

Obliczenia Naukowe

Lista nr 2

Eryk Krupa

244993

Zadanie 1

Pierwsza para wektorów (A):

x	2.718281828	-3.141592654	1.414213562	0.5772156649	0.3010299957
y	1486.2497	878366.9879	-22.37492	4773714.647	0.000185049

W drugiej parze wektorów (B) wartości pogrubione zostały usunięte.

Poniższe tabele przedstawiają sumy oraz błędy bezwzględne obliczone dla obu par wektorów w dwóch precyzjach Float32 i Float64 dla czterech algorytmów sumowania:

- A. w przód,
- B. w tył,
- C. od największego do najmniejszego,
- D. od najmniejszego do największego.

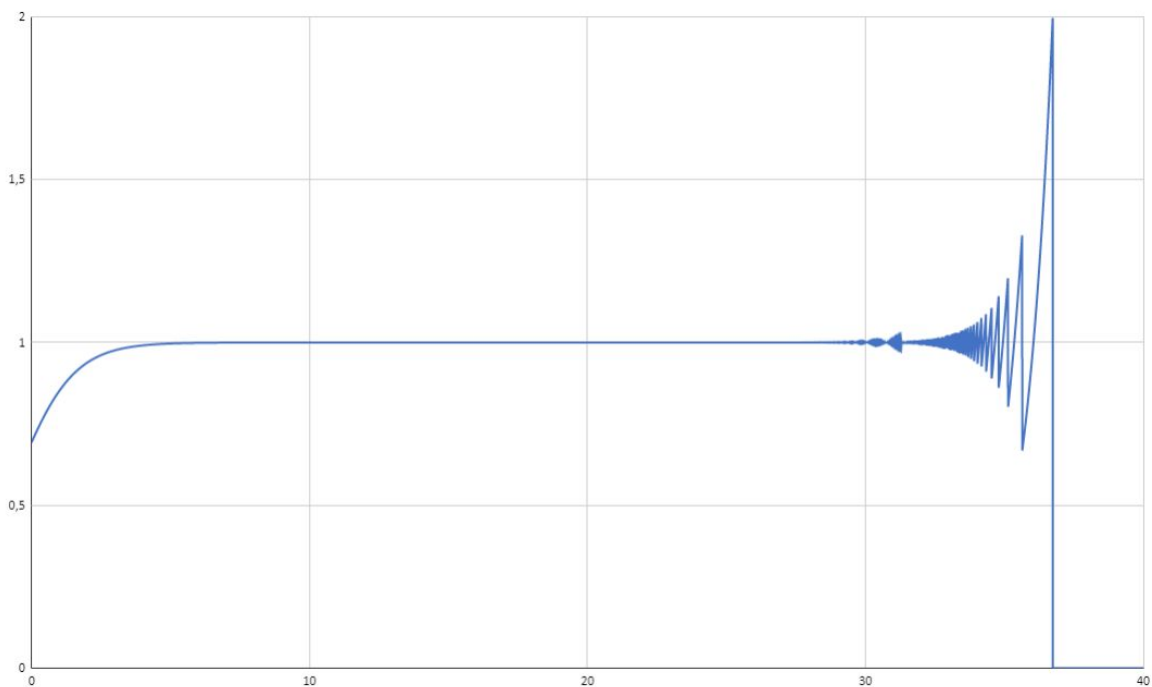
Float32		
Algorytm	Wektory A	Wektory B
"w przód"	-0.4999443	-0.4999443
"w tył"	-0.4543457	-0.4543457
"od największego do najmniejszego"	-0.25	-0.25
"od najmniejszego do największego"	-0.25	-0.25

Float64		
Algorytm	Wektory A	Wektory B
"w przód"	1.0251881368296672e-10	-0.004296342739891585
"w tył"	-1.5643308870494366e-10	-0.004296342998713953
"od największego do najmniejszego"	0.0	-0.004296342842280865
"od najmniejszego do największego"	0.0	-0.004296342842280865

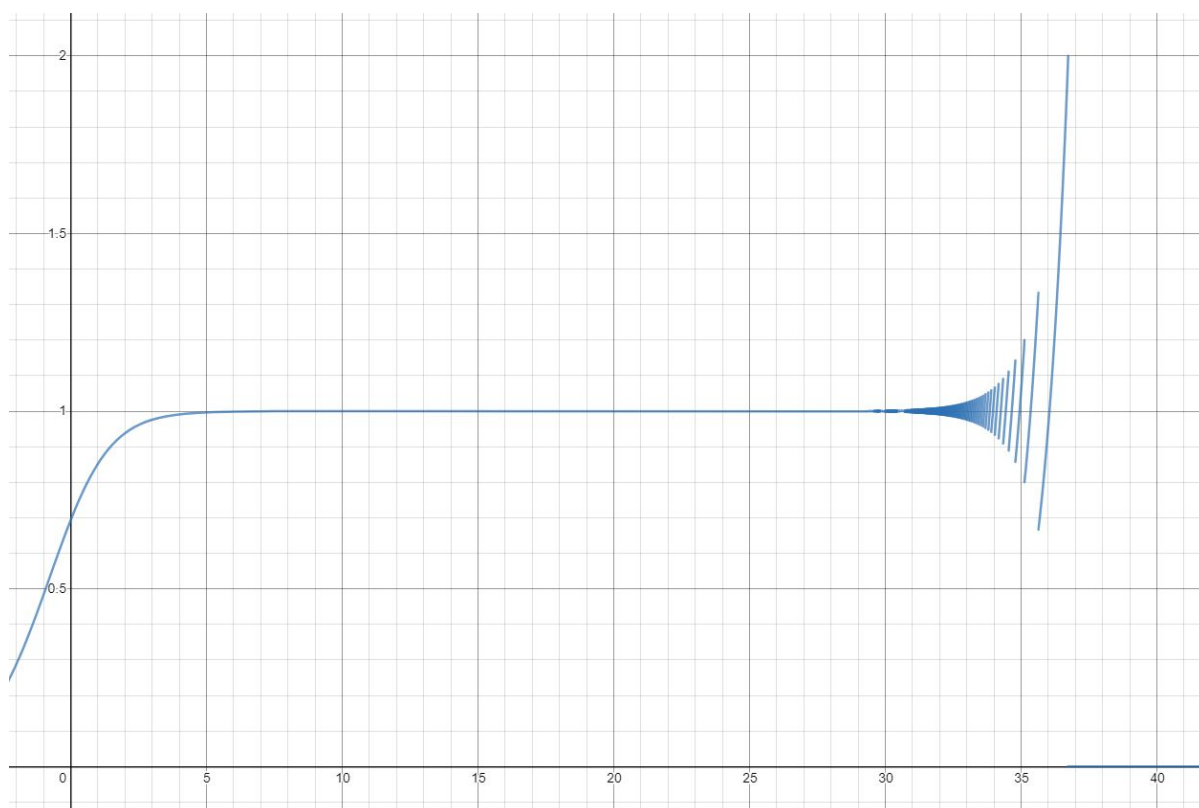
Można zauważyć, że wyniki dla Float32 są takie same dla obu par wektorów. Wynika to z faktu, że usunięte cyfry stały na tak mało znaczących pozycjach, że dla arytmetyki Float32 ich nieobecność nie robiła różnicy- i tak były usuwane. Z kolei w przypadku Float64 wyniki zmieniły się znacznie- stały się dokładniejsze, gdyż usunięcie ostatnich cyfr zmniejszyło błąd reprezentacji.

Zadanie 2

Wykres funkcji $f(x) = e^x \cdot \ln(1 + e^{-x})$ w dwóch programach do wizualizacji:



Wykres wygenerowany w programie Excel.



Wykres wygenerowany w programie Desmos.

Oba wykresy ujawniają nieoczekiwany sposób zachowania się funkcji dla x pomiędzy 30 a 40. Zachowanie to jest różne dla obu programów. Wykresy sugerują, że granicą funkcji jest równa 0. Nie jest to prawdą, ponieważ wartość granicy funkcji jest równa 1, co wynika z poniższego równania:

$$\lim_{x \rightarrow \infty} f(x) = \lim_{x \rightarrow \infty} e^x \cdot \ln(1 + e^{-x}) = \lim_{x \rightarrow \infty} \frac{1}{1 + e^{-x}} = 1$$

Przyjrzyjmy się dokładniej funkcji $f(x) = e^x \cdot \ln(1 + e^{-x})$. Wraz ze wzrostem wartości x , wzrasta różnica pomiędzy mnożonymi czynnikami e^x i $\ln(1 + e^{-x})$. Z tego powodu, funkcja zaczyna zachowywać się nienaturalnie dla $x > 30$. Dodatkowo, w pewnym momencie $1 + e^{-x}$ zaczyna być równe 1, gdyż 1 dodawane jest do coraz mniejszych wartości, przez co w pewnym momencie następuje pochłonięcie e^{-x} . Wynikiem logarytmu jak i całego równania staje się wtedy 0. Jest to zatem zadanie źle uwarunkowane.

Zadanie 3

Poniżej znajdują się błędy względne policzone za pomocą metody eliminacji Gaussa oraz metody inwersji dla macierzy Hilberta oraz macierzy losowej.

Macierz Hilberta			
Rozmiar	Rząd	Błąd względny (Gauss)	Błąd względny (Inwersja)
2	2	5.661048867003676e-16	1.4043333874306803e-15
3	3	8.022593772267726e-15	0.0
4	4	4.137409622430382e-14	0.0
5	5	1.6828426299227195e-12	3.3544360584359632e-12
6	6	2.618913302311624e-10	2.0163759404347654e-10
7	7	1.2606867224171548e-8	4.713280397232037e-9
8	8	6.124089555723088e-8	3.07748390309622e-7
9	9	3.8751634185032475e-6	4.541268303176643e-6
10	10	8.67039023709691e-5	0.0002501493411824886
11	10	0.00015827808158590435	0.007618304284315809
12	11	0.13396208372085344	0.258994120804705
13	11	0.11039701117868264	5.331275639426837
14	11	1.4554087127659643	8.71499275104814
15	12	4.696668350857427	7.344641453111494

Wartość błędu względnego dla macierzy Hilberta zależy od wskaźnika uwarunkowania. Można zauważyć, że im większy wskaźnik uwarunkowania tym większy błąd otrzymujemy. Widać, że metoda eliminacji Gaussa okazała się dokładniejsza. Macierz Hilberta jest źle uwarunkowana, przez co występują duże błędy.

Macierz losowa				
Rozmiar	Wsk. uw.	Rząd	Błąd względny (Gauss)	Błąd względny (Inwersja)
5	1.0	5	1.9860273225978183e-16	1.4895204919483638e-16
5	10.0	5	1.4043333874306804e-16	2.6737711109153337e-16
5	1000.0	5	1.6384725411432002e-15	3.891802844472395e-15
5	1.0e7	5	2.3845089422863355e-10	1.4840099941918116e-10

5	1.0e12	5	5.934902918663049e-6	3.51697710315433e-6
5	1.0e16	4	0.05917866059811854	0.06337281184585389
10	1.0	10	1.6467268631127714e-16	1.8242825833464351e-16
10	10.0	10	2.1355566272775288e-16	2.1925147983971603e-16
10	1000.0	10	1.7541664323074608e-14	2.2490827773791327e-14
10	1.0e7	10	1.249133552555659e-10	1.2253023704409647e-10
10	1.0e12	10	2.5748849213512094e-5	3.238715332123902e-5
10	1.0e16	9	0.21997136654358102	0.2573622963110564
20	1.0	20	4.47545209131181e-16	3.394814396577995e-16
20	10.0	20	4.124295487574583e-16	3.394814396577995e-16
20	1000.0	20	2.1086859037614256e-14	1.6876558589654652e-14
20	1.0e7	20	5.367000702677506e-10	5.219390330448221e-10
20	1.0e12	20	5.207124451586607e-5	4.9352674668529415e-5
20	1.0e16	19	0.11843516242578418	0.12833681044058443

W przypadku macierzy losowej sytuacja wygląda podobnie. Wartości błędów względnych rosną wraz ze wzrostem wskaźników uwarunkowania. Dane sugerują, że wskaźnik uwarunkowania jest istotniejszy nawet od wielkości macierzy. Oznacza to, że prowadząc obliczenia na macierzy źle uwarunkowanej należy spodziewać się dużych błędów.

Zadanie 4

Poniższa tabela prezentuje obliczone miejsca zerowe dla wielomianu Wilkinsona:

k	z_k	$ P(z_k) $	$ p(z_k) $	$ z_k - k $
1	0.99999999999996989	36352.0	5.517824e6	3.0109248427834245e-13
2	2.0000000000283182	181760.0	7.378697629901744e19	2.8318236644508943e-11
3	2.9999999995920965	209408.0	3.320413931687578e20	4.0790348876384996e-10
4	3.9999999837375317	3.106816e6	8.854437035384718e20	1.626246826091915e-8
5	5.000000665769791	2.4114688e7	1.8446752056545675e21	6.657697912970661e-7

6	5.999989245824773	1.20152064e8	3.320394888870126e21	1.0754175226779239e-5
7	7.000102002793008	4.80398336e8	5.423593016891272e21	0.00010200279300764947
8	7.999355829607762	1.682691072e9	8.26205014011023e21	0.0006441703922384079
9	9.002915294362053	4.465326592e9	1.196559421646318e22	0.002915294362052734
10	9.990413042481725	1.2707126784e10	1.6552601335207813e22	0.009586957518274986
11	11.025022932909318	3.5759895552e10	2.2478332979247994e22	0.025022932909317674
12	11.953283253846857	7.216771584e10	2.8869446884129956e22	0.04671674615314281
13	13.07431403244734	2.15723629056e11	3.807325552825022e22	0.07431403244734014
14	13.914755591802127	3.65383250944e11	4.612719853149547e22	0.08524440819787316
15	15.075493799699476	6.13987753472e11	5.901011420239329e22	0.07549379969947623
16	15.946286716607972	1.555027751936e12	7.01087410689741e22	0.05371328339202819
17	17.025427146237412	3.777623778304e12	8.568905825727875e22	0.025427146237412046
18	17.99092135271648	7.199554861056e12	1.0144799361089491e23	0.009078647283519814
19	19.00190981829944	1.0278376162816e13	1.1990376202486947e23	0.0019098182994383706
20	19.999809291236637	2.7462952745472e13	1.4019117414364248e23	0.00019070876336257925

Gdyby pierwiastki były obliczone dokładnie, w trzeciej i w czwartej kolumnie otrzymalibyśmy same zera. Tak się jednak nie dzieje, co więcej, są tam bardzo wysokie wartości, mimo tego, że popełnione błędy bezwzględne są bardzo bliskie zeru. Dla kolejnych wartości odstępstwo od zera rośnie, mimo tego, że błąd bezwzględny maleje. Błąd wynika m. in. z gigantycznych wartości w wielomianie naturalnym, których nie da się dokładnie przechować we Float64, mimo tego, że są to wartości całkowite. Wielomian w postaci iloczynowej zawiera jeszcze większe błędy. Wynikają one z tego, że nawet najmniejsze odchylenia od poprawnych wartości rosną do ogromnych rozmiarów, gdyż są wielokrotnie mnożone przez czynniki wielomianu.

Poniższa tabela prezentuje obliczone miejsca zerowe w eksperymencie Wilkinsona:

k	zk	P(zk)	p(zk)	zk-k
1	0.999999999998357	20992.0	3.012096e6	1.6431300764452317e-13
2	2.0000000000550373	349184.0	7.37869763029606e19	5.503730804434781e-11
3	2.99999999660342	2.221568e6	3.3204139201100146e20	3.3965799062229962e-9
4	4.000000089724362	1.046784e7	8.854437817429645e20	8.972436216225788e-8
5	4.99999857388791	3.9463936e7	1.8446726974084148e21	1.4261120897529622e-6

6	6.00002047667303	1.29148416e8	3.320450195282314e21	2.0476673030955794e-5
7	6.99960207042242	3.88123136e8	5.422366528916045e21	0.00039792957757978087
8	8.007772029099446	1.072547328e9	8.289399860984229e21	0.007772029099445632
9	8.915816367932559	3.065575424e9	1.1607472501770085e22	0.0841836320674414
10	10.095455630535774	7.143113638035824e9	1.7212892853671066e22	0.6519586830380406
11	10.095455630535774	7.143113638035824e9	1.7212892853671066e22	1.1109180272716561
12	11.793890586174369	3.357756113171857e10	2.8568401004080516e22	1.665281290598479
13	11.793890586174369	3.357756113171857e10	2.8568401004080516e22	2.045820276678428
14	13.992406684487216	1.0612064533081976e11	4.934647147685479e22	2.5188358711909045
15	13.992406684487216	1.0612064533081976e11	4.934647147685479e22	2.7128805312847097
16	16.73074487979267	3.315103475981763e11	8.484694713574187e22	2.9060018735375106
17	16.73074487979267	3.315103475981763e11	8.484694713574187e22	2.825483521349608
18	19.5024423688181	9.539424609817828e12	1.318194782057474e23	2.454021446312976
19	19.5024423688181	9.539424609817828e12	1.318194782057474e23	2.004329444309949
20	20.84691021519479	1.114453504512e13	1.591108408283123e23	0.8469102151947894

Zadanie to jest źle uwarunkowane, ponieważ w przypadku eksperymentu Wilkinsona niektóre pierwiastki uzyskują część urojoną.

Zadanie 5

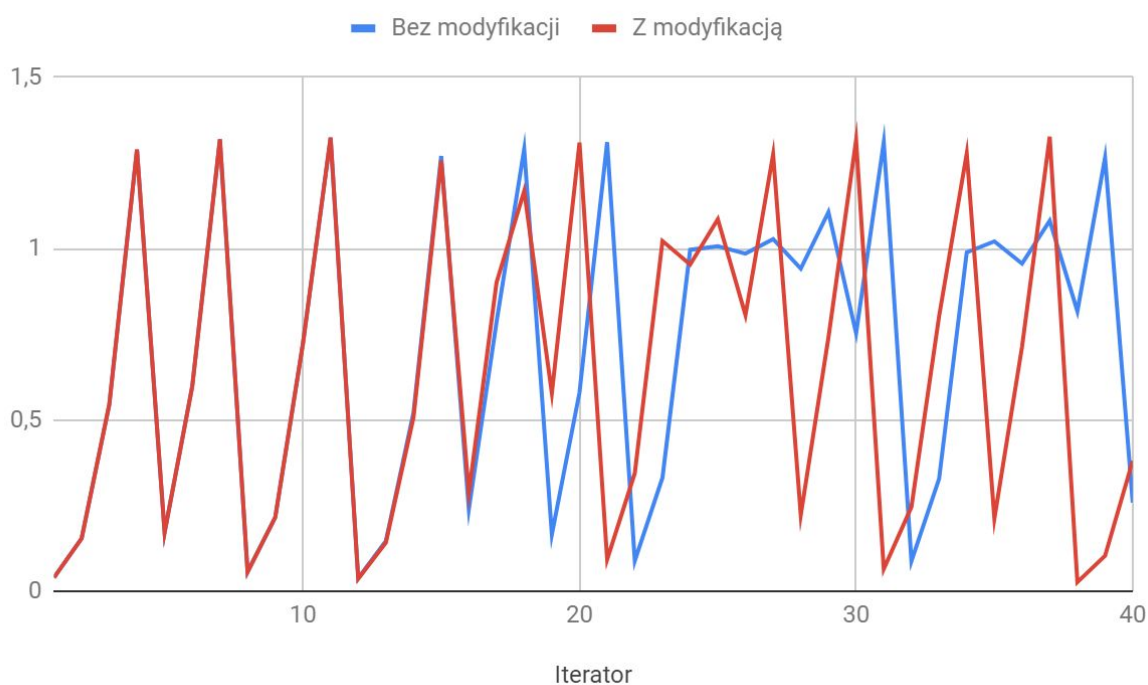
Rozważamy model populacji opisywany równaniem:

$$p_{n+1} = p_n + r p_n (1 - p_n), \text{ dla } n = 0, 1, 2, \dots,$$

dla podanej stałej $r = 3$ i $p_0 = 0.01$. Poniższa tabela oraz wykres przedstawiają pierwsze 40 iteracji, wraz z przypadkiem, gdzie zmodyfikowano wynik dziesiątej iteracji- obcięto czwarte miejsce po przecinku.

Iteracja	Bez modyfikacji	Z modyfikacją	Różnica
1.	0.0397	0.0397	0.0
2.	0.15407173	0.15407173	0.0
3.	0.5450726	0.5450726	0.0
4.	1.2889781	1.2889781	0.0
5.	0.1715188	0.1715188	0.0
6.	0.5978191	0.5978191	0.0
7.	1.3191134	1.3191134	0.0
8.	0.056273222	0.056273222	0.0
9.	0.21559286	0.21559286	0.0
10.	0.7229306	0.722	0.0009306101799011479
11.	1.3238364	1.324148	0.00031161308
12.	0.037716985	0.036488056	0.0012289286
13.	0.14660022	0.14195809	0.004642129
14.	0.521926	0.5073761	0.014549911
15.	1.2704837	1.2572129	0.013270855
16.	0.2395482	0.28709882	0.04755062
17.	0.7860428	0.9011181	0.11507529
18.	1.2905813	1.1684309	0.12215042
19.	0.16552472	0.5780312	0.41250646
20.	0.5799036	1.3097646	0.729861
21.	1.3107498	0.09260833	1.2181414
22.	0.088804245	0.34470442	0.25590017
23.	0.3315584	1.0223544	0.69079596
24.	0.9964407	0.9537921	0.042648613
25.	1.0070806	1.0860103	0.07892978
26.	0.9856885	0.805786	0.1799025
27.	1.0280086	1.2752707	0.24726212

28.	0.9416294	0.22213674	0.7194927
29.	1.1065198	0.7405127	0.3660071
30.	0.7529209	1.3169736	0.56405264
31.	1.3110139	0.06463623	1.2463777
32.	0.0877831	0.24601139	0.1582283
33.	0.3280148	0.80248076	0.47446597
34.	0.9892781	1.277997	0.28871894
35.	1.021099	0.21215892	0.80894005
36.	0.95646656	0.71360147	0.24286509
37.	1.0813814	1.3267246	0.24534321
38.	0.81736827	0.026303649	0.7910646
39.	1.2652004	0.10313895	1.1620615
40.	0.25860548	0.3806429	0.12203741

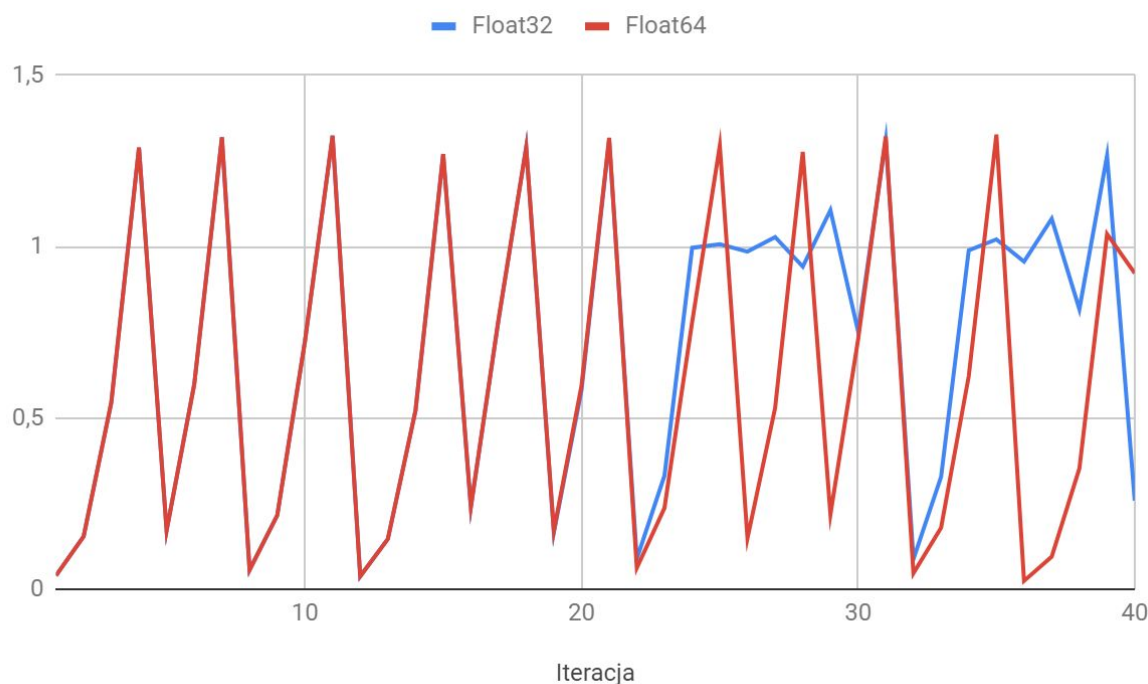


Na powyższym wykresie widać, że około 10. iteracji wyniki zaczynają się delikatnie “rozjeżdżać”. Kilkanaście iteracji później nie można już zauważyć żadnych zależności między wynikami. Drobna zmiana sprawiła, że wyniki całkowicie odbiegają od wartości prawdziwych.

Poniżej przedstawiono wyniki tej samej funkcji, ale w różnych arytmetykach zmiennopozycyjnych.

Iteracja	Float32	Float64	Różnica
1	0,0397	0,03970000148	0
2	0,15407173	0,1540717356	2,21E-09
3	0,5450726	0,5450726431	2,80E-08
4	1,2889781	1,288978014	8,62E-08
5	0,1715188	0,1715190955	2,93E-07
6	0,5978191	0,5978199817	8,92E-07
7	1,3191134	1,319113735	3,61E-07
8	0,056273222	0,05627180146	1,42E-06
9	0,21559286	0,2155876589	5,20E-06
10	0,7229306	0,7229165197	1,41E-05
11	1,3238364	1,323841195	4,75E-06
12	0,037716985	0,03769824951	1,87E-05
13	0,14660022	0,146529524	7,07E-05
14	0,521926	0,5217053918	0,0002205939976
15	1,2704837	1,27029202	0,0001917125096
16	0,2395482	0,2402426328	0,0006944264796
17	0,7860428	0,7878209634	0,001778153892
18	1,2905813	1,289298242	0,00128310306
19	0,16552472	0,1703230957	0,004798374521
20	0,5799036	0,5942625119	0,01435890932
21	1,3107498	1,317606248	0,006856479247
22	0,088804245	0,06216631591	0,02663792909
23	0,3315584	0,2370713111	0,09448709523
24	0,9964407	0,7796768248	0,2167638838
25	1,0070806	1,295019446	0,2879388908
26	0,9856885	0,1488516882	0,8368368193
27	1,0280086	0,5289362777	0,4990723025
28	0,9416294	1,276424353	0,3347949434
29	1,1065198	0,2179200246	0,8885997937
30	0,7529209	0,7292126869	0,02370823868

31	1,3110139	1,321597319	0,01058338238
32	0,0877831	0,04653085375	0,04125224447
33	0,3280148	0,1796280539	0,1483867371
34	0,9892781	0,6217135025	0,3675645756
35	1,021099	1,327270972	0,3061720011
36	0,95646656	0,02413918698	0,9323273686
37	1,0813814	0,09480864686	0,9865727933
38	0,81736827	0,3522685489	0,4650997201
39	1,2652004	1,036794804	0,2284055726
40	0,25860548	0,9223488193	0,6637433391

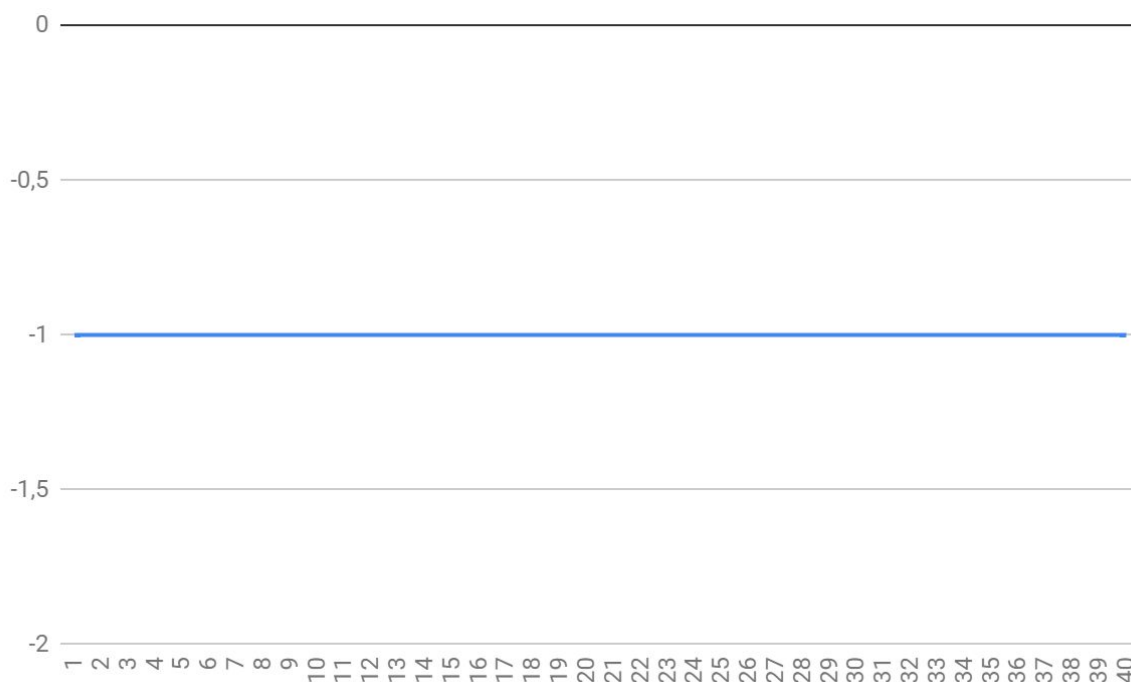


W kolejnym przypadku mamy klasyczną sytuację, gdzie rozbieżności w wynikach wynikają z różnicy w sposobie przechowywania liczb. Na początku, kiedy wartości można bez straty precyzji zapisać w obu arytmetykach, nie ma różnicy w wynikach. Wraz z kolejnymi iteracjami wyniki zaczynają od siebie odbiegać.

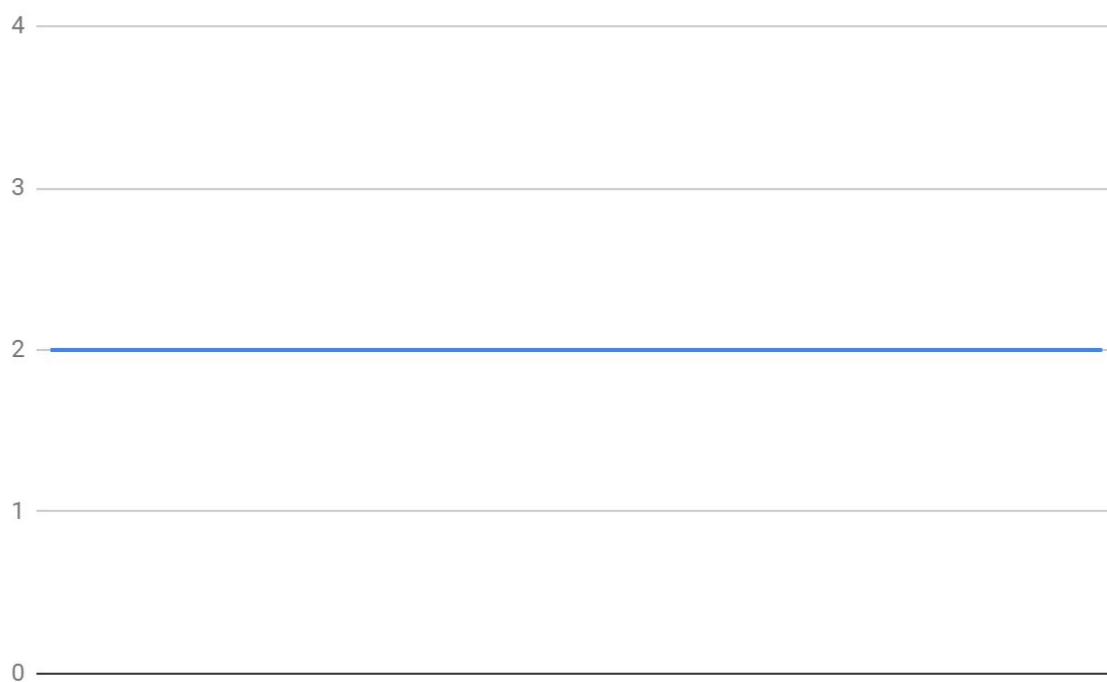
W obu powyższych przypadkach drobne błędy kumulują się i w sposób znaczny wpływają na wynik.

Zadanie 6

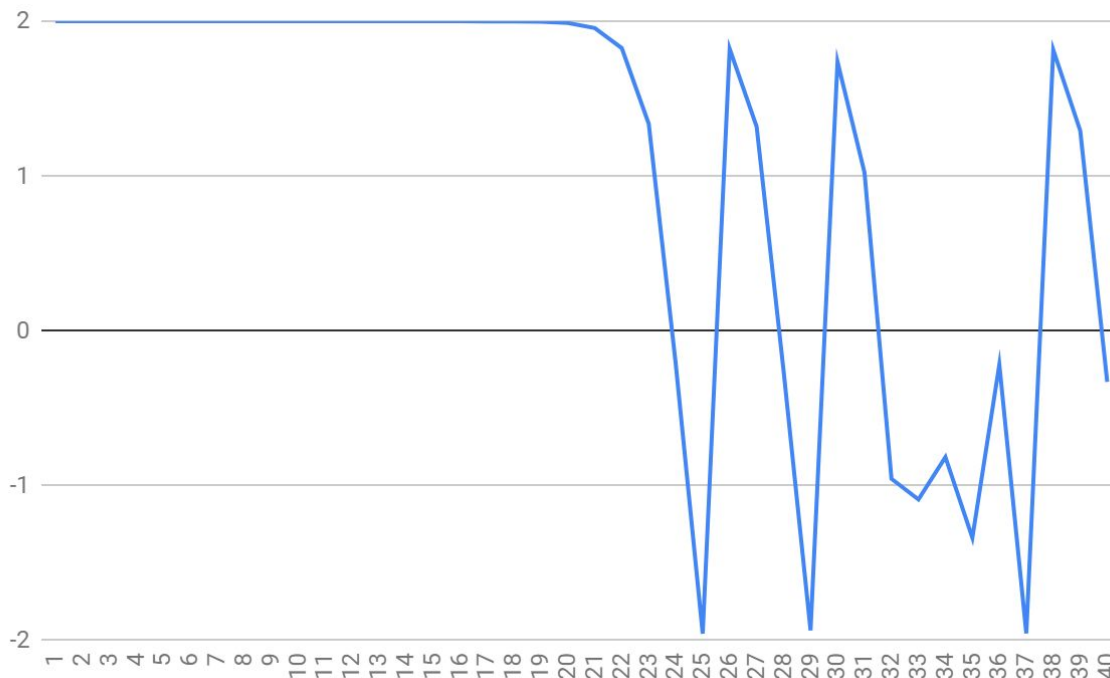
Poniżej znajduje się 7 wykresów dla siedmiu różnych par wartości c oraz x_n oraz równania: $x_{n+1} := x_n^2 + c$, dla $n = 0, 1, 2, \dots, 39, 40$.



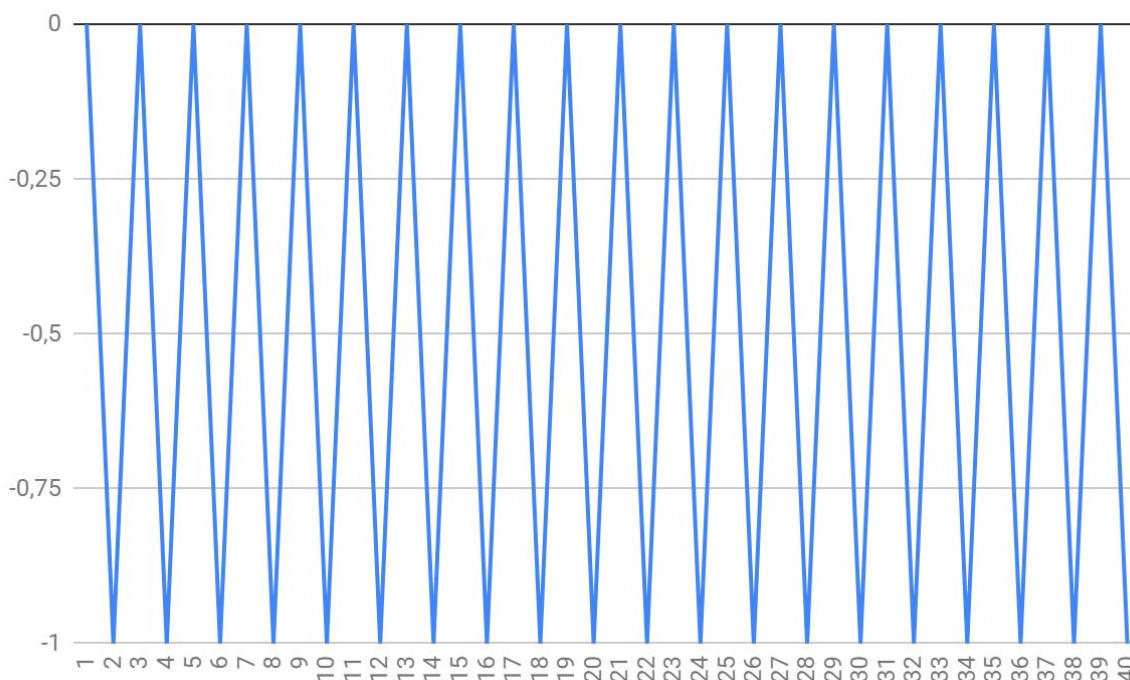
Dla $c = -2$ i $x_0 = 1$ wartość funkcji dla każdego x_n jest równa -1.



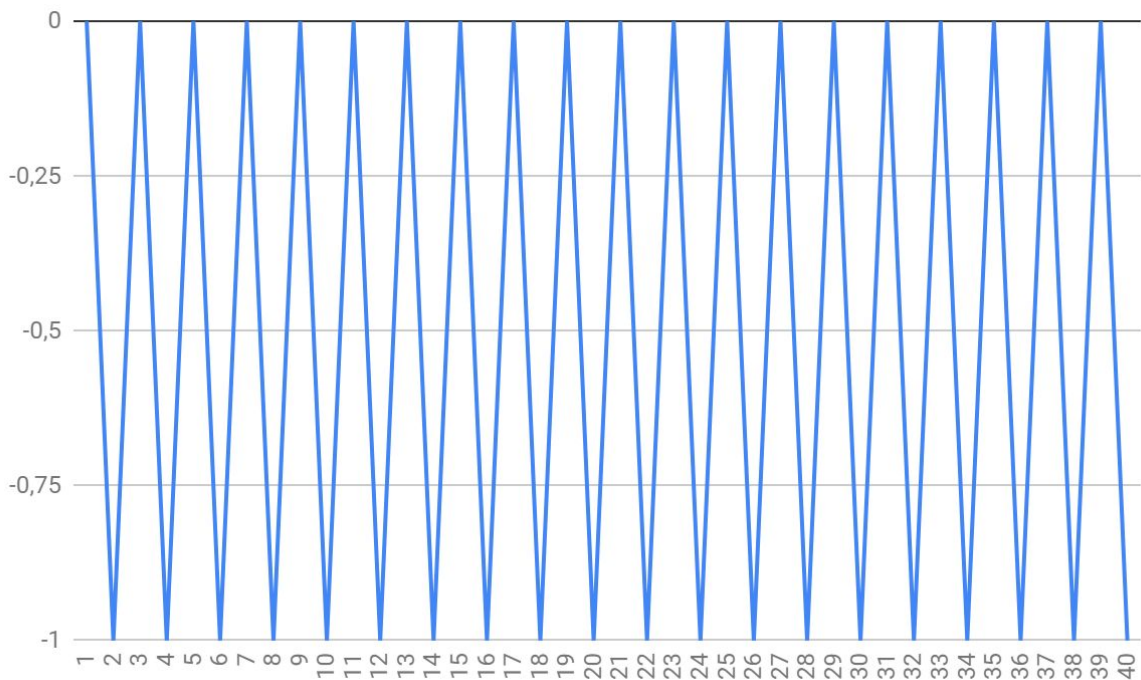
Dla $c = -2$ i $x_0 = 2$ wartość funkcji dla każdego x_n jest równa 2.



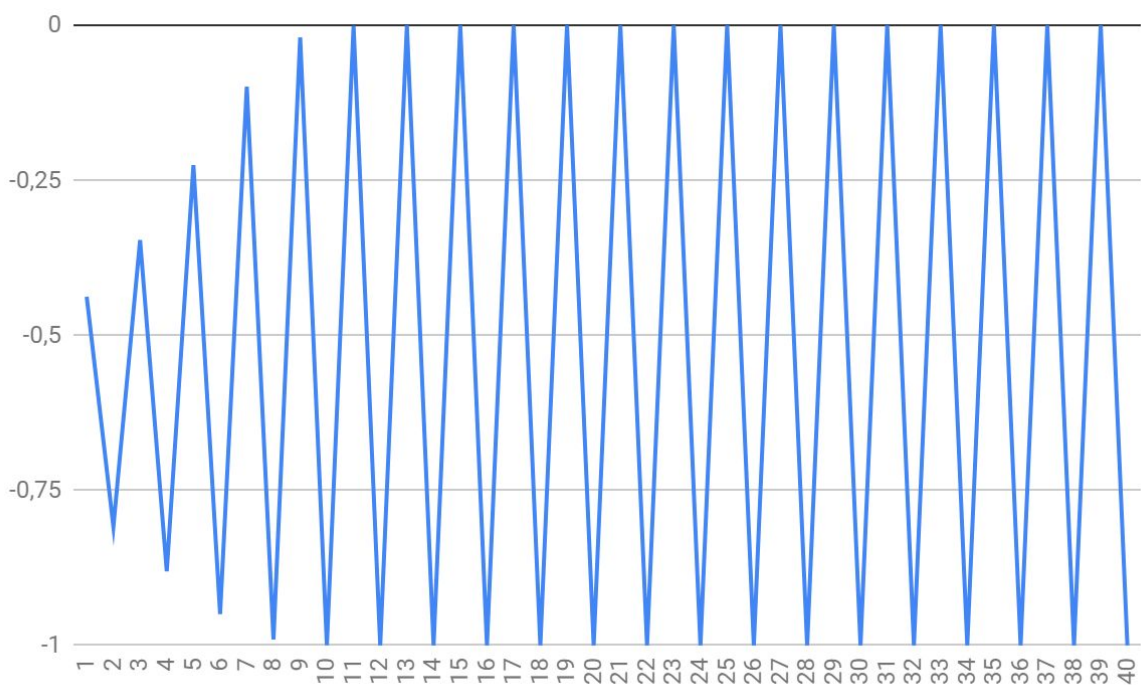
Dla $c = -2$ i $x_0 = 1.9999999999999999$ funkcja zachowuje się już zupełnie inaczej niż w poprzednim, wydawać by się mogło, niemal identycznym przykładzie. Po raz kolejny niewielkie zaburzenie wartości na początku nawarstwia się i coraz silniej wpływa na kolejne wyniki.



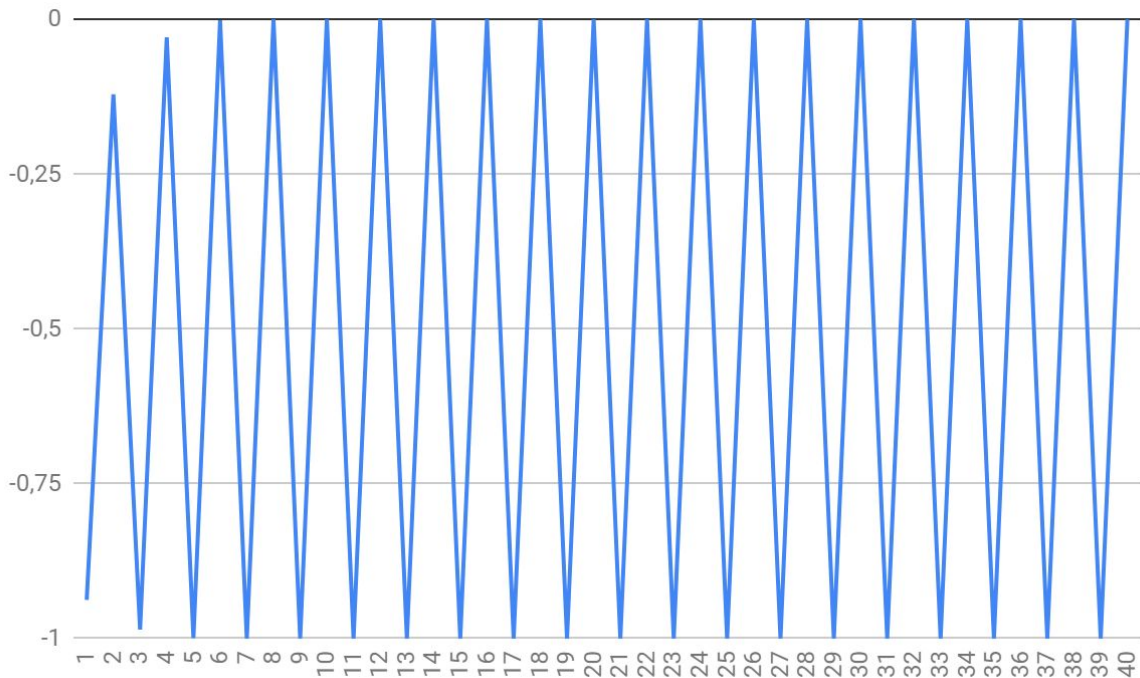
Dla $c = -1$ i $x_0 = 1$ funkcja przyjmuje naprzemiennie wartości -1 oraz 0.



Dla $c = -1$ i $x_0 = -1$ funkcja zachowuje się identycznie jak w przykładzie powyżej- przyjmuje naprzemiennie wartości -1 oraz 0.



Dla $c = -1$ i $x_0 = 0.75$ funkcja początkowo zwiększa swoją amplitudę, by ostatecznie ustabilizować się, i zmieniać wartości pomiędzy -1 a 0.



Dla $c = -1$ i $x_0 = 0.25$ funkcja podobnie jak w przykładzie poprzednim, początkowo zwiększa swoją amplitudę, by ostatecznie ustabilizować się i zmieniać wartości pomiędzy -1 a 0.

Każdy z wymienionych wyżej przykładów jest układem stabilnym od samego początku, lub stabilizuje się w czasie. Wyjątkiem jest przypadek dla $c = -2$ i $x_0 = 1.9999999999999999$ - układ ten jest chaotyczny, z każdą następną iteracją staje się coraz bardziej niestabilny, mimo tego, że wartość początkowa odbiegła od tej ze stabilnego przykładu drugiego tylko o wartość rzędu 10^{14} . Po raz kolejny pokazuje to, że drobne odchylenia na początku obliczania funkcji mogą w znaczny sposób wpływać na kolejne wyniki.