

Lista nr 2

1. Zadanie 1

1.1. Opis problemu

Tutaj wracam się do zadania z poprzedniej listy polegającego na obliczeniu iloczynu skalarnego różnymi algorytmami. Tym razem wektor x zostaje bardzo nieznacznie zaburzony.

$$\tilde{x} = [x_1, x_2, x_3, x_4 + \delta_1, x_5 + \delta_2]$$

$$\text{gdzie } \delta_1 = -9 \cdot 10^{-10} \text{ i } \delta_2 = -7 \cdot 10^{-10}$$

W tabeli umieściłem też poprzednie wyniki. Dla przypomnienia dokładny wynik działania $x \cdot y$ wynosił $-1.00657107 \cdot 10^{-11}$.

1.2. Wyniki

Float64				
Kolejność	Z zaburzeniem		Bez zaburzenia	
	Wynik	Błąd względny	Wynik	Błąd względny
Do przodu	-0.004296	$4.2683 \cdot 10^8$	$1.0251881 \cdot 10^{-10}$	11.185
W tył	-0.004296	$4.2683 \cdot 10^8$	$-1.5643309 \cdot 10^{-10}$	14.541
Od max do min	-0.004296	$4.2683 \cdot 10^8$	0.0	1.0
Od min do max	-0.004296	$4.2683 \cdot 10^8$	0.0	1.0

Tabela 1: Wyniki w precyzji Float64

Float32				
Kolejność	Z zaburzeniem		Bez zaburzenia	
	Wynik	Błąd względny	Wynik	Błąd względny
Do przodu	-0.4999	$4.9668 \cdot 10^{10}$	-0.4999443	$4.9668 \cdot 10^{10}$
W tył	-0.4543	$4.5138 \cdot 10^{10}$	-0.4543457	$4.5138 \cdot 10^{10}$
Od max do min	-0.5	$4.9674 \cdot 10^{10}$	-0.5	$4.9674 \cdot 10^{10}$
Od min do max	-0.5	$4.9674 \cdot 10^{10}$	-0.5	$4.9674 \cdot 10^{10}$

Tabela 2: Wyniki w precyzji Float32

1.3. Wnioski

Niezależnie od algorytmu, którym wyliczaliśmy wartość wyrażenia, te niewielkie zaburzenia przeniosły się na ogromny błąd względny wyniku. To eksperymentalnie potwierdza tezę przedstawioną na wykładzie, że liczenie iloczynu skalarnego jest zadaniem źle uwarunkowanym, gdy ma wartość bliską zeru, tj. wektory są prawie prostopadłe.

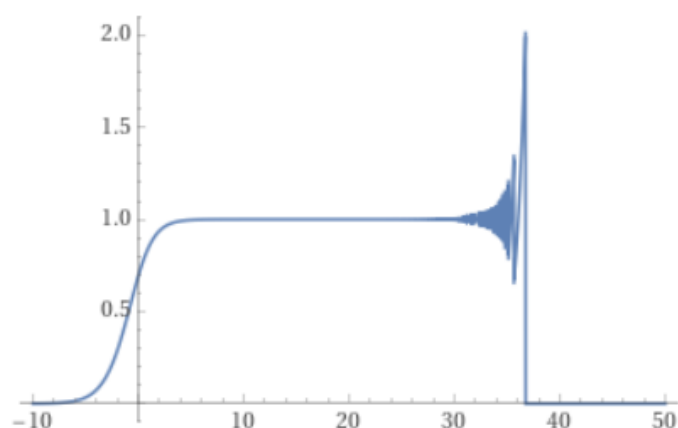
2. Zadanie 2

2.1. Opis problemu

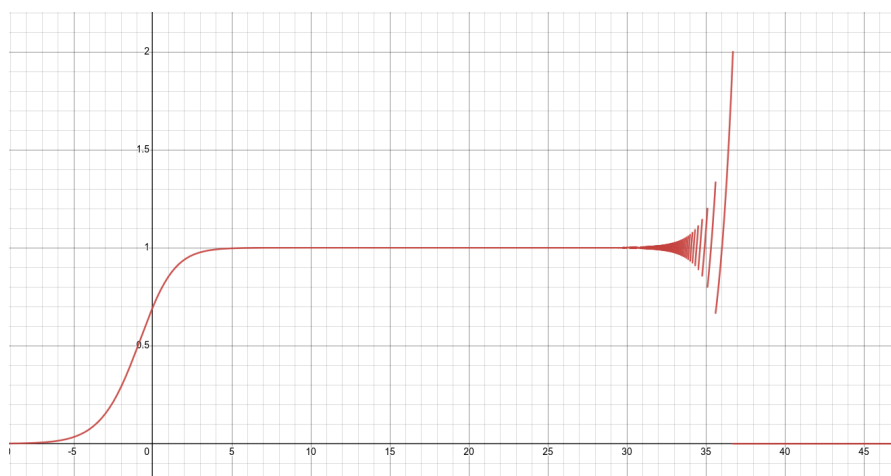
To zadanie polega na przetestowaniu, jak programy do wizualizacji radzą sobie z narysowaniem funkcji $f(x) = e^x \ln(1 + e^{-x})$ i spojrzeniu co się dzieje w granicy tej funkcji.

- Jako pierwszego programu użyłem Wolfram Alpha, któremu przekazałem polecenie `plot e^x * ln(1 + e^(-x)) from -10 to 50`.
- Drugą wizualizację wykonałem w internetowym kalkulatorze graficznym Desmos.
- Trzecią wizualizację przeprowadziłem w języku *Julia* poprzez policzenie i narysowanie wartości funkcji w zakresie $[-10, 50]$.

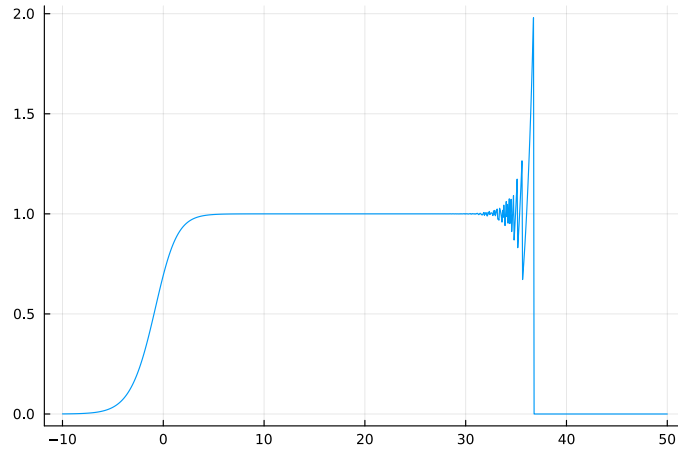
2.2. Wyniki



Rysunek 1: Wykres $f(x)$ narysowany przy pomocy Wolfram Alpha



Rysunek 2: Wykres $f(x)$ narysowany przy pomocy Desmos

Rysunek 3: Wykres $f(x)$ narysowany przy pomocy języka *Julia*

2.3. Wnioski

Żaden z wypróbowanych programów nie poradził sobie z narysowaniem tej funkcji. Granica w nieskończoności wynosi 1, a jednak według komputera funkcja osiąga wartość 0 dla każdego x większego od 38.

$$\lim_{x \rightarrow \infty} e^x \ln(1 + e^{-x}) = \lim_{x \rightarrow \infty} \frac{\ln(1 + e^{-x})}{e^{-x}} \stackrel{\text{d'H}}{=} \lim_{x \rightarrow \infty} -\frac{1}{e^x + 1} \cdot -\frac{1}{e^{-x}} = \lim_{x \rightarrow \infty} \frac{1}{1 + e^{-x}} = 1$$

W celu wyjaśnienia tego zjawiska przeprowadziłem analizę błędu wyliczania wartości tej funkcji. Dla uproszczenia założyłem, że jedynym działaniem, które wprowadziło swój błąd było dodawanie $1 + e^{-x}$ i ten błąd wynosił δ .

$$\begin{aligned} \frac{|e^x \ln((1 + e^{-x})(1 + \delta)) - e^x \ln(1 + e^{-x})|}{|e^x \ln(1 + e^{-x})|} &= \frac{|\ln(1 + e^{-x}) + \ln(1 + \delta) - \ln(1 + e^{-x})|}{|\ln(1 + e^{-x})|} = \\ &= \left| \frac{1}{\ln(1 + e^{-x})} \right| |\ln(1 + \delta)| \leq \left| \frac{1}{\ln(1 + e^{-x})} \right| \varepsilon \approx e^x \varepsilon \end{aligned}$$

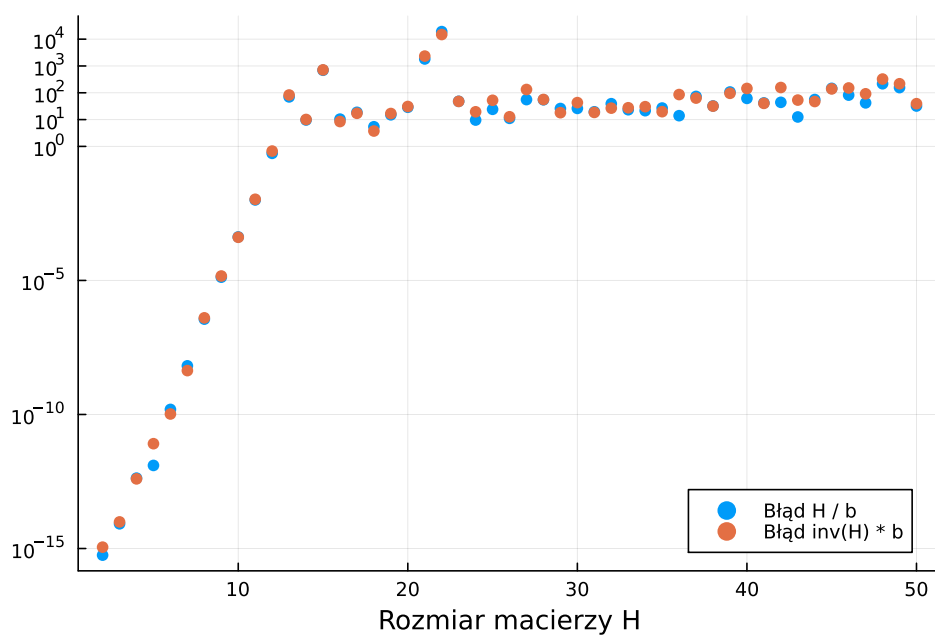
Stąd wyliczanie tej funkcji staje się niebezpiecznie nawet dla względnie niedużych x . W ogólności należy uważać na mnożenie ze sobą liczby bardzo dużej z liczbą bardzo małą.

3. Zadanie 3

3.1. Opis problemu

W tym zadaniu poddane testowi jest rozwiązywanie układu równań zadanego macierzą. Dla każdej macierzy generowany jest wektor prawych stron poprzez wymnożenie $A \cdot x$, gdzie $x = [1, 1, \dots, 1]$ będzie dokładnym rozwiązaniem układu równań. Następnie wyliczane są błędy względne wyników dla dwóch algorytmów - eliminacji Gaussa wywołanego poleceniem $A \setminus b$ oraz obliczania wyrażenia $A^{-1}b$. Eksperymenty zostały przeprowadzone na macierzach Hilberta o rosnących rozmiarach oraz losowo wygenerowanych macierzach z danym współczynnikiem uwarunkowania c . Wyniki losowe zostały uśrednione dla próbki 1000.

3.2. Wyniki



Rysunek 4: Wyniki dla macierzy Hilberta w formie wykresu

n	rank(A)	cond(A)	Błąd $A \setminus b$	Błąd $\text{inv}(A) * b$
2	2	19.2815	$5.66105 \cdot 10^{-16}$	$1.12402 \cdot 10^{-15}$
3	3	524.057	$8.35106 \cdot 10^{-15}$	$9.82553 \cdot 10^{-15}$
4	4	15513.7	$4.22673 \cdot 10^{-13}$	$3.96 \cdot 10^{-13}$
5	5	476607.0	$1.25683 \cdot 10^{-12}$	$8.12817 \cdot 10^{-12}$
6	6	$1.49511 \cdot 10^7$	$1.54351 \cdot 10^{-10}$	$1.04238 \cdot 10^{-10}$
7	7	$4.75367 \cdot 10^8$	$6.5208 \cdot 10^{-9}$	$4.32992 \cdot 10^{-9}$
8	8	$1.52576 \cdot 10^{10}$	$3.60105 \cdot 10^{-7}$	$4.02368 \cdot 10^{-7}$
9	9	$4.93153 \cdot 10^{11}$	$1.3217 \cdot 10^{-5}$	$1.46268 \cdot 10^{-5}$
10	10	$1.6025 \cdot 10^{13}$	0.000419417	0.000407149
11	10	$5.22478 \cdot 10^{14}$	0.0100491	0.010646
12	11	$1.64259 \cdot 10^{16}$	0.550211	0.669789
13	11	$4.49367 \cdot 10^{18}$	70.1556	82.6668
14	11	$3.21984 \cdot 10^{17}$	9.64964	10.0947
15	12	$3.36601 \cdot 10^{17}$	692.43	715.741
16	12	$2.24994 \cdot 10^{18}$	10.4147	8.44214
17	12	$6.26205 \cdot 10^{17}$	18.6758	17.158
18	12	$3.26663 \cdot 10^{18}$	5.40548	3.74241
19	13	$3.4623 \cdot 10^{18}$	15.0739	16.8477
20	13	$6.80697 \cdot 10^{18}$	28.7927	30.7512

Tabela 3: Wyniki dla macierzy Hilberta w formie tabeli

$n = 5$				
c	rank(A)	cond(A)	Błąd $A \setminus b$	Błąd $\text{inv}(A) * b$
1	5	1.0	$1.72531 \cdot 10^{-16}$	$1.69426 \cdot 10^{-16}$
10	5	10.0	$2.57554 \cdot 10^{-16}$	$2.60206 \cdot 10^{-16}$
10^3	5	1000.0	$1.78834 \cdot 10^{-14}$	$1.86758 \cdot 10^{-14}$
10^7	5	$1.0 \cdot 10^7$	$1.82978 \cdot 10^{-10}$	$1.94604 \cdot 10^{-10}$
10^{12}	5	$9.99999 \cdot 10^{11}$	$1.7274 \cdot 10^{-5}$	$1.81588 \cdot 10^{-5}$
10^{16}	4	$7.21813 \cdot 10^{16}$	0.205524	0.214423

Tabela 4: Wyniki dla macierzy R_5

$n = 10$				
c	rank(A)	cond(A)	Błąd $A \setminus b$	Błąd $\text{inv}(A) * b$
1	10	1.0	$2.76712 \cdot 10^{-16}$	$2.5242 \cdot 10^{-16}$
10	10	10.0	$3.46691 \cdot 10^{-16}$	$3.35893 \cdot 10^{-16}$
10^3	10	1000.0	$1.89975 \cdot 10^{-14}$	$1.94355 \cdot 10^{-14}$
10^7	10	$1.0 \cdot 10^7$	$1.78784 \cdot 10^{-10}$	$1.86803 \cdot 10^{-10}$
10^{12}	10	$1.0 \cdot 10^{12}$	$1.82771 \cdot 10^{-5}$	$1.88361 \cdot 10^{-5}$
10^{16}	9	$2.34787 \cdot 10^{16}$	0.197388	0.200439

Tabela 5: Wyniki dla macierzy R_{10}

$n = 20$				
c	rank(A)	cond(A)	Błąd $A \setminus b$	Błąd $\text{inv}(A) * b$
1	20	1.0	$4.81451 \cdot 10^{-16}$	$4.05122 \cdot 10^{-16}$
10	20	10.0	$5.10347 \cdot 10^{-16}$	$4.86107 \cdot 10^{-16}$
10^3	20	1000.0	$1.90056 \cdot 10^{-14}$	$1.93965 \cdot 10^{-14}$
10^7	20	$1.0 \cdot 10^7$	$1.91321 \cdot 10^{-10}$	$1.95645 \cdot 10^{-10}$
10^{12}	20	$1.0 \cdot 10^{12}$	$1.88181 \cdot 10^{-5}$	$1.93414 \cdot 10^{-5}$
10^{16}	19	$4.86352 \cdot 10^{16}$	0.212953	0.217771

Tabela 6: Wyniki dla macierzy R_{20}

3.3. Wnioski

Nawet dla niedużych macierzy nie da się rozwiązać układu równań z pewną dokładnością, jeżeli są złośliwe. To zadanie jest dla takich macierzy źle uwarunkowane, na co wskazuje fakt, że błędy są takie same niezależnie od użytego algorytmu. Wyniki wskazują, że współczynnik uwarunkowania może osiągnąć ogromne wartości. Macierze Hilberta są w tym przypadku szczególnie złośliwe, osiągając współczynnik uwarunkowania rzędu 10^{16} już dla niewielkiego $n = 12$. Można też zauważyć na losowo generowanych macierzach, że błędy w istocie zależały głównie od uwarunkowania, a nie miał na nie wpływ rozmiar macierzy.

4. Zadanie 4

4.1. Opis problemu

W tym zadaniu badam wielomian Wilkinsona wyznaczony wzorem $p(x) = \prod_{i=1}^{20} (x - i)$. Najpierw używam jego naturalnej $P(x)$ i przy użyciu pakietu `Polynomials` wyliczam jego pierwiastki z_k . Potem porównuję ze sobą wartości $|P(z_k)|$, $|p(z_k)|$ i błąd $|z_k - k|$. W drugiej części zadania dodatkowo zaburzyłem współczynnik a_{19} o -2^{-23} .

4.2. Wyniki

z_k	$ P(z_k) $	$ p(z_k) $	$ z_k - k $
0.999999999999997	35696.5	36626.4	$3.01092 \cdot 10^{-13}$
2.000000000000283	176253.0	181304.0	$2.83182 \cdot 10^{-11}$
2.99999999995921	279158.0	290172.0	$4.07903 \cdot 10^{-10}$
3.9999999837375	$3.02711 \cdot 10^6$	$2.04154 \cdot 10^6$	$1.62625 \cdot 10^{-8}$
5.0000006657698	$2.29175 \cdot 10^7$	$2.08946 \cdot 10^7$	$6.6577 \cdot 10^{-7}$
5.9999892458248	$1.29024 \cdot 10^8$	$1.12505 \cdot 10^8$	$1.07542 \cdot 10^{-5}$
7.000102002793	$4.80511 \cdot 10^8$	$4.57291 \cdot 10^8$	0.000102003
7.9993558296078	$1.63795 \cdot 10^9$	$1.55565 \cdot 10^9$	0.00064417
9.0029152943621	$4.87707 \cdot 10^9$	$4.68782 \cdot 10^9$	0.00291529
9.9904130424817	$1.36386 \cdot 10^{10}$	$1.26346 \cdot 10^{10}$	0.00958696
11.025022932909	$3.58563 \cdot 10^{10}$	$3.30013 \cdot 10^{10}$	0.0250229
11.953283253847	$7.53333 \cdot 10^{10}$	$7.38853 \cdot 10^{10}$	0.0467167
13.074314032447	$1.9606 \cdot 10^{11}$	$1.84762 \cdot 10^{11}$	0.074314
13.914755591802	$3.57513 \cdot 10^{11}$	$3.55143 \cdot 10^{11}$	0.0852444
15.075493799699	$8.21627 \cdot 10^{11}$	$8.4232 \cdot 10^{11}$	0.0754938
15.946286716608	$1.5515 \cdot 10^{12}$	$1.57073 \cdot 10^{12}$	0.0537133
17.025427146237	$3.69474 \cdot 10^{12}$	$3.31698 \cdot 10^{12}$	0.0254271
17.990921352716	$7.65011 \cdot 10^{12}$	$6.34485 \cdot 10^{12}$	0.00907865
19.001909818299	$1.14353 \cdot 10^{13}$	$1.22857 \cdot 10^{13}$	0.00190982
19.999809291237	$2.79241 \cdot 10^{13}$	$2.31831 \cdot 10^{13}$	0.000190709

Tabela 7: Wyniki dla niezaburzonego wielomianu

\tilde{z}_k	$ \tilde{P}(\tilde{z}_k) $	$ p(\tilde{z}_k) $	$ \tilde{z}_k - k $
0.9999999999998	20259.9	19987.9	$1.64313 \cdot 10^{-13}$
2.000000000006	346541.0	352369.0	$5.50373 \cdot 10^{-11}$
2.999999996604	$2.25806 \cdot 10^6$	$2.41624 \cdot 10^6$	$3.39658 \cdot 10^{-9}$
4.0000000897244	$1.05426 \cdot 10^7$	$1.12637 \cdot 10^7$	$8.97244 \cdot 10^{-8}$
4.9999985738879	$3.75783 \cdot 10^7$	$4.47574 \cdot 10^7$	$1.42611 \cdot 10^{-6}$
6.000020476673	$1.31409 \cdot 10^8$	$2.1421 \cdot 10^8$	$2.04767 \cdot 10^{-5}$
6.9996020704224	$3.93936 \cdot 10^8$	$1.78462 \cdot 10^9$	0.00039793
8.0077720290994	$1.18499 \cdot 10^9$	$1.8687 \cdot 10^{10}$	0.00777203
8.9158163679326	$2.22552 \cdot 10^9$	$1.37463 \cdot 10^{11}$	0.0841836
$10.0955 - 0.644933 \text{ im}$	$1.06779 \cdot 10^{10}$	$1.49007 \cdot 10^{12}$	0.651959
$10.0955 + 0.644933 \text{ im}$	$1.06779 \cdot 10^{10}$	$1.49007 \cdot 10^{12}$	1.11092
$11.7939 - 1.65248 \text{ im}$	$3.1402 \cdot 10^{10}$	$3.29628 \cdot 10^{13}$	1.66528
$11.7939 + 1.65248 \text{ im}$	$3.1402 \cdot 10^{10}$	$3.29628 \cdot 10^{13}$	2.04582
$13.9924 - 2.51882 \text{ im}$	$2.15767 \cdot 10^{11}$	$9.54602 \cdot 10^{14}$	2.51884
$13.9924 + 2.51882 \text{ im}$	$2.15767 \cdot 10^{11}$	$9.54602 \cdot 10^{14}$	2.71288
$16.7307 - 2.81262 \text{ im}$	$4.85011 \cdot 10^{11}$	$2.74211 \cdot 10^{16}$	2.906
$16.7307 + 2.81262 \text{ im}$	$4.85011 \cdot 10^{11}$	$2.74211 \cdot 10^{16}$	2.82548
$19.5024 - 1.94033 \text{ im}$	$4.5572 \cdot 10^{12}$	$4.25249 \cdot 10^{17}$	2.45402
$19.5024 + 1.94033 \text{ im}$	$4.5572 \cdot 10^{12}$	$4.25249 \cdot 10^{17}$	2.00433
20.8469	$8.75639 \cdot 10^{12}$	$1.37437 \cdot 10^{18}$	0.84691

Tabela 8: Wyniki dla zaburzonego wielomianu

4.3. Wnioski

Wielomian Wilkinsona okazał się wyjątkowo złośliwy przy próbie wyliczenia jego pierwiastków. W pierwszej części zadania mogłoby się wydawać, że różnice między wyliczonym pierwiastkiem a prawdziwym są niewielkie. Jednak podczas wyliczania wartości wielomianu w danym pierwiastku następuje kumulacja tego niewielkiego błędu i otrzymałem wartości znacznie oddalone od oczekiwanego 0.

Już na samym początku pojawiło się względnie niewielkie zaburzenie współczynników wielomianu. $a_4 = 8037811822645051776$, a jednak gdy zostanie to zapisane w komputerze w precyzji Float64, to po odczytaniu dostanie się $\tilde{a}_4 = 8037811822645051392$ - te wartości różnią się o 384.

W drugiej części zadania sam nieznacznie zaburzyłem jeden współczynnik, a to wystarczyło, żeby oryginalnie całkowite pierwiastki wielomianu stały się nie dosyć że zespolone, to jeszcze znacznie oddalone. To potwierdza tezę, że zadanie wyznaczania pierwiastków tego wielomianu jest źle uwarunkowane.

5. Zadanie 5

5.1. Opis problemu

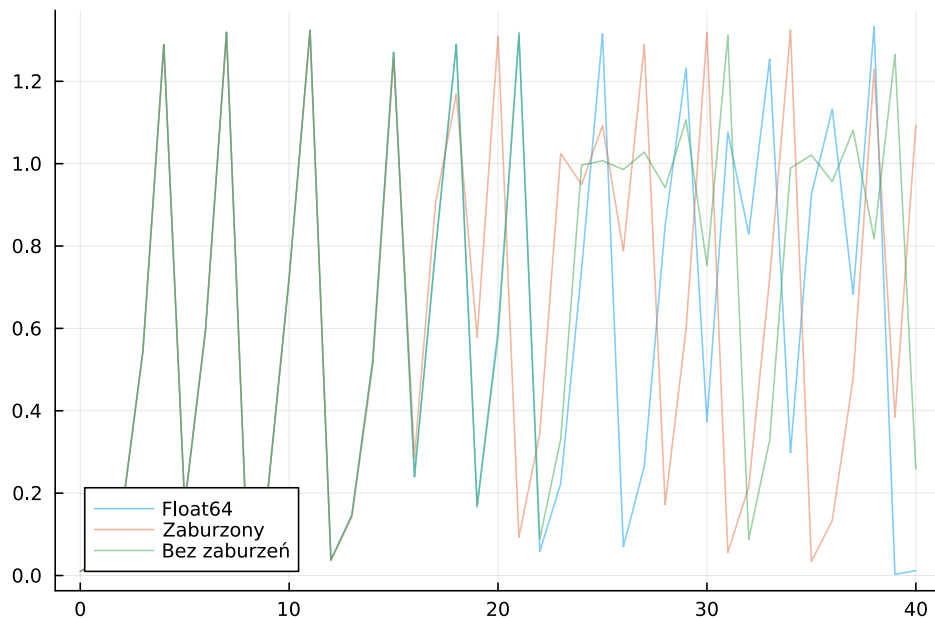
Rozważane jest równanie rekurencyjne modelu logistycznego.

$$p_{n+1} = p_n + rp_n(1 - p_n) \text{ z danym } p_0 \text{ i } r$$

Startując z początkowym stanem populacji $p_0 = 0.01$ i współczynnikiem wzrostu $r = 3$, wykonalem 40 iteracji tego procesu.

- W arytmetyce Float32.
- W arytmetyce Float32, ale z wprowadzonym zaburzeniem poprzez zaokrąglenie wyniku 10 iteracji do 3 miejsc po przecinku.
- W arytmetyce Float64.

5.2. Wyniki



Rysunek 5: Porównanie iteracji w formie wykresu

Iteracja	W Float64	W Float32 z zaburzeniem	W Float32	Komentarz
0	0.01	0.01	0.01	
1	0.0397	0.0397	0.0397	
2	0.15407173	0.15407173	0.15407173	
3	0.545072626044	0.5450726	0.5450726	Dla Float32 zostają ucięte ostatnie cyfry
4	1.28897800119	1.2889781	1.2889781	
...	
10	0.72291430118	0.722	0.7229306	Zostaje wprowadzone za- burzenie
11	1.32384194417	1.3241479	1.3238364	
12	0.0376952972547	0.036488414	0.037716985	
...	
17	0.788101190235	0.9010855	0.7860428	Zaburzony i niezaburzo- ny zaczynają się widocznie różnić
18	1.28909430279	1.1684768	1.2905813	
19	0.171084846702	0.577893	0.16552472	
20	0.596529312495	1.3096911	0.5799036	Zaburzony znajduje się nad 1, a reszta poniżej
21	1.31857558798	0.09289217	1.3107498	
22	0.0583776082594	0.34568182	0.088804245	Float64 i Float32 zaczynają się widocznie różnić
23	0.223286597599	1.0242395	0.33155844	
...	
30	0.374146489639	1.3191822	0.752921	Nie widać już żadnego po- wiązania między iteracja- mi
31	1.07662917143	0.05600393	1.3110139	
32	0.8291255674	0.21460637	0.0877831	
...	
38	1.33260564696	1.2292118	0.81736827	
39	0.00290915690285	0.3839622	1.2652004	
40	0.0116112380297	1.093568	0.25860548	

Tabela 9: Wyniki iteracji

5.3. Wnioski

Już po paru iteracjach od wprowadzenia zaburzenia można zauważyć znaczne różnice pomiędzy ciągiem zaburzonym i niezaburzonym. Te różnice powiększały wraz z kolejnymi iteracjami. Jeżeli p_n potrzebuje k cyfr do dokładnego zapisania, to wyrażenie $p_n(1 - p_n)$ potrzebuje już $2k$ cyfr. Ucinanie następuje bardzo szybko w precyzji Float32, a trochę później dla Float64 i wprowadza błąd obliczania. Wylizanie funkcji dla argumentu obciążonego błędem powoduje nałożenie jeszcze większego błędu, co spowodowało że wyniki eksperymentów rozbiegły się od siebie. Skoro zjawisko utraty dokładności występuje też w Float64, to w tej sytuacji nie stanowi ono lepszego przybliżenia prawdziwej wartości niż Float32. Proces ten można nazwać numerycznie niestabilnym.

6. Zadanie 6

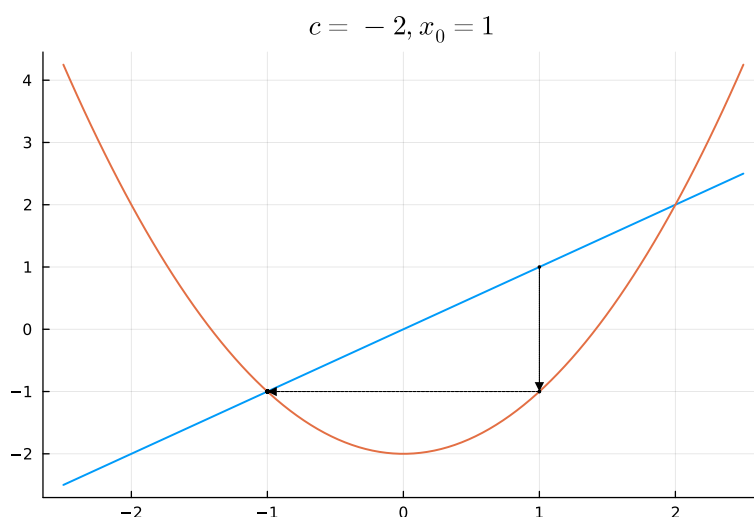
6.1. Opis problemu

Rozważane jest równanie rekurencyjne.

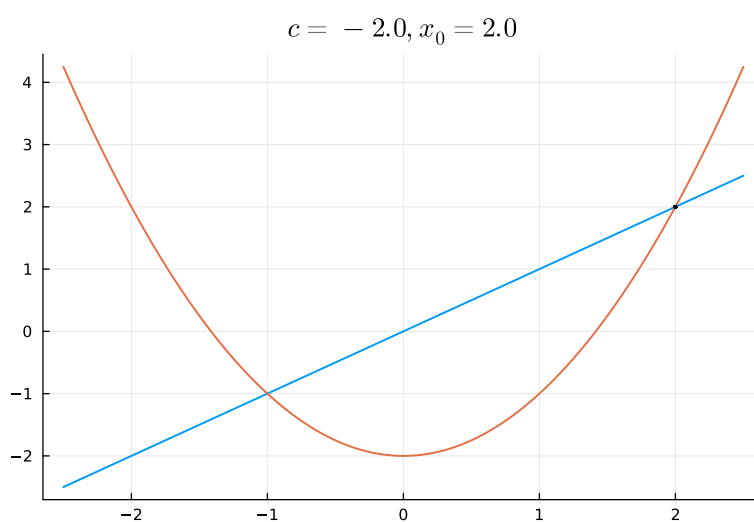
$$x_{n+1} = x_n^2 + c \text{ z danym } x_0 \text{ i } c$$

Zbadałem zachowanie tego równania dla danych wartości x_0 i c dla 40 pierwszych wyrazów, a następnie zwizualizowałem dla nich iterację graficzną.

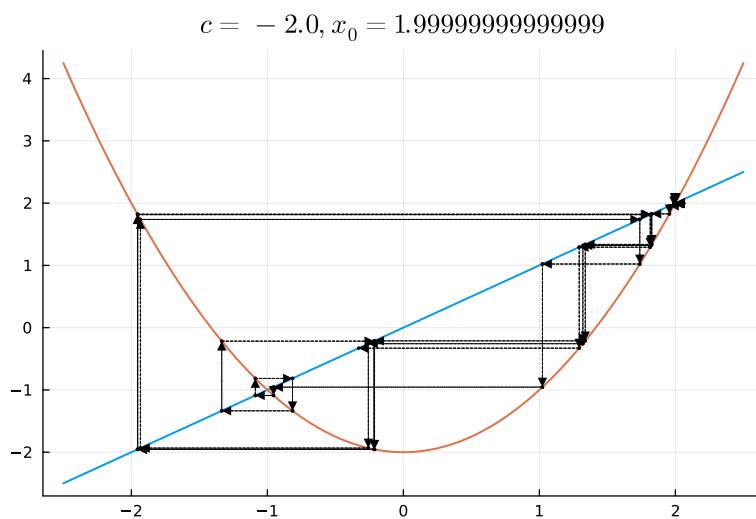
6.2. Wyniki



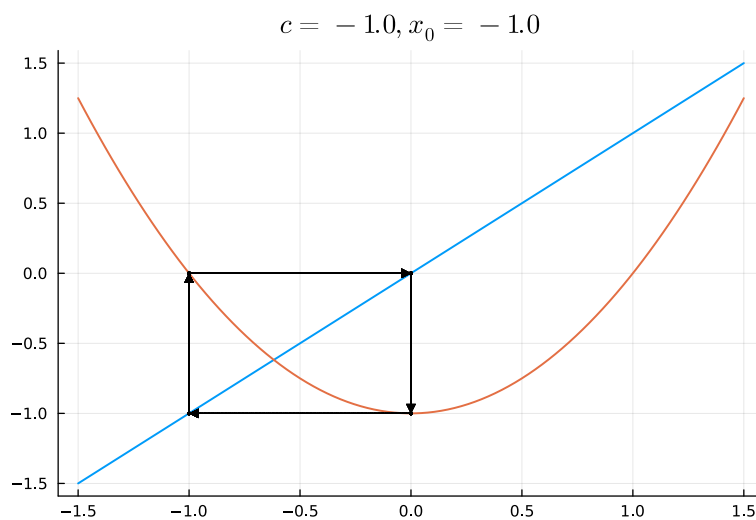
Rysunek 6: Iteracja graficzna $c = -2, x_0 = 1$



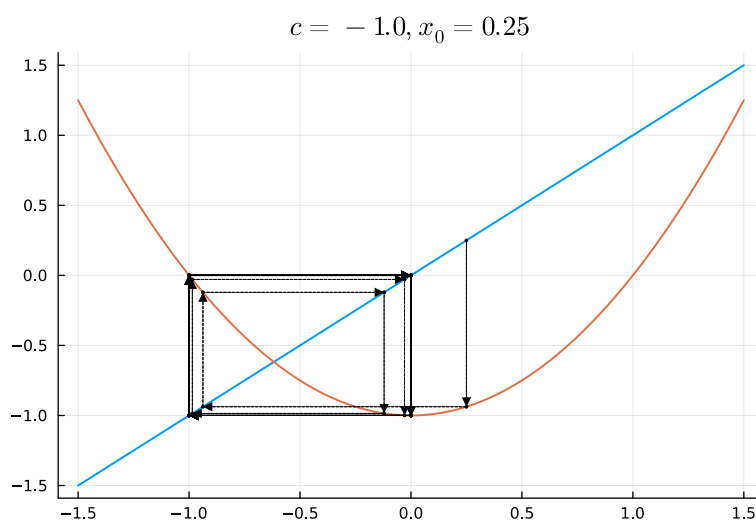
Rysunek 7: Iteracja graficzna $c = -2, x_0 = 2$



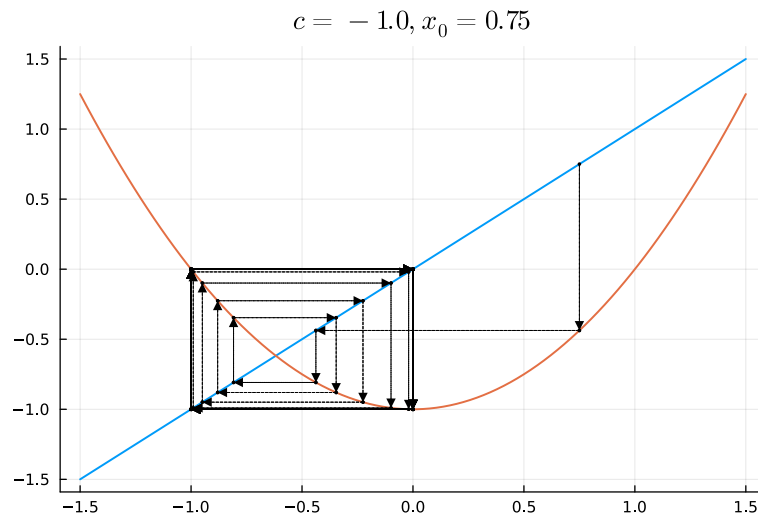
Rysunek 8: Iteracja graficzna $c = -2$ $x_0 = 1.9999999999999999$



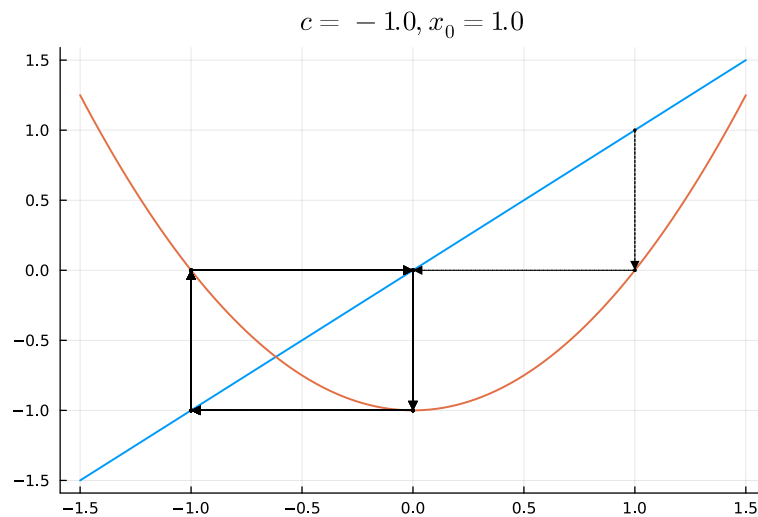
Rysunek 9: Iteracja graficzna $c = -1$ $x_0 = -1$



Rysunek 10: Iteracja graficzna $c = -1$ $x_0 = 0.25$



Rysunek 11: Iteracja graficzna $c = -1$ $x_0 = 0.75$



Rysunek 12: Iteracja graficzna $c = -1$ $x_0 = 1$

6.3. Wnioski

Z wykresu iteracji graficznej można odczytać punkty stałe danego równania rekurencyjnego. Dla $c = -2$ są to dokładnie -1 i 2 , w których kończymy na Rysunku 6 i 7 i nie możemy już ich opuścić. Natomiast bardzo niewielkie odchylenie się od 2 wywołało całkowity chaos, w którym nie udało się do niczego zbiec. Proces z $x_0 = 1.999999999999999$ stał się niestabilny. Nastąpiła duża czułość na warunki początkowe. Dla $c = -1$ punkty stałe znajdują się w $\frac{1 \pm \sqrt{5}}{2}$, jednak dla danych x_0 ciąg nie zbiegał do nich, ale do cyklu $0, -1, 0, -1, \dots$. Nastąpiło to dla każdego testowanego x_0 , co jest przejawem zachowania stabilnego.