Sprawozdanie z listy drugiej

Karolina Bąk

Listopad 2019

1 Zadanie 1

Zadanie polegało na powtórzeniu zadania 5 z poprzedniej listy dla nieco zaburzonych danych. Ponownie obliczyłam iloczyn skalarny dla wektora:

$$x = (2.718281828, -3.141592654, 1.414213562, 0.5772156649, 0.3010299957)$$

oraz:

$$y = (1486.2497, 878366.9879, -22.37492, 4773714.647, 0.000185049)$$

gdzie usunęłam ostatnie 9 z x_4 oraz ostatnie 7 z x_5. Uzyskałam następujące wyniki:

	Float32	Float64
1	-0.4999443	-0.004296342739891585
2	-0.4543457	-0.004296342998713953
3	-0.5	-0.004296342842280865
4	-0.5	-0.004296342842280865

Poprzednie wyniki wyglądały następująco:

	Float32	Float64
1	-0.4999443	1.0251881368296672*10^{-10}
2	-0.4543457	-1.5643308870494366*10^{-10}
3	-0.5	0
4	-0.5	0

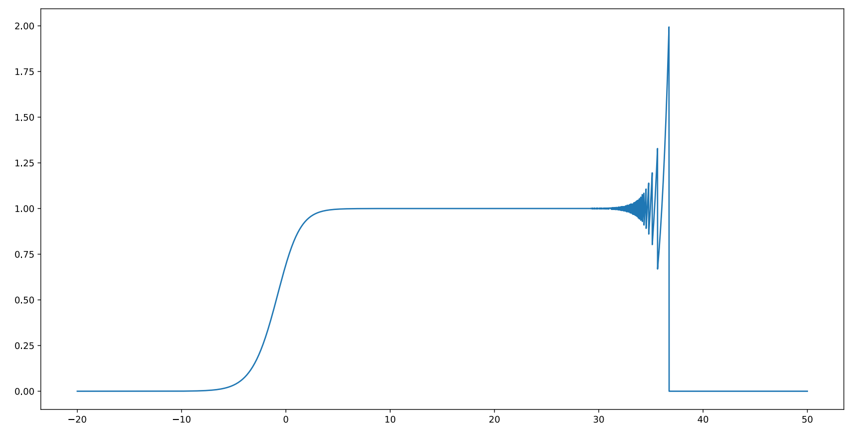
Niewielkie zmiany rzędu 10^{-10} spowodowały ogromną zmianę w wyniku (10^7 razy większy wynik). Potwierdza to wniosek z pierwszej listy, że zadanie jest źle uwarunkowane dla danych tego typu (różne znaki przy współrzędnych).

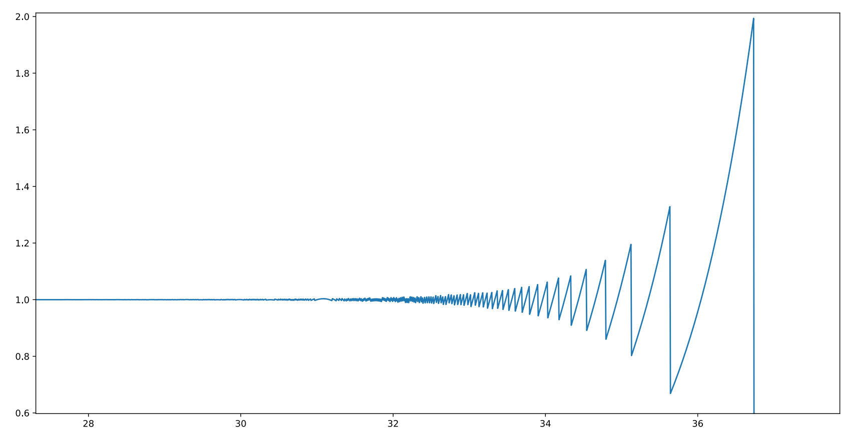
2 Zadanie 2

Celem drugiego zadania było przedstawienie graficznie wykresu funkcji:

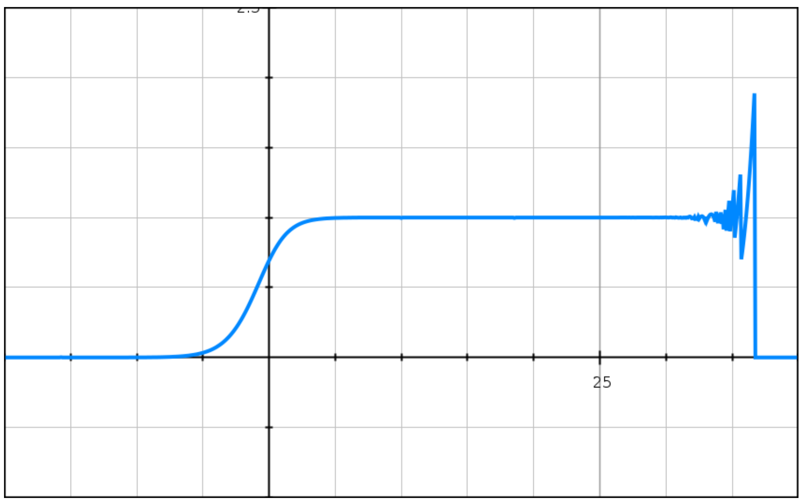
$$f(x) = e^x * ln(1 + e^{-x})$$

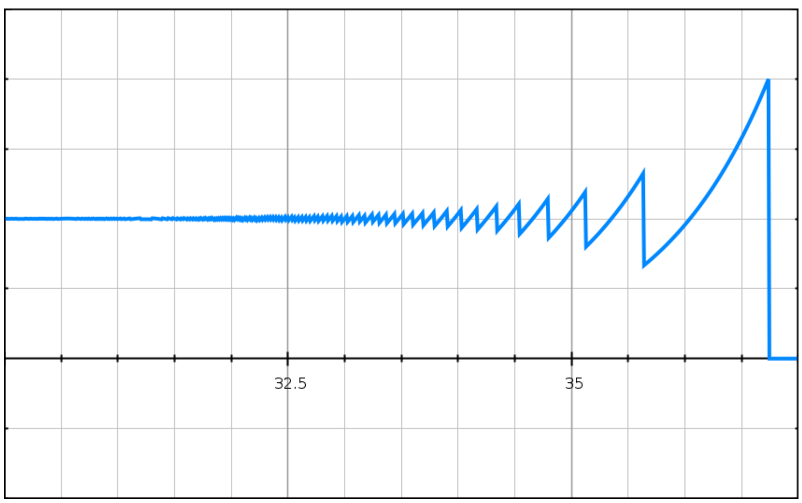
W co najmniej dwóch programach do wizualizacji. Jeden wykres wykonałam przy użyciu Pythona i paczki matplotlib, a drugi on-line na stronie graphsketch.com. Poniżej przedstawiam otrzymane wykresy w Pythonie:





Wykresy z graphsketch.com:





Procesory Intela przybliżają logarytm szeregiem Taylora. Dopóki reszty Peano zbiegają do zera jest ok. Od kiedy pojawi się błąd wynikający z ograniczeń arytmetyki zmiennoprzecinkowej w reszcie przestaje ona zbiegać do zera i szereg zaczyna przybliżać funkcję rozbieżnie, przez co oddalamy się coraz bardziej od dokładnej wartości funkcji aż dotrzemy do skrajnej wartości, gdzie zostanie ona uznana za zero maszynowe i następne wartości będą funkcją stałą f(x)=0.