Obliczenia naukowe

Sprawozdanie
Obliczenie naukowe
Lista 2

Monika Tworek

Indeks: 229776

Zad.1

Zadanie polegało na obliczeniu iloczynu skalarnego dwóch wektorów z poprzedniej listy ze zmienionymi danymi w pierwszym wektorze:
x_4=0.5772156649 x_5=0.3010299957
x_4=0.577215664 x_5=0.301029995

Sposób sumowania	niezmienione Float32	zmienione Float32
„w przód”	$1.0251881368296672e^{--10}$	-0.004296342739891585
„w tył”	-0.4543457	-0.4543457
Od największego	-0.5	-0.5
Od najmniejszego	-0.5	-0.5
	niezmienione Float64	zmienione Float64
„w przód”	$1.0251881368296672e^{-=10}$	-0.004296342739891585
„w tył”	$-1.5643308870494366e^{-10}$	-0.004296342998713953
Od największego	0.0	-0.004296342842280865
Od najmniejszego	0.0	-0.004296342842280865

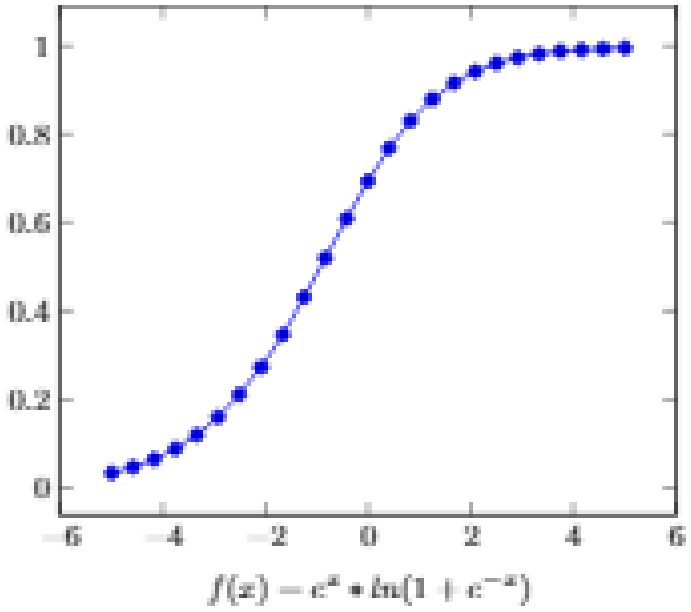
We Float32 zmiana nie miała wpływu ze względu na małą precyzję obliczeń. W przypadku dwukrotnie większej dokładności mieliśmy dużą różnicę w wynikach – między innymi nie uzyskaliśmy zer. To przykład zadania źle uwarunkowanego, gdzie minimalna zmiana na danych powoduje olbrzymie zmiany w wynikach. W tym zadaniu jest to spowodowane tym, że wektory są prawie prostopadłe.

Zad.2

Zadanie polegało na narysowaniu wykresu funkcji:

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

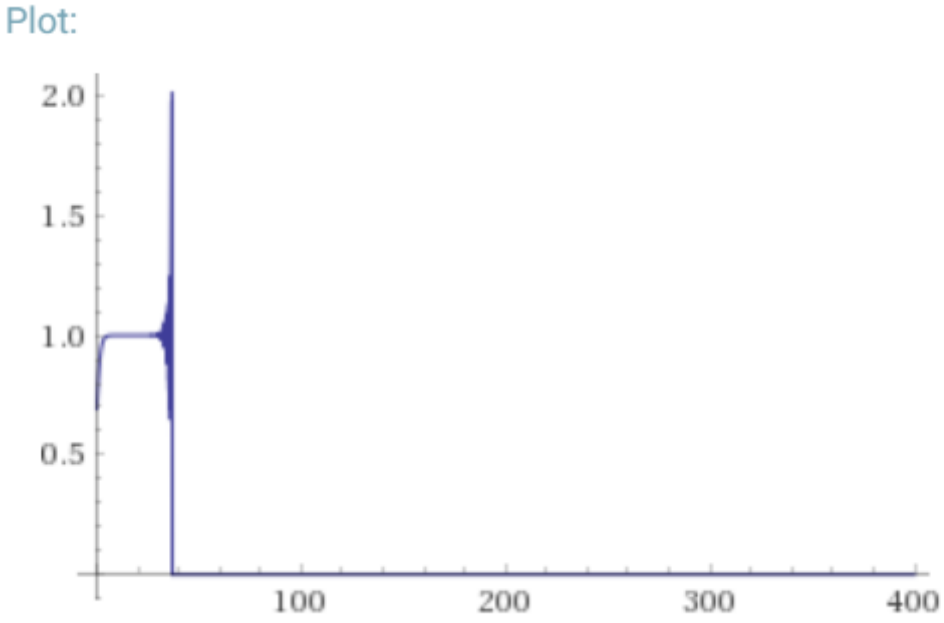
Jeżeli bierzemy pod uwagę mały zakres, to funkcja jest rysowana poprawnie:



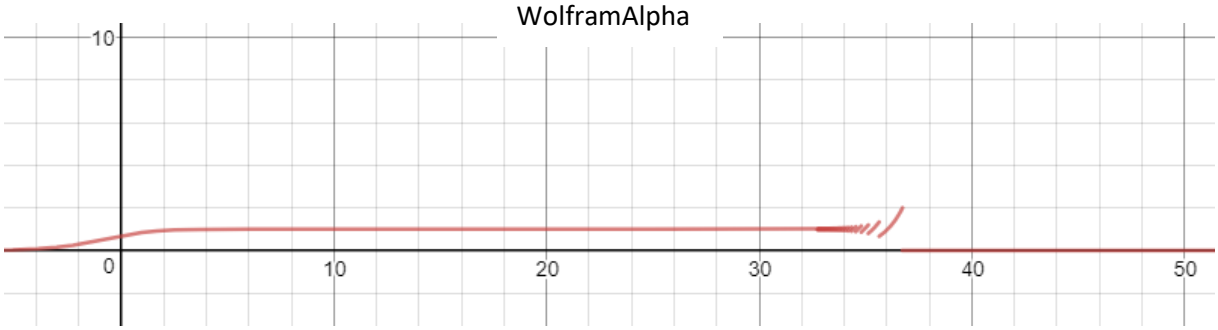
Wykres narysowany w latex

Granica wyliczona $\lim_{n\to\infty}(e^x \ln(1 + e^{-x}) = 1$

W przypadku gdy zwiększymy zakres niestety zaczyna się on psuć:

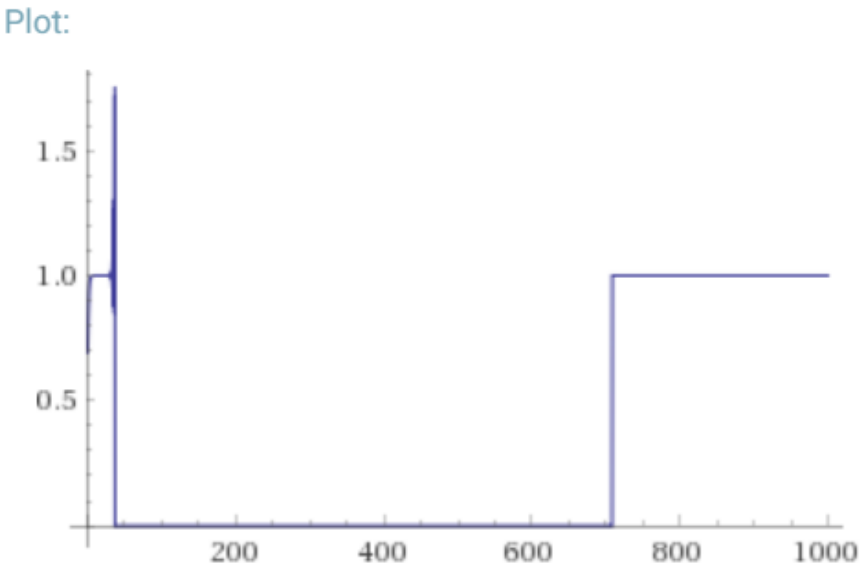


WolframAlpha



www.desmos.com

W obu przypadkach jest to spowodowane tym, że od pewnego momentu (zależnego od użytej pamięci) wynik psuje się ze względu na to, że e^x dąży do nieskończoności, a logarytm dąży do 0. Mnożymy wtedy dwie liczby jedną bardzo dużą, a drugą bardzo małą, co na poprzedniej liście zostało już pokazane, że zwraca to zły wynik, co możemy tu otrzymać. Tak się dzieje do momentu, gdy logarytm wynosi już 0, wtedy zgodnie z tym, że każda liczba przemnożona przez 0 daje 0. Bardzo ciekawy wypadek zauważa się w przypadku użycia wolframalpha, bo po pewnym momencie zwraca on „nagle” poprawny wynik:



Może być to spowodowane tym, że jest on przeznaczony do

wykonywania matematycznych działań i kiedy kończy mu się zakres, a wyliczył, że granica wynosi 1, więc tam po prostu umieszcza wykres.

Zad.3

Zadanie polegało na wyliczeniu błędu względnego dla macierzy Hilberta i losowej macierzy stopnia n dla układu równań liniowych:

$$Ax=b$$

gdzie A- macierz, x = (1,1,…,1). Do rozwiązania należało użyć algorytmu eliminacji Gaussa : $x = \frac{\mathrm{A}}{\mathrm{b}}$ i algorytmu $x = A^{-1}*b$.

Błąd względny liczony za pomocą wzoru:

$$\frac{|x - x'|}{|x|}$$

Wyniki błędu względnego dla macierzy Hilberta:

n	eliminacją Gaussa	$x = A^{-1}*b$
1	0.0	0.0
2	$5.661048867003676e^{-16}$	$1.4043333874306803e^{-15}$
3	$8.022593772267726e^{-15}$	0.0
4	$4.637277712035294e^{-13}$	$7.542470546988852e^{-13}$
5	$1.7697056701418277e^{-13}$	$7.45602798259539e^{-12}$
6	$3.496491467713994e^{-10}$	$3.533151828962887e^{-10}$
7	$1.3175049864850338e^{-8}$	$6.190844397992631e^{-9}$
8	$2.487433466002445e^{-7}$	$3.775275483015941e^{-7}$
9	$9.643625435772316e^{-6}$	$1.1659486044133412e^{-5}$
10	0.00022035288727930986	0.0003357158826776558
11	0.006022512934347414	0.01113776822564549
12	0.19509235225028912	0.16218620232347905
13	7.894191771622431	5.511855154155295
14	0.8270688593203056	3.3522039875276723
15	3.10349386243609	4.354299435453685
16	9.083139658689422	54.189834405860445

17	4.24328971542452	5.786281231941037
18	4.7860299021083	5.7599951815224495
19	6.114994252530053	12.309212980457932
20	19.122235961045973	17.030822563878868

Oraz dla macierzy losowej:

n	c	eliminacją Gaussa	x =A^{-1}*b
5	1	1.4043333874306804e^{-16}	2.482534153247273e^{-16}
5	10	6.807758675344951e^{-16}	6.358389842764979e^{-16}
5	10^3	1.6444198294488168e^{-14}	1.5485919901226002^{-14}
5	10^7	5.661108283973349e^{-10}	6.378986031188388e^{-10}
5	10^{12}	1.3805448628906103e^{-5}	8.529922399520072e^{-6}
5	10^{16}	0.3527640928749535	0.38514911884879083
10	1	3.14018491736755e^{-16}	3.4577699597798515e^{-16}
10	10	5.231024053648945e^{-16}	2.895107444979072e^{-16}
10	10^3	4.271883716981752e^{-14}	4.180528546095446e^{-14}
10	10^7	9.428734784179054e^{-11}	1.9573543111939664e^{-10}
10	10^{12}	1.014257908740652e^{-5}	9.987362531292952e^{-6}
10	10^{16}	0.022016453352644925	0.06106756951832125 6
20	1	6.15147718416645e^{-16}	5.67735533054474e^{-16}
20	10	8.752473334317946e^{-16}	7.288679687431566e^{-16}
20	10^3	9.920139380631545e^{-15}	6.8799796150562484e^{-15}
20	10^7	2.8281296078591995e^{-10}	2.0921383211768281e^{-10}
20	10^{12}	3.059617104941452e^{-6}	5.023791195322389e^{-6}
20	10^{16}	0.029125868925812037	0.09254185734751666

Zwłaszcza dla macierzy losowej widać, że zamiast na wielkość macierzy trzeba zwracać na uwarunkowanie. Jednak im większa macierz tym większe odchylenie od prawidłowego wyniku.

Zad.4

Używając funkcji root do obliczenia zer wielomianu w dwóch postaciach:

$P(x) = x^{20}-210x^{19} + \ldots$

$p(x) = (x-20)(x-19)\ldots(x-1)$

Pierwiastki zapisane zostały do tablicy z

k	$P(z_k)$	$p(z_k)$	$\lvert z_k-k\rvert$
1	36352.0	38400.0	$3.0109248427834245e^{-13}$
2	181760.0	198144.0	$2.8318236644508943e^{-11}$
3	209408.0	301568.0	$4.0790348876384996e^{-10}$
4	$-3.106816e^{6}$	$-2.844672e^{6}$	$1.626246826091915e^{-8}$
5	$-2.4114688e^{6}$	$-2.3346688e^{7}$	$6.657697912970661e^{-7}$
6	$-1.20152064e^{8}$	$-1.1882496e^{8}$	$1.0754175226779239e^{-5}$
7	$-4.80398336e^{8}$	$-4.78290944e^{8}$	0.00010200279300764947
8	$-1.682691072e^{9}$	$-1.67849728e^{9}$	0.0006441703922384079
9	$-4.465326592e^{9}$	$-4.457859584e^{9}$	0.002915294362052734
10	$-1.2707126784e^{10}$	$-1.2696907264e^{10}$	0.009586957518274986
11	$-3.5759895552e^{10}$	$-3.5743469056e^{10}$	0.025022932909317674
12	$-7.216771584e^{10}$	$-7.2146650624e^{10}$	0.04671674615314281
13	$-2.15723629056e^{11}$	$-2.15696330752e^{11}$	0.07431403244734014
14	$-3.65383250944e^{11}$	$-3.653447936e^{11}$	0.08524440819787316
15	$-6.13987753472e^{11}$	$-6.13938415616e^{11}$	0.07549379969947623
16	$-1.555027751936e^{12}$	$-1.554961097216e^{12}$	0.05371328339202819
17	$-3.777623778304e^{12}$	$-3.777532946944e^{12}$	0.025427146237412046
18	$-7.199554861056e^{12}$	$-7.1994474752e^{12}$	0.009078647283519814
19	$-1.0278376162816e^{13}$	$-1.0278235656704e^{13}$	0.0019098182994383706
20	$-2.7462952745472e^{13}$	$-2.7462788907008e^{13}$	0.0001907087633625792 5

Następnie powtarzamy eksperyment zmieniając współczynnik z - 210 na $-210-2^{-23}$. Warto zauważyć jak drobna zmiana jest to w zapisie binarnym:

11000000011010100100000000000000000000000**0**00000000000000000000000
11000000011010100100000000000000000000000**1**00000000000000000000000

k	$P(z_k)$	$p(z_k)$	$\|z_k-k\|$
1	20992.0	38400.0	$1.6431300764452317e^{-13}$
2	349184.0	198144.0	$5.503730804434781e^{-11}$
3	$2.221568e^{6}$	301568.0	$3.3965799062229962e^{-9}$
4	$1.046784e^{7}$	$-2.844672e^{6}$	$8.972436216225788e^{-8}$
5	$3.9463936e^{7}$	$-2.3346688e^{7}$	$1.4261120897529622e^{-6}$
6	$1.29148416e^{8}$	$-1.1882496e^{8}$	$2.0476673030955794e^{-5}$
7	$3.88123136e^{8}$	$-4.78290944e^{8}$	0.00039792957757978087
8	$1.072547328e^{9}$	$-1.67849728e^{9}$	0.0077720290994456 32
9	$3.065575424e^{9}$	$-4.457859584e^{9}$	0.0841836320674414
10	$6.226891264e^{9}$	$-1.2696907264e^{10}$	0.0954556305357741
11	$6.226891264e^{9}$	$-3.5743469056e^{10}$	0.9045443694642259
12	$1.2597167616e^{10}$	$-7.2146650624e^{10}$	0.2061094138256312
13	$1.2597167616e^{10}$	$-2.15696330752e^{11}$	1.2061094138256312
14	$6.2258671616e^{10}$	$-3.653447936e^{11}$	0.00759331551278386
15	$6.2258671616e^{10}$	$-6.13938415616e^{11}$	1.0075933155127839
16	$3.30830896128e^{11}$	$-1.554961097216e^{12}$	0.7307448797926703
17	$3.30830896128e^{11}$	$-3.777532946944e^{12}$	0.26925512020732967
18	$2.046021647872e^{12}$	$-7.1994474752e^{12}$	1.5024423688181017
19	$2.046021647872e^{12}$	$-1.0278235656704e^{13}$	0.5024423688181017
20	$1.114453504512e^{13}$	$-2.7462788907008e^{13}$	0.8469102151947894

Jak widać nawet minimalne odchylenie przekształca w bardzo duży wynik końcowy. W tym wielomianie zamiast zer otrzymujemy wyniki rzędu bilionów. Natomiast drobna zmiana powoduje ogromne różnice w wynikach między dwoma wielomianami (w obu nie uzyskujemy zer). To przypadek źle uwarunkowanego zadania. Warto

pamiętać, że wyniki zależą od ilości cyfr znaczących w przypadku arytmetyki Float64 wynosi ona od 15 do 17 cyfr znaczących.

Zad.5

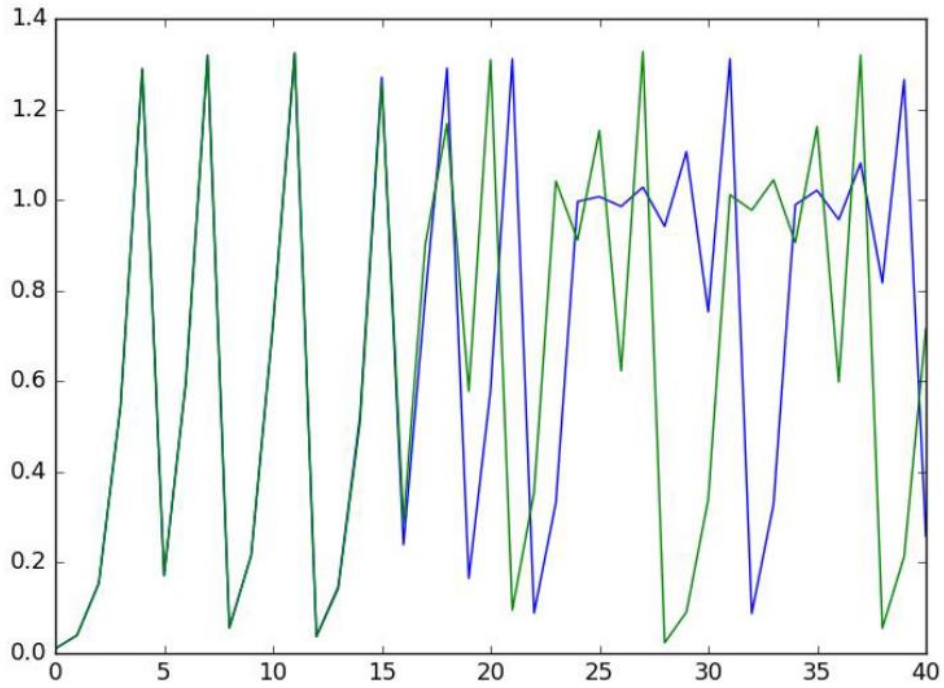
Zadanie polegało na przeprowadzeniu symulacji w dwóch wariantach mając model logistyczny:

$p_{n+1}=p_n + rp_n(1-p_n)$

i dane startowe: $p_0=0.01$ i r =3.

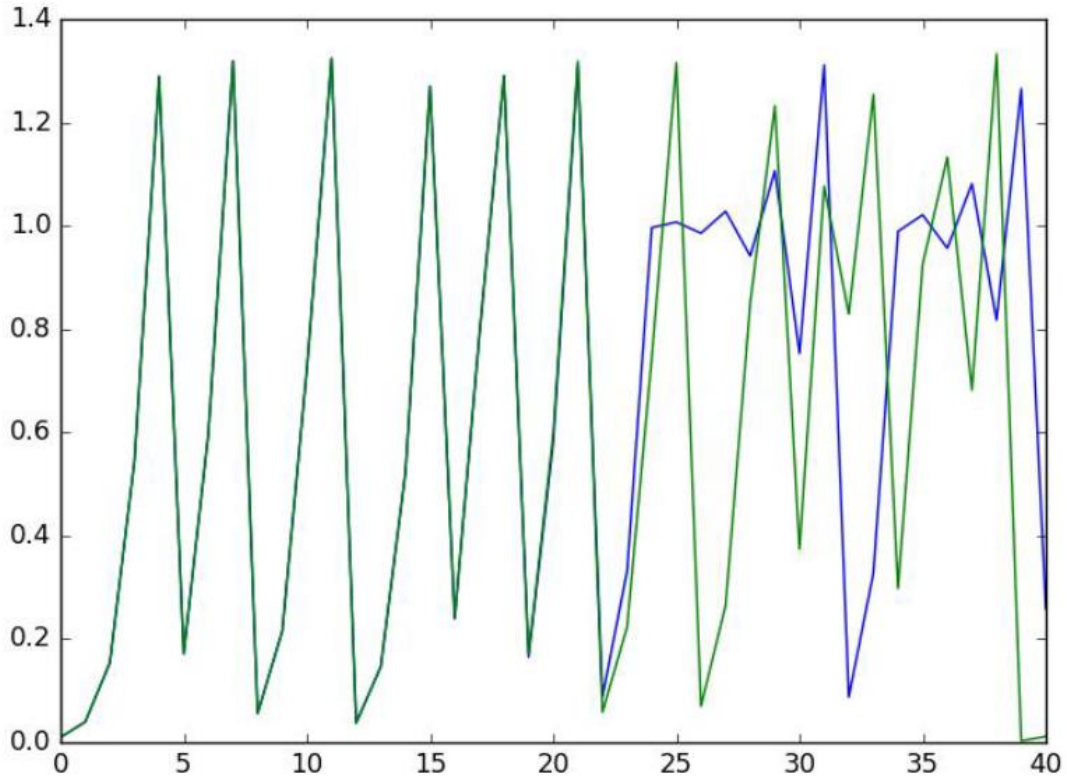
W pierwszej porównuje się arytmetykę Float32 i Float64, a w drugiej w arytmetyce Float32 porównuje się normalne liczenie z modyfikacją danych.

Analizę wyników porównam na wykresach ze względu na czytelność:



W przypadku gdy ucinamy dane (niebieska linia) widać, że zmiany mają wpływ już przy niewielkiej iteracji w stosunku do danych

niezmienianych(zielona linia). Ledwo zauważalny błąd wraz z kolejnymi iteracjami powoduje bardzo szybkie zmiany w wyniku. To przykład sprzężenia zwrotnego, czyli procesu, w którym dane wyjściowe są danymi wejściowymi do następnych obliczeń. Każdy błąd zostaje przeniesiony i spotęgowany. W przypadku porównywania arytmetyki Float32 i Float64 otrzymujemy:



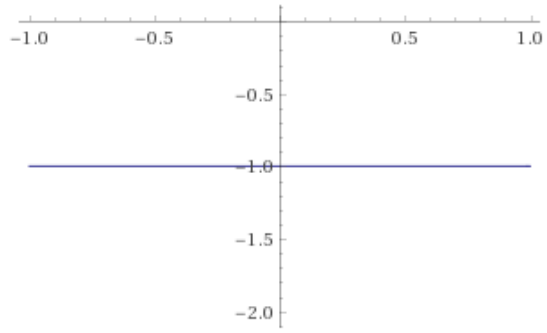
Różnice nie są tak zauważalne jak w poprzednim przypadku. Jednak i tutaj widać, że błędy na wyjściu są potęgowane przy kolejnych iteracjach w tym wypadku ze względu na precyzję obliczeń. W tym wykresie zieloną linią jest wykres z wyników uzyskanych z Float32, a niebieską z Float64. Widać, że gdy Float32 napotyka wynik zbliżony do 1.0 powoduje oscylowanie przy tym wyniku przy następnych obliczeniach. Choć stosowanie większej precyzji może pomóc, jednak po pewnym momencie błędy zniekształcają wynik do tego stopnia, że wyniki są bezużyteczne.

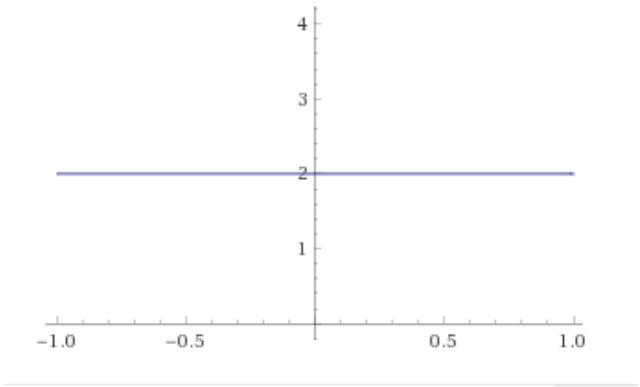
Zad.6

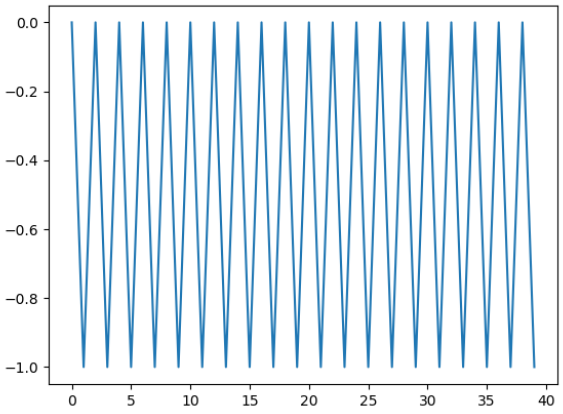
Rozważenie równania rekurencyjnego:

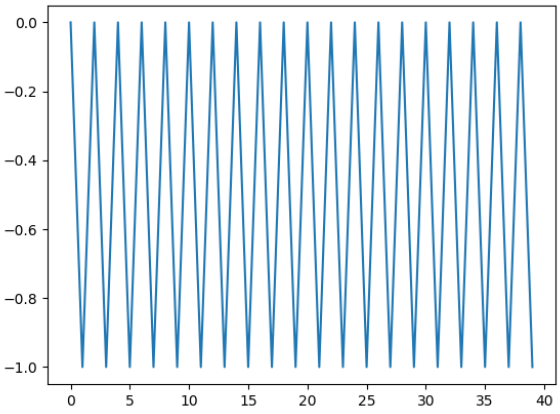
$$x_{n+1} = x_n^2+c$$

Dla przypadków 1,2,4,5 wyniki są przewidywalne – zgodne z wynikami:



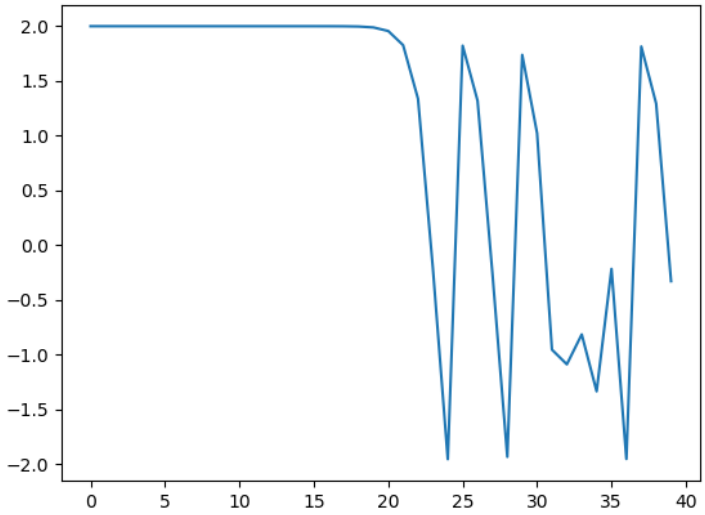




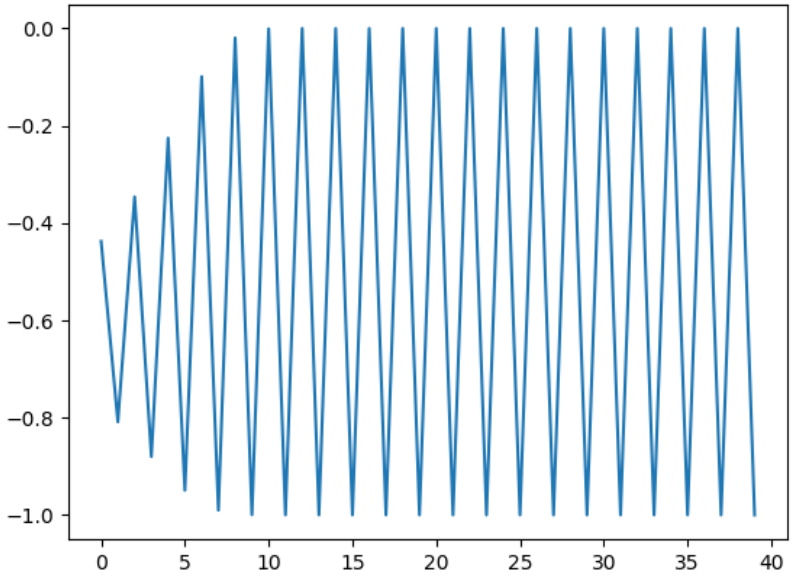


Dopiero w przypadku 3, 6 i 7 można zauważyć, że zachodzi chaos deterministyczny, gdzie błędy zaczynają się nawarstwiać, a za każdą iteracją błąd się zwiększa, bo błąd z wyjścia jest przenoszony do wejścia kolejnej iteracji:

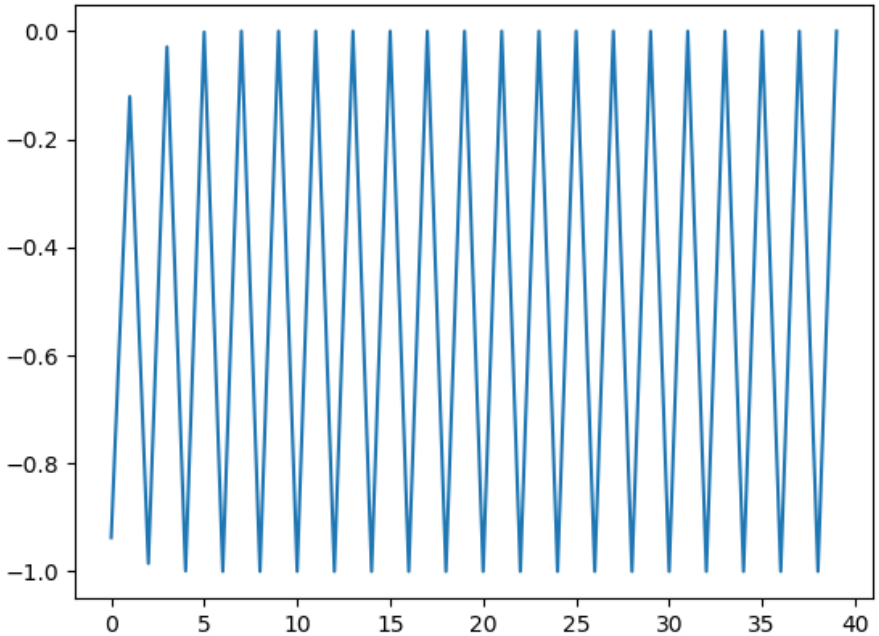
3.


2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0
5
10
15
20
25
30
35
40

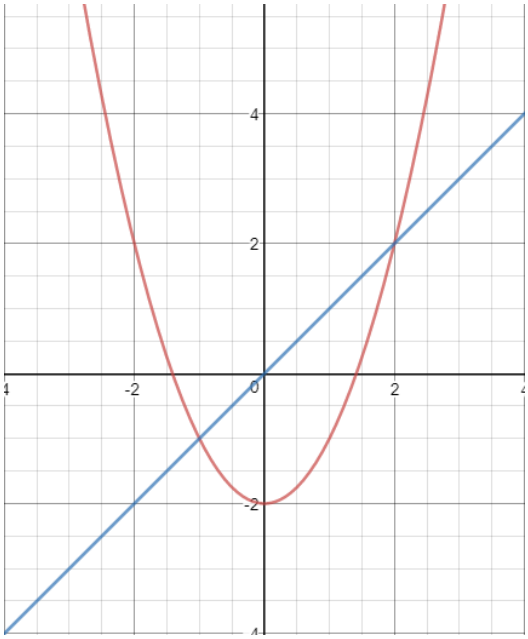
6.


0.0
−0.2
−0.4
−0.6
−0.8
−1.0
0
5
10
15
20
25
30
35
40

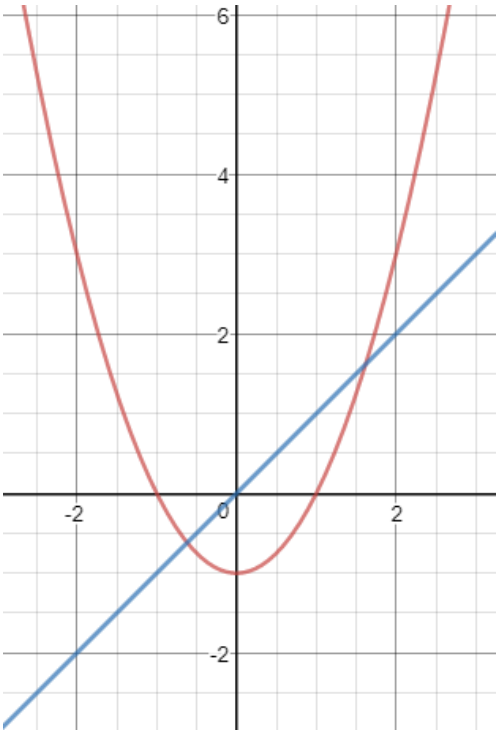
7.


0.0
−0.2
−0.4
−0.6
−0.8
−1.0
0
5
10
15
20
25
30
35
40

Można zauważyć, że niestabilność nie wynika z podnoszenia do kwadratu (pierwsze dwa eksperymenty), ale wartości początkowe mogą prowadzić do zachowań stabilnych. Warto w tym momencie zauażyć jak to wygląda przy funkcji $\Phi=x^2-2$



Na wykresie obok przedstawiono wykres funkcji $f(x) = x^2-2$ i $f(x) =x$.
Można zauważyć, że dla $x_0=1$ $\Phi(x)$ jest zbieżne do -1,
dla $x_0=2$ zbieżne do 2, a dla $x_0=1.99999999999999$ jest rozbieżna.



Można zauważyć, że dla $x_0=1$ v $x_0= -1$ podciąg wyrazów nieparzystych zbiega do 0, a parzystych do -1.
Dla $x_0=0.75$ i $x_0=0.25$ jest odwrotnie.