
Obliczenia Naukowe

Sprawozdanie z Laboratorium

LISTA 2

AUTOR:
KRZYSZTOF NOWAK
III ROK INF.
NR INDEKSU: 229807

12 listopada 2017

1 Zadanie 1

1.1 Opis problemu

Naszym zadaniem było powtórzenie zadania 5 z listy 1 z drobną modyfikacją danych: z x_4 mieliśmy usunąć ostatnią 9, a z x_5 ostatnią 7.

1.2 Rozwiązanie oraz wynik

Algorytm	Float32	Float64	Float32 błąd wzg.	Float64 błąd wzg.
a.	-0.4999443	-0.00429634273989	4.9668057661282e10	4.2682954615672e8
b.	-0.4543457	-0.004296342998713	4.51379655800e10	4.26829571869997e8
c.	-0.5	-0.00429634284228	4.967359135306e10	4.268295563288e8
d.	-0.5	-0.00429634284228	4.967359135306e10	4.268295563288e8

Dla porównania wyniki dla zadania 5 z listy 1

Algorytm	Float32	Float64	Float32 błąd wzg.	Float64 błąd wzg.
a.	-0.4999443	1.0251881368296672e-10	4.96680577e10	11.1849553
b.	-0.4543457	-1.5643308870494366e-10	4.5137965e10	14.541186645
c.	-0.5	0.0	4.9673591e10	-
d.	-0.5	0.0	4.9673591e10	-

1.3 Wnioski

W przypadku arytmetyki Float32 wprowadzone zmiany nie miały wpływu na wynik, ponieważ arytmetyka ta ma 7-8 cyfr znaczących, więc modyfikacja danych poprzez usunięcie cyfr z dalszych pozycji nie zmienia reprezentacji liczby. We Float64 natomiast możemy zauważyć zwiększenie się błędu w przypadku dwóch pierwszych wyników oraz otrzymanie wyników w kolejnych dwóch algorytmach, które na poprzedniej liście dały wyniki zupełnie odbiegające od innych, bardziej bliżonych do reszty algorytmów, aczkolwiek dalej dalekich od poprawnego rozwiązania.

2 Zadanie 2

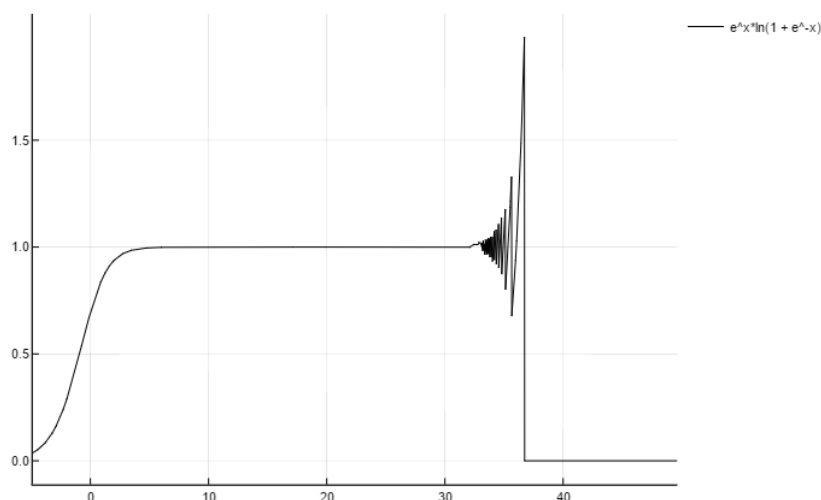
2.1 Opis problemu

Naszym zadaniem było narysowanie funkcji $f(x) = e^x \ln(1 + e^{-x})$ w co najmniej w dwóch programach do wizualizacji.

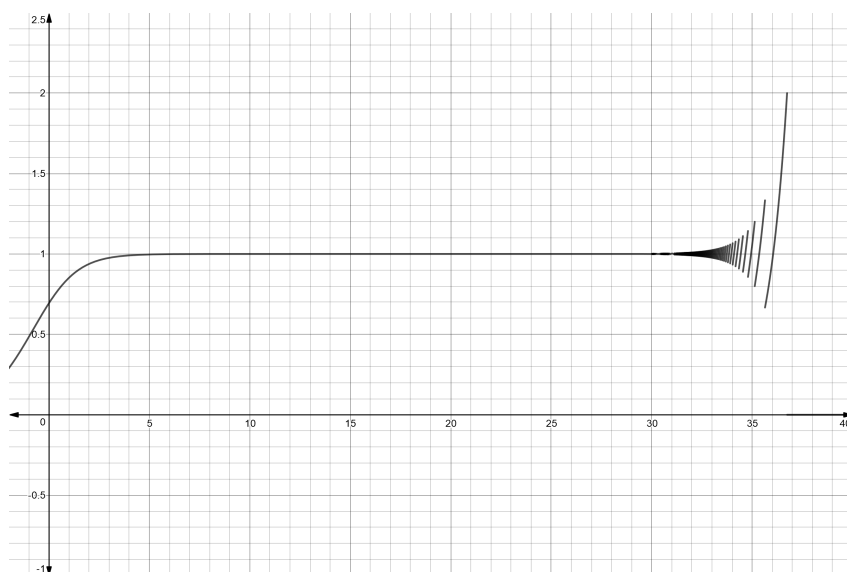
2.2 Rozwiązanie

Do wizualizacji tej funkcji użyłem możliwości języka Julia do generowania wykresów, oraz Desmos - strony internetowej oferującej kalkulator graficzny.

2.3 Wynik



Rysunek 1: Otrzymany wykres w języku Julia



Rysunek 2: Otrzymany wykres w Desmos

2.4 Wnioski

Pomimo tego, że $\lim_{x \rightarrow \infty} e^x \ln(1 + e^{-x}) = \lim_{x \rightarrow \infty} \frac{\ln(1 + e^{-x})}{e^{-x}} \stackrel{de\ l'H}{=} \lim_{x \rightarrow \infty} \frac{\frac{1}{1+e^{-x}} e^{-x}}{e^{-x}} = \lim_{x \rightarrow \infty} \frac{1}{1 + e^{-x}} = 1$ i rzeczywiście na początku wygląda jakby wykres miał dążyć do tej wartości, to jednak z powodu uwarunkowania funkcji, zaczyna dochodzić do dziwnych anomalii. Jest spowodowane to tym, że zaczynamy mnożyć bardzo dużą liczbę jaką jest e^x z bardzo małą: $\ln(1 + e^{-x})$ co powoduje błąd, dodatkowo po pewnym czasie logarytm zaczyna być mniejszy od zera maszynowego, więc wartość całej funkcji w arytmetyce Float64 wynosi 0.

3 Zadanie 3

3.1 Opis problemu

Naszym zadaniem było rozwiązanie układu równań liniowych $Ax = b$ przy pomocy eliminacji Gaussa, oraz za wykorzystaniem inwersji, a następnie porównać z dokładnym rozwiązaniem dla dwóch sposobów generowania macierzy: tworzenie macierzy Hilberta o danym stopniu, oraz tworzenie losowej macierzy o danym stopniu oraz uwarunkowaniu.

3.2 Rozwiązanie i wyniki

Dla macierzy Hilberta:

n	Cond	Rząd	Gauss błąd wzg.	Inwersja błąd wzg.
2	19.28147006790397	2	$5.661048867003676e - 16$	$1.1240151438116956e - 15$
3	524.0567775860644	3	$8.022593772267726e - 15$	$9.825526038180824e - 15$
4	15513.73873892924	4	$4.4515459601812086e - 13$	$2.950477637286781e - 13$
5	476607.25024259434	5	$1.6828426299227195e - 12$	$8.500055777753297e - 12$
6	$1.4951058642254665e7$	6	$2.618913302311624e - 10$	$3.3474135070361745e - 10$
7	$4.75367356583129e8$	7	$1.2606867224171548e - 8$	$5.163959183577243e - 9$
8	$1.5257575538060041e10$	8	$1.026543065687064e - 7$	$2.698715074276819e - 7$
9	$4.931537564468762e11$	9	$4.83235712050215e - 6$	$9.175846868614517e - 6$
10	$1.6024416992541715e13$	10	0.0006329153722983848	0.00045521422517408853
11	$5.222677939280335e14$	11	0.011543958596122112	0.00804446677343116
12	$1.7514731907091464e16$	11	0.2975640310734787	0.34392937091205217
13	$3.344143497338461e18$	11	2.375017867706776	5.585796893150773
14	$6.200786263161444e17$	12	5.281004646755168	4.800641929017436
15	$3.674392953467974e17$	12	1.177294734836712	4.8273577212576475
16	$7.865467778431645e17$	12	20.564655823804095	31.736467496266126
17	$1.263684342666052e18$	12	17.742214635179074	15.910335962604142
18	$2.2446309929189128e18$	12	4.2764564411159425	6.281223433472033
19	$6.471953976541591e18$	13	22.119937292648906	22.92561401563632
20	$1.3553657908688225e18$	13	14.930069669294001	21.53949860251383

Dla losowej macierzy o danym uwarunkowaniu:

n	Cond	Rząd	Gauss błąd wzg.	Inwersja błąd wzg.
5	1.0	5	0.0	$1.5700924586837752e - 16$
5	1000.0	5	$1.7872590803983013e - 14$	$2.608962690553374e - 14$
5	$1.0e7$	5	$1.8698717615339606e - 10$	$1.1745759774299514e - 10$
5	$1.0e12$	5	$1.6062935334094203e - 5$	$1.5337456100956596e - 5$
5	$1.0e16$	4	0.15489274332998698	0.15703566122376023
10	1.0	10	$3.020133145511626e - 16$	$2.1642230995786354e - 16$
10	1000.0	10	$4.4313060788193306e - 15$	$3.0323522101633274e - 15$
10	$1.0e7$	10	$1.8545138145554434e - 10$	$1.4123838415462525e - 10$
10	$1.0e12$	10	$1.7998453164569674e - 5$	$2.3966104839261076e - 5$
10	$1.0e16$	9	0.05041672106322188	0.07441383556954216
20	1.0	20	$5.545561229666063e - 16$	$4.983652532311798e - 16$
20	1000.0	20	$8.192362705716001e - 16$	$4.245641492672234e - 15$
20	$1.0e7$	20	$4.2639013046040924e - 11$	$2.5322527734235123e - 11$
20	$1.0e12$	20	$1.2770052605025184e - 5$	$1.2130797284055973e - 5$
20	$1.0e16$	19	0.21212436679371685	0.21354052535581688

3.3 Wnioski

Jak można zauważyć, im większy jest wskaźnik uwarunkowania, tym generowany błąd jest większy. Złośliwość macierzy Hilberta polega na tym, że zawiera ona wiele liczb, które nie są skończone w arytmetyce zmiennopozycyjnej, więc kiedy są wykorzystywane podczas obliczeń wielokrotnie, błąd ten jest powielany. Dlatego też im większy stopień tej macierzy, tym też większy jego wskaźnik uwarunkowania, oraz związany z tym błąd rośnie. Podobna sytuacja dzieje się dla macierzy wygenerowanych losowo, dla podanych wskaźników uwarunkowania. Przez ten chaos w wynikach, nie jesteśmy w stanie określić, która z metod obliczania niewiadomej jest lepsza.

4 Zadanie 4

4.1 Opis problemu

Naszym zadaniem było obliczenie pierwiastków wielomianu Wilkinsona, porównania z rzeczywistymi rozwiązaniami, oraz powtórzenie eksperymentu Wilkinsona, polegającego na zaburzeniu jednego ze współczynników o 2^{-23} .

4.2 Rozwiązanie

Dla podanych współczynników za pomocą funkcji Poly z Polynomials stworzono wielomian P, następnie obliczono pierwiastki tego wielomianu za pomocą funkcji roots zapisane w z, potem stworzono wielomian w postaci iloczynowej używając funkcji poly. Obliczono wartości funkcji dla obliczonych pierwiastków, oraz policzono różnicę między rzeczywistymi pierwiastkami, a obliczonymi. Kroki zostały powtórzone dla zaburzonego wielomianu.

4.3 Wyniki

Dla wielomianu Wilkinsona:

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

Dla zaburzonego wielomianu Wilkinsona:

k	$ P(z_k) $
1	20992.0
2	349184.0
3	2.221568e6
4	1.046784e7
5	3.9463936e7
6	1.29148416e8
7	3.88123136e8
8	1.072547328e9
9	3.065575424e9
10	7.143113638035824e9
11	7.143113638035824e9
12	3.357756113171857e10
13	3.357756113171857e10
14	1.0612064533081976e11
15	1.0612064533081976e11
16	3.315103475981763e11
17	3.315103475981763e11
18	9.539424609817828e12
19	9.539424609817828e12
20	1.114453504512e13

4.4 Wnioski

Z powodu wysokiego współczynnika uwarunkowania wielomianu Wilkinsona, oraz niedokładności arytmetyki zmiennopozycyjnej, nawet wyliczone przez program “pierwiastki” dają błędne wyniki, natomiast niewielka zmiana współczynnika powoduje, że niektóre pierwiastki są zespolone i dają równie błędne wyniki.

5 Zadanie 5

5.1 Opis problemu

Naszym zadaniem było rozważenie rekurencji: $p_{n+1} := p_n + rp_n(1 - P_n)$, która to jest modelem logistycznym, modelem wzrostu populacji, oraz wykonanie eksperymentów

5.2 Rozwiązanie i wyniki

n	Float32	Float32 obcięte	Float64
1	0.0397	0.0397	0.0397
2	0.15407172	0.15407172	0.154071730000000002
3	0.5450726	0.5450726	0.5450726260444213
4	1.288978	1.288978	1.2889780011888006
5	0.17151925	0.17151925	0.17151914210917552
6	0.5978204	0.5978204	0.5978201201070994
7	1.3191139	1.3191139	1.3191137924137974
8	0.05627131	0.05627131	0.056271577646256565
9	0.21558586	0.21558586	0.21558683923263022
10	0.72291166	0.722	0.722914301179573
11	1.3238429	1.3241479	1.3238419441684408
12	0.037691634	0.03648846	0.03769529725473175
13	0.14650455	0.14195961	0.14651838271355924
14	0.5216274	0.50738084	0.521670621435246
15	1.2702242	1.2572174	1.2702617739350768
16	0.24048816	0.2870828	0.24035217277824272
17	0.7884489	0.90108156	0.7881011902353041
18	1.2888407	1.1684823	1.2890943027903075
19	0.17203197	0.5778765	0.17108484670194324
20	0.5993429	1.3096823	0.5965293124946907
21	1.3197359	0.09292625	1.3185755879825978
22	0.05383516	0.34579912	0.058377608259430724
23	0.20664598	1.0244654	0.22328659759944824
24	0.69847625	0.94927347	0.7435756763951792
25	1.3302977	1.0937335	1.315588346001072
26	0.0121148545	0.78617495	0.07003529560277899
27	0.04801911	1.2904866	0.26542635452061003
28	0.18515894	0.1658795	0.8503519690601384
29	0.63778424	0.58097	1.2321124623871897
30	1.3308307	1.3113016	0.37414648963928676
31	0.0099917315	0.08667082	1.0766291714289444
32	0.03966742	0.3241478	0.8291255674004515
33	0.15394917	0.98137575	1.2541546500504441
34	0.5446956	1.0362079	0.29790694147232066
35	1.2887025	0.92365116	0.9253821285571046
36	0.17254764	1.1352103	1.1325322626697856
37	0.6008725	0.674734	0.6822410727153098
38	1.3203467	1.3331381	1.3326056469620293
39	0.051440563	0.0007807463	0.0029091569028512065
40	0.19782385	0.0031211565	0.011611238029748606

5.3 Wnioski

W tym zadaniu powtarzaliśmy eksperyment Lorenza, tak jak autorzy książki “Fraktale. Granice chaosu”. Pomimo stworzenia solidnego modelu, który opisuje dane zjawisko, przy ograniczonych zasobach, nasze działania będą generować pewien błąd. Jeżeli zignorujemy ten błąd może dojść do tego, że chociaż wszystkie nasze obliczenia będą poprawne, to z powodu zatracenia dokładności, otrzymywane wyniki przestaną być skorelowane z rzeczywistym wynikiem. Zjawisko to

naukowcy nazywają deterministycznym chaosem.

6 Zadanie 6

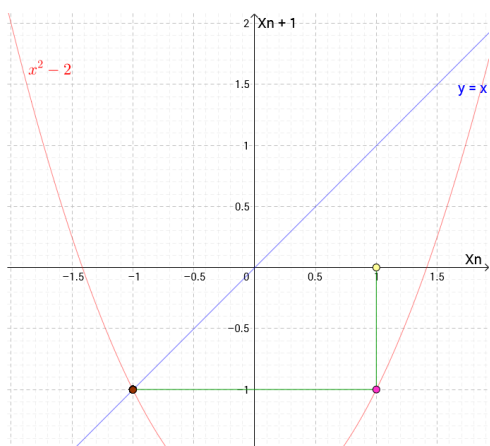
6.1 Opis problemu

Naszym zadaniem było rozwiązanie rekurencji: $x_{n+1} := x_n^2 + c$, dla podanych wartości x_0 oraz c .

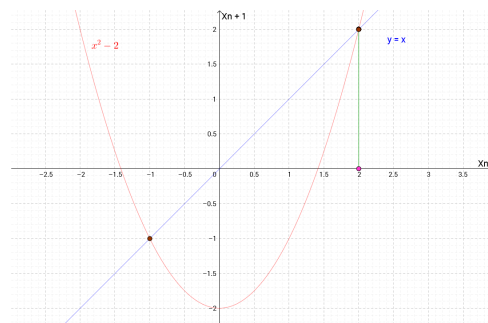
6.2 Rozwiązanie i wyniki

n	1)	2)	3)	4)	5)	6)	7)
1	-1.0	2.0	1.999999999999	0.0	0.0	-0.4375	-0.9375
2	-1.0	2.0	1.999999999998	-1.0	-1.0	-0.80859375	-0.12109375
3	-1.0	2.0	1.9999999999936	0.0	0.0	-0.3461761474609	-0.98533630371093
4	-1.0	2.0	1.9999999999974	-1.0	-1.0	-0.8801620749291	-0.02911236858926
5	-1.0	2.0	1.99999999998976	0.0	0.0	-0.2253147218565	-0.99915246999512
6	-1.0	2.0	1.99999999995907	-1.0	-1.0	-0.9492332761147	-0.00169434170264
7	-1.0	2.0	1.9999999998362	0.0	0.0	-0.0989561875165	-0.99999712920619
8	-1.0	2.0	1.99999999934516	-1.0	-1.0	-0.9902076729522	-5.74157936927e - 6
9	-1.0	2.0	1.99999999738065	0.0	0.0	-0.01948876442658	-0.99999999996703
10	-1.0	2.0	1.9999999895226	-1.0	-1.0	-0.999620188061	-6.59314824957e - 11
11	-1.0	2.0	1.99999995809041	0.0	0.0	-0.00075947962064	-1.0
12	-1.0	2.0	1.99999983236193	-1.0	-1.0	-0.9999994231901	0.0
13	-1.0	2.0	1.99999932944778	0.0	0.0	-1.1536182557e - 6	-1.0
14	-1.0	2.0	1.99999731779157	-1.0	-1.0	-0.999999999998	0.0
15	-1.0	2.0	1.99998927117349	0.0	0.0	-2.6616486792e - 12	-1.0
16	-1.0	2.0	1.99995708480908	-1.0	-1.0	-1.0	0.0
17	-1.0	2.0	1.9998283410780	0.0	0.0	0.0	-1.0
18	-1.0	2.0	1.99931339377896	-1.0	-1.0	-1.0	0.0
19	-1.0	2.0	1.99725404654394	0.0	0.0	0.0	-1.0
26	-1.0	2.0	1.8220620963151	-1.0	-1.0	-1.0	0.0
27	-1.0	2.0	1.3199102828284	0.0	0.0	0.0	-1.0
28	-1.0	2.0	-0.25783684528373	-1.0	-1.0	-1.0	0.0
29	-1.0	2.0	-1.93352016121412	0.0	0.0	0.0	-1.0
30	-1.0	2.0	1.73850021382151	-1.0	-1.0	-1.0	0.0
31	-1.0	2.0	1.02238299345743	0.0	0.0	0.0	-1.0
32	-1.0	2.0	-0.95473301468900	-1.0	-1.0	-1.0	0.0
33	-1.0	2.0	-1.08848487066284	0.0	0.0	0.0	-1.0
34	-1.0	2.0	-0.81520068633809	-1.0	-1.0	-1.0	0.0
35	-1.0	2.0	-1.33544784099389	0.0	0.0	0.0	-1.0
36	-1.0	2.0	-0.216579063984746	-1.0	-1.0	-1.0	0.0
37	-1.0	2.0	-1.9530935090434	0.0	0.0	0.0	-1.0
38	-1.0	2.0	1.81457425506781	-1.0	-1.0	-1.0	0.0
39	-1.0	2.0	1.29267972715492	0.0	0.0	0.0	-1.0
40	-1.0	2.0	-0.32897912300267	-1.0	-1.0	-1.0	0.0

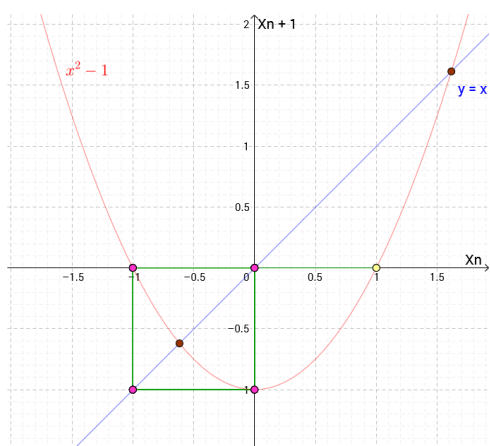
Dodatkowo przeprowadzona została iteracja graficzna:



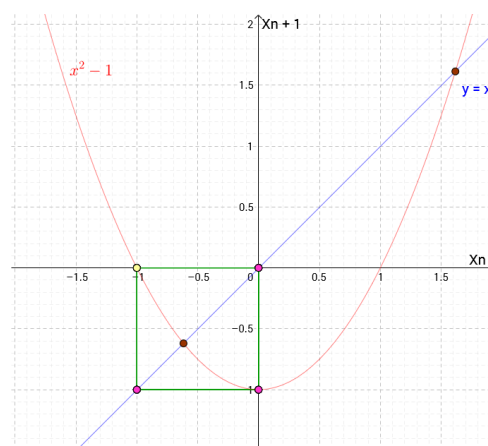
Rysunek 3: Iteracja dla 1)



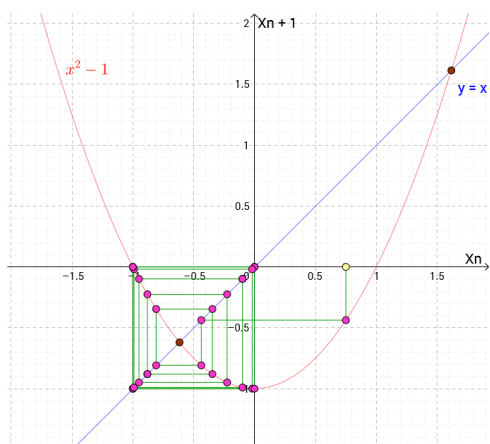
Rysunek 4: Iteracja dla 2)



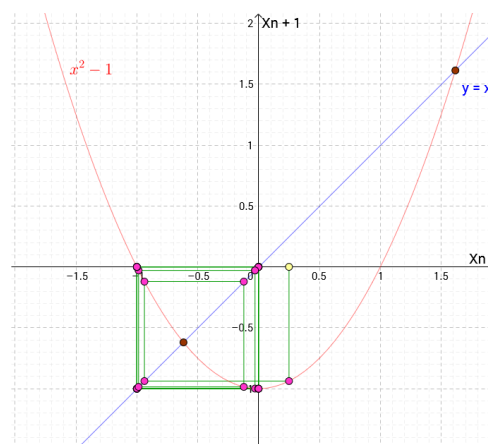
Rysunek 5: Iteracja dla 4)



Rysunek 6: Iteracja dla 5)



Rysunek 7: Iteracja dla 6)



Rysunek 8: Iteracja dla 7)

6.3 Wnioski

Podobnie jak w zadaniu 5, mamy tutaj do czynienia z chaosem deterministycznym, przykłady 1) i 2) są stabilne i nie generują błędów. 3) natomiast jest wykazuje zachowanie chaotyczne, nieuporządkowane, jeżeli przekroczymy odpowiednią liczbę iteracji, 4) oraz 5) są stabilne i nie generują błędów, 6) i 7) generują pewien błąd jednakże, ten błąd jest przez nas porządkany, ponieważ otrzymujemy przewidywalne wyniki, a niewielkie błędy mogą być pomijane, bo satysfakcjonujące wyniki możemy otrzymać w skończonej liczbie iteracji.