

## Lista 2

### Obliczenia naukowe

Patryk Majewski  
250134

### Zadanie 1

Celem zadania było sprawdzenie, jak zmieni się wynik iloczynu skalarnego dwóch wektorów (obliczanego za pomocą algorytmów z zadania 5 z listy 1) po naniesieniu pozornie niewielkich zmian na dane. Wektory w oryginalnej wersji zadania mają następującą postać:

$$x = [2.718281828, -3.141592654, 1.414213562, 0.5772156649, 0.3010299957]$$
$$y = [1486.2497, 878366.9879, -22.37492, 4773714.647, 0.000185049]$$

Natomiast po naniesieniu zmian opisanych w zadaniu:

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

### Wyniki

Przyjrzyjmy się teraz uzyskanym wynikom:

Float32			Float64		
algorytm	$x \cdot y$	$x' \cdot y$	algorytm	$x \cdot y$	$x' \cdot y$
naprzód	-0.4999443	-0.4999443	naprzód	1.0251881368296672e-10	-0.004296342739891585
wstecz	-0.4543457	-0.4543457	wstecz	-1.5643308870494366e-10	-0.004296342998713953
malejąco	-0.5	-0.5	malejąco	0.0	-0.004296342842280865
rosnąco	-0.5	-0.5	rosnąco	0.0	-0.004296342842280865

W arytmetyce Float32 uzyskaliśmy wyniki identyczne do tych z poprzedniej próby, ponieważ usuwane z liczb najmniej znaczące cyfry i tak znajdowały się na granicy precyzji. Algorytmy działające na arytmetyce Float64, podobnie jak poprzednio, zwracają wyniki różniące się od siebie nawzajem o wartości rzędu  $10^{-10}$ .

Oszacujmy jeszcze dokładną wartość  $x' \cdot y$ . Z poprzedniej listy wiemy, że  $x \cdot y = -1.00657107 \cdot 10^{-11}$ , zatem:

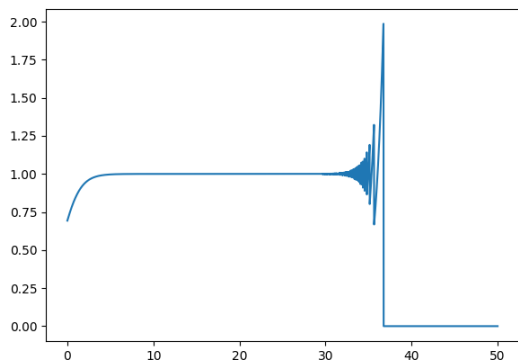
$$x' \cdot y = -1.00657107 \cdot 10^{-11} - 9 \cdot 10^{-10} \cdot 4773714.647 - 7 \cdot 10^{-10} \cdot 0.000185049 = -0.004296343192495245$$

### Wnioski

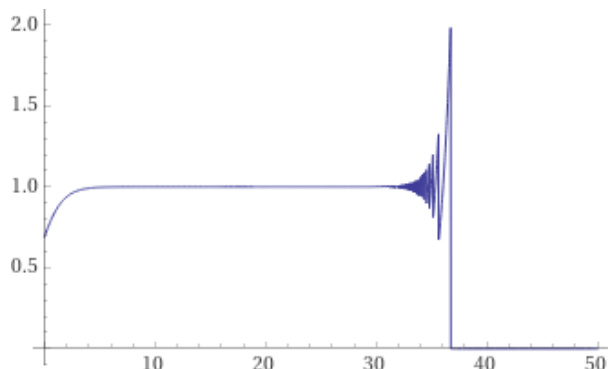
Okazuje się, że o ile błędy bezwzględne w obu próbach mają podobny rząd wielkości, precyzyjna wartość iloczynu zmieniała się znacząco, więc błędy względne zdecydowanie zmalały. Zmiany w inpucie rzędu  $10^{-10}$  spowodowały, że rząd wielkości wyniku zmienił się z  $10^{-11}$  aż do  $10^{-3}$ . Możemy zatem orzec, że stosowane przez nas algorytmy wcale nie są takie nieporadne, ale raczej zadanie jest źle uwarunkowane – nieduże zaburzenie wejścia powoduje spore zmiany uzyskanych wyników.

## Zadanie 2

Celem zadania było narysowanie wykresu funkcji  $e^x \ln(1 + e^{-x})$  w dwóch różnych programach, a następnie porównanie wykresów z wyliczoną granicą w nieskończoności.



Rysunek 1: Wykres w pyplot



Rysunek 2: Wykres w WolframAlpha

Obliczmy wartość granicy funkcji w nieskończoności:

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

Jak widać, wykresy z obu programów nie odzwierciedlają rzeczywistego przebiegu funkcji.

## Wnioski

Zadanie jest źle uwarunkowane. Niewielkie błędy wynikające z ograniczonej precyzji arytmetyki powodują duże wahania wyników (widoczne na wykresie dla wartości  $x$  między 30 a 40). Dla wartości  $x$  mniej więcej od 36 funkcja na obu wykresach spada do 0. Wynika to z faktu, że dla dużych  $x$  zachodzi  $1 + x \approx 1$ , zatem  $\ln(1 + x) \approx 0$ . Jest to niezgodne ze stanem faktycznym (oraz, co za tym idzie, wyliczoną przez nas granicą).

Problem występuje w dwóch znanych i godnych zaufania programach, co potwierdza tezę o złym uwarunkowaniu zadania.

## Zadanie 3

Zadanie polegało na porównaniu metod rozwiązywania układów równań liniowych  $\mathbf{Ax} = \mathbf{b}$  pod względem błędów w obliczeniach. Porównaniem objęliśmy następujące macierze  $\mathbf{A}$ :

- macierze Hilberta stopnia  $n \in \{1, 4, 7, 10, \dots, 52\}$
- macierze losowe stopnia  $n \in \{5, 10, 20\}$  o współczynniku uwarunkowania  $c \in \{1, 10, 10^3, 10^7, 10^{12}, 10^{16}\}$

Układy równań rozwiązywać będziemy za pomocą dwóch metod:

- metody eliminacji Gaussa –  $\mathbf{x} = \mathbf{A}\mathbf{b}$
- metody z macierzą odwrotną –  $\mathbf{x} = \text{inv}(\mathbf{A}) * \mathbf{b}$

Na początku wektor  $\mathbf{b}$  generujemy przez podstawienie do wzoru  $\mathbf{Ax} = \mathbf{b}$  wartości  $\mathbf{x} = (1, \dots, 1)^T$ . Mamy zatem wzorcowy  $\mathbf{x}$ , dzięki któremu wyznaczymy błędy dla obu metod.

## Wyniki

Uzyskane błędy względne dla obu metod wraz z wartościami zwracanymi przez funkcje `rank(A)` oraz `cond(A)` umieszczone zostały w tabelach poniżej.

n	cond(A)	rank(A)	błąd met. Gaussa	błąd met. inwersji
1	1.0	1	0.0	0.0
4	15513.73873892924	4	4.137409622430382e-14	0.0
7	4.75367356583129e8	7	1.2606867224171548e-8	4.713280397232037e-9
10	1.6024416992541715e13	10	8.67039023709691e-5	0.0002501493411824886
13	3.344143497338461e18	11	0.11039701117868264	5.331275639426837
16	7.865467778431645e17	12	54.15518954564602	29.84884207073541
19	6.471953976541591e18	13	9.720589712655698	12.233761393757726
22	1.0361032753348465e19	13	27.234738419698047	24.624027562307738
25	1.3719347461445998e18	13	10.15919484338797	16.93987792970947
28	9.235324245161374e18	14	11.411352448884246	16.937647181923893
31	2.3508867005384925e19	14	12.320597478907297	19.8025699875531
34	4.5616405243414067e18	14	207.42023282963822	256.1422882376434
37	6.763982849658887e18	15	24.22230865170609	67.88591352534833
40	6.507249058549335e18	15	29.420548088052648	45.91591143758286
43	3.895762014266483e19	15	18.044584163261476	25.181811215901565
46	2.394794042936751e19	15	13.46125704739375	18.917439464506856
49	6.148071066691518e18	16	36.45744218491094	38.17923960801737
52	7.605434953598917e18	16	219.4197163854475	440.07012340256716

Tabela 1: Wyniki uzyskane dla macierzy Hilberta

n	c	rank(A)	błąd met. Gaussa	błąd met. inwersji
5	1.0000000000000004	5	2.0471501066083611e-16	2.8522145930998397e-16
5	9.999999999999995	5	2.2752801345137457e-16	1.85775845048325e-16
5	1000.0000000000031	5	4.280689436990162e-14	4.40307229178804e-14
5	1.0000000000484556e7	5	3.524581781059975e-10	3.4973069293889684e-10
5	1.000015859505686e12	5	1.243757562677427e-5	1.3453231792131314e-5
5	8.53020946828605e15	4	0.23288970884402505	0.28168369339562416
10	1.0000000000000007	10	1.9229626863835638e-16	1.7554167342883504e-16
10	9.999999999999984	10	5.063396036227354e-16	4.965068306494546e-16
10	999.999999999952	10	3.597705077052124e-15	4.961094661755162e-15
10	1.0000000001188556e7	10	2.1250582444454513e-10	1.650095112024925e-10
10	9.999517518895454e11	10	3.216999433836553e-5	3.323045278291586e-5
10	9.485933085623844e15	9	0.10457053813876756	0.07108945728798328
20	1.0000000000000016	20	4.800992561380035e-16	3.5975337699988616e-16
20	9.999999999999998	20	3.484402727788097e-16	3.528343968566428e-16
20	999.9999999999326	20	3.502119297771989e-14	2.978180247710411e-14
20	9.99999997573659e6	20	2.881361515888184e-10	2.441582527490554e-10
20	1.0000081002731069e12	20	1.9123537356740552e-5	1.72595950381795e-5
20	9.289295115200694e15	19	0.047259805043657496	0.07164535323818984

Tabela 2: Wyniki uzyskane dla macierzy losowych o ustalonych współczynnikach uwarunkowania

## Wnioski

Macierze Hilberta osiągają bardzo duże wskaźniki uwarunkowania nawet przy stosunkowo niewielkich rozmiarach. Dane zdają się sugerować, że metoda Gaussa zazwyczaj umożliwia nam uzyskanie dokładniejszych wyników dla tego typu macierzy. Obie metody skutkują jednak ogromnymi błędami względnymi.

W przypadku losowych macierzy o ustalonym wskaźniku uwarunkowania trudno jest zaobserwować poważne różnice między dwoma algorytmami – błędy prawie w każdym przypadku okazywały się być tego samego, niewielkiego rzędu. Co więcej, rzędy błędów są podobne dla macierzy o różnych rozmiarach, jeśli ich współczynnik uwarunkowania jest zbliżony.

Taki stan rzeczy sugeruje, że algorytmy działają sprawnie, zatem rozwiązanie układu równań liniowych, w którym współczynniki określa macierz Hilberta jest zadaniem źle uwarunkowanym. Jeśli znamy współczynnik uwarunkowania macierzy A, jesteśmy w stanie oszacować błędy względne rozwiązań, które uzyskamy.

## Zadanie 4

W zadaniu rozpatrywany jest złośliwy wielomian Wilkinsona:

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

W podpunkcie (a) sprawdzimy, jak pakiet `Polynomials` radzi sobie z wyznaczaniem jego pierwiastków. W podpunkcie (b) powtórzymy eksperyment samego Wilkinsona, który polega na marginalnym zaburzeniu jednego ze współczynników wielomianu.

### Porównanie dwóch postaci wielomianu

Celem zadania było znalezienie pierwiastków wielomianu za pomocą funkcji `roots` z pakietu `Polynomials`, a następnie sprawdzenie, jaką wartość osiąga dla nich wielomian zapisany w postaci iloczynowej,  $p(x)$ , i naturalnej,  $P(x)$ . Należało także sprawdzić, jak dalekie od faktycznych są znalezione pierwiastki.

k	$z_k$	$ P(z_k) $	$ p(z_k) $	$ z_k - k $
1	0.999999999996989	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	5.99989245824773	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 3: Wyznaczone wartości dla znalezionych pierwiastków wielomianu

### Wnioski

Jak widać w tabeli 3, wyznaczone pierwiastki nie są do końca dokładne, ale bardzo zbliżone do właściwych. Tabela 4 potwierdza przypuszczenia dotyczące przyczyn takiego stanu rzeczy – prawdziwe pierwiastki nie zerują  $P(x)$ , ponieważ wielomian w postaci naturalnej przechowywany jest niedokładnie z uwagi na ograniczenia arytmetyki Float64. Niektóre współczynniki wielomianu przy niższych potęgach  $x$  są bowiem ogromnymi liczbami, których nie sposób jest zapisać bez pominięcia kilku cyfr znaczących.

Tabela 3 informuje nas też o tym, na czym polega tytułowa złośliwość wielomianu. Bardzo niewielkie oddalenie od faktycznych pierwiastków powoduje, że wartości funkcji windują się do ogromnych wartości (rzędu nawet  $10^{23}$ ). Sugeruje to, że problem wyznaczania pierwiastków wielomianu Wilkinsona jest źle uwarunkowany.

k	$ P(k) $	$ p(k) $
1	0.0	0
2	8192.0	0
3	27648.0	0
4	622592.0	0
5	2.176e6	0
6	8.84736e6	0
7	2.4410624e7	0
8	5.89824e7	0
9	1.45753344e8	0
10	2.27328e8	0
11	4.79074816e8	0
12	8.75003904e8	0
13	1.483133184e9	0
14	2.457219072e9	0
15	3.905712e9	0
16	6.029312e9	0
17	9.116641408e9	0
18	1.333988352e10	0
19	1.9213101568e10	0
20	2.7193344e10	0

Tabela 4: Wyznaczone wartości dla faktycznych pierwiastków wielomianu

## Eksperyment Wilkinsona

Celem zadania było zaburzenie współczynnika  $-210$  stojącego przy  $x^{19}$  o wartość  $-2^{-23}$  i wyjaśnienie zjawiska będącego efektem tego działania.

k	$z_k$	$ P(z_k) $
1	0.9999999999998357 + 0.0im	20259.872313418207
2	2.0000000000550373 + 0.0im	346541.4137593836
3	2.99999999660342 + 0.0im	2.2580597001197007e6
4	4.000000089724362 + 0.0im	1.0542631790395478e7
5	4.99999857388791 + 0.0im	3.757830916585153e7
6	6.000020476673031 + 0.0im	1.3140943325569446e8
7	6.99960207042242 + 0.0im	3.939355874647618e8
8	8.007772029099446 + 0.0im	1.184986961371896e9
9	8.915816367932559 + 0.0im	2.2255221233077707e9
10	10.095455630535774 - 0.6449328236240688im	1.0677921232930157e10
11	10.095455630535774 + 0.6449328236240688im	1.0677921232930157e10
12	11.793890586174369 - 1.6524771364075785im	3.1401962344429485e10
13	11.793890586174369 + 1.6524771364075785im	3.1401962344429485e10
14	13.992406684487216 - 2.5188244257108443im	2.157665405951858e11
15	13.992406684487216 + 2.5188244257108443im	2.157665405951858e11
16	16.73074487979267 - 2.812624896721978im	4.850110893921027e11
17	16.73074487979267 + 2.812624896721978im	4.850110893921027e11
18	19.5024423688181 - 1.940331978642903im	4.557199223869993e12
19	19.5024423688181 + 1.940331978642903im	4.557199223869993e12
20	20.84691021519479 + 0.0im	8.756386551865696e12

Tabela 5: Wyznaczone pierwiastki zaburzonego wielomianu i wartości w nich

## Wnioski

Wyniki zawarte w tabeli 5 potwierdzają przypuszczenia z podpunktu (a). Złe uwarunkowanie problemu znajdowania pierwiastków wielomianu Wilkinsona powoduje, że marginalne zaburzenie skutkuje pojawieniem się rozwiązań zespolonych. Jednocześnie wartości funkcji dla tych "pierwiastków" są bardzo duże.

## Zadanie 5

Celem zadania było wyznaczenie czterdziestu pierwszych wyrazów ciągu  $p_n$  określonego równaniem

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

dla  $p_0 = 0.01$  i  $r = 3$  na trzy sposoby:

1. w arytmetyce Float32
2. w arytmetyce Float32, obcinając  $p_{10}$  do trzech miejsc po przecinku i następny wyraz wyznaczając już z obciętej wartości
3. w arytmetyce Float64

## Wyniki

Wyniki doświadczenia zostały zamieszczone w tabeli 6.

## Wnioski

Wyniki pokazują, że w początkowych iteracjach wyniki w obu arytmetykach są zbieżne. W okolicach 19-tej iteracji model z obcięciem zaczyna zdecydowanie odstawać od pozostałych dwóch. Z kolei około iteracji 22 również modele bez przycięcia zaczynają się rozbiegać. Taki stan rzeczy spowodowany jest ogromną i szybko postępującą kumulacją błędów przy obliczaniu kolejnych wyrazów, która wywodzi się między innymi z faktu podnoszenia poprzedniej wartości do kwadratu. Pierwsze wyrazy wyznaczone są precyzyjnie, kolejne pamiętamy w zaokrągleniu wynikającym z precyzji, a dokładność następnych trudno ocenić ze względu na kumulujący się błąd.

Proces wyznaczania kolejnych wyrazów ciągu  $p_n$  jest niestabilny, ponieważ niewielkie błędy popełnione w początkowej fazie przenoszą się z ogromną siłą na utratę dokładności dalszych wyników. Co więcej, obserwując, że Float32 bez obciążenia pozostał powiązany z Float64 dłużej niż jego odpowiednik z odcięciem, możemy wnioskować, że na tempo drastycznej utraty dokładności mają wpływ zarówno moment popełnienia błędu jak i jego rozmiar. W pewnym momencie również arytmetyka Float64 zacznie popełniać błędy związane z precyzją, więc mimo że jej wyniki dla większych  $n$  są bardziej godne zaufania, nie możemy mieć pewności, że mają związek z rzeczywistością.

n	Float32	Float32 z obciążeniem	Float64
0	0.01	0.01	0.01
1	0.0397	0.0397	0.0397
2	0.15407173	0.15407173	0.15407173000000002
3	0.5450726	0.5450726	0.5450726260444213
4	1.2889781	1.2889781	1.2889780011888006
5	0.1715188	0.1715188	0.17151914210917552
6	0.5978191	0.5978191	0.5978201201070994
7	1.3191134	1.3191134	1.3191137924137974
8	0.056273222	0.056273222	0.056271577646256565
9	0.21559286	0.21559286	0.21558683923263022
10	0.7229306	<b>0.722</b>	0.722914301179573
11	1.3238364	1.3241479	1.3238419441684408
12	0.037716985	0.036488414	0.03769529725473175
13	0.14660022	0.14195944	0.14651838271355924
14	0.521926	0.50738037	0.521670621435246
15	1.2704837	1.2572169	1.2702617739350768
16	0.2395482	0.28708452	0.24035217277824272
17	0.7860428	0.9010855	0.7881011902353041
18	1.2905813	1.1684768	1.2890943027903075
19	0.16552472	0.577893	0.17108484670194324
20	0.5799036	1.3096911	0.5965293124946907
21	1.3107498	0.09289217	1.3185755879825978
22	0.088804245	0.34568182	0.058377608259430724
23	0.3315584	1.0242395	0.22328659759944824
24	0.9964407	0.94975823	0.7435756763951792
25	1.0070806	1.0929108	1.315588346001072
26	0.9856885	0.7882812	0.07003529560277899
27	1.0280086	1.2889631	0.26542635452061003
28	0.9416294	0.17157483	0.8503519690601384
29	1.1065198	0.59798557	1.2321124623871897
30	0.7529209	1.3191822	0.37414648963928676
31	1.3110139	0.05600393	1.0766291714289444
32	0.0877831	0.21460639	0.8291255674004515
33	0.3280148	0.7202578	1.2541546500504441
34	0.9892781	1.3247173	0.29790694147232066
35	1.021099	0.034241438	0.9253821285571046
36	0.95646656	0.13344833	1.1325322626697856
37	1.0813814	0.48036796	0.6822410727153098
38	0.81736827	1.2292118	1.3326056469620293
39	1.2652004	0.3839622	0.0029091569028512065
40	0.25860548	1.093568	0.011611238029748606

Tabela 6: Wartości  $p_n$  wyznaczone z rekurencyjnego równania  $p_{n+1} = p_n + rp_n(1 - p_n)$  w kolejnych iteracjach

## Zadanie 6

Rozpatrywane są ciągi rekurencyjne wyrażone wzorem

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

Należy przeprowadzić obserwację zachowania wyrazów ciągu, gdy

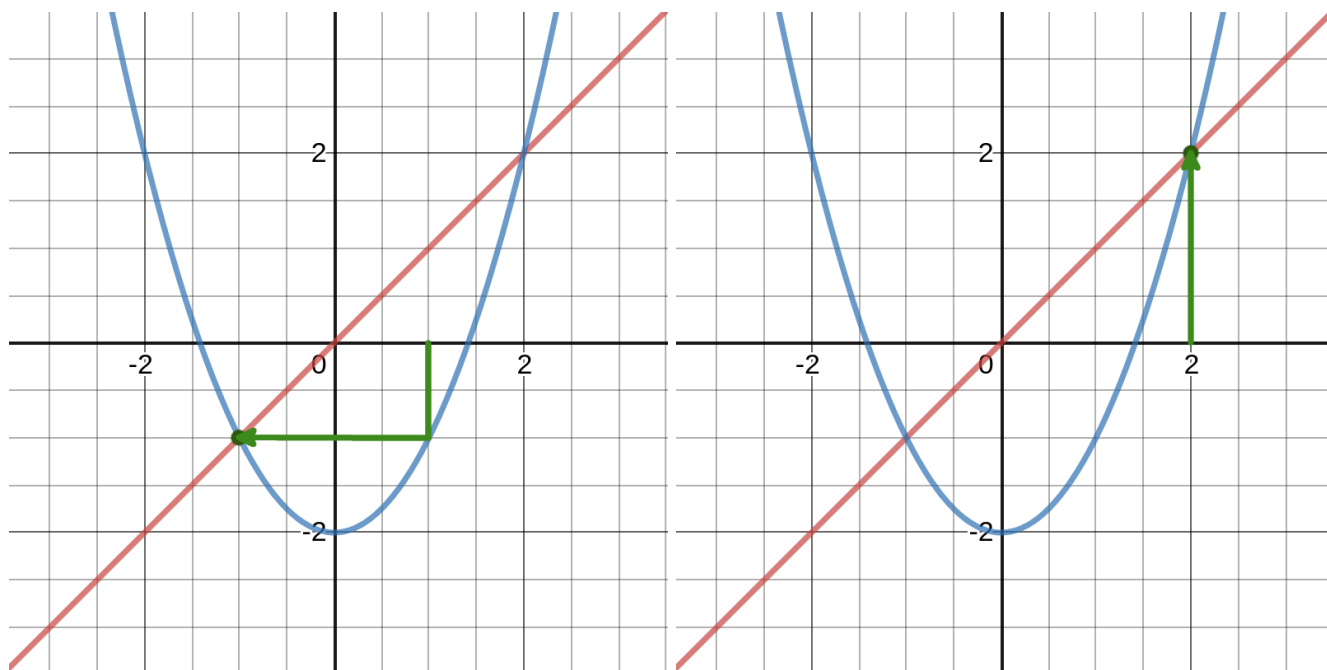
- $c = -2$  oraz  $x_0 \in \{1, 2, 1.9999999999999999\}$
- $c = -1$  oraz  $x_0 \in \{1, -1, 0.75, 0.25\}$

## Wyniki

Wyznaczone wyrazy ciągów zostały zamieszczone w tabelach 7 (dla  $c = -2$ ) i 8 (dla  $c = -1$ ). Graficzne iteracje dla  $c = -2$  przedstawiono na rysunkach 3 i 4, a dla  $c = -1$  na rysunku 5.

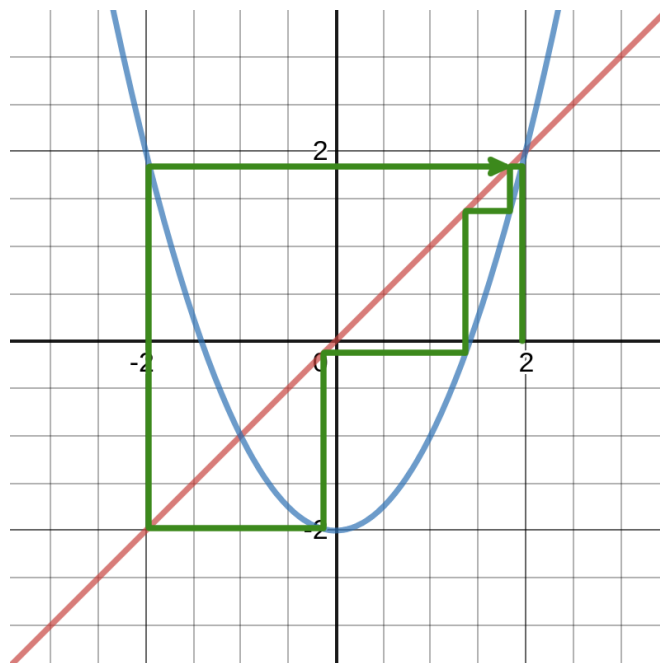
## Wnioski

Proces wyznaczania kolejnych wyrazów ciągów określonych rekurencyjnie może mieć różną stabilność w zależności od przyjętych parametrów – od całkowicie stabilnego (na przykład dla  $x_0 = 1$  przy obu wartościach  $c$ ) przez stabilizujący się ( $c = -1$ ,  $x_0 = 0.75$  i  $0.25$ ) do niestabilnego (jak przy  $c = -2$ ,  $x_0 = 1.99...$  czy też w przykładach z zadania 5). W przypadku stopniowej stabilizacji wartość parametrów wpływa także na jej tempo (w  $c = -1$  szybciej dla  $x_0 = 0.25$  niż  $0.75$ ). Oznacza to, że w celu uzyskania wiarygodnych i użytecznych wyników musimy dokonać doboru odpowiednich wartości parametrów.

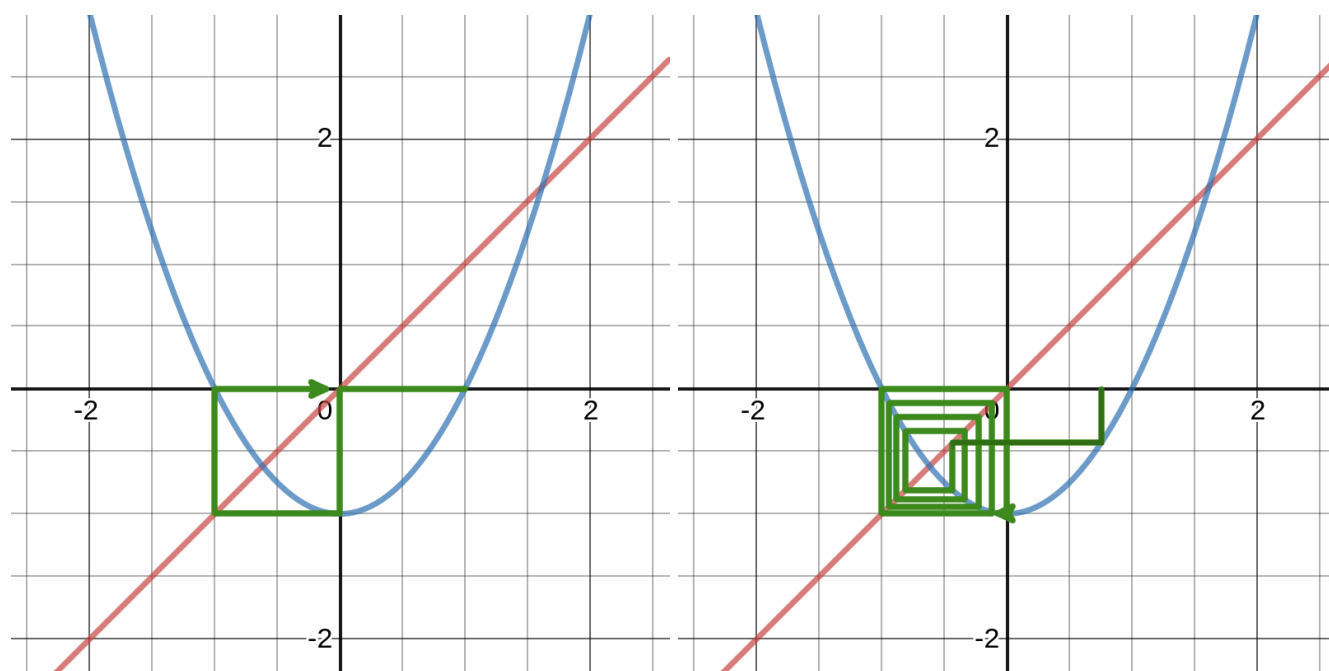


Rysunek 3: Graficzne iteracje wyznaczania  $x_{n+1} = x_n^2 - 2$  dla  $x_0 = 1$  (po lewo) i  $x_0 = 2$  (po prawo). Widzimy, jak strzałki wędrują do punktów stałych, z których już nie ma wyjścia, stąd widoczny w tabeli 7 brak zmian w kolejnych wyrazach tych ciągów.





Rysunek 4: Graficzna iteracja wyznaczania  $x_{n+1} = x_n^2 - 2$  dla  $x_0 \approx 1.99$ . Widzimy, jak strzałka wędruje kolejno między okolicami wartości 1.9, 1.7,  $-0.2$ ,  $-1.9$  i raczej się nie stabilizuje, co mniej więcej zgadza się z danymi z tabeli 7.



Rysunek 5: Graficzne iteracje wyznaczania  $x_{n+1} = x_n^2 - 1$  dla  $x_0 = 1$  (po lewo) i  $x_0 = 0.75$  (po prawo). Dla jedynki wyraźnie widać natychmiastowe zacyklenie się procesu. Dla 0.75 widoczna jest stopniowa stabilizacja procesu aż do momentu, w którym wchodzimy w identyczny cykl. Ścieżki pokrywają się z danymi z tabeli 8.

n	$x_0 = 1$	$x_0 = 2$	$x_0 = 1.999999999999999$
1	-1.0	2.0	1.999999999999996
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	1.999999999590727
7	-1.0	2.0	1.99999999836291
8	-1.0	2.0	1.999999993451638
9	-1.0	2.0	1.999999973806553
10	-1.0	2.0	1.99999989522621
11	-1.0	2.0	1.999999580904841
12	-1.0	2.0	1.999998323619383
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 7: Wyrazy ciągów  $x_{n+1} = x_n^2 + c$  dla  $c = -2$  i ustalonego  $x_0$

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 8: Wyrazy ciągów  $x_{n+1} = x_n^2 + c$  dla  $c = -1$  i ustalonego  $x_0$