Obliczenia naukowe
Lista 1

Dominik Kaczmarek, nr albumu 261757

22 maja 2023

Spis treści

1 Zadanie 1

1.1 Opis problemu

Zadanie polegało na powtórzeniu eksperymentu z zadania 5 z poprzedniej listy po wprowadzeniu niewielikich zmian w początkowych tablicach. Z x_4 usuwamy ostatnią cyfrę 9 oraz z x_5 usuwamy ostatnią cyfrę 7. Stare tablice wyglądały następująco:

$$x = [2.718281828, -3.141592654, 1.414213562, 0.57721566\mathbf{4}\mathbf{9}, 0.301029995\mathbf{7}]$$

$$y = [1486.2497, 878366.9879, -22.37492, 4773714.647, 0.000185049]$$

Tablice po modyfikacji:

$$x' = [2.718281828, -3.141592654, 1.414213562, 0.577215664, 0.301029995]$$

$$y = [1486.2497, 878366.9879, -22.37492, 4773714.647, 0.000185049]$$

1.2 Rozwiązanie

W pliku *z1.jl* znajdują się algorytmy do liczenia poniższych sum ze zmodyfikowanymi danymi.

a) "w przód": $\sum_{i=1}^{n} x_i y_i$

b) "w tył": $\sum_{i=n}^{1} x_i y_i$

c) dodanie dodatnich liczb w porządku od największej do najmniejszej oraz ujemnych w porządku od najmniejszej do największej, a następnie dodanie do siebie obliczonych sum częściowych

d) metoda przeciwna do sposobu *3*

1.3 Wyniki i interpretacja

Tabela 1: `Float32`

alg	$x \cdot y$	$x' \cdot y$
a	-0.4999443	-0.4999443
b	-0.4543457	-0.4543457
c	-0.5	-0.5
d	-0.5	-0.5

Tabela 2: `Float64`

alg	$x \cdot y$	$x' \cdot y$
a	$1.0251881368296672 \cdot 10^{-10}$	-0.004296342739891585
b	$-1.5643308870494366 \cdot 10^{-10}$	-0.004296342998713953
c	0.0	-0.004296342842280865
d	0.0	-0.004296342842280865

Możemy zauważyć, że wyniki uzykane w arytmetyce `Float32` są identyczne dla tablic x i x'. Ostatnie cyfry usunięte w x_4 oraz x_5 i tak wcześniej nie mieściły się w precyzji `Float32`, dlatego wyniki pozostały bez zmian. Inaczej jest w przypadku wyników w arytmetyce `Float64`. Różnice miedzy wynikami algorytmów dla $x' \cdot y$ są rzędu 10^{-10}, ale porównując wyniki między $x \cdot y$ i $x' \cdot y$ widać znaczącą różnicę. Wyniki $x \cdot y$ są rzędu 10^{-10} natomiast dla $x' \cdot y$ są rzędu 10^{-3}.

1.4 Wnioski

Pozornie mało znacząca zmiana (usunięte cyfry zjadowały się na 10 miejscu po przecinku) dwóch wartości w tablicy x spowodowała znaczącą róznicę miedzy wynikami $x \cdot y$ i $x' \cdot y$. Rząd wielkości wyników zmienił się z 10^{-10} do 10^{-3} co przy tak małej zmianie inputu jest nieporządanym zjawiskiem. Świadczy to o **złym uwarunkowaniu zadania**.

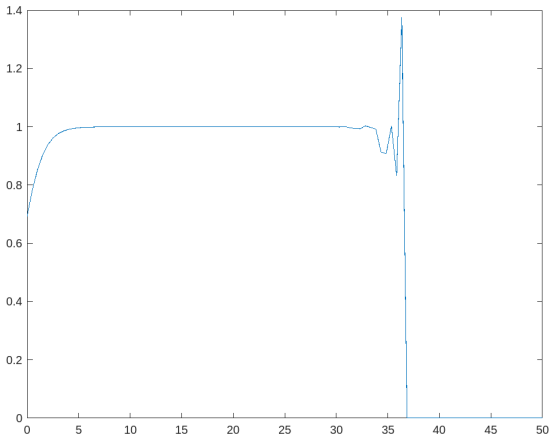
2 Zadanie 2

2.1 Opis problemu

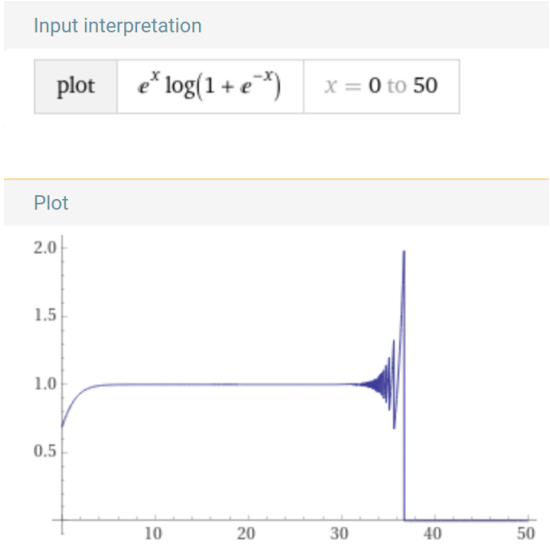
- Narysować wykres funkcji $f(x) = e^x \ln(1 + e^{-x})$ w conajmniej dwóch programach do wizualizacji.
- Policzyć granicę funckji $\lim_{x\to\infty} f(x)$
- Porównać otrzymane wykresy z obliczoną granicą

2.2 Rozwiązanie

Do wygenerowania wykresów funckji $f(x)$ użyłem programów `Matlab` oraz `Wolfram Alpha`.



Wykres 1: Wykres wygenerowany przy użyciu `Matlaba`



Wykres 2: Wykres wygenerowany przy użyciu `Wolfram Alpha`

Liczenie granicy $\lim_{x\to\infty} f(x)$:

$$\lim_{x\to\infty} f(x) = \lim_{x\to\infty} e^x \ln(1 + e^{-x}) = \lim_{x\to\infty} \frac{\ln(1 + e^{-x})}{e^{-x}} = \left(\frac{0}{0}\right)$$

Otrzymujemy wyrażenie $\left(\frac{0}{0}\right)$, więc możemy użyć reguły *de l'Hospitala*. Obliczmy zatem pochodną licznika i mianownika. Niech $g(x) = \ln(1 + e^{-x})$, $h(x) = e^{-x}$.

$$g'(x) = \frac{-e^{-x}}{e^{-x} + 1}$$
$$h'(x) = -e^{-x}$$

Mając policzone pochodne wstawiamy je do liczonej granicy.

$$\lim_{x\to\infty} f(x) = \lim_{x\to\infty} \frac{g'(x)}{h'(x)} = \lim_{x\to\infty} \frac{\frac{-e^{-x}}{e^{-x}+1}}{-e^{-x}} = \lim_{x\to\infty} \frac{1}{e^{-x} + 1} = \frac{1}{1} = 1$$

2.3 Wnioski

Licząc granicę podanej funkcji $f(x)$ w teorii nasze wykresy powinny zbiegać do jedynki przy $x \to \infty$. Dzieje się tak jednak tylko do pewnego momentu. Dla $x > 30$ nasz wykres zaczyna *"wariować"* i rozbieżności względem oczekiwanego wyniku stają się coraz większe. Ostatecznie funkcja przyjmuje wartości równe 0. Wynika to z faktu, że dla większych wartości x funkcja e^{-x} przyjmuje wartości bliskie 0, co z kolei sprawia, że funkcja $g(x) = \ln(1 + e^{-x})$ także przyjmuje wartości bliskie 0. Ze względu na ograniczoną precyzje arytmetyki `Float64` dla $x > 40$ funkcja $g(x)$ zwraca wynik $g(x) = 0$.

3 Zadanie 3

3.1 Opis problemu

W tym zadaniu naszym celem było rozwiązanie układu równań liniowych

$$\boldsymbol{Ax} = \boldsymbol{b}$$

dla danej macierzy współczynników $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ i wektora prawych stron $b \in \mathbb{R}^n$.
Macierz $\boldsymbol{A}$ generować w następujący sposób:

(a) $\boldsymbol{A} = \boldsymbol{H}_n$, gdzie $\boldsymbol{H}_n$ jest macierzą *Hilberta* stopnia n wygenerowaną za pomocą funkcji $\boldsymbol{A} = hilb(n)$ (źródła w języku `Julia` znajdowały się na stronie domowej *dr. Zielińskiego*),

(b) $\boldsymbol{A} = \boldsymbol{R}_n$, gdzie $\boldsymbol{R}_n$ jest losową macierzą stopnia n z zadanym wskaźnikiem uwarunkowania c wygenerowaną za pomocą funkcji $\boldsymbol{A} = matcond(n, c)$ (źródła w języku `Julia` znajdowały się na stronie domowej *dr. Zielińskiego*).

Wektor $\boldsymbol{b}$ zadany jest następująco $\boldsymbol{b} = \boldsymbol{Ax}$, gdzie $\boldsymbol{A}$ jest wygenerowaną macierzą, a $x = (1, \ldots, 1)^T$. Zatem wiemy jakie jest rozwiązanie dokładne $\boldsymbol{Ax} = \boldsymbol{b}$ dla $\boldsymbol{A}$ i $\boldsymbol{b}$.

Rozwiązać $\boldsymbol{Ax} = \boldsymbol{b}$ za pomocą dwóch algorytmów:

(i) eliminacji Gaussa `(x=A/b)`,

(ii) $x = \boldsymbol{A}^{-1}\boldsymbol{b}$ `(x=inv(A)*b)`.

Eksperymenty wykonać dla macierzy Hilberta $\boldsymbol{H}_n$ z rosnącym stopniem $n > 1$ oraz dla macierzy losowej $\boldsymbol{R_n}$, $n = 5, 10, 20$ z rosnącym wskaźnikiem uwarunkowania $c = 1, 10, 10^3, 10^7, 10^{12}, 10^{16}$. Porównać obliczony $\tilde{\boldsymbol{x}}$ z rozwiązaniem dokładnym $\boldsymbol{x} = (1, ..., 1)^T$,tj. **policzyć błędy względne**.

3.2 Rozwiązanie

Stworzyłem dwie funkcje liczące $\boldsymbol{x}$ metodami (i) oraz (ii), korzystające z generatorów macierzy ze strony dr. Zielińskiego. Dla $\boldsymbol{H}_n$ wykonałem testy dla $n = 2, 4, 6, .., 32$, natomiast dla $\boldsymbol{R}_n$ dla n i c zadanych w poleceniu. Dokładne rozwiązanie znajduje się w pliku *z3.jl*.

3.3 Wyniki i interpretacja

n	$rank(\boldsymbol{H}_n)$	$cond(\boldsymbol{H}_n)$	błąd met. Gaussa	błąd met. inwersji
2	2	19.28147006790397	5.661048867003676e-16	1.4043333874306803e-15
4	4	15513.73873892924	4.137409622430382e-14	0.0
6	6	1.4951058642254734e7	2.618913302311624e-10	2.0163759404347654e-10
8	8	1.5257575538060041e10	6.124089555723088e-8	3.07748390309622e-7
10	10	1.602441698742836e13	8.67039023709691e-5	0.0002501493411824886
12	11	1.7515952300879806e16	0.13396208372085344	0.258994120804705
14	11	6.200786281355982e17	1.4554087127659643	8.71499275104814
16	12	7.046389953630175e17	54.15518954564602	29.84884207073541
18	12	2.2477642911280653e18	10.257619124632317	24.762070989128866
20	13	1.1484020388436145e18	108.31777346206205	114.34403152557572
22	13	1.093638219471544e19	17.003425713362045	102.60161611995359
24	13	2.1608428256899587e18	39.638573210187644	43.34914763015038
26	14	5.825077809111695e18	63.80426636186403	100.78434642499187
28	14	5.889050001520599e18	276.91498822022265	290.1167291705239
30	14	5.512691295390715e18	24.80615905441871	59.97231132227779
32	14	4.150195034190458e18	36.582441571177284	67.4381226943068
34	14	4.56520144655429e18	88.87380459381126	95.99116506490786
36	15	4.3381250003280466e18	15.563379312608332	19.599011323097056

Tabela 3: Wyniki eksperymetów dla macierzy $\boldsymbol{H}_n$

n	c	$rank(\boldsymbol{R}_n)$	$cond(\boldsymbol{R}_n)$	błąd met. Gaussa	błąd met. inwersji
5	1	5	1.0000000000000004	7.021666937153402e-17	2.220446049250313e-16
5	10	5	9.999999999999996	1.719950113979703e-16	4.965068306494546e-17
5	10^3	5	1000.0000000000083	1.7325100555155983e-14	1.795790461201834e-14
5	10^7	5	1.000000000167412e7	4.853951289935605e-11	1.041250292910165e-10
5	10^{12}	5	1.0000141022729923e12	4.197979517919719e-6	6.565824309562113e-6
5	10^{16}	4	1.0332139199743814e16	0.12063837255405567	0.14756114662128203
10	1	10	1.0000000000000009	3.2368285245694683e-16	3.1791949213824894e-16
10	10	10	9.999999999999998	4.440892098500626e-16	1.7554167342883504e-16
10	10^3	10	999.9999999999978	2.579339843768069e-14	2.997256743867112e-14
10	10^7	10	9.999999991479073e6	2.6845434166603924e-11	5.617114676168066e-11
10	10^{12}	10	1.0001037283817148e12	1.7835143002023235e-5	1.9004817374591476e-5
10	10^{16}	9	1.4873033093706182e16	0.2245006676364469	0.18802011195614152
20	1	20	1.0000000000000009	4.447825579847492e-16	4.1317602818954126e-16
20	10	20	9.999999999999996	4.399061727369024e-16	4.2494139782533216e-16
20	10^3	20	999.9999999999884	2.1642230995786356e-15	6.466989556846191e-15
20	10^7	20	9.999999994900774e6	2.666728644265561e-10	2.8250009341778073e-10
20	10^{12}	20	9.999538639467982e11	5.294673355771642e-5	5.618752538414722e-5
20	10^{16}	19	1.8408804806608844e16	0.12364852917255352	0.11065103860222009

Tabela 4: Wyniki eksperymetów dla macierzy $\boldsymbol{R}_n$

3.4 Wnioski

Macierz Hilberta $\boldsymbol{H}_n$ jest macierzą bardzo źle uwarunkową, na co wskazują bardzo duże wartości $cond(\boldsymbol{H}_n)$ (wskaźnik uwarunkowania macierzy) już dla małych n. Metoda Gaussa zwraca wyniki podobne do metody inwersji (różnią się co najwyżej o jeden rząd wielkości), jednak dla $\boldsymbol{H}_n$ metoda Gaussa wydaje się nieco lepsza od metody inwersji, ponieważ w ogóle zwraca mniejszy błąd względny (istnieją pojedyncze przypadki gdzie jest odwrotnie $n \in \{4, 6, 16\}$). Inkrementując n, dla $n \leqslant 12$ obydwa błędy względne szybko rosną, gdy $12 \leqslant x \leqslant 16$ tempo wzrostu błędów spada do jednego rzędu wielkości, natomiast dla $x \geqslant 16$ błędy oscylują między rzędami wielkości 10 i 10^2.

W przypadku macierzy losowych $\boldsymbol{R}_n$ o ustalonym wskaźniku uwarunkowania c błędy orzymane dwoma metodami również mają podobne wartości dla tych samych n i c. Możemy również zaobserwować, że dla tych samych wartości c błędy mają podobny rzęd wielkości (np. dla $c = 10^{12}$ $\boldsymbol{R}_5 = 4.197979517919719 \cdot 10^{-6}$, $\boldsymbol{R}_{10} = 1.7835143002023235 \cdot 10^{-5}$, $\boldsymbol{R}_{20} = 5.294673355771642 \cdot 10^{-5}$).

Na podstawie macierzy $\boldsymbol{R}_n$ widać, że wielkość błędu zależy od wskaźnika uwarunkowania macierzy ($cond(\boldsymbol{H}_n)$) a nie od stopnia macierzy n. Zatem znając wskaźnik uwarunkowania macierzy jesteśmy w stanie oszacować błędy względne, które otrzymamy.

4 Zadanie 4

4.1 Opis problemu

(a) Mając złośliwy wielomian Wilkinsona $P(x)$ (podany w treści zadania) i jego postać iloczynową

$$p(x) = \prod_{k=1}^{20}(x - k)$$

obliczyć pierwiastki $P(x)$ korzystając z funkcji `roots` (z pakietu *Polynomials*). Sprawdzić obliczone pierwiastki z_k $(k = 1, 2, 3, \ldots, 20)$, obliczając $|P(z_k)|$, $|p(z_k)|$, $|k - z_k|$.

(b) Powtórzyć eksperyment Wilkinsona, tj. w $P(x)$ zmienić współczynnik -210 na $-210 - 2^{-23}$. Wyjaśnić zjawisko.

4.2 Rozwiązanie

- Do wygenerowania wielomianu $P(x)$ użyłem funkcji `P = Polynomial(reverse(wspolczynniki))`, gdzie *`wspolczynniki`* to tablica z pliku *wielomian.txt*

- Do wygenerowania wielomianu $p(x)$ użyłem funkcji `p = fromroots(1:20)`

- Do wyliczenia pierwiastków z_k użyłem funkcji `z = roots(P)`

Dokładne rozwiązanie znajduje się w pliku *z4.jl*.

4.3 Wyniki i interpretacja

k	z_k	$\|P(z_k)\|$	$\|p(z_k)\|$	$\|k - z_k\|$
1	0.9999999999996989	35696.50964788257	5.518479490350445e6	3.0109248427834245e-13
2	2.0000000000283182	176252.60026668405	7.37869762990174e19	2.8318236644508943e-11
3	2.9999999995920965	279157.6968824087	3.3204139316875795e20	4.0790348876384996e-10
4	3.9999999837375317	3.0271092988991085e6	8.854437035384718e20	1.626246826091915e-8
5	5.000000665769791	2.2917473756567076e7	1.8446752056545688e21	6.657697912970661e-7
6	5.999989245824773	1.2902417284205095e8	3.320394888870117e21	1.0754175226779239e-5
7	7.000102002793008	4.805112754602064e8	5.423593016891273e21	0.00010200279300764947
8	7.999355829607762	1.6379520218961136e9	8.262050140110275e21	0.0006441703922384079
9	9.002915294362053	4.877071372550003e9	1.196559421646277e22	0.002915294362052734
10	9.990413042481725	1.3638638195458128e10	1.655260133520688e22	0.009586957518274986
11	11.025022932909318	3.585631295130865e10	2.24783329792479e22	0.025022932909317674
12	11.953283253846857	7.533332360358197e10	2.886944688412679e22	0.04671674615314281
13	13.07431403244734	1.9605988124330817e11	3.807325552826988e22	0.07431403244734014
14	13.914755591802127	3.5751347823104315e11	4.612719853150334e22	0.08524440819787316
15	15.075493799699476	8.21627123645597e11	5.901011420218566e22	0.07549379969947623
16	15.946286716607972	1.5514978880494067e12	7.010874106897764e22	0.05371328339202819
17	17.025427146237412	3.694735918486229e12	8.568905825736165e22	0.025427146237412046
18	17.99092135271648	7.650109016515867e12	1.0144799361044434e23	0.009078647283519814
19	19.00190981829944	1.1435273749721195e13	1.1990376202371257e23	0.0019098182994383706
20	19.999809291236637	2.7924106393680727e13	1.4019117414318134e23	0.00019070876336257925

Tabela 5: Wyniki dla podpunktu `(a)`

Analizując wartości $|k - z_k|$ dla podpunktu `(a)` widzimy, że różnica rośnie do wiersza $n = 14$, po czym spowrotem maleje. Dodatkowo $|k - z_{14-i}| \approx |k - z_{14-i}|, i = 1, \ldots, 6$. Kluczowy jest również fakt, że $(\bigvee k \in \{1, \ldots, 20\})(|k - z_k| \leqslant 0.1)$, czyli można powiedzieć, że otrzymane pierwiastki są zbliżone do właściwych. Pomimo tego wyniki funkcji $P(z_k)$ oraz $p(z_k)$ nie są nawet zbliżone do 0, wręcz przeciwnie są liczbami bardzo dużymi (rzędu nawet 10^13 dla $P(z_k)$ i 10^23 dla $p(z_k)$).

k	z_k	$\lvert P(z_k)\rvert$	$\lvert p(z_k)\rvert$	$\lvert k - z_k\rvert$
1	0.9999999999998357 + 0.0im	20259.872313418207	3.0131001276845885e6	1.6431300764452317e-13
2	2.0000000000550373 + 0.0im	346541.4137593836	7.37869763029606e19	5.503730804434781e-11
3	2.99999999660342 + 0.0im	2.2580597001197007e6	3.320413920110016e20	3.3965799062229962e-9
4	4.000000089724362 + 0.0im	1.0542631790395478e7	8.854437817429642e20	8.972436216225788e-8
5	4.99999857388791 + 0.0im	3.757830916585153e7	1.844672697408419e21	1.4261120897529622e-6
6	6.000020476673031 + 0.0im	1.3140943325569446e8	3.320450195282313e21	2.0476673030955794e-5
7	6.99960207042242 + 0.0im	3.939355874647618e8	5.422366528916004e21	0.00039792957757978087
8	8.007772029099446 + 0.0im	1.184986961371896e9	8.289399860984408e21	0.007772029099445632
9	8.915816367932559 + 0.0im	2.2255221233077707e9	1.160747250177049e22	0.0841836320674414
10	10.095455630535774 - 0.6449328236240688im	1.0677921232930157e10	1.7212892853670706e22	0.6519586830380407
11	10.095455630535774 + 0.6449328236240688im	1.0677921232930157e10	1.7212892853670706e22	1.1109180272716561
12	11.793890586174369 - 1.6524771364075785im	3.1401962344429485e10	2.8568401004080956e22	1.665281290598479
13	11.793890586174369 + 1.6524771364075785im	3.1401962344429485e10	2.8568401004080956e22	2.0458202766784277
14	13.992406684487216 - 2.5188244257108443im	2.157665405951858e11	4.934647147686795e22	2.518835871190904
15	13.992406684487216 + 2.5188244257108443im	2.157665405951858e11	4.934647147686795e22	2.7128805312847097
16	16.73074487979267 - 2.812624896721978im	4.850110893921027e11	8.484694713563005e22	2.9060018735375106
17	16.73074487979267 + 2.812624896721978im	4.850110893921027e11	8.484694713563005e22	2.825483521349608
18	19.5024423688181 - 1.940331978642903im	4.557199223869993e12	1.3181947820607215e23	2.4540214463129764
19	19.5024423688181 + 1.940331978642903im	4.557199223869993e12	1.3181947820607215e23	2.0043294443099486
20	20.84691021519479 + 0.0im	8.756386551865696e12	1.5911084081430876e23	0.8469102151947894

Tabela 6: Wyniki dla podpunktu `(b)` z podmienionym współczynnikiem

Inaczej sytuacja wygląda dla $|k-z_k|$ w podpunkcie `(b)`. Różnica $|k-z_k|$ rośnie aż do wartości ≈ 2.9 dla $n = 16$. Istnieje zatem znacząca różnica między rzeczywistymi pierwiastkami k a wyliczonymi z_k. Wynika to z faktu, że w kolumnie z_k pojawiły się pierwiastki urojone, co jest chyba najbardziej zaskakujące. Podobnie jak przed modyfikacją funkcje $P(z_k)$ oraz $p(z_k)$ osiągają bardzo duże wartości (zbliżone do odpowiedników z podpunktu (1)).

4.4 Wnioski

Niewielka różnica między z_k a faktycznymi pierwiastkami sprawia, że funkcja zwraca bardzo duże liczby, czyli wyznaczanie pierwiastków wielomianu Wilkinsona jest źle uwarunkowane.
Wyznaczanie pierwiastków wielomianu Wilkinsona jest źle uwarunkowane ze względu na zaburzenia współczynników.
Dla niewielkiego względnego zaburzenia z_{19} w wielomianie $P(x)$ pojawiają się pierwiastki zespolone.

5 Zadanie 5

5.1 Opis problemu

Mając równanie rekurencyjne (model logistyczny, model wzrostu populacji)

$$p_{n+1} := p_n + rp_n(1 - p_n), \text{ dla } n = 0, 1, \ldots \tag{1}$$

Przeprowadzić następujące eksperymenty dla danych $p_0 = 0.01$ i $r = 3$:

1. wykonać 40 iteracji wyrażenia `(1)` w arytmetyce `Float32`;
2. to samo co w punkcie powyżej, tylko że z niewielką modyfikacją. Najpierw wykonać 10 iteracji wyrażenia `(1)`, po czym obciąć p_{10} do trzech miejsc po przecinku ($p_{10} = 0.722$) i kontynuować obliczenia do 40-stej itearcji;
3. wykonać 40 iteracji wyrażenia `(1)` w arytmetyce `Float64`.

Porównać wyniki otrzymane w punktach `1` i `2` oraz w punktach `1` i `3`.

5.2 Rozwiązanie

Napisałem algorytmy wykonujące zadane iteracje. Dokładne rozwiązanie znajduje się w pliku *z5.jl*.

5.3 Wyniki i interpretacja

n	p_n sposobem 1	p_n sposobem 2
1	0.0397	0.0397
2	0.15407173	0.15407173
3	0.5450726	0.5450726
4	1.2889781	1.2889781
5	0.1715188	0.1715188
6	0.5978191	0.5978191
7	1.3191134	1.3191134
8	0.056273222	0.056273222
9	0.21559286	0.21559286
10	0.7229306	**0.722**
11	1.3238364	1.3241479
12	0.037716985	0.036488414
13	0.14660022	0.14195944
14	0.521926	0.50738037
15	1.2704837	1.2572169
16	0.2395482	0.28708452
17	0.7860428	0.9010855
18	1.2905813	1.1684768
19	0.16552472	0.577893
20	0.5799036	1.3096911
21	1.3107498	0.09289217
22	0.088804245	0.34568182
23	0.3315584	1.0242395
24	0.9964407	0.94975823
25	1.0070806	1.0929108
26	0.9856885	0.7882812
27	1.0280086	1.2889631
28	0.9416294	0.17157483
29	1.1065198	0.59798557
30	0.7529209	1.3191822
31	1.3110139	0.05600393
32	0.0877831	0.21460639
33	0.3280148	0.7202578
34	0.9892781	1.3247173
35	1.021099	0.034241438
36	0.95646656	0.13344833
37	1.0813814	0.48036796
38	0.81736827	1.2292118
39	1.2652004	0.3839622
40	0.25860548	1.093568

Tabela 7: Porównanie wyników eskperymentów 1 i 2

Przeprowadzane są dwie grupy iteracji startujące z tego samego punktu. Dla 10 iteracji w prawej kolumnie obcinamy wartość p_n do trzech miejsc po przecinku i w takiej postaci bierzemy do kolejnego kroku obliczeń. Chociaż przez kilka kolejnych kroków otrzymywane wyniki są całkiem do siebie zbliżone to niedługo potem tracą wszelką korelację.

n	p_n `Float32`	p_n `Float64`
1	0.0397	0.0397
2	0.15407173	0.15407173000000002
3	0.5450726	0.5450726260444213
4	1.2889781	1.2889780011888006
5	0.1715188	0.17151914210917552
6	0.5978191	0.5978201201070994
7	1.3191134	1.3191137924137974
8	0.056273222	0.056271577646256565
9	0.21559286	0.21558683923263022
10	0.7229306	0.722914301179573
11	1.3238364	1.3238419441684408
12	0.037716985	0.03769529725473175
13	0.14660022	0.14651838271355924
14	0.521926	0.521670621435246
15	1.2704837	1.2702617739350768
16	0.2395482	0.24035217277824272
17	0.7860428	0.7881011902353041
18	1.2905813	1.2890943027903075
19	0.16552472	0.17108484670194324
20	0.5799036	0.5965293124946907
21	1.3107498	1.3185755879825978
22	0.088804245	0.058377608259430724
23	0.3315584	0.22328659759944824
24	0.9964407	0.7435756763951792
25	1.0070806	1.315588346001072
26	0.9856885	0.07003529560277899
27	1.0280086	0.26542635452061003
28	0.9416294	0.8503519690601384
29	1.1065198	1.2321124623871897
30	0.7529209	0.37414648963928676
31	1.3110139	1.0766291714289444
32	0.0877831	0.8291255674004515
33	0.3280148	1.2541546500504441
34	0.9892781	0.29790694147232066
35	1.021099	0.9253821285571046
36	0.95646656	1.1325322626697856
37	1.0813814	0.6822410727153098
38	0.81736827	1.3326056469620293
39	1.2652004	0.0029091569028512065
40	0.25860548	0.011611238029748606

Tabela 8: Porównanie wyników eskperymentów `1` i `3`

Sytuacja wygląda podobnie dla drugiego eksperymentu, gdzie porównujemy wyniki w arytmetykach `Float32` i `Float64`. Ta druga ma większą precyzję, jednak po kilku iteracjach zaczynają pojawiać się niedokładności wynikające z zaokrągleń, które z każdym koljnym krokiem są coraz większe. Analogicznie jak w pierwszym przykładzie po przekroczniu pewnego n (≈ 20) wartości p_n w obu kolumnach tracą jakąkolwiek korelacje.

5.4 Wnioski

Intuicja może podpowiadać, że wyniki w arytmetyce `Float64` są bardziej wiarygodne od obu wyników w arytmetyce `Float64`. W rzeczywistości jednak po wykonaniu kilkunastu iteracji każdy z nich jest równie wiarygodny jak wynik otrzymany przy użyciu generatora liczb losowych. Przyczyną jest szybka kumulacja błędu, której nie jesteśmy w stanie uniknąć przez ograniczenia arytmetyk (możemy ją jedynie opóźnić). Podsumowując **"Chaos zwycięża każdy komputer"** i nie jesteśmy w stanie przewidzieć wszystkich wyników.

6 Zadanie 6

6.1 Opis problemu

Mając równanie rekurencyjne

$$x_{n+1} := x_n^2 + c \text{ dla } n = 0, 1, \ldots, \tag{2}$$

gdzie c jest pewną daną stałą, przeprowadzić następujące eksperymenty. Dla danych:

1. $c = -2$ i $x_0 = 1$
2. $c = -2$ i $x_0 = 2$
3. $c = -2$ i $x_0 = 1.99999999999999$
4. $c = -1$ i $x_0 = 1$
5. $c = -1$ i $x_0 = -1$
6. $c = -1$ i $x_0 = 0.75$
7. $c = -1$ i $x_0 = 0.25$

wykonać, w języku `Julia` w arytmetyce `Float64`, 40 iteracji wyrażenia `(2)`. Zaobserwować zachowanie generowanych ciągów.

6.2 Rozwiązanie

Wykonałem po 40 iteracji równania `(2)` dla każdej pary zadanych danych. Dokładne rozwiązanie znajduje się w pliku *z6.jl*.

6.3 Wyniki i interpretacja

Sprządziłem dwie tabele. W pierwszej z nich znalazły się wyniki iteracji dla par danych, gdzie $c = -2$ (tj. przykład `1, 2, 3`), a w drugiej gdzie $c = -1$ (tj. przykład `4, 5, 6, 7`).
Oprócz tego dla każdego przykład wykonałem iterację graficzną przu użyciu programu `GeoGebra`.

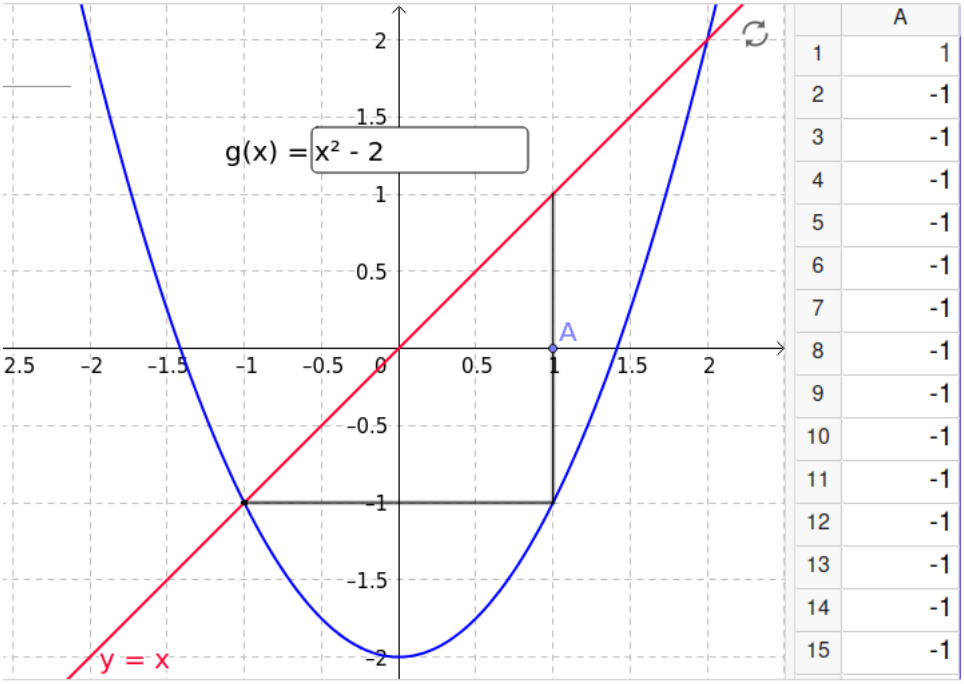
n	$x_0 = 1$	$x_0 = 2$	$x_0 = 1.99999999999999$
1	-1.0	2.0	1.99999999999996
2	-1.0	2.0	1.9999999999998401
3	-1.0	2.0	1.9999999999993605
4	-1.0	2.0	1.999999999997442
5	-1.0	2.0	1.9999999999897682
6	-1.0	2.0	1.9999999999590727
7	-1.0	2.0	1.999999999836291
8	-1.0	2.0	1.9999999993451638
9	-1.0	2.0	1.9999999973806553
10	-1.0	2.0	1.999999989522621
11	-1.0	2.0	1.9999999580904841
12	-1.0	2.0	1.9999998323619383
13	-1.0	2.0	1.9999993294477814
14	-1.0	2.0	1.9999973177915749
15	-1.0	2.0	1.9999892711734937
16	-1.0	2.0	1.9999570848090826
17	-1.0	2.0	1.999828341078044
18	-1.0	2.0	1.9993133937789613
19	-1.0	2.0	1.9972540465439481
20	-1.0	2.0	1.9890237264361752
21	-1.0	2.0	1.9562153843260486
22	-1.0	2.0	1.82677862987391
23	-1.0	2.0	1.3371201625639997
24	-1.0	2.0	-0.21210967086482313
25	-1.0	2.0	-1.9550094875256163
26	-1.0	2.0	1.822062096315173
27	-1.0	2.0	1.319910282828443
28	-1.0	2.0	-0.2578368452837396
29	-1.0	2.0	-1.9335201612141288
30	-1.0	2.0	1.7385002138215109
31	-1.0	2.0	1.0223829934574389
32	-1.0	2.0	-0.9547330146890065
33	-1.0	2.0	-1.0884848706628412
34	-1.0	2.0	-0.8152006863380978
35	-1.0	2.0	-1.3354478409938944
36	-1.0	2.0	-0.21657906398474625
37	-1.0	2.0	-1.953093509043491
38	-1.0	2.0	1.8145742550678174
39	-1.0	2.0	1.2926797271549244
40	-1.0	2.0	-0.3289791230026702

Tabela 9: Wyniki dla podpunktów `1, 2, 3` ($c = -2$)

n	$x_0 = 1$	$x_0 = -1$	$x_0 = 0.75$	$x_0 = 0.25$
1	0.0	0.0	-0.4375	-0.9375
2	-1.0	-1.0	-0.80859375	-0.12109375
3	0.0	0.0	-0.3461761474609375	-0.9853363037109375
4	-1.0	-1.0	-0.8801620749291033	-0.029112368589267135
5	0.0	0.0	-0.2253147218564956	-0.9991524699951226
6	-1.0	-1.0	-0.9492332761147301	-0.0016943417026455965
7	0.0	0.0	-0.0989561875164966	-0.9999971292061947
8	-1.0	-1.0	-0.9902076729521999	-5.741579369278327e-6
9	0.0	0.0	-0.01948876442658909	-0.9999999999670343
10	-1.0	-1.0	-0.999620188061125	-6.593148249578462e-11
11	0.0	0.0	-0.0007594796206411569	-1.0
12	-1.0	-1.0	-0.9999994231907058	0.0
13	0.0	0.0	-1.1536182557003727e-6	-1.0
14	-1.0	-1.0	-0.9999999999986692	0.0
15	0.0	0.0	-2.6616486792363503e-12	-1.0
16	-1.0	-1.0	-1.0	0.0
17	0.0	0.0	0.0	-1.0
18	-1.0	-1.0	-1.0	0.0
19	0.0	0.0	0.0	-1.0
20	-1.0	-1.0	-1.0	0.0
21	0.0	0.0	0.0	-1.0
22	-1.0	-1.0	-1.0	0.0
23	0.0	0.0	0.0	-1.0
24	-1.0	-1.0	-1.0	0.0
25	0.0	0.0	0.0	-1.0
26	-1.0	-1.0	-1.0	0.0
27	0.0	0.0	0.0	-1.0
28	-1.0	-1.0	-1.0	0.0
29	0.0	0.0	0.0	-1.0
30	-1.0	-1.0	-1.0	0.0
31	0.0	0.0	0.0	-1.0
32	-1.0	-1.0	-1.0	0.0
33	0.0	0.0	0.0	-1.0
34	-1.0	-1.0	-1.0	0.0
35	0.0	0.0	0.0	-1.0
36	-1.0	-1.0	-1.0	0.0
37	0.0	0.0	0.0	-1.0
38	-1.0	-1.0	-1.0	0.0
39	0.0	0.0	0.0	-1.0
40	-1.0	-1.0	-1.0	0.0

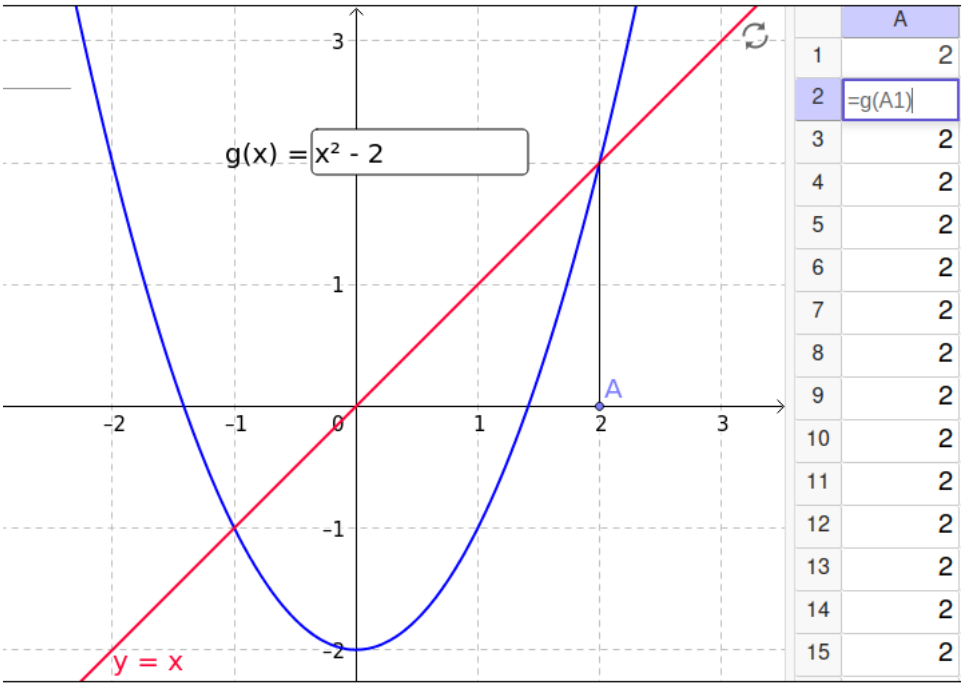
Tabela 10: Wyniki dla podpunktów `4, 5, 6, 7` ($c = -1$)

Rysunek 1: Iteracja graficzna dla `(1)`, $c = -2, x_0 = 1$



	A
1	1
2	-1
3	-1
4	-1
5	-1
6	-1
7	-1
8	-1
9	-1
10	-1
11	-1
12	-1
13	-1
14	-1
15	-1

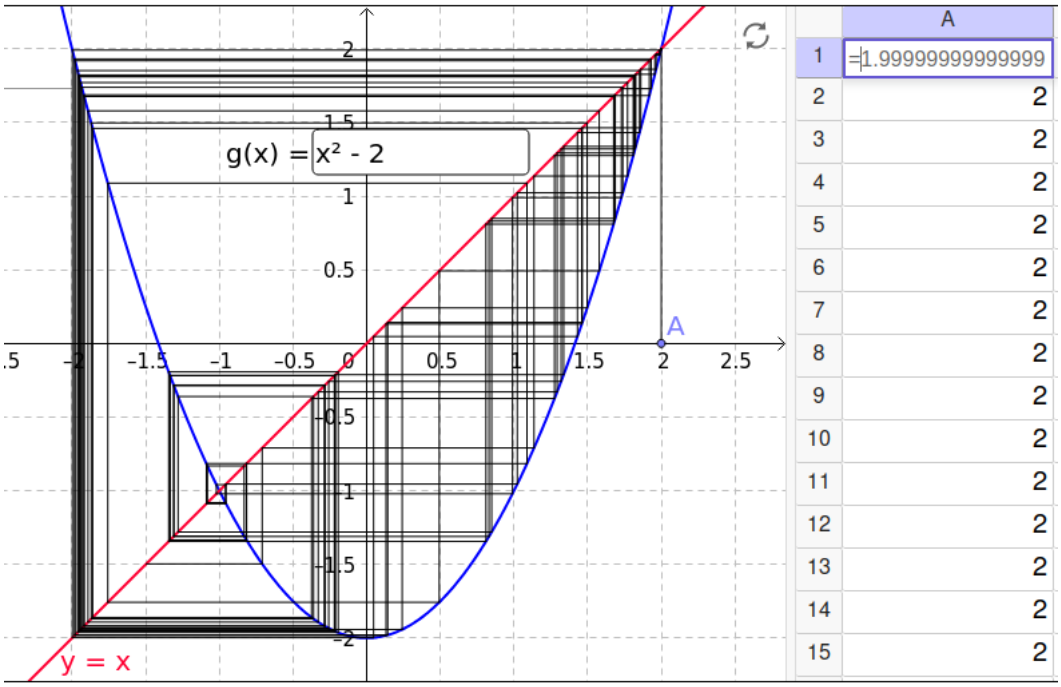
Rysunek 2: Iteracja graficzna dla `(2)`, $c = -2, x_0 = 2$



	A
1	2
2	=g(A1)
3	2
4	2
5	2
6	2
7	2
8	2
9	2
10	2
11	2
12	2
13	2
14	2
15	2

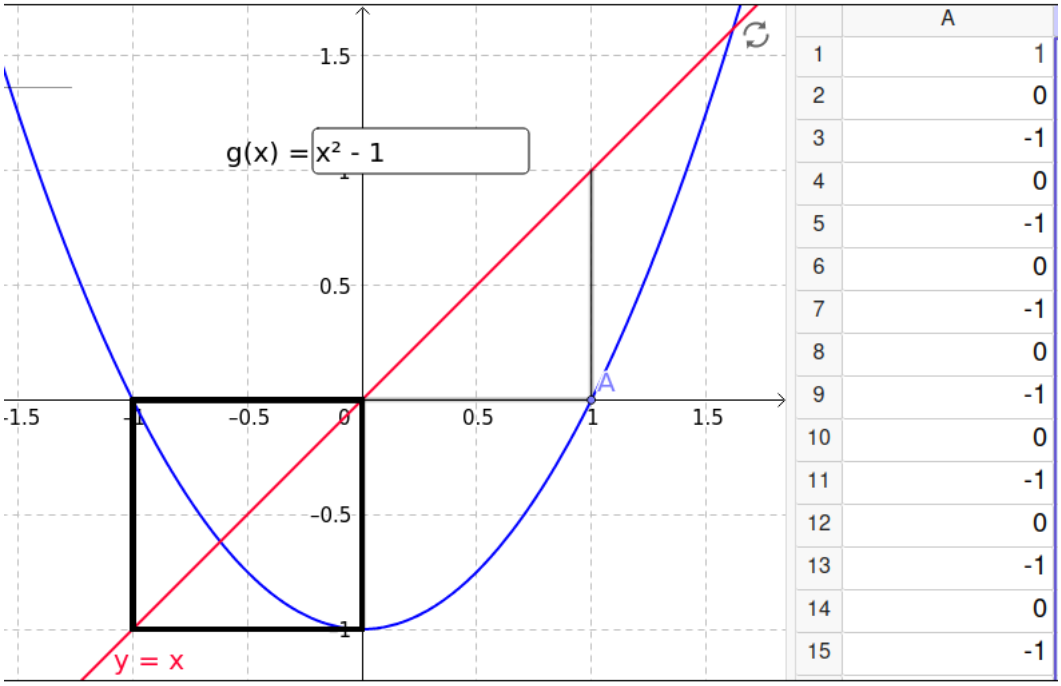
O tym że układ sprzężenia zwrotnego jest w stanie idealnie stabilnym dla przykładów (1) i (2) świadczy "mała liczba kresek na wykresie", ponieważ wyniki nakładają się na siebie.

Rysunek 3: Iteracja graficzna dla `(3)`, $c = -2, x_0 = 1.99999999999999$

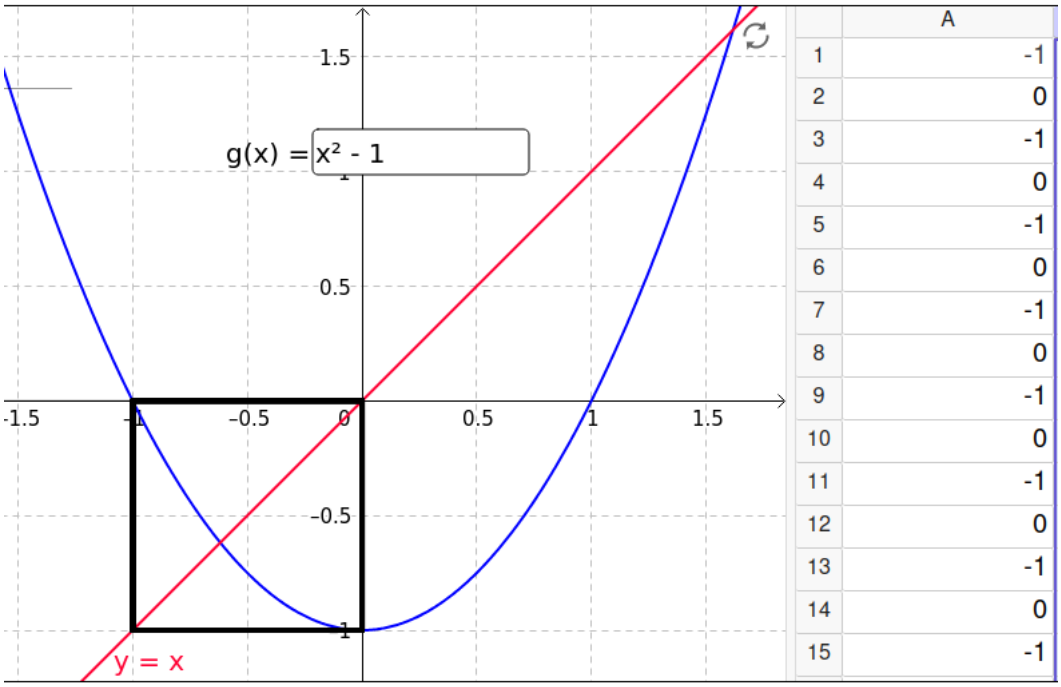


Zarówno z analizy tabel jak i wykresów możemy zauważyć że jedynie dla przykładu `(3)` wyniki nie ulegają stabilizacji.

Rysunek 4: Iteracja graficzna dla `(4)`, $c = -1, x_0 = 1$



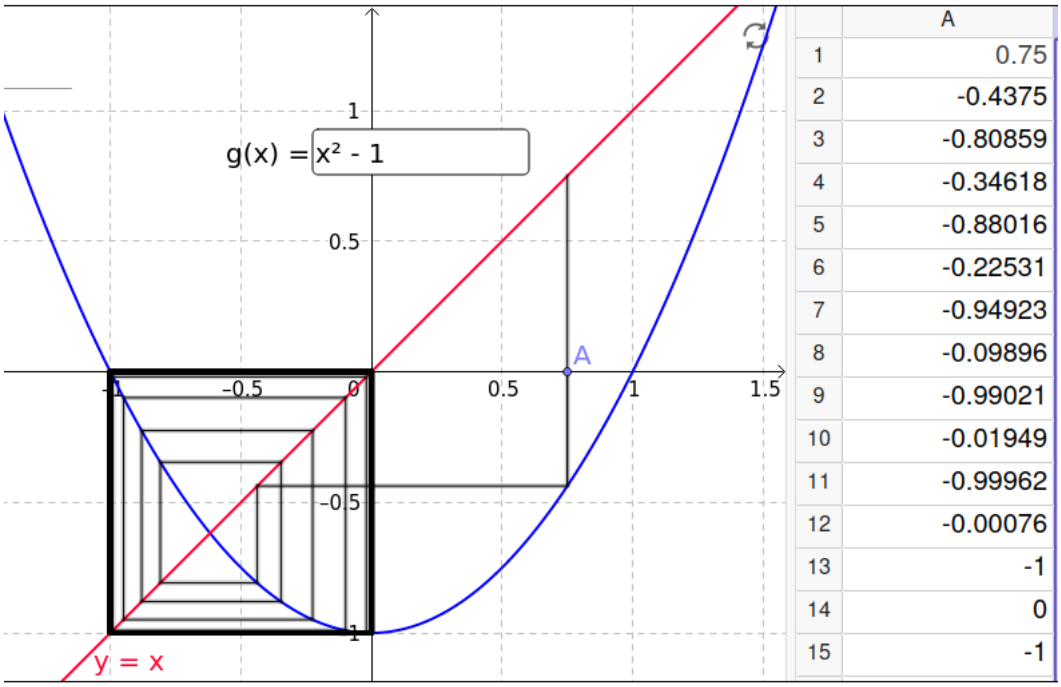
Rysunek 5: Iteracja graficzna dla `(5)`, $c = -1, x_0 = -1$



	A
1	-1
2	0
3	-1
4	0
5	-1
6	0
7	-1
8	0
9	-1
10	0
11	-1
12	0
13	-1
14	0
15	-1

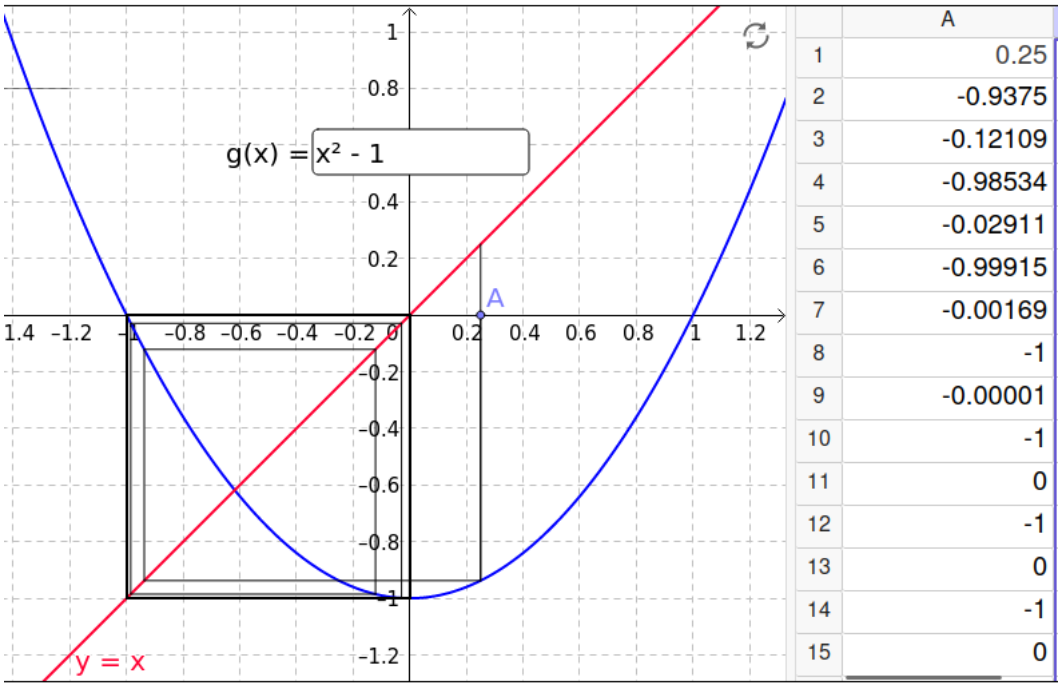
W przykładach (4) i (5) sytuacja jest analogiczna do przykładów (1) i (2)

Rysunek 6: Iteracja graficzna dla `(6)`, $c = -1, x_0 = 0.75$



	A
1	0.75
2	-0.4375
3	-0.80859
4	-0.34618
5	-0.88016
6	-0.22531
7	-0.94923
8	-0.09896
9	-0.99021
10	-0.01949
11	-0.99962
12	-0.00076
13	-1
14	0
15	-1

Rysunek 7: Iteracja graficzna dla (7), $c = -1, x_0 = 0.25$



	A
1	0.25
2	-0.9375
3	-0.12109
4	-0.98534
5	-0.02911
6	-0.99915
7	-0.00169
8	-1
9	-0.00001
10	-1
11	0
12	-1
13	0
14	-1
15	0

Ostatnie dwa przykłady reprezentują układ sprzężenia, który dopiero po kilku iteracjach osiagaja stablilizacje ("Czarne linie zaczynają się pokrywać od pewnego momentu").

6.4 Wnioski

Proces wyznaczania kolejnych wyrazów ciągów określonych rekrencyjnie może mieć różną stabliność w zależności od przyjętych parametrów. Od idealnie stabilnego (przykłady 1, 2, 4, 5) przez stabilizujący się (przykład 6 i 7) do niestabilnego (przykład 3, a także przykłady z poprzedniego zadania). W przypadku stopniowej stabilizacji wartość parametrów wpływa także na jej szybkość (w przykładzie (7) nastąpiła szybciej niż w (6)). Oznacza to, że w celu uzyskania wiarygodnych i użytecznych wyników musimy właściwie wybrać parametry początkowe.