Obliczenia Naukowe

Lista II

Kacper Pieniążek, 236606

Wtorek TP 15:15

11 listopada 2018

Spis treści

1 Zadanie 1

1.1 Cel zadania

Celem jest sprawdzenie, jaki wpływ ma zmiana danych wejściowych na wyniki obliczanych iloczynów skalarnych z zadania 5 poprzedniej listy.

1.2 Rozwiązanie

Do rozwiązania wykorzystany został program z poprzedniej listy, gdzie zostało przedstawione działanie programu, dokładne dane wejściowe oraz wykorzystane algorytmy. Modyfikacji uległy jedynie dane wejściowe. Usunięta została ostatnia cyfra z x_4 oraz x_5 i tak samo, jak dla niezmienionych danych, obliczono iloczyn skalarny na cztery sposoby.

1.2.1 Wyniki

Tablica 1: Niezmodyfikowane dane wejściowe

Sposób	Float32	Float64
I	-0.4999443	1.0251881368296672e-10
II	-0.4543457	-1.5643308870494366e-10
III	-0.5	0.0
IV	-0.5	0.0

Tablica 2: Nowe dane wejściowe

Sposób	Float32	Float64
I	-0.4999443	-0.004296342739891585
II	-0.4543457	-0.004296342998713953
III	-0.5	-0.004296342842280865
IV	-0.5	-0.004296342842280865

1.3 Wnioski

Wprowadzone zmiany w danych wejściowych nie mają żadnego wpływu na wynik dla $Float32$. Wynika to z tego, że wprowadzone zmiany są zbyt małe dla tego typu. Według standardu $IEEE754$ $Float32$ oferuje precyzję do 7 cyfr dziesiętnych, a zmiana w x_5 jest rzędu $2 \cdot 10^{-8}$. Oznacza to, że nawet po modyfikacji dane wejściowe wymagają zbyt dużej precyzji obliczeń dla typu $Float32$.
Zmiany wyników dla typu $Float64$ pokazują jednoznacznie, że zmiany miały znaczenie dla tej precyzji. Dodatkowo, wyniki czterech metod nie są już tak rozbieżne, jak wcześniej.
Obliczanie iloczynu skalarnego jest źle uwarunkowane dla par x_i, y_i t. że $x_i \cdot y_i < 0$, co tłumaczy dosyć duże różnice wyników przy tak niewielkich zmianach.

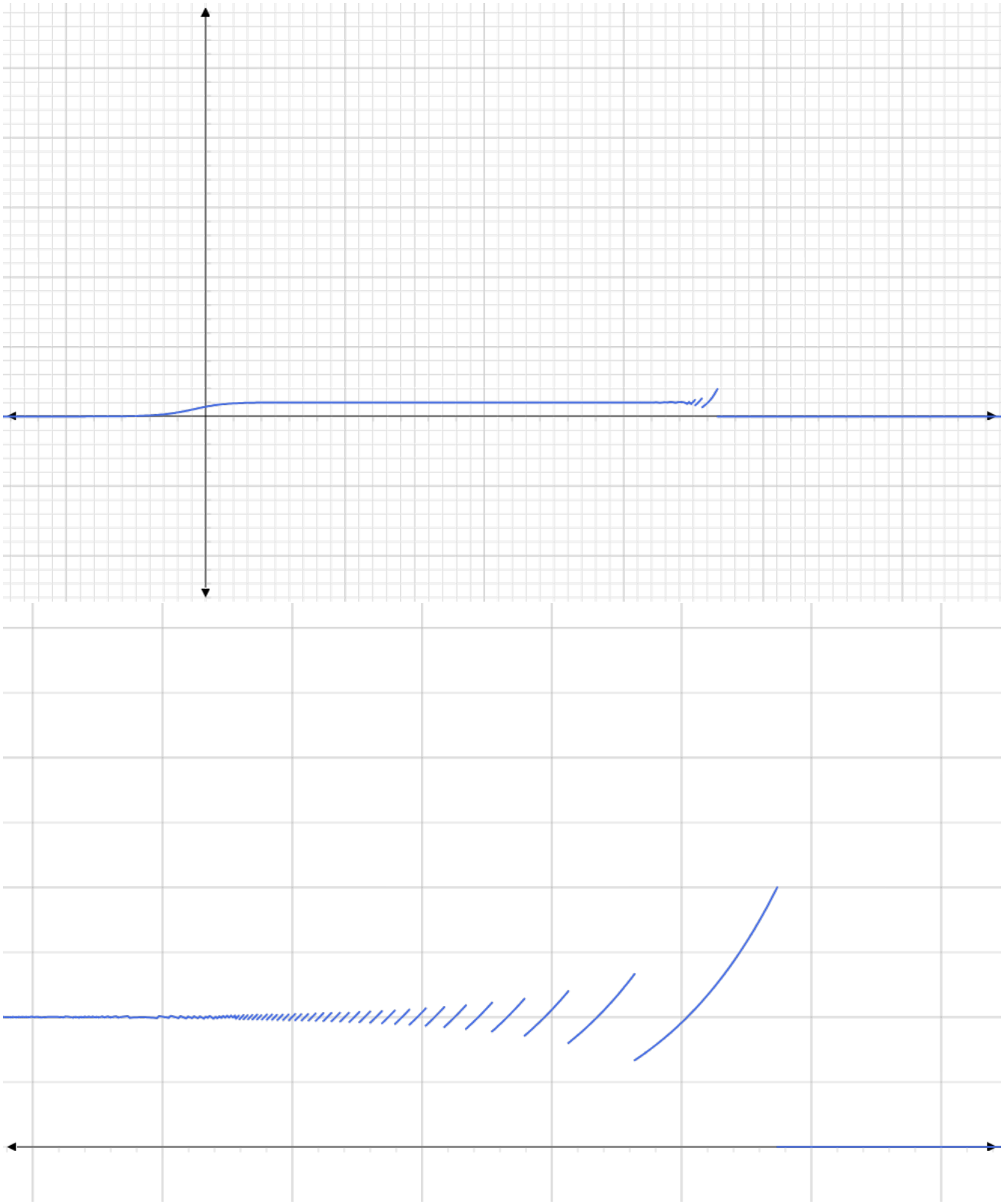
2 Zadanie 2

2.1 Cel zadania

Celem zadania jest obserwacja zachowania funkcji $f(x) = e^x ln(1 + e^{-x})$ dla kolejnych wartości, policzenie granicy $lim_{x \to \text{inf}} f(x)$ oraz porównanie wykresu funkcji z policzoną granicą.

2.2 Rozwiązanie

$lim_{x \to \text{inf}}\ e^x ln(1 + e^{-x}) = 1$
Do narysowania funkcji wykorzystano silnik Wolfram Alpha oraz bibliotekę Pyplot. Dla wartości x około 35 widać wyraźne zniekształcenia funkcji.



2.3 Wnioski

Na wykresach granica wydaje się zbiegać do 1 dla niewielkich x, jednak dla wystarczająco dużych x orzymujemy $ln(1 + e^{-x}) = 0$. Dzieje się tak dlatego, że e^{-x} zbiega do 0 i dla wartości bardzo bliskich granicy w końcu $e^{-x} = 0$, co jest wynikiem zaokrąglania obliczeń oraz ograniczonej precyzji.
Wyrażenie e^{-x} jest źle uwarunkowane, można zauważyć wahania w wartościach dla x, zanim funkcja osiąga wartość 0.

3 Zadanie 3

3.1 Cel zadania

Zadanie polega na rozwiązaniu układu równań liniowych postaci

$$\mathbf{Ax} = \mathbf{b}$$

gdzie $\mathbf{A} \in \mathbb{R}^{n \times n}$ jest macierzą współczynników, $\mathbf{b} \in \mathbb{R}^{n}$ jest wektorem prawych stron.
Znane są dokładne wartości dla $\mathbf{A}$ oraz $\mathbf{b}$.
Macierz $\mathbf{A}$ jest generowana jako macierz Hilberta $H_n(\forall n \in \{2, 3, \cdots, 20\})$ oraz jako macierz losowa $R_n, n \in \{5, 10, 20\}$ ze wskaźnikiem uwarunkowania $c \in \{1, 10, 10^3, 10^7, 10^{12}, 10^{16}\}$.
Wektor $\mathbf{b}$ jest zadany jako $\mathbf{b} = \mathbf{Ax}$, gdzie $\mathbf{x} = (1, \cdots, 1)^T$.

3.2 Rozwiązanie

Za generowanie odpowiednich macierzy odpowiadają funkcje dołączone do zadania. Pierwszy sposób liczenia $\mathbf{x}$ odbywa się za pomocą eliminacji Gaussa. W języku Julia $A\backslash b$. Drugi sposób to przekształcenie wyrażenia do postaci $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, zaimplementowane jako $x = inv(A) * b$.
Błędy względne zostały obliczone przy użyciu normy $\frac{norm(x,1)-norm(x_0,1)}{norm(x,1)}$, gdzie x to wartość dokładna $(1, \cdots, 1)^T$, x_0 to wartość obliczona w zadaniu.

3.2.1 Wyniki

Tablica 3: Wyniki dla macierzy Hilberta H_n

n	cond(A)	rank(A)	$\lvert e_{gauss} \rvert$	$\lvert e_{inv} \rvert$
2	19.28147006790397	2	5.551115123125783e-16	1.0547118733938987e-15
3	524.0567775860644	3	6.994405055138486e-15	8.178642948071987e-15
4	15513.73873892924	4	3.7067571234672414e-13	2.4993895841873837e-13
5	476607.25024259434	5	1.3548051569500785e-12	6.04918337643312e-12
6	1.4951058642254665e7	6	1.992921383475732e-10	2.348237530351109e-10
7	4.75367356583129e8	7	9.265715845379369e-9	2.3892148418421957e-9
8	1.5257575538060041e10	8	7.296422475933095e-8	1.34114088723436e-7
9	4.931537564468762e11	9	3.348640447746002e-6	6.10150057503114e-6
10	1.6024416992541715e13	10	0.0004282990960899014	0.0003092186946361153
11	5.222677939280335e14	11	0.007636021054455983	0.005133252578037045
12	1.7514731907091464e16	11	0.1928599575355822	0.20347770551840463
13	3.344143497338461e18	11	1.5118132061604694	2.4681875707947545
14	6.200786263161444e17	12	3.132118996548091	2.6789253859688307
15	3.674392953467974e17	12	0.7804591637192894	3.0374592003923455
16	7.865467778431645e17	12	14.501848726499087	18.722663938999176
17	1.263684342666052e18	12	11.422172075251698	9.626610203699938
18	2.2446309929189128e18	12	2.616100179986132	3.525611314165886
19	6.471953976541591e18	13	14.935729436732025	14.217555962909666
20	1.3553657908688225e18	13	10.015300078430128	10.05004960000515
21	3.290126328601399e18	13	13.984874752102568	17.126295607542968

Tablica 4: Wyniki dla macierzy losowych R_n

n	cond(A)	rank(A)	$\|e_{gauss}\|$	$\|e_{inv}\|$
5	10^0	5	2.2204460492503132e-17	1.3322676295501878e-16
5	10^1	5	2.886579864025407e-16	3.108624468950438e-16
5	10^3	5	1.6875389974302379e-15	1.5987211554602255e-15
5	10^7	5	2.6691449050986195e-11	6.293703336268664e-11
5	10^{12}	5	1.554312234475219e-16	7.62939453125e-6
5	10^{16}	4	0.43960885200414734	0.4875
10	10^0	10	1.4432899320127036e-16	2.55351295663786e-16
10	10^1	10	2.331468351712829e-16	3.108624468950438e-16
10	10^3	10	2.589040093425865e-14	2.0095036745715333e-14
10	10^7	10	4.575483769642119e-10	4.700268618762493e-10
10	10^{12}	10	1.7563734017211895e-5	1.735687255859375e-5
10	10^{16}	9	0.14261350023112038	0.1412109375
20	10^0	20	3.885780586188048e-16	3.275157922644212e-16
20	10^1	20	3.6637359812630164e-16	4.440892098500626e-16
20	10^3	20	2.707833957060757e-14	2.6334490144108712e-14
20	10^7	20	2.0347719065227922e-10	2.3101165425032377e-10
20	10^{12}	20	1.043664711670167e-5	1.5103816986083985e-5
20	10^{16}	19	0.13200934197172365	0.15966796875

3.3 Wnioski

Na podstawie zaprezentowanych wyników można zauważyć, że błędy względne rosną dla coraz wyższych wskaźników uwarunkowania macierzy. Uwarunkowanie macierzy ma znaczny wpływ na wynik równania $Ax = b$, więc uwarunkowanie tego zadania jest zależne od wskaźnika. Stąd wniosek, że macierz Hilberta H_n jest przykładem macierzy źle uwarunkowanej.

4 Zadanie 4

4.1 Cel zadania

Celem jest obserwacja, jak zachowuje się Perfidny wielomian Wilkinsona przy niewielkich zmianach jednego ze współczynników $a_{19} = -210$ na $a'_{19} = -210 - 2^{-23}$ oraz obliczenie pierwiastków wielomianu i ich sprawdzenie, tj. $\forall z_k$ wyznaczyć $|P(z_k)|$, $|p(z_k)|$, $|z_k - k|$, gdzie P to wielomian Wilkinsona w postaci naturalnej, p to postać iloczynowa.

4.2 Rozwiązanie

Do rozwiązania wykorzystano pakiet **Polynomials** do języka Julia. W rozwiązaniu znajdują się funkcje *poly*, *polyval*, *Poly*, *roots*.
poly tworzy wielomian iloczynowy na podstawie podanych miejsc zerowych,
Poly tworzy wielomian z podanych współczynników,
polyval rozwiązuje wielomian dla podanego x,
roots wyznacza pierwiastki danego wielomianu.
Dodatkowo, oznaczono $\Delta = (0, \delta, 0, 0, \cdots, 0)$, gdzie $\delta = -2^{-23}$ jest współczynnikiem stojącym przy 19. potędzie wielomianu.

4.2.1 Wyniki

Tablica 5: Obliczone wartości dla wielomianu Wilkinsona P

n	$\lvert P(x)\rvert$	$\lvert p(x)\rvert$	$\lvert z_k - k\rvert$
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

Tablica 6: Obliczone wartości dla wielomianu $P + \Delta$

n	$\lvert P(x)\rvert$	$\lvert p(x)\rvert$	$\lvert z_k - k\rvert$
1	20992.0	22016.0	1.6431300764452317e-13
2	349184.0	365568.0	5.503730804434781e-11
3	2.221568e6	2.295296e6	3.3965799062229962e-9
4	1.046784e7	1.0729984e7	8.972436216225788e-8
5	3.9463936e7	4.3303936e7	1.4261120897529622e-6
6	1.29148416e8	2.06120448e8	2.0476673030955794e-5
7	3.88123136e8	1.757670912e9	0.00039792957757978087
8	1.072547328e9	1.8525486592e10	0.007772029099445632
9	3.065575424e9	1.37174317056e11	0.0841836320674414
10	7.143113638035824e9	1.4912633816754019e12	0.6519586830380406
11	7.143113638035824e9	1.4912633816754019e12	1.1109180272716561
12	3.357756113171857e10	3.2960214141301664e13	1.665281290598479
13	3.357756113171857e10	3.2960214141301664e13	2.045820276678428
14	1.0612064533081976e11	9.545941595183662e14	2.5188358711909045
15	1.0612064533081976e11	9.545941595183662e14	2.7128805312847097
16	3.315103475981763e11	2.7420894016764064e16	2.9060018735375106
17	3.315103475981763e11	2.7420894016764064e16	2.825483521349608
18	9.539424609817828e12	4.2525024879934694e17	2.454021446312976
19	9.539424609817828e12	4.2525024879934694e17	2.004329444309949
20	1.114453504512e13	1.3743733197249713e18	0.8469102151947894

4.3 Wnioski

Obliczone pierwiastki są jedynie przybliżoną wartością, ponieważ *Float*64 ma ograniczoną precyzję do 17 cyfr znaczących. Powoduje to zaokrąglenie niektórych współczynników, co bezpośrednio wpływa na wyznaczane wartości.
Eksperyment odnosi się do złośliwej natury wielomianu Wilkinsona. Okazuje się, że po zaburzeniu a_{19} o tak niewielką wartość $P(x)$ ma zera zespolone. Powstały pierwiastki podwójne, które rozbiły się na pierwiastki zespolone. Wprowadzona zmiana była niewielka, wręcz minimalna dla tej precyzji, jednak przesunięcia pierwiastków są znaczące.

Wielomian Wilkinsona to przykład zadania źle uwarunkowanego ze względu na zaburzenia współczynników.

5 Zadanie 5

5.1 Cel zadania

Celem jest przeprowadzenie eksperymentów na równaniu rekurencyjnym

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

gdzie r jest pewną stałą.
Pierwszy eksperyment polega na wykonaniu 40 iteracji powyższej rekurencji dla $p0 = 0.01$, $r = 3$, następnie dla tych samych danych wykonanie 40 iteracji z przerwą po 10 iteracjach na obcięcie wartości p_{10} do 3 miejsc po przecinku. Typem badanych jest $Float32$.
W drugim eksperymencie dla danych $p_0 = 0.01$, $r = 3$ należy porównać wyniki dane przez $Float32$ oraz $Float64$ po 40 iteracjach wyrażenia.

5.2 Rozwiązanie

Implementacja w języku Julia. Równanie jest obliczane prostą funkcją, która wyznacza iteracyjnie kolejne wyrażenia p_k dla $k = 0, 1, \cdots, 40$.
Dla drugiego eksperymentu, po 10 iteracjach wywoływana jest funkcja $floor(p, 3)$ w celu obcięcia wartości p_{10} do 3. miejsc po przecinku. Następnie program dalej iteruje po p aż do $k = 40$.

5.2.1 Wyniki

Sposób	Typ	p_{40}
I	Float32	0.25860548
II	Float32	1.093568
I	Float64	0.011611238029748606

5.3 Wnioski

Przedstawione wyrażenie rekurencyjne jest numerycznie niestabilne. Przez ograniczoną precyzję typów maszynowych dochodzi do zaokrągleń kolejnych wartości, a błędy nawarstwiają się, kiedy do dalszych obliczeń zostają przekazane niedokładne wartości. Dlatego można zaobserwować istotne różnice pomiędzy każdym z wyników. Jednorazowe obcięcie p_{10} spowodowało, że dalsze obliczenia mogły mieć co najwyżej 3 cyfry znaczące.

6 Zadanie 6

6.1 Cel zadania

Celem jest przeprowadzenie obliczeń równania rekurencyjnego

$$x_{n+1} = x_n^2 + c, n = 0, 1, \cdots 40$$

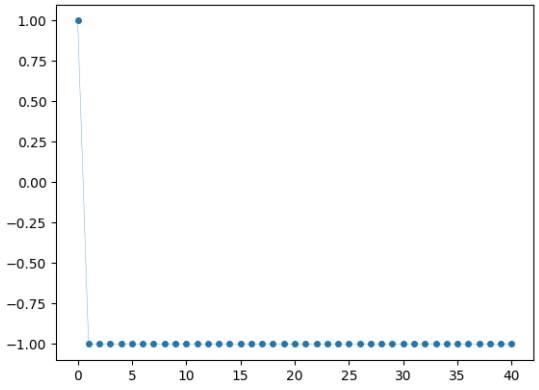
Na podstawie obliczeń należy przeprowadzić obserwację zachowania się powstałych ciągów.

6.2 Rozwiązanie

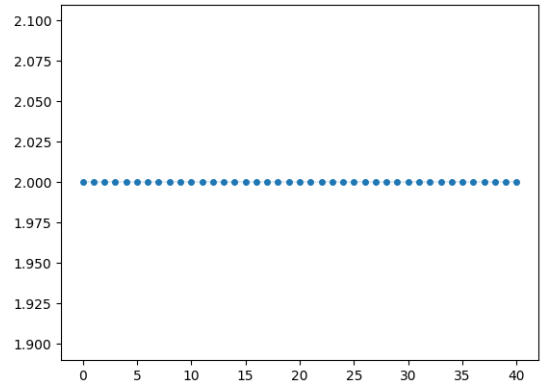
Implementacja w języku Julia z wykorzystaniem pakietu $PyPlot$. Równanie jest obliczane w sposób iteracyjny dla kolejnych wartości, wszystkie wyliczone x_k dla $k = 0, \cdots, 40$ zapisywane są do tablicy, aby pod koniec działania pętli narysować wykres wartości dla każdego wykonanego kroku iteracji.

6.2.1 Wyniki

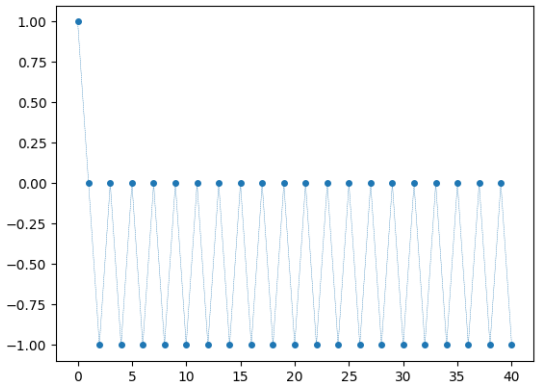
$c = -2\ x_0 = 1$



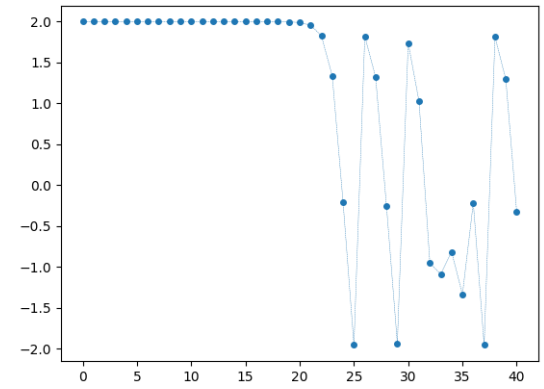
$c = -2\ x_0 = 2$



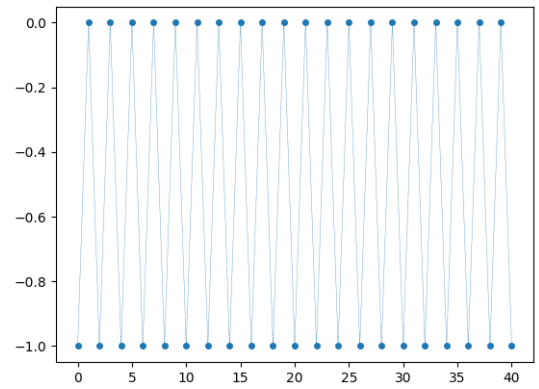
$c = -1\ x_0 = 1$



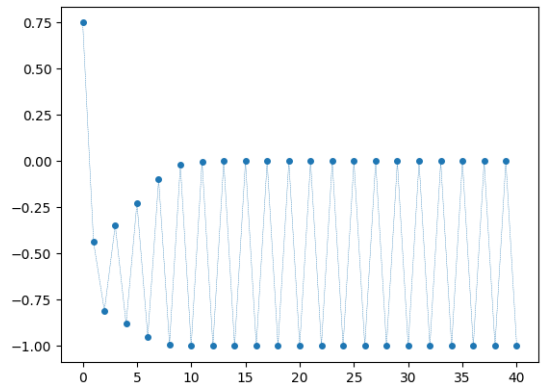
$c = -2\ x_0 = 1.99999999999999$



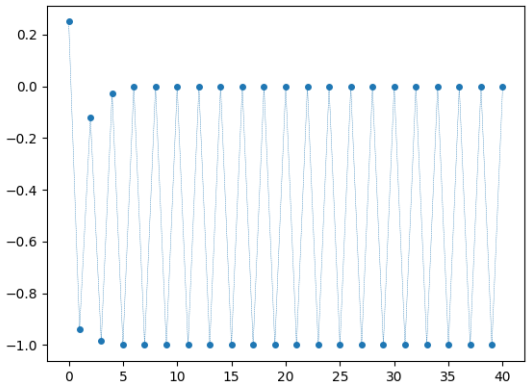
$c = -1\ x_0 = -1$



$c = -1\ x_0 = 0.75$



$c = -1\ x_0 = 0.25$



6.3 Wnioski

Na wykresach możemy zobaczyć, że dla $c = -2$ ciąg x_n ma punkty stałe $x = 2$, do którego zbiega również np. dla wartości 0, oraz $x = -1$. Dla wartości $|x| > 2$ ciąg rozbiega się do ∞, dla $x \in (1, 2)$ zbieżność zależy od tego, do jakiej wartości zostanie zaokrąglona liczba przez utratę precyzji.
Dla $c = -1$ ciąg nie ma punktów stałych; rozbiega się do ∞ lub krąży po przedziale $[0, -1]$.