

POLITECHNIKA WROCŁAWSKA

ANNA MODRZEJEWSKA

236642

Obliczenia naukowe

Lista nr 2

10 listopada 2018

Zadanie 1

1) Opis problemu

Należy powtórzyć zadanie z poprzedniej listy z nieznacznie zmienionymi danymi (usunięcie cyfry rzędu 10^{-10} z dwóch wektorów).

2) Rozwiązanie

Zaimplementowano 4 algorytmy obliczania iloczynu skalarnego dwóch wektorów. Algorytm A - w przód, algorytm B - w tył, algorytm C - od największego do najmniejszego, algorytm D - od najmniejszego do największego.

3) Otrzymane wyniki

Porównanie wyników wszystkich algorytmów (wykonanych w arytmetykach Float32 i Float64) wraz z wynikami otrzymanymi na poprzedniej liście:

Float64:

	bez modyfikacji danych	z modyfikacją danych
A	$1.02518813683 * 10^{-10}$	-0.004296342739891585
B	$-1.5643308870494 * 10^{-10}$	-0.004296342998713953
C	0.0	-0.004296342842280865
D	0.0	-0.004296342842280865

Float32:

	bez modyfikacji danych	z modyfikacją danych
A	-0.4999443	-0.4999443
B	-0.4543457	-0.4543457
C	-0.5	-0.5
D	-0.5	-0.5

4) Analiza wyników

Można zauważyć, że niewielkie zaburzenia danych wejściowych w przypadku arytmetyki Float32 nie spowodowały zmiany wyników. Wynika to z bardzo małego błędu względem poprzednich danych (rzędu 10^{-10}) w stosunku do precyzji obliczeń, przez co liczby $x_4 = 0.5772156649$ i $x'_4 = 0.577215664$ mają tę samą reprezentację bitową. Z kolei w przypadku arytmetyki Float64 widać wyraźne różnice w otrzymanych wynikach. Algorytmy C i D nie zwróciły wartości zerowych. Można stwierdzić, że zadanie obliczenia iloczynu skalarnego wektorów prawie prostopadłych (iloczyn skalarny bliski 0) jest źle uwarunkowane.

Zadanie 2

1) Opis problemu

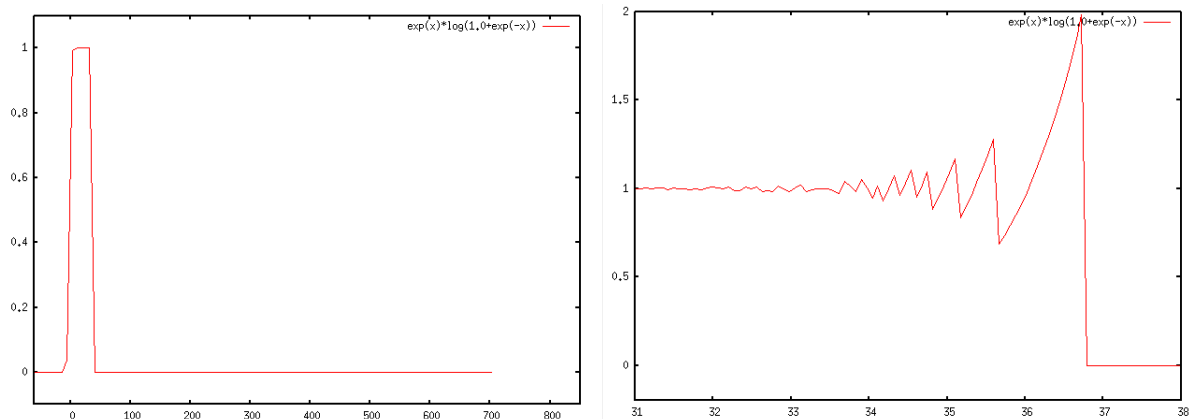
Należy narysować wykres funkcji $f(x) = e^x + \ln(1 + e^{-x})$ przy pomocy dwóch programów, a następnie policzyć granicę funkcji $f(x)$ oraz porównać wynik z wykresem.

2) Rozwiązanie

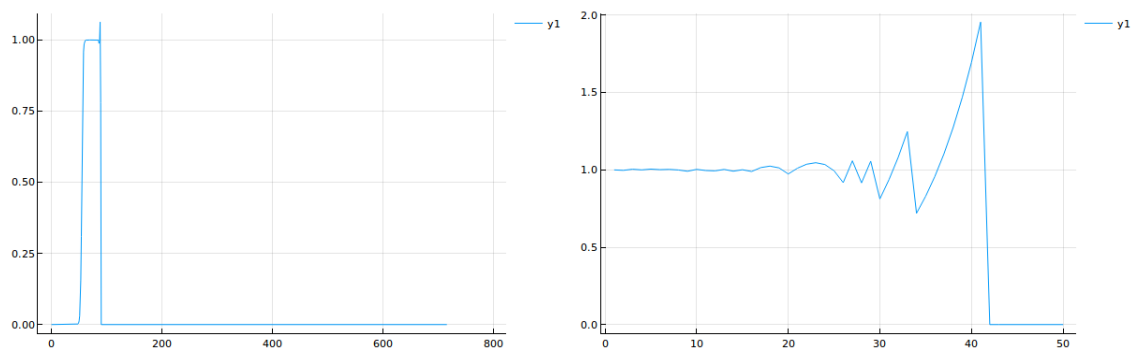
Obliczenie granicy funkcji $f(x)$:

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

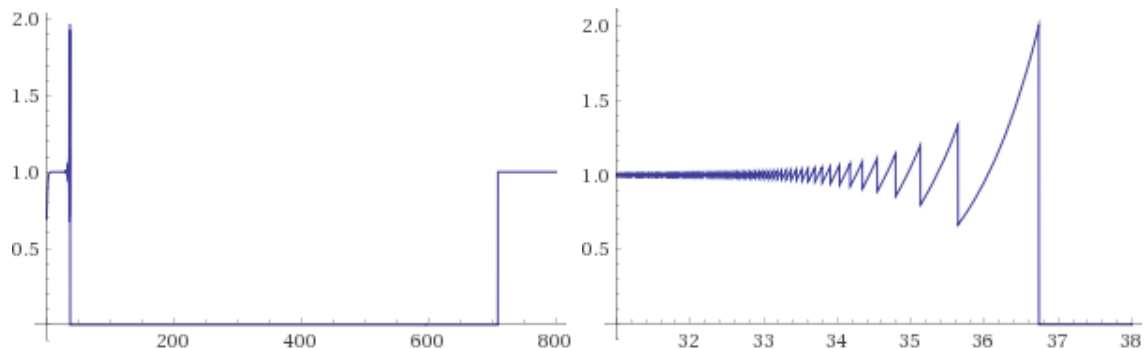
Wykresy wygenerowane za pomocą programu **gnuplot** ($x \in \langle -50, 800 \rangle$ oraz $x \in \langle 31, 38 \rangle$):



Za pomocą Plotly w języku Julia ($x \in \langle 0, 800 \rangle$ oraz $x \in \langle 0, 80 \rangle$):



Za pomocą Wolfram Alpha:



3) Analiza wyników

Wykresy wykazują nietypowe zachowanie w okolicy $31 \leq x < 37$, gdzie bardzo mała zmiana wartości x powoduje wysokie oscylacje. Może to świadczyć o złym uwarunkowaniu zadania.

W okolicy $x = 37$ wykresy osiągają wartość 0. Jest to związane z bardzo małą wartością $e^{-37} \approx$

$8.5330476 \cdot 10^{17}$ co jest mniejsze od wartości epsilon maszynowego, zatem $\ln(1 \oplus e^{-37}) = \ln(1) = 0$. Na wykresie wygenerowanym przez matematyczny program **Wolfram Alpha** funkcja przyjmuje stałą wartość 1 dla $x \leq 710$. Wynika to z tego, że w arytmetyce Float64 $e^{710} = \infty$, a $e^{-750} = 0$, przez co wyrażenie $e^x + \ln(1 + e^{-x})$ staje się nieoznaczone ($0 \cdot \infty$), dlatego wykres przyjmuje dla tego zakresu x wartość $f(x)$ równą wyliczonej granicy $f(x)$.

Zadanie 3

1) Opis problemu

Należy wygenerować macierz na dwa sposoby - jako macierz Hilberta oraz jako losowa macierz z zadanyim wskaźnikiem uwarunkowania c , a następnie rozwiązać układ równań liniowych $Ax = b$ dla danej macierzy A i wektora b za pomocą eliminacji Gaussa ($x = A \backslash b$) oraz $x = A^{-1}b$. Policzyc błędy względne otrzymanych rozwiązań.

2) Rozwiązanie

Wygenerowano macierze Hilberta H_n dla $n \in \{1, 2, \dots, 20\}$ za pomocą funkcji `hilb(n)` oraz losową R_n^c dla $n \in \{5, 10, 20\}$ z rosnącym wskaźnikiem uwarunkowania $c \in \{1, 10, 10^3, 10^7, 10^{12}, 10^{16}\}$ za pomocą funkcji `matcond(n)`.

Wskaźnik uwarunkowania dla wygenerowanej macierzy można sprawdzić za pomocą funkcji `cond(A)`. Błąd względny jest liczony ze wzoru:

$$\delta = \frac{\|x - \hat{x}\|}{\|x\|}$$

3) Otrzymane wyniki

Błędy względne dla macierzy Hilberta:

n	c	δ (Gauss)	δ (Macierz odwrotna)
1	1.0e+00	0.000000e+00	0.000000e+00
2	1.9e+01	5.661049e-16	1.404333e-15
3	5.2e+02	8.022594e-15	0.000000e+00
4	1.6e+04	4.137410e-14	0.000000e+00
5	4.8e+05	1.682843e-12	3.354436e-12
6	1.5e+07	2.618913e-10	2.016376e-10
7	4.8e+08	1.260687e-08	4.713280e-09
8	1.5e+10	6.124090e-08	3.077484e-07
9	4.9e+11	3.875163e-06	4.541268e-06
10	1.6e+13	8.670390e-05	2.501493e-04
11	5.2e+14	1.582781e-04	7.618304e-03
12	1.8e+16	1.339621e-01	2.589941e-01
13	3.3e+18	1.103970e-01	5.331276e+00
14	6.2e+17	1.455409e+00	8.714993e+00
15	3.7e+17	4.696668e+00	7.344641e+00
16	7.9e+17	5.415519e+01	2.984884e+01
17	1.3e+18	1.370724e+01	1.051694e+01
18	2.2e+18	9.134135e+00	7.575476e+00
19	6.5e+18	9.720590e+00	1.223376e+01
20	1.4e+18	7.549915e+00	2.206270e+01

Błędy względne dla macierzy losowej:

n	c	δ (Gauss)	δ (Macierz odwrotna)
5	1.0e+00	1.922963e-16	9.930137e-17
	1.0e+01	5.551115e-16	5.874748e-16
	1.0e+03	1.807435e-14	1.438739e-14
	1.0e+07	2.625613e-10	2.925623e-10
	1.0e+12	1.509270e-05	3.411969e-06
	1.0e+16	1.854288e-01	1.581139e-01
10	1.0e+00	2.328823e-16	2.248030e-16
	1.0e+01	4.033641e-16	2.937374e-16
	1.0e+03	2.332716e-14	2.698760e-14
	1.0e+07	7.419062e-11	1.014468e-10
	1.0e+12	3.318006e-06	3.647103e-06
	1.0e+16	2.614957e-01	3.052479e-01
20	1.0e+00	2.905732e-16	4.468561e-16
	1.0e+01	6.497414e-16	4.041273e-16
	1.0e+03	1.910468e-14	1.595171e-14
	1.0e+07	2.475013e-10	2.322159e-10
	1.0e+12	1.541623e-05	1.294256e-05
	1.0e+16	6.487102e-02	5.733012e-02

4) Analiza wyników

W przypadku macierzy Hilberta, zarówno liczenie metodą eliminacji Gaussa jak i za pomocą macierzy odwrotnej, generuje coraz większy błąd wraz ze wzrostem stopnia macierzy. Wraz ze wzrostem stopnia rośnie również współczynnik uwarunkowania macierzy. W przypadku macierzy losowej można zauważyć związek współczynnika uwarunkowania macierzy z błędem względnym. Te same współczynniki generują podobne błędy niezależnie od stopnia macierzy oraz sposobu liczenia.

Zadanie 4

1) Opis problemu

Należy policzyć 20 pierwiastków wielomianu Wilkinsona z postaci iloczynowej (p_x) i ogólnej (P_x), a także $|P(z_k)|$, $|p(z_k)|$, $|z_k - k|$ (błąd bezwzględny) dla $k \in \{1, 2, \dots, 20\}$. Następnie powtórzyć obliczenia dla zmienionego współczynnika przy x^{19} z -210 na $-210 - 2^{-23}$.

2) Rozwiązanie

Do rozwiązania zadania można wykorzystać pakiet `Polynomials`.

Funkcja `Poly` tworzy wielomian dla zadanych współczynników, funkcja `roots` zwraca pierwiastki zadanego wielomianu, funkcja `poly` tworzy wielomian dla zadanych pierwiastków, a funkcja `polyval` wylicza wartość wielomianu dla zadanego x .

3) Otrzymane wyniki

k	z_k	$ P(z_k) $	$ p(z_k) $	$ z_k - k $
1	0.999999999996989	36352.0	38400.0	3.0109248427834245e-13
2	2.0000000000283182	181760.0	198144.0	2.8318236644508943e-11
3	2.999999995920965	209408.0	301568.0	4.0790348876384996e-10
4	3.9999999837375317	3.106816e6	2.844672e6	1.626246826091915e-8
5	5.000000665769791	2.4114688e7	2.3346688e7	6.657697912970661e-7
6	5.999989245824773	1.20152064e8	1.1882496e8	1.0754175226779239e-5
7	7.000102002793008	4.80398336e8	4.78290944e8	0.00010200279300764947
8	7.999355829607762	1.682691072e9	1.67849728e9	0.0006441703922384079
9	9.002915294362053	4.465326592e9	4.457859584e9	0.002915294362052734
10	9.990413042481725	1.2707126784e10	1.2696907264e10	0.009586957518274986
11	11.025022932909318	3.5759895552e10	3.5743469056e10	0.025022932909317674
12	11.953283253846857	7.216771584e10	7.2146650624e10,	0.04671674615314281
13	13.07431403244734	2.15723629056e11	2.15696330752e11	0.07431403244734014
14	13.914755591802127	3.65383250944e11	3.653447936e11	0.08524440819787316
15	15.075493799699476	6.13987753472e11	6.13938415616e11	0.07549379969947623
16	15.946286716607972	1.555027751936e12	1.554961097216e12	0.05371328339202819
17	17.025427146237412	3.777623778304e12	3.777532946944e12	0.025427146237412046
18	17.99092135271648	7.199554861056e12	7.1994474752e12	0.009078647283519814
19	19.00190981829944	1.0278376162816e13	1.0278235656704e13	0.0019098182994383706
20	19.999809291236637	2.7462952745472e13	2.7462788907008e13	0.00019070876336257925

Mimo że błąd bezwzględny wyliczenia pierwiastka za pomocą funkcji `roots` wydaje się być mały, znacznie wpływa na końcowy wynik (różnice między $P(x)$ a $p(x)$ są zauważalnie duże, ponieważ niedokładnie wyliczony x jest podnoszony do potęgi ≤ 20).

Po zamianie współczynnika przy x^{19} na $-230 - 2^{23}$ wyniki prezentują się następująco:

k	z_k	$ P(z_k) $	$ p(z_k) $	$ z_k - k $
1	0.9999999999998357	20992.0	22016.0	1.643130076445e-13
2	2.0000000000550373	349184.0	365568.0	5.5037308e-11
3	2.99999999660342	2.221568e6	2.295296e6	3.396579906223e-9
4	4.000000089724362	1.046784e7	1.0729984e7	8.972436216226e-8
5	4.99999857388791	3.9463936e7	4.3303936e7	1.426112089753e-6
6	6.000020476673031	1.29148416e8	2.06120448e8	2.047667303096e-5
7	6.99960207042242	3.88123136e8	1.757670912e9	0.00039792957758
8	8.007772029099446im	1.072547328e9	1.8525486592e10	0.0077720290994
9	8.915816367932559	3.065575424e9	1.37174317056e11	0.084183632067
10	10.095455630535774 - 0.65im	7.143113638035824e9	1.4912633816754019e12	0.651958683038
11	10.095455630535774 + 0.65im	7.143113638035824e9	1.4912633816754019e12	1.110918027272
12	11.793890586174369 - 1.65im	3.357756113171857e10	3.2960214141301664e13	1.6652812906
13	11.793890586174369 + 1.65im	3.357756113171857e10	3.2960214141301664e13	2.04582027668
14	13.992406684487216 - 2.52im	1.0612064533081976e11	9.545941595183662e14	2.518835871191
15	13.992406684487216 + 2.52im	1.0612064533081976e11	9.545941595183662e14	2.712880531285
16	16.73074487979267 - 2.81im	3.315103475981763e11	2.7420894016764064e16	2.906001873538
17	16.73074487979267 + 2.81im	3.315103475981763e11	2.7420894016764064e16	2.82548352135
18	19.5024423688181 - 1.94im	9.539424609817828e12	4.2525024879934694e17	2.45402144631
19	19.5024423688181 + 1.94im	9.539424609817828e12	4.2525024879934694e17	2.00432944431
20	20.84691021519479	1.114453504512e13	1.3743733197249713e18	0.846910215195

Zmiana współczynnika była bardzo niewielka, lecz spowodowała zmniejszenie wyników funkcji wielomianowej niekiedy dwukrotnie. Można wywnioskować, że to zadanie jest źle uwarunkowane.

Zadanie 5

1) Opis problemu

Należy zbadać *model wzrostu populacji* dany równaniem rekurencyjnym:

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

gdzie r jest pewną daną stałą, $r(1-p_n)$ jest czynnikiem wzrostu populacji, a p_0 jest wielkością populacji stanowiącą procent maksymalnej populacji dla danego stanu środowiska.

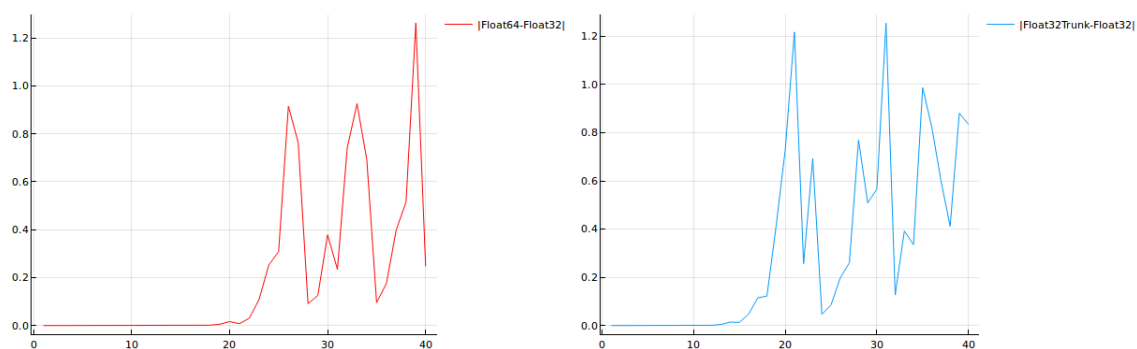
Dla danych $p_0 = 0.01$, $r = 3$ zostanie przeprowadzonych 40 iteracji równania rekurencyjnego w arytmetykach Float64, Float32 oraz z obcięciem wyniku po 10. iteracji do 3 cyfr po przecinku.

2) Otrzymane wyniki

p_n	Float64	Float32	Float32 z obcięciem
p_0	0.01	0.01	0.01
p_1	0.0397	0.0397	0.0397
p_2	0.15407173000000002	0.15407173	0.15407173
p_3	0.5450726260444213	0.5450726	0.5450726
p_4	1.2889780011888006	1.2889781	1.2889781
p_5	0.17151914210917552	0.1715188	0.1715188
p_9	0.21558683923263022	0.21559286	0.21559286
p_{10}	0.722914301179573	0.7229306	0.722
p_{11}	1.3238419441684408	1.3238364	1.3241479
p_{12}	0.03769529725473175	0.037716985	0.036488414
...
p_{15}	1.2702617739350768	1.2704837	1.2572169
...
p_{20}	0.5965293124946907	0.5799036	1.3096911
...
p_{25}	1.315588346001072	1.0070806	1.0929108
...
p_{30}	0.37414648963928676	0.7529209	1.3191822
...
p_{35}	0.9253821285571046	1.021099	0.034241438
...
p_{38}	1.3326056469620293	0.81736827	1.2292118
p_{39}	0.0029091569028512065	1.2652004	0.3839622
p_{40}	0.011611238029748606	0.25860548	1.093568

3) Porównanie wyników i wnioski

Różnice między odpowiadającymi wartościami p_n otrzymanymi w arytmetyce Float64 a Float32 oraz między Float32 a Float32 z modyfikacją danych po 10. iteracji (wykresy przy użyciu biblioteki `Plots` w języku `Julia`):



Analizując czerwony wykres, można zauważyć, że do $x \leq 20$ różnice precyzji spowodowały niezauważalne różnice wyników. Dla $x > 20$ różnice zaczęły znacznie narastać, czego przyczyną jest obliczanie wyniku poprzez sprzężenie zwrotne - wynik poprzedniej operacji jest daną wejściową dla kolejnej iteracji. Błąd generowany przez niewystarczającą precyzję arytmetyki Float32 jest przenoszony na kolejne wyniki i powiększany po każdej iteracji.

Na niebieskim wykresie widać, że drobna modyfikacja danych (błąd względny rzędu 0.13%) spowodowała duże odchylenia między odpowiadającymi wynikami. Po 40. iteracji błąd względny wyniósł około 76%.

Zadanie 6

1) Opis problemu

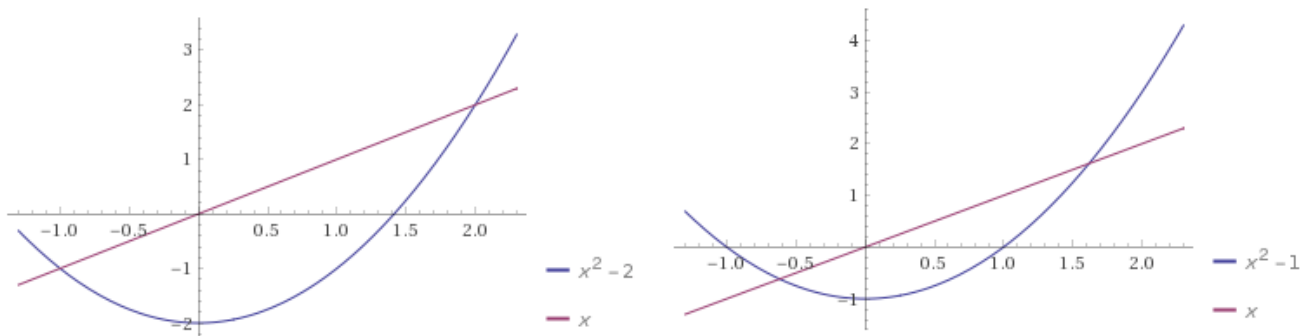
Należy policzyć wartość równania rekurencyjnego $x_{n+1} = x_n^2 + c$ dla danej stałej c , $n \in \{1, 2, \dots, 40\}$ oraz siedmiu różnych zestawów danych c , x_0 :

1. $x_0 = 1, c = -2$
2. $x_0 = 2, c = -2$
3. $x_0 = 1.99999999, c = -2$
4. $x_0 = 1, c = -1$
5. $x_0 = -1, c = -1$
6. $x_0 = 0.75, c = -1$
7. $x_0 = 0.25, c = -1$

2) Otrzymane wyniki

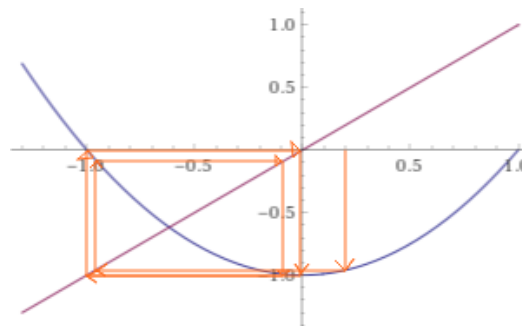
zestaw \ iteracja	1	2	3	4	5	6	7
1	-1.0	2.0	2.0	0.0	0.0	-0.4375	-0.9375
2	-1.0	2.0	2.0	-1.0	-1.	-0.808594	-0.121094
3	-1.0	2.0	2.0	0.0	0.0	-0.346176	-0.985336
4	-1.0	2.0	2.0	-1.0	-1.0	-0.880162	-0.0291124
5	-1.0	2.0	2.0	0.0	0.0	-0.225315	-0.999152
10	-1.0	2.0	2.0	-1.0	-1.0	-0.99962	-6.59315e-11
15	-1.0	2.0	1.99999	0.0	0.0	-2.66165e-12	-1.0
20	-1.0	2.0	1.98902	-1.0	-1.0	-1.0	0.0
25	-1.0	2.0	-1.95501	0.0	0.0	0.0	-1.0
30	-1.0	2.0	1.7385	-1.0	-1.0	-1.0	0.0
35	-1.0	2.0	-1.33545	0.0	0.0	0.0	-1.0
40	-1.0	2.0	-0.328979	-1.0	-1.0	-1.0	0.0

Wykresy funkcji $x^2 - 2$ (zestawy 1, 2, 3) oraz $x^2 - 1$ (zestawy 4, 5, 6, 7) wraz z prostą x :



Równanie przyjmuje stałe wartości dla zestawu 1. i 2. Natomiast w 3. przypadku minimalna zmiana x_0 w stosunku do 2. przypadku generuje rozbieżność ciągu. Świadczy to o złym uwarunkowaniu zadania. Dla tych przypadków funkcja przyjmuje stałe wartości -1 i 2 . Do stabilnych rozwiązań będą prowadzić tylko te wartości początkowe x_0 , dla których funkcja jest zbieżna do tych punktów.

Iteracja graficzna dla 7. zestawu danych, $x_0 = 0.25$:



Widać, jak po kilku iteracjach równanie dla kolejnych x przyjmuje wartości na przemian 0.0 i -1.0 . Jest to stabilny zestaw danych. Trzeci zestaw, dla którego kolejne iteracje będą rozbiegały, charakteryzuje się niestabilnością.