzyskania wyników wykorzystano język Python w wersji 3.10, bibliotekę numpy oraz bibliotekę matplotlib. Numpy zawiera w sobie typy zmiennych, matplotlib oferuje generowanie wykresów. Poniższe tabele oraz wykresy obrazują wyniki dla podanych wyżej typów dla  $k \in [0, 34]$ . Dla typu Double i Long double został dodany jeden wykres ponieważ wartości te są identyczne.

	Float	Double	Long double
0	5.5	5.5	5.5
1	5.5454545	5.545454545454546	5.545454545454546
2	5.5901566	5.5901639344262435	5.5901639344262435
3	5.6332855	5.633431085044251	5.633431085044251
4	5.6720505	5.674648620514802	5.674648620514802
5	5.6675262	5.713329052462441	5.713329052462441
6	4.941246	5.74912092113604	5.74912092113604
7	-10.562157	5.781810945409518	5.781810945409518
8	160.5037	5.81131466923334	5.81131466923334
9	102.190025	5.83766396240722	5.83766396240722
10	100.12508	5.861078484508624	5.861078484508624

11	100.00732	5.883542934069212	5.883542934069212
12	100.00043	5.935956716634138	5.935956716634138
13	100.00002	6.534421641135182	6.534421641135182
14	100.0	15.413043180845833	15.413043180845833
15	100.0	67.47239836474625	67.47239836474625
16	100.0	97.13715118465481	97.13715118465481
17	100.0	99.82469414672073	99.82469414672073
18	100.0	99.98953968869486	99.98953968869486
19	100.0	99.9993761416421	99.9993761416421
20	100.0	99.99996275956511	99.99996275956511
21	100.0	99.99999777513808	99.99999777513808
22	100.0	99.99999986698653	99.99999986698653
23	100.0	99.9999999920431	99.9999999920431
24	100.0	99.9999999952378	99.9999999952378
25	100.0	99.999999999715	99.999999999715
26	100.0	99.999999999983	99.999999999983
27	100.0	99.999999999999	99.999999999999
28	100.0	100.0	100.0
29	100.0	100.0	100.0
30	100.0	100.0	100.0
31	100.0	100.0	100.0
32	100.0	100.0	100.0
33	100.0	100.0	100.0
34	100.0	100.0	100.0



# Wnioski

Wartości  $x$  dla każdego typu danych odbiegają od wartości jakie powinny uzyskać według dokładnej arytmetyki. W przypadku typu danych float, wartości te przez niedokładność nie zachowują nawet warunku o ciągu rosnącym. Dla typów double i long double które posiadają większą dokładność widzimy że po zaledwie paru iteracjach wartości skaczą do nowego limitu, 100, dzieje się tak przez niemożliwość dokładnego zapisania wartości  $61/11$  co powiela błąd w każdej kolejnej iteracji.