

# Obliczenia naukowe: Sprawozdanie 1

Szymon Hładyszewski

8 listopada 2025

## 1 Zadanie 1

### 1.1 Opis problemu

Zadanie polegało na powtórzeniu obliczeń z zadania 5 z listy 1, ale bez ostatnich cyfr w  $x_4$  i  $x_5$ . Poniżej znajdują się dwa oryginalne wektory:

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

Dane zaburzonego wektora x:

- $x_{noised} = [2.718281828, -3.141592654, 1.414213562, 0.577215664, 0.301029995]$

Obliczenia polegały na wyznaczeniu iloczynu skalarnego tych wektorów czterema sposobami:

1. w przód
2. w tył
3. od największego do najmniejszego
4. od najmniejszego do największego

### 1.2 Rozwiązanie

Kod rozwiązania zadania znajduje się w pliku `zadanie1.jl`.

### 1.3 Wyniki i interpretacja

Poniższa tabela ilustruje wyniki obliczeń iloczynu skalarnego dla wektorów z pełną precyzją oraz z obciętymi ostatnimi cyframi. Widzimy różnice w wynikach dla Float64 po obcięciu ostatnich cyfr, co wskazuje na wpływ precyzji obliczeń.

Przypadek	Pełna wartość	Ucięta wartość
<b>Float32</b>		
w przód	-0.4999443	-0.4999443
w tył	-0.4543457	-0.4543457
od największego do najmniejszego	-0.5	-0.5
od najmniejszego do największego	-0.5	-0.5
<b>Float64</b>		
w przód	$1.0251881368 \times 10^{-10}$	-0.00429634274
w tył	$-1.5643308870 \times 10^{-10}$	-0.00429634299
od największego do najmniejszego	0.0	-0.00429634284
od najmniejszego do największego	0.0	-0.00429634284

Tabela 1: Porównanie wyników iloczynów skalarnych (Float32 i Float64) przed i po zmianie

## 1.4 Wnioski

Dla Float32 wyniki się nie różnią, co wynika ze zbyt małej precyzyji tej arytmetyki dla tak mało znaczących zmian. Jednakże dla Float64 mimo tych z pozoru nic nieznaczących detali, widzimy niemałe różnice w wynikach. Widać zatem jak bardzo wrażliwe są te algorytmy na nawet najmniejsze zmiany danych.

## 2 Zadanie 2

### 2.1 Opis problemu

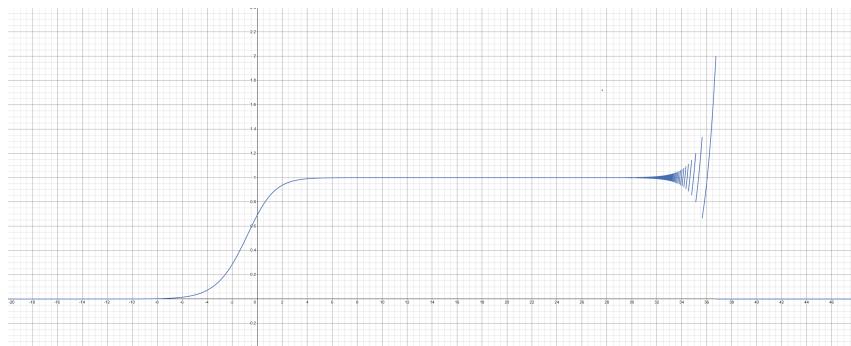
Zadanie polegało na znalezieniu granicy funkcji  $f(x) = e^x \times \ln(1 + e^{-x})$  w nieskończoności oraz zwizualizowaniu wykresu w wybranych dwóch programach do wizualizacji.

### 2.2 Rozwiązanie

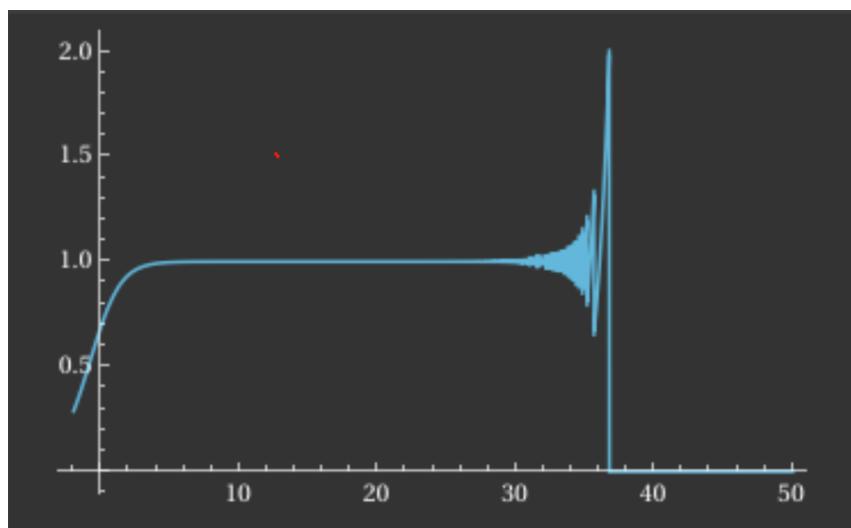
Kod rozwiązania zadania znajduje się w pliku `zadanie2.jl`.

### 2.3 Wyniki i interpretacja

Według programu Julia, granica funkcji  $f(x)$  w nieskończoności wynosi 1. Wykresy funkcji wygenerowane w programach Desmos oraz Wolfram Alpha przedstawiono na rysunkach poniżej. Jak widzimy na poniższych wykresach, znacznie odbiegają one od uzyskanego wyniku.



Rysunek 1: Wykres funkcji  $f(x) = e^x \times \ln(1 + e^{-x})$  w programie Desmos



Rysunek 2: Wykres funkcji  $f(x) = e^x \times \ln(1 + e^{-x})$  w programie Wolfram Alpha

## 2.4 Wnioski

Jak widać na wykresach, obie wizualizacje zaczynają dziwnie oscylować w przedziale od 30 do 40, mimo że granica wynosi 1. Wynika to z faktu, że dla dużego  $x$  mnożymy bardzo duże  $e^x$  oraz bardzo małe  $\ln(1+e^{-x})$ , co prowadzi do dużego znaczenia błędu wynikającego z arytmetyki w stosunku do rzeczywistej wartości. Dla odpowiednio dużego  $x$ ,  $\ln(1 + e^{-x})$  zostaje zaokrąglane do 0, przez co wartość funkcji od pewnego momentu również spada do zera.

## 3 Zadanie 3

### 3.1 Opis problemu

Zadanie polegało na wygenerowaniu macierzy  $A$  oraz rozwiązaniu układu równań liniowych postaci  $Ax = b$  za pomocą dwóch metod:

- eliminacja Gaussa
- obliczenie odwrotności macierzy  $A$  i skorzystanie ze wzoru  $x = A^{-1}b$

Macierz  $A$  miała być wygenerowana na dwa sposoby:

- macierz Hilberta
- macierz losowa o zadanym wskaźniku uwarunkowania

### 3.2 Rozwiązanie

Kod rozwiązania zadania znajduje się w pliku `zadanie3.jl`. Funkcje generujące macierz  $A$  zostały wzięte ze strony wykładowcy na jego prośbę i zastosowane w tymże programie.

### 3.3 Wyniki i interpretacja

$n$	errGauss	errInv	cond( $A$ )	rank( $A$ )
1	0.0	0.0	1.0	1
2	$5.66 \times 10^{-16}$	$1.40 \times 10^{-15}$	19.281	2
3	$8.02 \times 10^{-15}$	0.0	524.06	3
4	$4.14 \times 10^{-14}$	0.0	$1.55 \times 10^4$	4
5	$1.68 \times 10^{-12}$	$3.35 \times 10^{-12}$	$4.77 \times 10^5$	5
6	$2.62 \times 10^{-10}$	$2.02 \times 10^{-10}$	$1.50 \times 10^7$	6
7	$1.26 \times 10^{-8}$	$4.71 \times 10^{-9}$	$4.75 \times 10^8$	7
8	$6.12 \times 10^{-8}$	$3.08 \times 10^{-7}$	$1.53 \times 10^{10}$	8
9	$3.88 \times 10^{-6}$	$4.54 \times 10^{-6}$	$4.93 \times 10^{11}$	9
10	$8.67 \times 10^{-5}$	$2.50 \times 10^{-4}$	$1.60 \times 10^{13}$	10
11	$1.58 \times 10^{-4}$	$7.62 \times 10^{-3}$	$5.22 \times 10^{14}$	10
12	0.13396	0.25899	$1.75 \times 10^{16}$	11
13	0.11040	5.33128	$3.19 \times 10^{18}$	11
14	1.45541	8.71499	$6.20 \times 10^{17}$	11
15	4.69667	7.34464	$3.68 \times 10^{17}$	12
16	54.1552	29.8488	$7.05 \times 10^{17}$	12
17	13.7072	10.5169	$1.25 \times 10^{18}$	12
18	10.2576	24.7621	$2.25 \times 10^{18}$	12
19	102.160	109.946	$6.47 \times 10^{18}$	13
20	108.318	114.344	$1.15 \times 10^{18}$	13

Tabela 2: Wyniki dla macierzy Hilberta

cond( $A$ )	$n$	errGauss	errInv	rank( $A$ )
1	5	$1.79 \times 10^{-16}$	$9.93 \times 10^{-17}$	5
1	10	$3.49 \times 10^{-16}$	$2.19 \times 10^{-16}$	10
1	20	$5.13 \times 10^{-16}$	$4.29 \times 10^{-16}$	20
10	5	$3.18 \times 10^{-16}$	$2.05 \times 10^{-16}$	5
10	10	$6.27 \times 10^{-16}$	$4.85 \times 10^{-16}$	10
10	20	$4.74 \times 10^{-16}$	$5.61 \times 10^{-16}$	20
$10^3$	5	$9.01 \times 10^{-14}$	$1.18 \times 10^{-14}$	5
$10^3$	10	$3.09 \times 10^{-14}$	$3.55 \times 10^{-14}$	10
$10^3$	20	$5.38 \times 10^{-14}$	$5.67 \times 10^{-14}$	20
$10^7$	5	$5.22 \times 10^{-10}$	$5.17 \times 10^{-10}$	5
$10^7$	10	$2.80 \times 10^{-10}$	$3.44 \times 10^{-10}$	10
$10^7$	20	$3.96 \times 10^{-10}$	$3.51 \times 10^{-10}$	20
$10^{12}$	5	$8.16 \times 10^{-6}$	$8.20 \times 10^{-6}$	5
$10^{12}$	10	$7.89 \times 10^{-6}$	$1.14 \times 10^{-5}$	10
$10^{12}$	20	$4.36 \times 10^{-5}$	$4.29 \times 10^{-5}$	20
$10^{16}$	5	0.18657	0.15068	4
$10^{16}$	10	$3.72 \times 10^{-16}$	0.05559	9
$10^{16}$	20	0.48812	0.48500	19

Tabela 3: Błędy względne rozwiązań układów dla macierzy losowych przy rosnącym cond( $A$ )

Analizując powyższe tabele, widzimy, że wraz ze wzrostem rozmiaru macierzy Hilberta, błąd względny zarówno dla metody eliminacji Gaussa, jak i metody wykorzystującej odwrotność macierzy rośnie znacząco.

### 3.4 Wnioski

W macierzy Hilberta zarówno wskaźnik uwarunkowania jak i błędy względne dla poszczególnych metod rosną bardzo szybko wraz ze wzrostem rozmiaru macierzy. Na podstawie tych obserwacji można stwierdzić, że macierz Hilberta jest źle uwarunkowana. Jeśli chodzi o macierze losowe, widać, że ich wielkość błędu względnego mrtodami Gaussa i inwersji zależy przede wszystkim od wskaźnika uwarunkowania, a nie od rozmiaru.

## 4 Zadanie 4

### 4.1 Opis problemu

Zadanie polegało na znalezieniu 20 miejsc zerowych wielomianu Wilkinsona stosując metodę roots() dla jego postaci naturalnej i sprawdzeniu ich dla tejże postaci oraz iloczynowej i wyjaśnienie rozbieżności. Następnie należało zmienić współczynnik przy  $x^{19}$  z  $-210$  na  $-210 - 2^{-23}$  i wyjaśnić zjawisko.

Oto wielomian Wilkinsona w postaci iloczynowej:

$$P(x) = (x - 1)(x - 2)(x - 3)\dots(x - 20)$$

oraz w postaci naturalnej:

$$\begin{aligned} p(x) = & x^{20} - 210x^{19} + 20615x^{18} - 1256850x^{17} + 53327946x^{16} - 1672280820x^{15} + 40171771630x^{14} \\ & - 756111184500x^{13} + 11310276995381x^{12} - 135585182899530x^{11} + 1307535010540395x^{10} \\ & - 10142299865511450x^9 + 63030812099294896x^8 - 311333643161390640x^7 + 1206647803780373360x^6 \\ & - 3599979517947607200x^5 + 8037811822645051776x^4 - 12870931245150988800x^3 + 13803759753640704000x^2 \\ & - 8752948036761600000x + 2432902008176640000 \quad (1) \end{aligned}$$

### 4.2 Rozwiązanie

Kod rozwiązania zadania znajduje się w pliku `zadanie4.jl`.

### 4.3 Wyniki i interpretacja

Poniższa tabela przedstawia znalezione miejsca zerowe i ich sprawdzenia:

$k$	$r_k$	$ P(r_k) $	$ p(r_k) $	$ r_k - k $
1	0.99999999999996989	$3.57 \times 10^4$	$5.52 \times 10^6$	$3.01 \times 10^{-13}$
2	2.00000000000283182	$1.76 \times 10^5$	$7.38 \times 10^{19}$	$2.83 \times 10^{-11}$
3	2.9999999995920965	$2.79 \times 10^5$	$3.32 \times 10^{20}$	$4.08 \times 10^{-10}$
4	3.9999999837375317	$3.03 \times 10^6$	$8.85 \times 10^{20}$	$1.63 \times 10^{-8}$
5	5.000000665769791	$2.29 \times 10^7$	$1.84 \times 10^{21}$	$6.66 \times 10^{-7}$
6	5.999989245824773	$1.29 \times 10^8$	$3.32 \times 10^{21}$	$1.08 \times 10^{-5}$
7	7.000102002793008	$4.81 \times 10^8$	$5.42 \times 10^{21}$	$1.02 \times 10^{-4}$
8	7.999355829607762	$1.64 \times 10^9$	$8.26 \times 10^{21}$	$6.44 \times 10^{-4}$
9	9.002915294362053	$4.88 \times 10^9$	$1.20 \times 10^{22}$	$2.92 \times 10^{-3}$
10	9.990413042481725	$1.36 \times 10^{10}$	$1.66 \times 10^{22}$	$9.59 \times 10^{-3}$
11	11.025022932909318	$3.59 \times 10^{10}$	$2.25 \times 10^{22}$	$2.50 \times 10^{-2}$
12	11.953283253846857	$7.53 \times 10^{10}$	$2.89 \times 10^{22}$	$4.67 \times 10^{-2}$
13	13.07431403244734	$1.96 \times 10^{11}$	$3.81 \times 10^{22}$	$7.43 \times 10^{-2}$
14	13.914755591802127	$3.58 \times 10^{11}$	$4.61 \times 10^{22}$	$8.52 \times 10^{-2}$
15	15.075493799699476	$8.22 \times 10^{11}$	$5.90 \times 10^{22}$	$7.55 \times 10^{-2}$
16	15.946286716607972	$1.55 \times 10^{12}$	$7.01 \times 10^{22}$	$5.37 \times 10^{-2}$
17	17.025427146237412	$3.69 \times 10^{12}$	$8.57 \times 10^{22}$	$2.54 \times 10^{-2}$
18	17.99092135271648	$7.65 \times 10^{12}$	$1.01 \times 10^{23}$	$9.08 \times 10^{-3}$
19	19.00190981829944	$1.14 \times 10^{13}$	$1.20 \times 10^{23}$	$1.91 \times 10^{-3}$
20	19.999809291236637	$2.79 \times 10^{13}$	$1.40 \times 10^{23}$	$1.91 \times 10^{-4}$

Tabela 4: Analiza pierwiastków wielomianu Wilkinsona

Powyzsza tabela pokazuje, że wyliczone pierwiastki wielomianu Wilkinsona różnią się od rzeczywistych wartości, a błąd rośnie wraz z każdym kolejnym pierwiastkiem.

$k$	$r_k$	$ P(r_k) $	$ p(r_k) $	$ r_k - k $
1	0.9999999999998357	$2.03 \times 10^4$	$3.01 \times 10^6$	$1.64 \times 10^{-13}$
2	2.00000000000550373	$3.47 \times 10^5$	$7.38 \times 10^{19}$	$5.50 \times 10^{-11}$
3	2.999999999660342	$2.26 \times 10^6$	$3.32 \times 10^{20}$	$3.40 \times 10^{-9}$
4	4.000000089724362	$1.05 \times 10^7$	$8.85 \times 10^{20}$	$8.97 \times 10^{-8}$
5	4.99999857388791	$3.76 \times 10^7$	$1.84 \times 10^{21}$	$1.43 \times 10^{-6}$
6	6.000020476673031	$1.31 \times 10^8$	$3.32 \times 10^{21}$	$2.05 \times 10^{-5}$
7	6.99960207042242	$3.94 \times 10^8$	$5.42 \times 10^{21}$	$3.98 \times 10^{-4}$
8	8.007772029099446	$1.18 \times 10^9$	$8.29 \times 10^{21}$	$7.77 \times 10^{-3}$
9	8.915816367932559	$2.23 \times 10^9$	$1.16 \times 10^{22}$	$8.42 \times 10^{-2}$
10	$10.0954556305 \pm 0.6449i$	$1.07 \times 10^{10}$	$1.72 \times 10^{22}$	$6.52 \times 10^{-1}$
11	$10.0954556305 \pm 0.6449i$	$1.07 \times 10^{10}$	$1.72 \times 10^{22}$	1.11
12	$11.7938905862 \pm 1.6525i$	$3.14 \times 10^{10}$	$2.86 \times 10^{22}$	1.67
13	$11.7938905862 \pm 1.6525i$	$3.14 \times 10^{10}$	$2.86 \times 10^{22}$	2.05
14	$13.9924066845 \pm 2.5188i$	$2.16 \times 10^{11}$	$4.93 \times 10^{22}$	2.52
15	$13.9924066845 \pm 2.5188i$	$2.16 \times 10^{11}$	$4.93 \times 10^{22}$	2.71
16	$16.7307448798 \pm 2.8126i$	$4.85 \times 10^{11}$	$8.48 \times 10^{22}$	2.91
17	$16.7307448798 \pm 2.8126i$	$4.85 \times 10^{11}$	$8.48 \times 10^{22}$	2.83
18	$19.5024423688 \pm 1.9403i$	$4.56 \times 10^{12}$	$1.32 \times 10^{23}$	2.45
19	$19.5024423688 \pm 1.9403i$	$4.56 \times 10^{12}$	$1.32 \times 10^{23}$	2.00
20	20.8469102152	$8.76 \times 10^{12}$	$1.59 \times 10^{23}$	0.85

Tabela 5: Analiza pierwiastków zaburzonego wielomianu Wilkinsona

W zaburzonym wielomianie widzimy, że niewielka zmiana bardzo znacząco wpłynęła na uzyskane rezultaty. Pojawiła się również część urojona dla pierwiastków od 10 do 19.

## 4.4 Wnioski

W przypadku oryginalnego wielomianu Wilkinsona widzimy, że początkowo pierwiastki są niemal dokładnie wyznaczane, jednak wraz ze wzrostem ich wartości, błędy stają się coraz większe. W rezultacie wartości wielomianu dla tych wyznaczonych miejsc zerowych są bardzo duże i rosną dla coraz większego  $k$ , ponieważ te błędy się kumulują. Widzimy zatem jak wyznaczanie pierwiastków Wilkinsona jest źle uwarunkowane. Podobna sytuacja jest przy niewielkim zaburzeniu wartości współczynnika przy  $x_{19}$ , jednak tutaj błędy są jeszcze większe, a dla pierwiastków od 10 do 19 pojawia się część urojona. Pokazuje to, jak bardzo wielomian ten jest wrażliwy na niewielkie zmiany. Jest to spowodowane ograniczeniami wynikającymi z arytmetyki Float64.

# 5 Zadanie 5

## 5.1 Opis problemu

Zadanie polegało na wykonaniu 40 iteracji zadanego równania rekurencyjnego dla arytmetyki Float64 i Float32 oraz wykonaniu dlaFloat32 10 iteracji, obcięciu wyniku do 0.722 oraz kontynuacji iteracji aż do 40 wykonień.

Oto równanie rekurencyjne:

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

z warunkami:

$$r = 3$$

$$p_0 = 0.01$$

## 5.2 Rozwiązanie

Kod rozwiązania zadania znajduje się w pliku `zadanie5.jl`.

## 5.3 Wyniki i interpretacja

Poniższa tabela przedstawia wyniki uzyskane dla poszczególnych przypadków:

<i>n</i>	Float32 <i>p</i>	Float64 <i>p</i>	Float32_cut <i>p</i>
1	0.0397	0.0397	–
2	0.15407173	0.15407173	–
3	0.5450726	0.54507263	–
4	1.2889781	1.28897800	–
5	0.1715188	0.17151914	–
6	0.5978191	0.59782012	–
7	1.3191134	1.31911379	–
8	0.05627322	0.05627158	–
9	0.21559286	0.21558684	–
10	0.7229306	0.72291430	–
11	1.3238364	1.32384194	1.3241479
12	0.03771699	0.03769530	0.03648841
13	0.14660022	0.14651838	0.14195944
14	0.52192600	0.52167062	0.50738037
15	1.2704837	1.27026177	1.2572169
16	0.2395482	0.24035217	0.28708452
17	0.7860428	0.78810119	0.9010855
18	1.2905813	1.28909430	1.1684768
19	0.16552472	0.17108485	0.5778930
20	0.5799036	0.59652931	1.3096911
21	1.3107498	1.31857559	0.09289217
22	0.08880425	0.05837761	0.34568182
23	0.3315584	0.22328660	1.0242395
24	0.9964407	0.74357568	0.94975823
25	1.0070806	1.31558835	1.0929108
26	0.9856885	0.07003530	0.7882812
27	1.0280086	0.26542635	1.2889631
28	0.9416294	0.85035197	0.17157483
29	1.1065198	1.23211246	0.59798557
30	0.7529209	0.37414649	1.3191822
31	1.3110139	1.07662917	0.05600393
32	0.0877831	0.82912557	0.21460639
33	0.3280148	1.25415465	0.72025780
34	0.9892781	0.29790694	1.3247173
35	1.0210990	0.92538213	0.03424144
36	0.95646656	1.13253226	0.13344833
37	1.0813814	0.68224107	0.48036796
38	0.81736827	1.33260565	1.2292118
39	1.2652004	0.00290916	0.3839622
40	0.25860548	0.01161124	1.0935680

Tabela 6: Porównanie wartości *p* dla Float32, Float64 i Float32\_cut

Analizując powyższą tabelę, widzimy znaczne różnice w wynikach dla różnych arytmetyk oraz po obcięciu wartości w przypadku Float32.

#### 5.4 Wnioski

Dla Float32 i Float64 wyniki są do siebie w miarę podobne aż do 21 iteracji. Później zaczynają znacznie od siebie odbiegać, aFloat32 z obcięciem już od 17 iteracji mocno się różni od reszty. Można zatem wysnuć wniosek, że model logistyczny jest bardzo wrażliwy na dokładność obliczeń, a najdrobniejsze zmiany rażąco wpływają na końcowe wyniki. Widać tutaj jak duży wpływ na wynik mają błędy zaokrągleń i precyzja arytmetyki, szczególnie w przypadku iteracyjnych obliczeń, gdzie błąd może się kumulować. Ostateczne wyniki dla 40. iteracji są skrajnie różne dla każdego z przypadków.

## 6 Zadanie 6

### 6.1 Opis problemu

Zadanie polegało na wykonaniu 40 iteracji zadanego równania rekurencyjnego z ustaloną stałą i pierwszą wartością dla Float64 i zaobserwować zachowanie.

Oto równanie rekurencyjne:

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

z warunkami:

1.  $c = -2$  oraz  $x_0 = 1$
2.  $c = -2$  oraz  $x_0 = 2$
3.  $c = -2$  oraz  $x_0 = 1.999999999999999$
4.  $c = -1$  oraz  $x_0 = -1$
5.  $c = -1$  oraz  $x_0 = 0.75$
6.  $c = -1$  oraz  $x_0 = 0.25$

### 6.2 Rozwiązanie

Kod rozwiązania zadania znajduje się w pliku `zadanie6.jl`. Interpretacja graficzna została wygenerowana z użyciem kalkulatora graficznego GeoGebra.

### 6.3 Wyniki i interpretacja

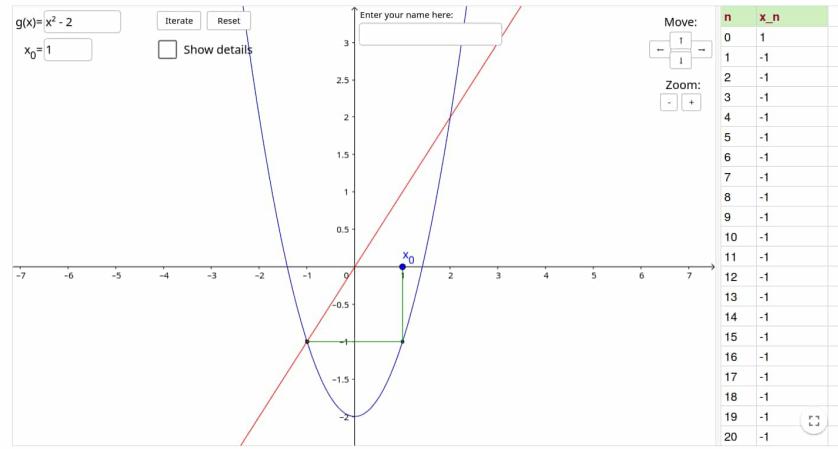
Poniższe tabele przedstawiają wyniki uzyskane dla poszczególnych iteracji:

$n$	$x_0 = -1.0$	$x_0 = 0.75$	$x_0 = 0.25$
1	-1.0	0.75	0.25
2	0.0	-0.4375	-0.9375
3	-1.0	-0.8086	-0.1211
4	0.0	-0.3462	-0.9853
5	-1.0	-0.8802	-0.0291
6	0.0	-0.2253	-0.9992
7	-1.0	-0.9492	-0.0017
8	0.0	-0.0990	-0.999997
9	-1.0	-0.9902	-5.74e-6
10	0.0	-0.0195	-1.0
11	-1.0	-0.9996	-6.59e-11
12	0.0	-7.59e-4	-1.0
13	-1.0	-1.0	0.0
14	0.0	-1.15e-6	-1.0
15	-1.0	-1.0	0.0
16	0.0	-2.66e-12	-1.0
17	-1.0	-1.0	0.0
18	0.0	0.0	-1.0
19	-1.0	-1.0	0.0
20	0.0	0.0	-1.0
21	-1.0	-1.0	0.0
22	0.0	0.0	-1.0
23	-1.0	-1.0	0.0
24	0.0	0.0	-1.0
25	-1.0	-1.0	0.0
26	0.0	0.0	-1.0
27	-1.0	-1.0	0.0
28	0.0	0.0	-1.0
29	-1.0	-1.0	0.0
30	0.0	0.0	-1.0
31	-1.0	-1.0	0.0
32	0.0	0.0	-1.0
33	-1.0	-1.0	0.0
34	0.0	0.0	-1.0
35	-1.0	-1.0	0.0
36	0.0	0.0	-1.0
37	-1.0	-1.0	0.0
38	0.0	0.0	-1.0
39	-1.0	-1.0	0.0
40	0.0	0.0	-1.0

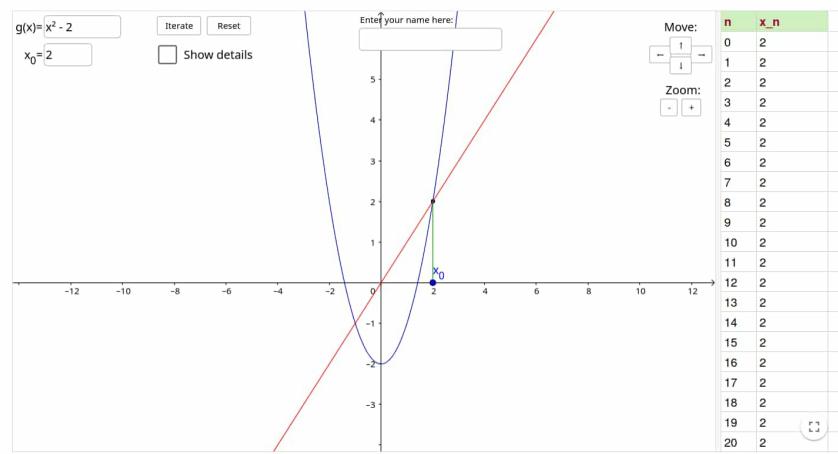
Tabela 7: Iteracje dla  $c = -1$  i różnych wartości początkowych  $x_0$

$n$	$x_0 = 1.0$	$x_0 = 2.0$	$x_0 = 1.999999999999999$
1	1.0	2.0	2.000000000000000
2	-1.0	2.0	1.999999999999996
3	-1.0	2.0	1.999999999999984
4	-1.0	2.0	1.999999999999936
5	-1.0	2.0	1.999999999999744
6	-1.0	2.0	1.99999999998977
7	-1.0	2.0	1.99999999995907
8	-1.0	2.0	1.99999999983629
9	-1.0	2.0	1.99999999934516
10	-1.0	2.0	1.9999999738066
11	-1.0	2.0	1.9999998952262
12	-1.0	2.0	1.9999995809048
13	-1.0	2.0	1.9999983236194
14	-1.0	2.0	1.9999932944778
15	-1.0	2.0	1.9999731779157
16	-1.0	2.0	1.9998927117349
17	-1.0	2.0	1.9995708480908
18	-1.0	2.0	1.99982834107804
19	-1.0	2.0	1.99931339377896
20	-1.0	2.0	1.99725404654395
21	-1.0	2.0	1.98902372643618
22	-1.0	2.0	1.95621538432605
23	-1.0	2.0	1.82677862987391
24	-1.0	2.0	1.33712016256400
25	-1.0	2.0	-0.21210967086482
26	-1.0	2.0	-1.95500948752562
27	-1.0	2.0	1.82206209631517
28	-1.0	2.0	1.31991028282844
29	-1.0	2.0	-0.25783684528374
30	-1.0	2.0	-1.93352016121413
31	-1.0	2.0	1.73850021382151
32	-1.0	2.0	1.02238299345744
33	-1.0	2.0	-0.95473301468901
34	-1.0	2.0	-1.08848487066284
35	-1.0	2.0	-0.81520068633810
36	-1.0	2.0	-1.33544784099389
37	-1.0	2.0	-0.21657906398475
38	-1.0	2.0	-1.95309350904349
39	-1.0	2.0	1.81457425506782
40	-1.0	2.0	1.29267972715492

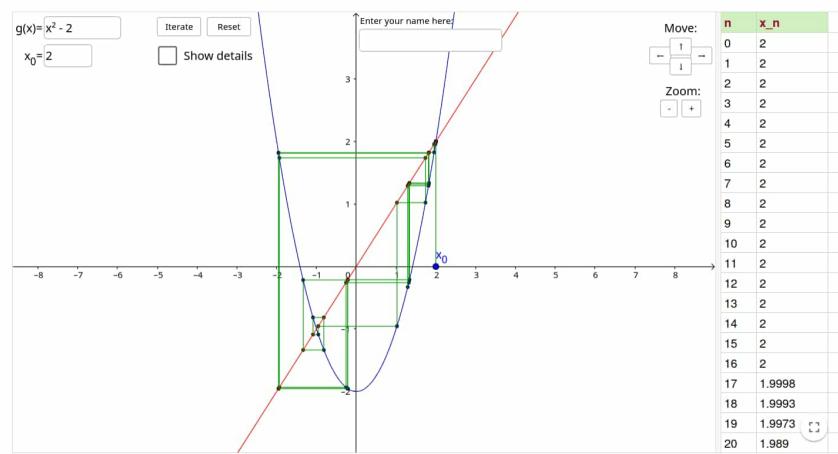
Tabela 8: Iteracje dla  $c = -2$  i różnych wartości początkowych  $x_0$



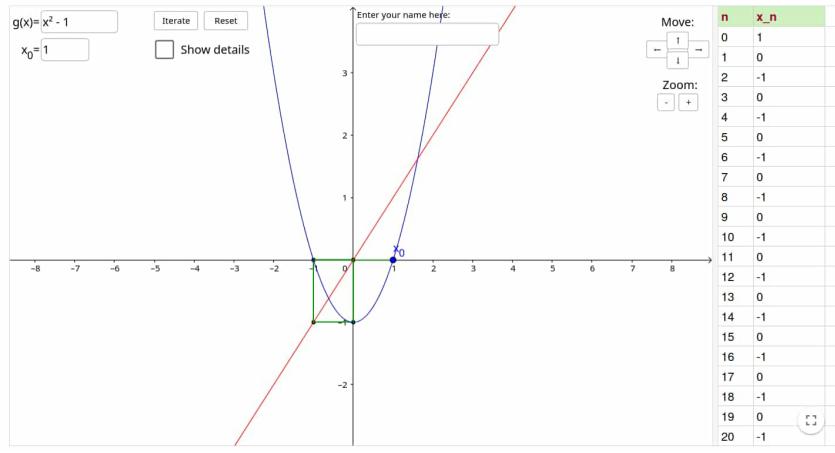
Rysunek 3: Wykres dla podpunktu 1



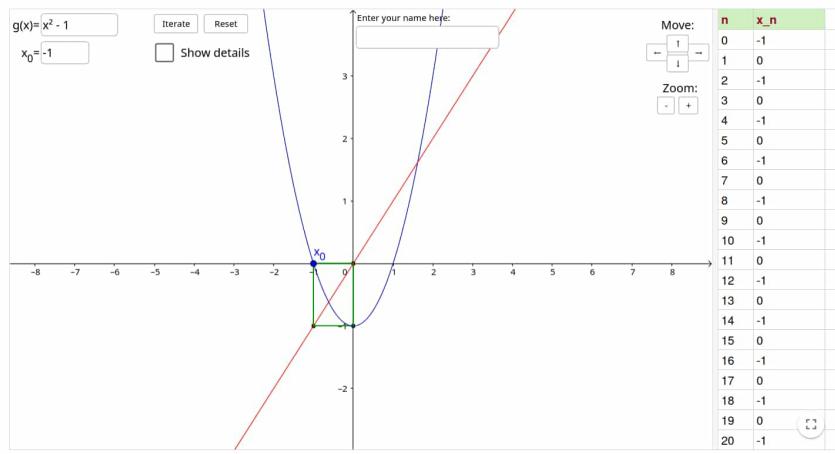
Rysunek 4: Wykres dla podpunktu 2



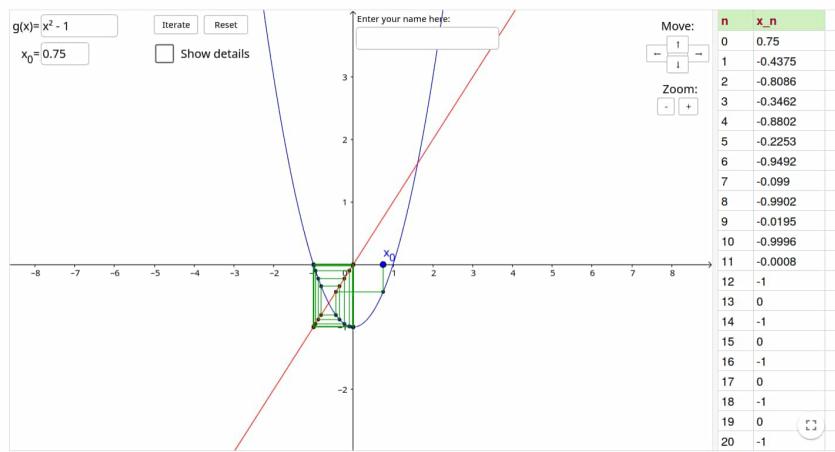
Rysunek 5: Wykres dla podpunktu 3



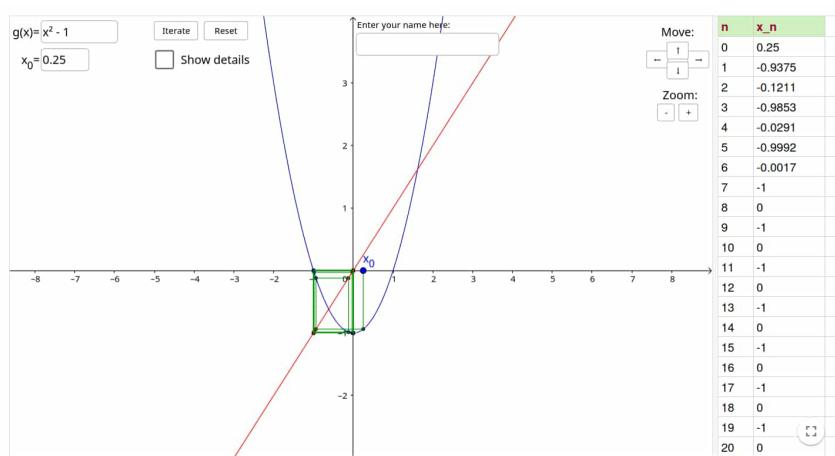
Rysunek 6: Wykres dla podpunktu 4



Rysunek 7: Wykres dla podpunktu 5



Rysunek 8: Wykres dla podpunktu 6



Rysunek 9: Wykres dla podpunktu 7

#### **6.4 Wnioski**

Dla podpunktu 1 i 2 widzimy, że kolejne iteracje nie wpływają na zmianę wartości - pozostają one stałe. Graficzne iteracje dla tych przypadków prowadzą do zachowania stabilnego. W podpunkcie 3. mimo dużego podobieństwa do podpunktu 2, widzimy, że uzyskane wartości są znacznie różne. Z kolejnymi iteracjami wyniki coraz bardziej odbiegają od początkowej wartości. Następne podpunkty, gdzie stała  $c = -1$ , z każdą kolejną iteracją wyniki zaczynają się coraz bardziej stabilizować, wchodząc w cykl  $(0, -1)$ . W tym wypadku wartość  $x_0$  decyduje jedynie o tym, jak szybko osiągnięty zostanie tenże cykl.