Obliczenia Naukowe Lista 2

Paweł Prusisz

25.10.2021

1 Zadanie 1

1.1 Opis problemu

W zadaniu 1 należy ponownie wykonać zadanie 5 z listy 1, ale tym razem dokonując niewielkiej zmiany danych dla 2 wartości w jednym z wektorów. Wektory w tym zadaniu wyglądają następująco:

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

$y = [1486.2497, 878366.9879, -22.37492, 4773714.647, 0.000185049]$

Wyniki obliczeń wektora skalarnego mamy następnie porównać z wynikami z zadania 1.5.

1.2 Rozwiązanie

W tabeli znajdują się wyniki z zadania 1.5 oraz wyniki policzone na potrzeby zadania 2.1

	A	B	C	D
1.5 Float32	-0.2499443	-0.2043457	-0.25	-0.25
1.5 Float64	1.0251881368296672e-10	-1.5643308870494366e-10	0.0	0.0
2.1 Float32	-0.2499443	-0.2043457	-0.25	-0.25
2.1 Float64	-0.004296342739891585	-0.004296342998713953	-0.004296342842280865	-0.004296342842280865

Jak widzimy w przypadku Float32, wyniki pozostały takie same, wynika to z niewielkiej prezycji obliczeń tego typu. W przypadku Float32, który gwarantuje prezycję do ok. 7 miejsc po przecinku w systemie dziesiętnym, usunięcie 10 cyfry po przecinku nie było wystarczająco znaczące, żeby wpłynąć na ostateczny wynik obliczeń.

Wyniki obliczeń we Float64 różnią sie od siebie diametralnie. Porównując wyniki z obliczeń metodami A i B mamy do czynienia z wynikami rzędu 10^{-10} w przypadku zadania z listy 1, oraz wartościami rzędu $10-3$ dla zadania z listy 2. Przykład ten pokazuje jak wrażliwy jest algorytm obliczania iloczynu skalarnego na niewielkie zmiany danych. Mamy tu do czyienia z zadaniem źle uwarunkowanym. Na błąd w otrzymanych wynikach wpływa również fakt, iż wektory X i Y są prawie prostopadłe.

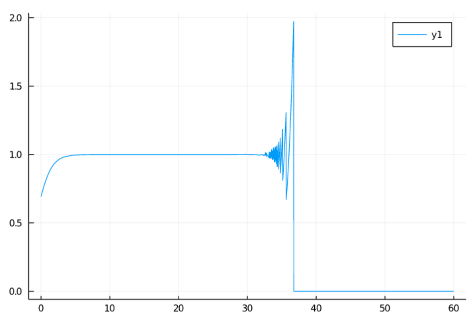
2 Zadanie 2

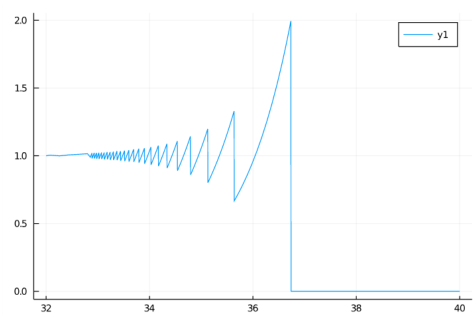
2.1 Opis problemu

W zadaniu 2 należy używając co najmniej 2 różnych narzędzi do wizualizacji, wygenerować wykres funkcji $f(x) = e^x * ln(1 + e^{-x})$, a następnie porównać z wartością $\lim_{x\to\infty} f(x)$.

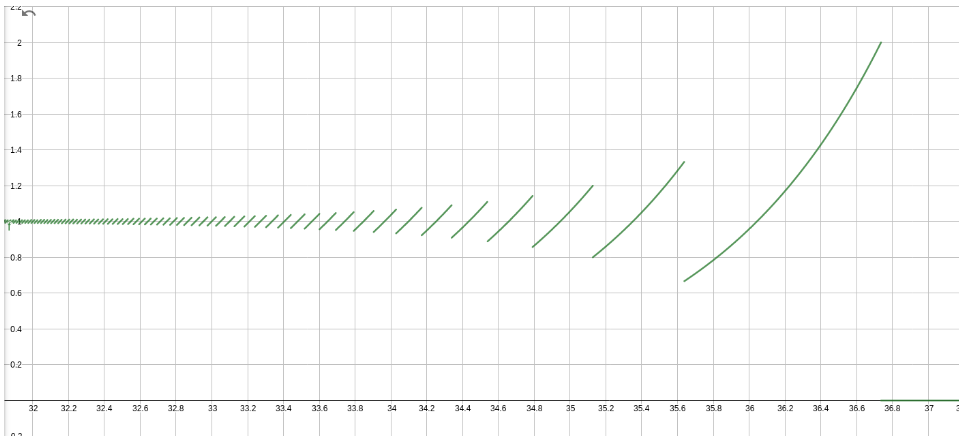
2.2 Rozwiązanie

Poniżej wykresy wygenerowane za pomocą pakietu Plots do języka Julia:





Oraz wykresy wygenerowane za pomocą programu geogebra:



Widzimy, że wartości na wykresach od $x \geq 37$ są równe zero, a granica $\lim_{x\to\infty} f(x) = 1$.

Błąd zaczyna się pojawiać, gdy duże wartości szybko rosnącej funkcji e^x mnożymy z bliskimi zeru wartościami funkcji $ln(1-e^{-x})$. Dla x z przedziału $[30, 40]$ prezentują się następująco:

x	$ln(1+e^{-x})$
30	9.348077867343381e-14
31	3.441691376337926e-14
32	1.2656542480726704e-14
33	4.6629367034256464e-15
34	1.7763568394002489e-15
35	6.661338147750937e-16
36	2.2204460492503128e-16
37	0.0
38	0.0
39	0.0
40	0.0

Jak widać od $x = 37$ Julia zaczyna przyjmować wartość logarytmu jako 0, co pokazuje dlaczego ostatecznie wykres zbiega do 0 zamiast do 1. Jest to kolejny przykład zadania źle uwarunkowanego.

3 Zadanie 3

3.1 Opis problemu

W zadaniu 3 należy rozwiązać układ równań liiowych w postaci $Ax = b$. Macierz A jest generowana na 2 różne sposoby:

(a) $A = H_n$ gdzie H_n jest macierzą Hilberta stopnia n wygenerowaną za pomocą funkcji $A = hilb(n)$

(b) $A = R_n$ gdzie R_n jest losową macierzą stopnia n z zadanym wskaźnikiem uwarunkowania c wygenerowaną za pomocą funkcji $A = matcond(n, c)$

W zadaniu wektor b jest zadany następująco: $b = Ax$, gdzie A jest wygenerowaną macierzą, a $x = (1, ..., 1)^T$

3.2 Rozwiązanie

Tabele poniżej zawierają wyniki zwrócone przez program oraz błędy względne dla obu metod rozwiązywania.

Wyniki dla $A = H_n$

n	cond(A)	Metoda macierzy odwrotnej	Metoda eliminacji Gaussa
2	19.28147006790397	1.4043333874306803e-15	5.661048867003676e-16
3	524.0567775860644	0.0	8.022593772267726e-15
4	15513.73873892924	0.0	4.137409622430382e-14
5	476607.25024259434	3.3544360584359632e-12	1.6828426299227195e-12
6	1.4951058642254665e7	2.0163759404347654e-10	2.618913302311624e-10
7	4.75367356583129e8	4.713280397232037e-9	1.2606867224171548e-8
8	1.5257575538060041e10	3.07748390309622e-7	6.124089555723088e-8
9	4.931537564468762e11	4.541268303176643e-6	3.8751634185032475e-6
10	1.6024416992541715e13	0.0002501493411824886	8.67039023709691e-5
11	5.222677939280335e14	0.007618304284315809	0.00015827808158590435
12	1.7514731907091464e16	0.258994120804705	0.13396208372085344
13	3.344143497338461e18	5.331275639426837	0.11039701117868264
14	6.200786263161444e17	8.71499275104814	1.4554087127659643
15	3.674392953467974e17	7.344641453111494	4.696668350857427
16	7.865467778431645e17	29.84884207073541	54.15518954564602
17	1.263684342666052e18	10.516942378369349	13.707236683836307
18	2.2446309929189128e18	7.575475905055309	9.134134521198485
19	6.471953976541591e18	12.233761393757726	9.720589712655698
20	1.3553657908688225e18	22.062697257870493	7.549915039472976

Wyniki dla $A = R_n$

n	c	Metoda macierzy odwrotnej	Metoda eliminacji Gaussa
5	1.0	9.930136612989092e-17	2.0471501066083611e-16
5	10.0	2.579925170969555e-16	1.3136335981433191e-16
5	1000.0	5.016753763159116e-14	5.0539883677237826e-14
5	1.0e7	6.507814330688532e-11	1.1781096184934976e-11
5	1.0e12	1.8374015165393515e-5	2.3344948097423716e-5
5	1.0e16	0.03797309002973355	0.02697104191450584
10	1.0	3.1006841635969763e-16	2.696722356863272e-16
10	10.0	2.406906162008981e-16	3.439900227959406e-16
10	1000.0	3.7033951582531006e-14	3.551165968336269e-14
10	1.0e7	9.090580394919659e-11	8.129420332163008e-11
10	1.0e12	4.023782384891318e-5	3.6364358182678266e-5
10	1.0e16	0.04899191279722858	0.013572604100535774
20	1.0	4.2130001622920406e-16	5.4672143489065705e-16
20	10.0	5.063396036227354e-16	4.820209419629775e-16
20	1000.0	1.2563879461962553e-14	8.651052721635365e-15
20	1.0e7	1.7885393104053008e-10	2.3672558048186985e-10
20	1.0e12	3.664641696861926e-5	3.744353418841211e-5
20	1.0e16	0.06136046122033206	0.02650423094707032

W przypadku macierzy Hilberta H_n błąd dla obu metod rośnie wraz z rozmiarem macierzy n. Podobnie zachoduje się wskaźnik uwarunkowania.

W przypadku Macierzy R_n, podobnie jak dla c, błąd rośnie wraz ze wzrostem wartości n, jednak jest znacznie wolniejszy wzrost. Błąd ten jest porównywalny z H_n dopiero, gdy rośnie wartość c.

Ostatecznie widzimy, że w przypadku macierzy Hilberta H_n zadanie jest źle uwarunkowane. W przypadku R_n, jeżeli wartość c nie jest zbyt duża, wyniki są obarczone względnie małym błędem, nawet jeśli wielkość macierzy rośnie.

4 Zadanie 4

4.1 Opis problemu

W zadaiu 4 należy zbadać problem obliczania miejsc zerowych wielomianu zaproponowanego przez wilkinsona. Wielomian dany jest w postaci kanonicznej (1) oraz w postaci iloczynowej (2)

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

(2) $p(x) = (x-20)(x-19)(x-18)(x-17)(x-16)(x-15)(x-14)(x-13)(x-12)(x-11)(x-10)(x-9)(x-8)(x-7)(x-6)(x-5)(x-4)(x-3)(x-2)(x-1)$

Dodatkowo mamy sprawdzić wpływ zmiany współczynnika 210 na $210 - 2^{-23}$ na obliczone przez nas miejsca zerowe.

4.2 Rozwiązanie

Wyniki obliczeń dla przypadku wielomianu bez zaburzeń przezentują się następująco:

n	z_k	$\|P(z_k)\|$	$\|p(z_k)\|$	$\|z_k - k\|$
1	0.9999999999996989	36352.0	36626.425482422805	3.0109248427834245e-13
2	2.0000000000283182	181760.0	181303.93367257662	2.8318236644508943e-11
3	2.9999999995920965	209408.0	290172.2858891686	4.0790348876384996e-10
4	3.9999999837375317	3.106816e6	2.0415372902750901e6	1.626246826091915e-8
5	5.000000665769791	2.4114688e7	2.0894625006962188e7	6.657697912970661e-7
6	5.999989245824773	1.20152064e8	1.1250484577562995e8	1.0754175226779239e-5
7	7.000102002793008	4.80398336e8	4.572908642730946e8	0.00010200279300764947
8	7.999355829607762	1.682691072e9	1.5556459377357383e9	0.0006441703922384079
9	9.002915294362053	4.465326592e9	4.687816175648389e9	0.002915294362052734
10	9.990413042481725	1.2707126784e10	1.2634601896949205e10	0.009586957518274986
11	11.025022932909318	3.5759895552e10	3.300128474498415e10	0.025022932909317674
12	11.953283253846857	7.216771584e10	7.388525665404988e10	0.04671674615314281
13	13.07431403244734	2.15723629056e11	1.8476215093144193e11	0.07431403244734014
14	13.914755591802127	3.65383250944e11	3.5514277528420844e11	0.08524440819787316
15	15.075493799699476	6.13987753472e11	8.423201558964254e11	0.07549379969947623
16	15.946286716607972	1.555027751936e12	1.570728736625802e12	0.05371328339202819
17	17.025427146237412	3.777623778304e12	3.3169782238892363e12	0.025427146237412046
18	17.99092135271648	7.199554861056e12	6.34485314179128e12	0.009078647283519814
19	19.00190981829944	1.0278376162816e13	1.228571736671966e13	0.0019098182994383706
20	19.999809291236637	2.7462952745472e13	2.318309535271638e13	0.00019070876336257925

W przypadku wielomianu z zaburzonym 2 czynnikiem wyniki prezentują sie następująco:

n	z_k	$\|P(z_k)\|$	$\|p(z_k)\|$	$\|z_k - k\|$
1	0.9999999999998357 + 0.0im	20496.0	19987.872313406835	1.6431300764452317e-13
2	2.0000000000550373 + 0.0im	339570.0	352369.4138087958	5.503730804434781e-11
3	2.99999999660342 + 0.0im	2.2777455e6	2.4162415582518433e6	3.3965799062229962e-9
4	4.000000089724362 + 0.0im	1.0488020625e7	1.1263702300292023e7	8.972436216225788e-8
5	4.99999857388791 + 0.0im	4.1239073125e7	4.475744423806908e7	1.4261120897529622e-6
6	6.000020476673031 + 0.0im	1.406328934140625e8	2.1421031658039317e8	2.0476673030955794e-5
7	6.99960207042242 + 0.0im	4.122812662421875e8	1.7846173427860644e9	0.00039792957757978087
8	8.007772029099446 + 0.0im	1.0307901272578125e9	1.8686972170009857e10	0.007772029099445632
9	8.915816367932559 + 0.0im	2.1574055781816406e9	1.3746309775142993e11	0.0841836320674414
10	10.095455630535774 - 0.6449328236240688im	9.384147605647182e9	1.490069535200058e12	0.6519586830380407
11	10.095455630535774 + 0.6449328236240688im	9.384147605647182e9	1.490069535200058e12	1.1109180272716561
12	11.793890586174369 - 1.6524771364075785im	3.0012060598372482e10	3.2962792355717145e13	1.665281290598479
13	11.793890586174369 + 1.6524771364075785im	3.0012060598372482e10	3.2962792355717145e13	2.0458202766784277
14	13.992406684487216 -2.5188244257108443im	2.0030917431984006e11	9.546022365750216e14	2.518835871190904
15	13.992406684487216 + 2.5188244257108443im	2.0030917431984006e11	9.546022365750216e14	2.7128805312847097
16	16.73074487979267 - 2.812624896721978im	1.1583329328642004e12	2.742106076928478e16	2.9060018735375106
17	16.73074487979267 + 2.812624896721978im	1.1583329328642004e12	2.742106076928478e16	2.825483521349608
18	19.5024423688181 - 1.940331978642903im	5.867381806750561e12	4.2524858765203725e17	2.4540214463129764
19	19.5024423688181 + 1.940331978642903im	5.867381806750561e12	4.2524858765203725e17	2.0043294443099486
20	20.84691021519479 + 0.0im	9.550552334336e12	1.37437435599976e18	0.8469102151947894

Wartości wielomianów $P(x)$ oraz $p(x)$ dla znalezionych "miejsc zerowych" nie mają się nawet blisko 0. Pokazuje to jak bardzo mały błąd w obliczeniu miejsca zerowego np. $z_1 = 0.9999999999996989$, który jest wartością bardzo bliską 1, daje nam dla $P(z_1) = 36352.0$, zamiast spodziewanego wyniku 0. Co więcej, gdy jeden ze współczynników wielomianu, -210, zamienimy na $-210 - 2^{-23}$, to dostajemy pierwiaski zespolone.

Błędne wartości w podpunkcie a) wynikają z ograniczeń arytmetyki Float64, która pozwala na zapamiętywanie z dokładnością do 16 miejsc znaczących, a współczynniki wielomianu mają nawet do 19 cyfr. Oznacza to, że współczynniki zapamiętywanego wielomianu są pamiętane z błędem. W podpunkcie b) widzimy jak zmienienie jednego ze współczynników o wartość 2^{-23} znacząco zmienia wynik, a nawet daje nam wartości zespolone, co pokazuje, że zadanie jest źle uwarunkowane.

5 Zadanie 5

5.1 Opis problemu

W zadaniu 5 należy przeanalizowac równanie rekurencyjne modelujące wzrost populacji:

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

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

Naszym zadaniem jest przeprowadzenia 2 ekspetymentów.

1. Dla daych $p_0 = 0.01$ i $r = 3$ wykonać 40 iteracji, a następnie wykonać 40 iteracji z niewielką modyfikacją wyniku w 10 iteracji.

2. Dla daych $p_0 = 0.01$ i $r = 3$ wykonać 40 iteracji wyrażenia w arytmetyce Float32 i Float64. Porównać otrzymane wyniki.

5.2 Rozwiązanie

Wyniki obliczeń prezentują się następująco:

n	Float32	Float32 obcięcie	Float64
1	0.03969999939873814	0.03969999939873814	0.0397
2	0.15407172773817313	0.15407172773817313	0.15407173000000002
3	0.5450726190880153	0.5450726190880153	0.5450726260444213
4	1.2889779961136554	1.2889779961136554	1.2889780011888006
5	0.1715191610590978	0.1715191610590978	0.17151914210917552
6	0.5978201764051411	0.5978201764051411	0.5978201201070994
7	1.3191138156693423	1.3191138156693423	1.3191137924137974
8	0.0562714866081746	0.0562714866081746	0.056271577646256565
9	0.21558650581741648	0.21558650581741648	0.21558683923263022
10	0.7229133987979771	0.722	0.722914301179573
11	1.3238422487069792	1.3241479990501404	1.3238419441684408
12	0.037694096443262826	0.03648822603508983	0.03769529725473175
13	0.1465138510530293	0.14195873222279592	0.14651838271355924
14	0.5216564785609494	0.5073780839282732	0.521670621435246
15	1.2702494693699669	1.2572147755609153	1.2702617739350768
16	0.2403967341758202	0.2870921265776132	0.24035217277824272
17	0.7882151672960809	0.9011028388818839	0.7881011902353041
18	1.2890112193175571	1.1684523768045643	1.2890943027903075
19	0.17139510669062252	0.5779666366375508	0.17108484670194324
20	0.5974515789700204	1.309730247351835	0.5965293124946907
21	1.31896114823877	0.09274102692244313	1.3185755879825978
22	0.05686906126507618	0.3451614134658845	0.058377608259430724
23	0.21777397467279178	1.0232364498262363	0.22328659759944824
24	0.7288193865568098	0.9519073025459458	0.7435756763951792
25	1.3217444515641055	1.0892466722628869	1.315588346001072
26	0.0459526205349281	0.7976117499442289	0.07003529560277899
27	0.1774755521376311	1.2818934888296303	0.26542635452061003
28	0.6154094937308534	0.19782120520711688	0.8503519690601384
29	1.3254514400012178	0.6738851331396788	1.2321124623871897
30	0.03134120060096546	1.3331770145586672	0.37414648963928676
31	0.12241798983853196	0.0006252017919865516	1.0766291714289444
32	0.44471346664580713	0.0024996345361040966	0.8291255674004515
33	1.1855436643348343	0.009979793625974134	1.2541546500504441
34	0.5256333172059385	0.039620385661445434	0.29790694147232066
35	1.2736621163529978	0.15377221776589672	0.9253821285571046
36	0.22800290551359947	0.5441511861936599	1.1325322626697856
37	0.7560556472864678	1.2883032044667382	0.6822410727153098
38	1.3093621637645918	0.1740373779491542	1.3326056469620293
39	0.09416082736348597	0.6052824850263665	0.0029091569028512065
40	0.35004452522461527	1.3220292800663853	0.011611238029748606

Jak widać zaokrąglenie w 10 iteracji spowodowało, że już od 19 iteracji wyniki obliczenń są całkowicie różne od siebie. Widzimy w tym przypadku jak stosunkowo niewielki błąd może nawarstwiać się w kolejnych iteracjach całkowicie przekłamując wyniki obliczeń. Raz popełniony błąd propaguje na kolejne wartości ciągu które są zależne od poprzednich. Otrzymane w ten sposób wartości są tak samo pewne jak liczby zwracane przez generator liczb losowych.

Podobną sytuacje można zaobserwować porównując wyniki liczone w arytmetyce Float32 i Floar64. Od 26 iteracji wyniki zaczynają się znacząco rozjeżdzać. Brak korelacji pomiędzy wynikami od momentu gdzieś $20-22$ iteracji obrazuje pojęcie chaosu deterministycznego. Mamy tutaj do czynienia ze zjawiskiem czułej zależności od warunków początkowych.

6 Zadanie 6

6.1 Opis problemu

W zadaniu 6 należuuy rozważyć następujące równanie rekurencyjne:

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

dla $n \in N$, gdzie c jest pewną stałą.
Eksperyment należy przeprowadzić dla następujących danych:

1. $c = -2$ i $x_0 = 1$
2. $c = -2$ i $x_0 = 2$
3. $c = -2$ i $x_0 = 1.99999999999999$
4. $c = -1$ i $x_0 = 1$
5. $c = -1$ i $x_0 = -1$
6. $c = -1$ i $x_0 = 0.75$
7. $c = -1$ i $x_0 = 0.25$

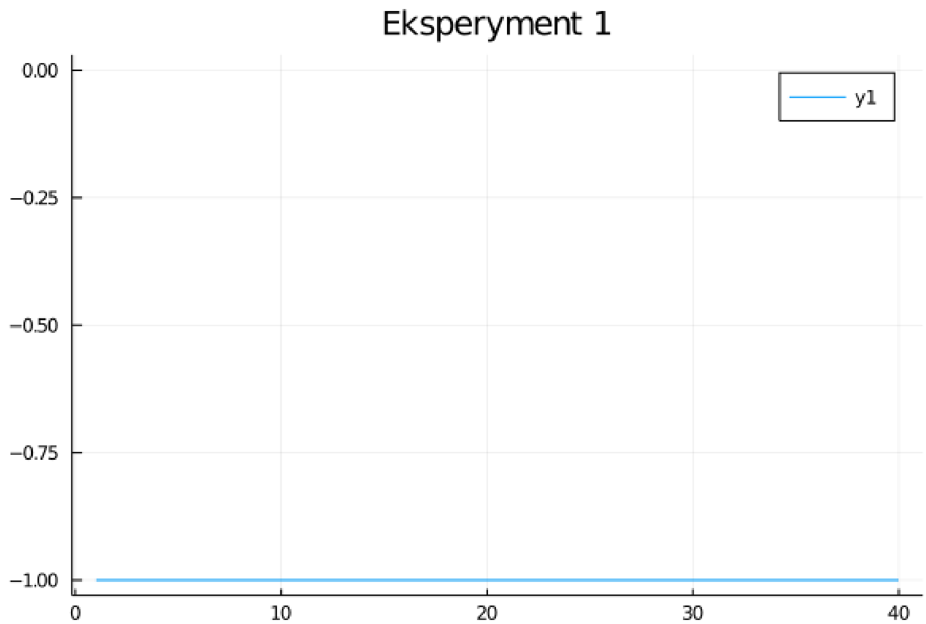
Dla każdego eksperymentu należy przeprowadzić 40 iteracji.

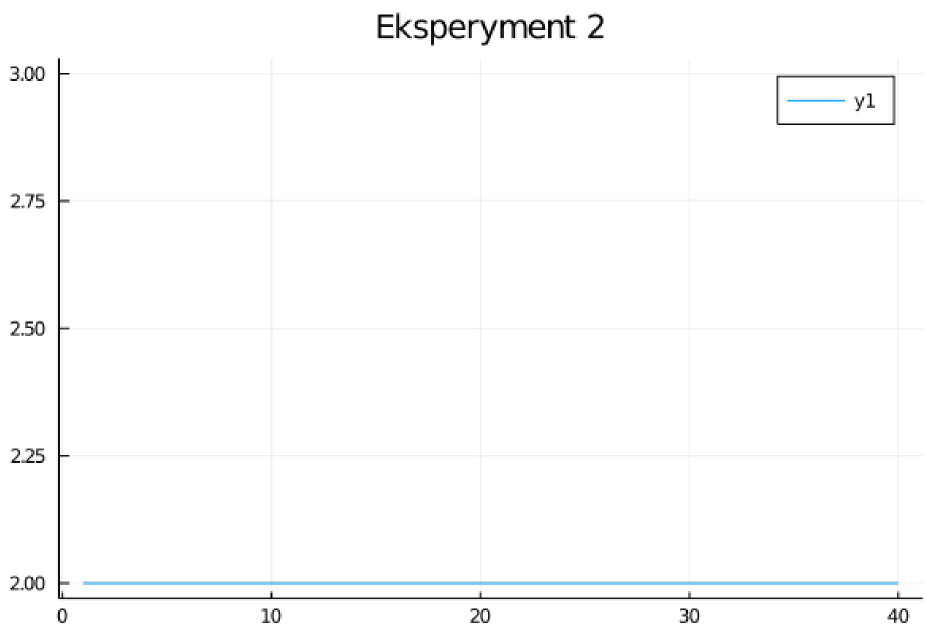
6.2 Rozwiązanie

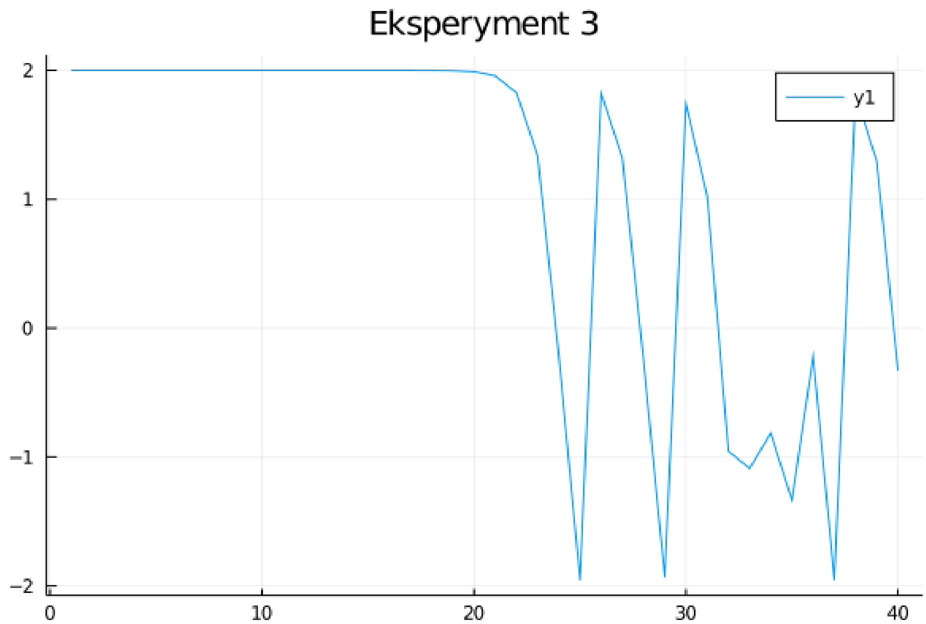
Wyniki iteracji dla kolejnych eksperymentów prezentują się następująco:

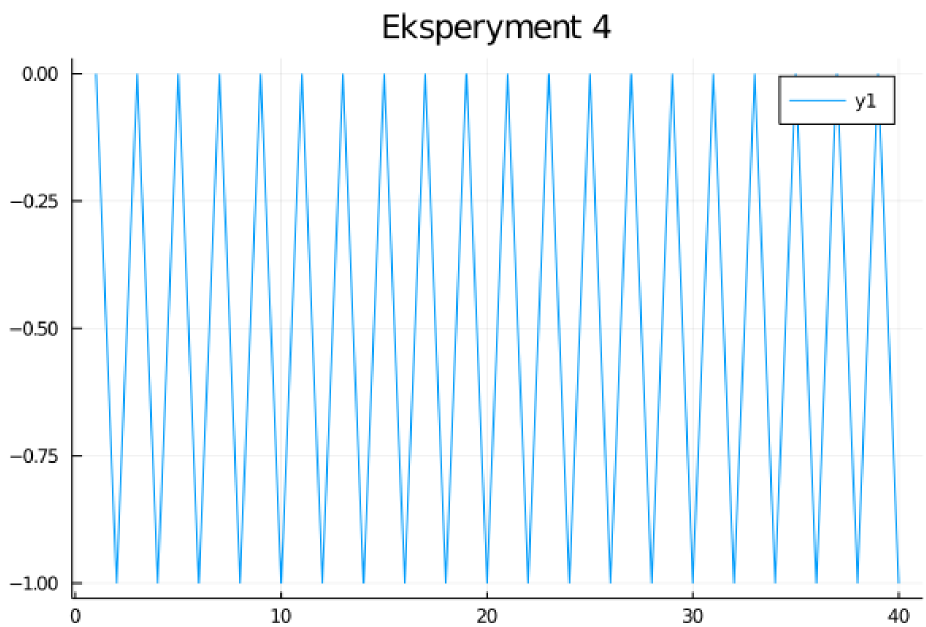
n	1	2	3	4	5	6	7
1	-1.0	2.0	1.99999999999996	0.0	0.0	-0.4375	-0.9375
2	-1.0	2.0	1.9999999999998401	-1.0	-1.0	-0.80859375	-0.12109375
3	-1.0	2.0	1.9999999999993605	0.0	0.0	-0.3461761474609375	-0.9853363037109375
4	-1.0	2.0	1.999999999997442	-1.0	-1.0	-0.8801620749291033	-0.029112368589267135
5	-1.0	2.0	1.9999999999897682	0.0	0.0	-0.2253147218564956	-0.9991524699951226
6	-1.0	2.0	1.9999999999590727	-1.0	-1.0	-0.9492332761147301	-0.0016943417026455965
7	-1.0	2.0	1.999999999836291	0.0	0.0	-0.0989561875164966	-0.9999971292061947
8	-1.0	2.0	1.9999999993451638	-1.0	-1.0	-0.9902076729521999	-5.741579369278327e-6
9	-1.0	2.0	1.9999999973806553	0.0	0.0	-0.01948876442658909	-0.9999999999670343
10	-1.0	2.0	1.999999989522621	-1.0	-1.0	-0.999620188061125	-6.593148249578462e-11
11	-1.0	2.0	1.9999999580904841	0.0	0.0	-0.0007594796206411569	-1.0
12	-1.0	2.0	1.9999998323619383	-1.0	-1.0	-0.9999994231907058	0.0
13	-1.0	2.0	1.9999993294477814	0.0	0.0	-1.1536182557003727e-6	-1.0
14	-1.0	2.0	1.9999973177915749	-1.0	-1.0	-0.9999999999986692	0.0
15	-1.0	2.0	1.9999892711734937	0.0	0.0	-2.6616486792363503e-12	-1.0
16	-1.0	2.0	1.9999570848090826	-1.0	-1.0	-1.0	0.0
17	-1.0	2.0	1.999828341078044	0.0	0.0	0.0	-1.0
18	-1.0	2.0	1.9993133937789613	-1.0	-1.0	-1.0	0.0
19	-1.0	2.0	1.9972540465439481	0.0	0.0	0.0	-1.0
20	-1.0	2.0	1.9890237264361752	-1.0	-1.0	-1.0	0.0
21	-1.0	2.0	1.9562153843260486	0.0	0.0	0.0	-1.0
22	-1.0	2.0	1.82677862987391	-1.0	-1.0	-1.0	0.0
23	-1.0	2.0	1.3371201625639997	0.0	0.0	0.0	-1.0
24	-1.0	2.0	-0.21210967086482313	-1.0	-1.0	-1.0	0.0
25	-1.0	2.0	-1.9550094875256163	0.0	0.0	0.0	-1.0
26	-1.0	2.0	1.822062096315173	-1.0	-1.0	-1.0	0.0
27	-1.0	2.0	1.319910282828443	0.0	0.0	0.0	-1.0
28	-1.0	2.0	-0.2578368452837396	-1.0	-1.0	-1.0	0.0
29	-1.0	2.0	-1.9335201612141288	0.0	0.0	0.0	-1.0
30	-1.0	2.0	1.7385002138215109	-1.0	-1.0	-1.0	0.0
31	-1.0	2.0	1.0223829934574389	0.0	0.0	0.0	-1.0
32	-1.0	2.0	-0.9547330146890065	-1.0	-1.0	-1.0	0.0
33	-1.0	2.0	-1.0884848706628412	0.0	0.0	0.0	-1.0
34	-1.0	2.0	-0.8152006863380978	-1.0	-1.0	-1.0	0.0
35	-1.0	2.0	-1.3354478409938944	0.0	0.0	0.0	-1.0
36	-1.0	2.0	-0.21657906398474625	-1.0	-1.0	-1.0	0.0
37	-1.0	2.0	-1.953093509043491	0.0	0.0	0.0	-1.0
38	-1.0	2.0	1.8145742550678174	-1.0	-1.0	-1.0	0.0
39	-1.0	2.0	1.2926797271549244	0.0	0.0	0.0	-1.0
40	-1.0	2.0	-0.3289791230026702	-1.0	-1.0	-1.0	0.0

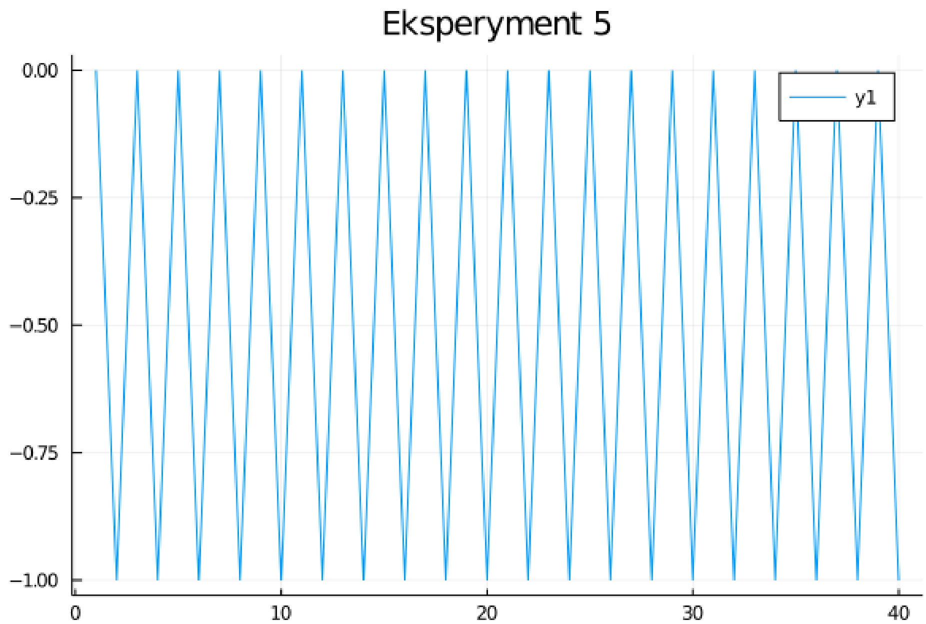
Wykresy dla odpowienich eksperymentów:

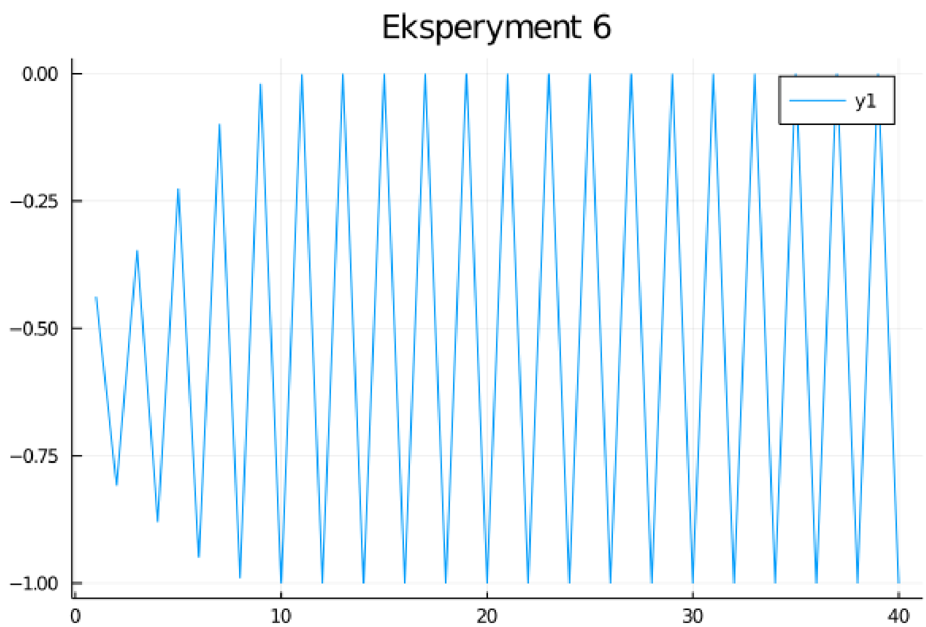

Eksperyment 1
0.00
−0.25
−0.50
−0.75
−1.00
0
10
20
30
40
y1

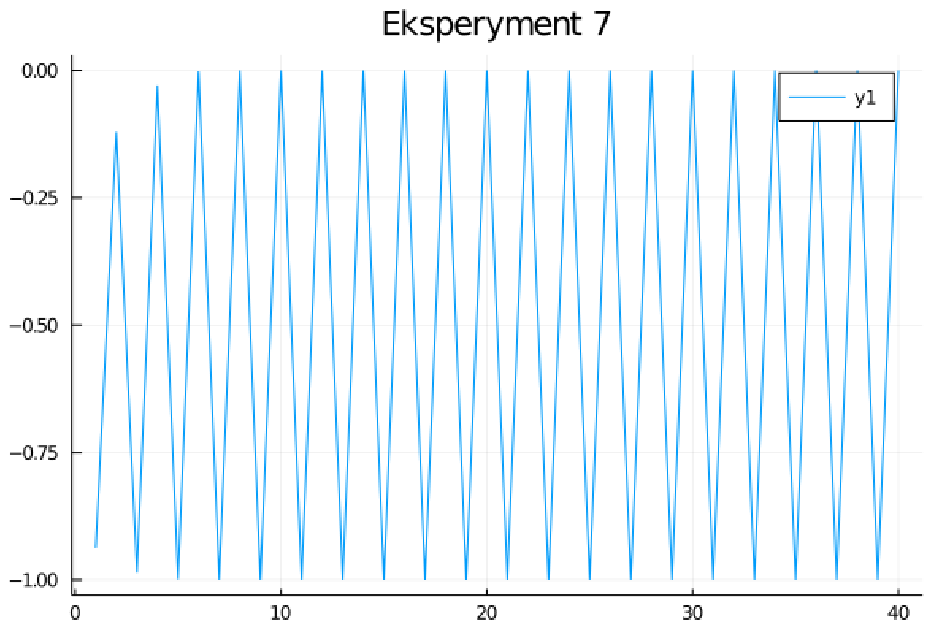

Eksperyment 2
3.00
2.75
2.50
2.25
2.00
0
10
20
30
40
y1


Eksperyment 3
y1
2
1
0
−1
−2
0
10
20
30
40


Eksperyment 4
y1
0.00
−0.25
−0.50
−0.75
−1.00
0
10
20
30
40


Eksperyment 5
y1
0.00
−0.25
−0.50
−0.75
−1.00
0
10
20
30
40


Eksperyment 6
y1
0.00
−0.25
−0.50
−0.75
−1.00
0
10
20
30
40



W przypadku eksperymentów 4, 5, 6 i 7 widzimy reguralne wahania wartości funkcji na przedziale wartości $[-1, 0]$.

W przypadku wykresów 1 i 2 widzimy linię prostą, a wykres 3 stopniowo wyłamuje się z okolic wartości 2.0 i zaczyna losowo skakać między wartościami.

Dla eksperymentów 4 i 5 mamy do czynienia z ciągami, których podciągi zbiegają do 0 lub -1 w zależności od parzystości n.

Podobnie ma się sytuacja w eksperymencie 6 z tą różnicą, że zjawisko to zaczyna dopiero zachodzić dla $n > 16$, a dla początkowych wartości oscylują one między 0 a -1.

Dla eksperymentu 7 mamy podobną sytuacje jak w eksperymencie 6 z tą różnicą, że oscylacja między 0 a -1 zaczyna się dla $n > 10$ oraz wartość 0 jest przyjmowana dla parzystych wartości n, przeciwnie do przypadku 6 gdzie dla parzystych n eksperyment 6 przyjmował wartości -1.