

Lista 2

Obliczenia Naukowe

Zadanie 1.

Opis zadania

Zadaniem było zbadanie, jak wpływ na wynik iloczynu skalarnego dwóch wektorów (obliczanego przy użyciu algorytmów z zadania 5 z listy 1) mają pozornie niewielkie zmiany danych. Wektory w pierwotnej wersji zadania miały postać:

$$x = [2.718281828, -3.141592654, 1.414213562, 0.5772156649, 0.3010299957]$$

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

Natomiast po naniesieniu opisanych zmian wektor x przyjął postać:

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

Wyniki

Float32

Algorytm	$x * y$	$x' * y$
forward	-0.4999443	-0.4999443
backward	-0.4543457	-0.4543457
descending_order	-0.5	-0.5
ascending_order	-0.5	-0.5

Float64

Algorytm	$x * y$	$x' * y$
forward	1.0251881368296672e-10	-0.004296342739891585
backward	-1.5643308870494366e-10	-0.004296342998713953
descending_order	0.0	-0.004296342842280865
ascending_order	0.0	-0.004296342842280865

Wnioski

Wygląda na to, że choć błędy bezwzględne w obu próbach są podobnego rzędu wielkości, precyzyjna wartość iloczynu uległa znaczącej zmianie. W rezultacie błędy względne znacznie się zmniejszyły. Nawet drobne modyfikacje wejścia na poziomie 10^{-10} spowodowały zmianę rzędu wielkości wyniku od 10^{-11} do 10^{-3} . Możemy więc wnioskować, że zastosowane przez nas algorytmy nie są nieefektywne, ale raczej to zadanie jest źle uwarunkowane – niewielkie zaburzenia wejścia mają znaczący wpływ na otrzymane wyniki.

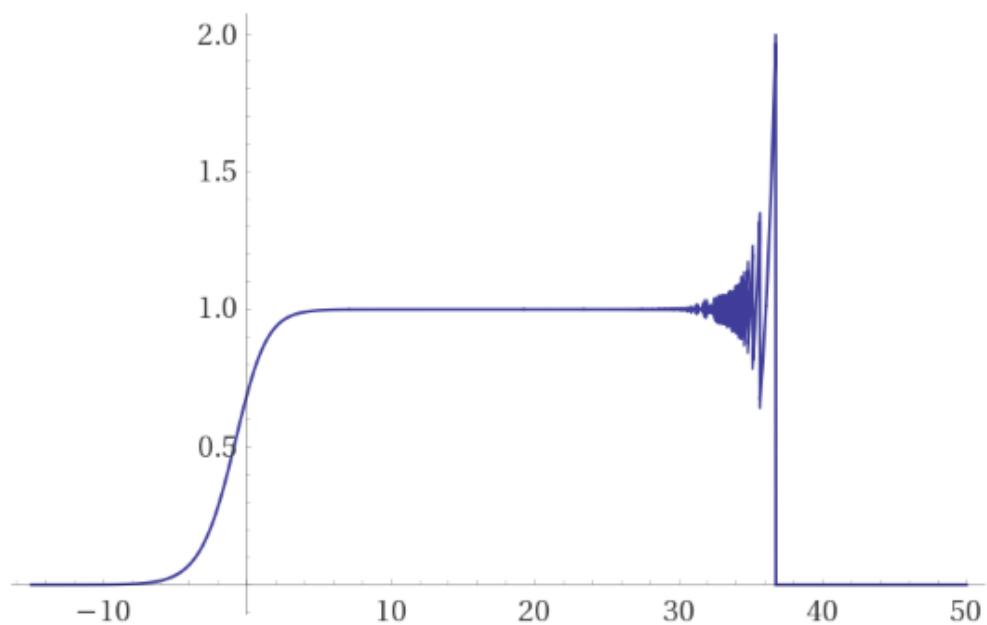
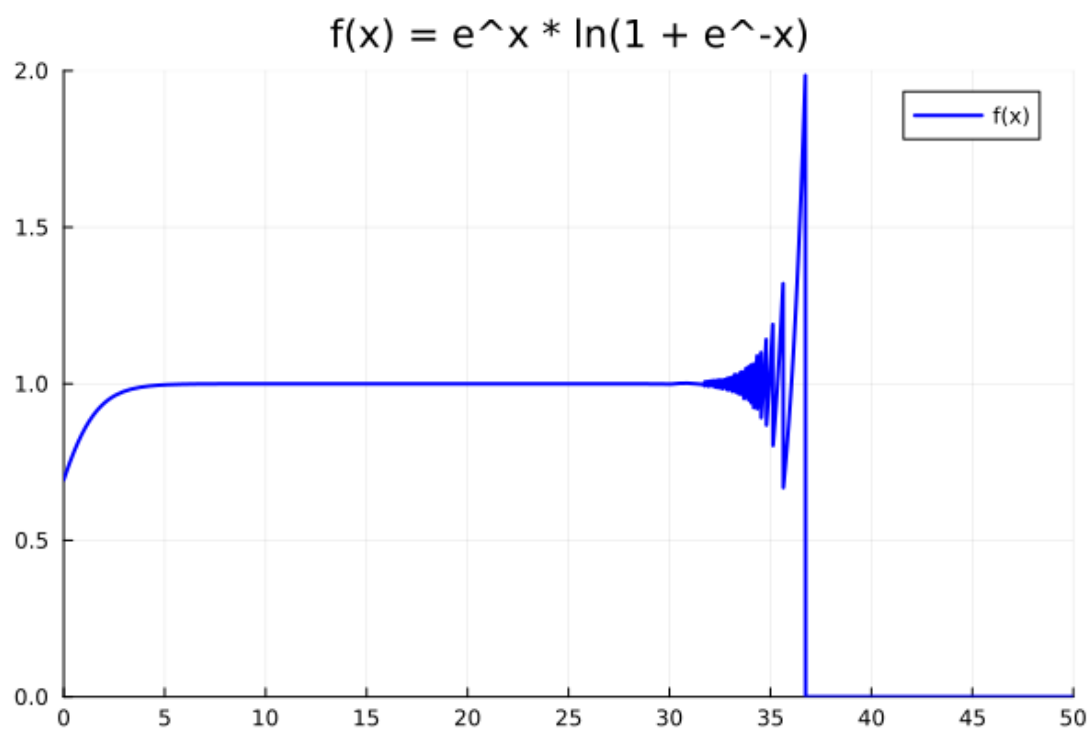
Zadanie 2.

Opis zadania

Celem zadania było narysowanie wykresu funkcji $f(x) = e^x \ln(1 + e^{-x})$ w dwóch różnych programach, a następnie porównanie wykresów z wyliczoną granicą w nieskończoności.

Wyniki

$$\lim_{x \rightarrow \infty} e^x \ln(1 + e^{-x}) = \lim_{x \rightarrow \infty} \frac{\ln(1 + e^{-x})}{e^{-x}} = \lim_{x \rightarrow \infty} \frac{-e^{-x}}{(1 + e^{-x})(-e^{-x})} = \lim_{x \rightarrow \infty} \frac{1}{1 + e^{-x}} = 1$$



Computed by Wolfram|Alpha

Wnioski

Zadanie jest uznawane za źle uwarunkowane. Nawet niewielkie błędy spowodowane ograniczoną precyzją arytmetyki prowadzą do znacznych fluktuacji wyników, co jest zauważalne na wykresie dla wartości x między 30 a 40. Dla wartości x około 36 funkcja na obu wykresach gwałtownie spada do zera. To zjawisko wynika z faktu, że dla dużych x zachodzi przybliżenie $1 + x \approx 1$, co skutkuje $\ln(1 + x) \approx 0$. Jest to niezgodne z rzeczywistością (i, co za tym idzie, z obliczoną przez nas granicą).

Problem ten występuje w dwóch powszechnie stosowanych i wiarygodnych programach, co potwierdza tezę o złym uwarunkowaniu tego zadania.

Zadanie 3.

Opis Zadania

Zadanie polegało na porównaniu dwóch metod rozwiązywania układów równań liniowych $Ax = b$ pod względem błędów w obliczeniach. Badanie obejmowało macierze A o określonych własnościach:

1. Macierze Hilberta

Stopnie n przyjmujące wartości z sekwencji $\{1, 4, 7, 10, \dots, 52\}$.

2. Macierze losowe

Stopnie $n \in \{5, 10, 20\}$ i współczynnikiem uwarunkowania $c \in \{1, 10, 10^3, 10^7, 10^{12}, 10^{16}\}$.

Do rozwiązania układów równań używano dwóch metod:

1. Metoda eliminacji Gaussa

$$x = Ab$$

2. Metoda z macierzą odwrotną

$$x = \text{inv}(A) \cdot b$$

Na początku generowano wektor b poprzez podstawienie wartości $x = (1, \dots, 1)^T$ do wzoru $Ax = b$. Dzięki temu uzyskiwano wzorcowe x , co umożliwiało wyznaczenie błędów dla obu metod.

Wyniki

Macierz Hilberta:

Wielkość	Rząd	Współczynnik uwarunkowania	Błąd Metody Gaussa	Błąd Macierzy odwrotnej
1	1	1.0	0.0	0.0
2	2	19.28147006790397	0.3071475584169753	0.3071475584169748

Wielkość	Rząd	Współczynnik uwarunkowania	Błąd Metody Gaussa	Błąd Macierzy odwrotnej
3	3	524.0567775860644	0.38125383099933113	0.381253830999332
4	4	15513.73873892924	0.41953891475305755	0.4195389147530581
5	5	476607.2502422687	0.4433763869895056	0.4433763869897754
6	6	1.49510586424659e7	0.4598223580900048	0.45982235810041816
7	7	4.753673565921816e8	0.47193854478614516	0.47193854464773893
8	8	1.5257575538072489e10	0.48128264147259203	0.4812826504848323
9	9	4.931537556012197e11	0.4887361838475567	0.48873590121629773
10	10	1.602441350036382e13	0.49483788867877526	0.4948399362295172
11	10	5.222703245009594e14	0.49993670962971726	0.500096134186326
12	11	1.760619121841585e16	0.5120560847629834	0.5396757497648286
13	11	3.1905581877988255e18	0.5132192380957149	3.5717327555088345
14	11	9.27636978936766e17	1.086457534475591	5.723671796732953
15	12	3.67568286586649e17	3.1258537952116305	4.847438869932798
16	12	7.063115212292111e17	35.450491051218	19.566695883780312
17	12	8.07124989431416e17	8.963215243216949	6.883091679282024
18	12	1.4135073701749765e18	6.700575476688344	16.130244647018774
19	13	5.190132496359103e18	66.39030366996684	71.44698595999716
20	13	1.3193976166344822e18	70.25375920097635	74.15157916514092

Losowa macierz:

Wielkość	Wskaźnik Uwarunkowania	Rząd	Współczynnik uwarunkowania	Błąd Metody Gaussa	Błąd oc
5	1.0	5	1.0000000000000004	0.32268634029159465	0.32268
5	10.0	5	9.999999999999998	0.9632109792366139	0.96321
5	1000.0	5	999.9999999999918	0.9984798081909074	0.99847
5	1.0e7	5	9.99999999828954e6	1.0000000093139296	1.00000
5	1.0e12	5	1.0000031822661598e12	1.0000000000012037	1.00000
5	1.0e16	4	9.25080669240908e15	1.0	1.0
10	1.0	10	1.0000000000000016	0.16305631927440734	0.16305
10	10.0	10	10.000000000000004	0.8619208856678274	0.86192
10	1000.0	10	999.999999999167	0.9996654543680095	0.99966
10	1.0e7	10	1.0000000001399769e7	0.999999844499018	0.99999
10	1.0e12	10	1.0000404116851804e12	0.9999999999988204	0.99999

Wielkość	Wskaźnik Uwarunkowania	Rząd	Współczynnik uwarunkowania	Błąd Metody Gaussa	Błąd oc
10	1.0e16	9	6.977083262832289e15	1.0	1.0
20	1.0	20	1.0000000000000013	0.20573054880527553	0.20573
20	10.0	20	9.999999999999993	0.6163539312368534	0.61635
20	1000.0	20	999.9999999999989	0.9996975287126516	0.99969
20	1.0e7	20	9.99999997355066e6	1.0000000252645709	1.00000
20	1.0e12	20	1.0000330403806237e12	0.9999999999999826	0.99999
20	1.0e16	19	8.849855710748998e15	1.0	1.0

Wnioski

Wnioski z przeprowadzonych testów sugerują, że macierze Hilberta wykazują bardzo duże wskaźniki uwarunkowania, nawet dla relatywnie niewielkich rozmiarów. Dane zdają się wskazywać, że metoda eliminacji Gaussa zazwyczaj prowadzi do uzyskania bardziej precyzyjnych wyników dla tego rodzaju macierzy. Jednak obie metody skutkują ogromnymi błędami względnymi.

W przypadku losowych macierzy o ustalonym wskaźniku uwarunkowania trudno zauważyć istotne różnice między oboma algorytmami. Błędy są prawie zawsze tego samego, niewielkiego rzędu. Co więcej, rzędy błędów są podobne dla macierzy o różnych rozmiarach, o ile ich współczynniki uwarunkowania są zbliżone.

Ten stan rzeczy sugeruje, że algorytmy działają efektywnie. Rozwiązanie układu równań liniowych, w którym współczynniki są określone przez macierz Hilberta, jest zadaniem źle uwarunkowanym. Znając współczynnik uwarunkowania macierzy A , możemy oszacować błędy względne uzyskanych rozwiązań.

Zadanie 4.

Opis zadania

W zadaniu rozpatrywany jest złośliwy wielomian Wilkinsona:

$$f(x) = (x - 1)(x - 2)(x - 3) \dots (x - 20)$$

(a) Wyznaczanie pierwiastków za pomocą Pakietu Polynomials:

W tym podpunkcie skoncentrujemy się na zbadaniu skuteczności pakietu Polynomials w wyznaczaniu pierwiastków wielomianów. Wykorzystamy funkcje dostępne w pakiecie do identyfikacji pierwiastków różnych wielomianów i ocenimy, jak dobrze radzi sobie z tym zadaniem.

(b) Eksperyment Wilkinsona:

Eksperyment Wilkinsona polega na marginalnym zaburzeniu jednego ze współczynników wielomianu. W naszym przypadku, powtórzymy ten eksperyment, gdzie wprowadzimy niewielką zmianę w jednym z współczynników wielomianu, a następnie sprawdzimy, jak pakiet Polynomials reaguje na to zaburzenie. Analizujemy, czy niewielka zmiana w współczynniku prowadzi do istotnych różnic w pierwiastkach wyznaczonych przez pakiet.

Wyniki

Wyniki w tabelach są wartościami bezwzględnymi dla większej przejrzystości danych.

Wyznaczanie pierwiastków za pomocą Pakietu Polynomials:

k	z_k	$P(z_k)$	$p(z_k)$	$z_k - k$
1	0.999999999998084	23323.616390897252	24347.616390897056	1.916244940503
2	2.0000000000114264	64613.550791712885	80997.5507918065	1.142641536944
3	2.9999999998168487	18851.098984644806	101795.09897958105	1.831512719263
4	3.999999983818672	2.6359390809003003e6	2.37379508196076e6	1.618132783320
5	5.000000688670983	2.3709842874839526e7	2.306984278668964e7	6.886709833509
6	5.999988371602095	1.2641076289358065e8	1.2508366146559621e8	1.162839790502
7	7.000112910766231	5.2301629899144447e8	5.2055763533357024e8	0.000112910766
8	7.999279406281878	1.798432141726085e9	1.7942387274786062e9	0.000720593718
9	9.003273831140774	5.121881552672067e9	5.115158339932115e9	0.003273831140
10	9.989265687778465	1.4157542666785017e10	1.4147337313301882e10	0.010734312221
11	11.027997558569794	3.586354765112257e10	3.584844758752149e10	0.027997558569
12	11.94827395840048	8.510931555828575e10	8.508835580052173e10	0.051726041599
13	13.082031971969954	2.2136146301419052e11	2.2133136429365445e11	0.082031971969
14	13.906800565193148	3.812024574451268e11	3.811638891032201e11	0.093199434806
15	15.081439299377482	8.809029239560208e11	8.808498064028993e11	0.081439299377
16	15.942404318674466	1.6747434633806333e12	1.6746775852847332e12	0.057595681325
17	17.026861831476396	3.3067827086376123e12	3.3066967643946914e12	0.026861831476
18	17.99048462339055	6.166202940769282e12	6.16609562105193e12	0.009515376609
19	19.001981084996206	1.406783619602919e13	1.4067702719729383e13	0.001981084996
20	19.999803908064397	3.284992217648231e13	3.2849758339688676e13	0.000196091935

Eksperyment Wilkinsona:

k	z_k	$P(z_k)$	$p(z_k)$	$z_k - k$
1	0.999999999999805 +	2168.9361669986724	2168.9361669986324	1.95399252334

k	z_k	$P(z_k)$	$p(z_k)$	$z_k -$
	0.0im			
2	1.999999999985736 + 0.0im	29948.438957395843	13564.438957442584	1.42641454203
3	3.000000000105087 + 0.0im	239010.53520956426	48546.53517157824	1.05087050172
4	3.9999999950066143 + 0.0im	939293.8049425513	1.201437802324579e6	4.99338570492
5	5.000000034712704 + 0.0im	7.44868039679552e6	4.333159313828003e6	3.47127038224
6	6.000005852511414 + 0.0im	1.4689332508961653e7	5.5871717686950475e7	5.85251141416
7	6.999704466216799 + 0.0im	5.817946400915084e7	1.2508396452132258e9	0.00029553378
8	8.007226654064777 + 0.0im	1.3954205929609105e8	1.742241327643653e10	0.00722665406
9	8.917396943382494 + 0.0im	2.459617755654851e8	1.3471875977601944e11	0.08260305661
10	10.09529034477879 - 0.6432770896263527im	2.291018560461982e9	1.4798961582209915e12	0.65029659682
11	10.09529034477879 + 0.6432770896263527im	2.291018560461982e9	1.4798961582209915e12	1.11009232692
12	11.793588728372308 - 1.6522535463872843im	2.077690789102519e10	3.2933037979480902e13	1.66509681238
13	11.793588728372308 + 1.6522535463872843im	2.077690789102519e10	3.2933037979480902e13	2.04581766974
14	13.99233053734825 - 2.5188196443048287im	9.390730597798799e10	9.54583157311972e14	2.51883132051
15	13.99233053734825 + 2.5188196443048287im	9.390730597798799e10	9.54583157311972e14	2.71290437474
16	16.73073008036981 - 2.8126272986972136im	9.592356563898315e11	2.7421143234744428e16	2.90600047685
17	16.73073008036981 + 2.8126272986972136im	9.592356563898315e11	2.7421143234744428e16	2.82548732274
18	19.50243895868367 - 1.9403320231930836im	5.050467401799687e12	4.2524516249528954e17	2.45401939372
19	19.50243895868367 + 1.9403320231930836im	5.050467401799687e12	4.2524516249528954e17	2.00432863255
20	20.84690887410499 + 0.0im	4.858653129933677e12	1.3743527553999101e18	0.84690887410

Wnioski

(a) Wyznaczanie pierwiastków za pomocą Pakietu Polynomials:

Jak można zauważyć w przedstawionych wynikach, obliczone pierwiastki nie są całkowicie dokładne, ale są bardzo bliskie rzeczywistym wartościom. Co ciekawe rzeczywiste pierwiastki nie zerują wielomianu $P(x)$, ponieważ z powodu ograniczeń arytmetyki Float64, wielomian w postaci naturalnej jest przechowywany niedokładnie.

Niektóre współczynniki wielomianu dla niższych potęg x są ogromnymi liczbami, co powoduje trudność w ich precyzyjnym zapisie bez utraty znaczącej liczby cyfr. Informacje zawarte w tabeli dotyczące oddalenia od rzeczywistych pierwiastków ukazują istotę zjawiska znanej jako "złośliwość wielomianu". Nawet niewielkie odchylenie od rzeczywistych pierwiastków prowadzi do znacznych wzrostów wartości funkcji (nawet rzędu 10^{23}). Wskazuje to na to, że problem wyznaczania pierwiastków wielomianu Wilkinsona jest źle uwarunkowany, co sprawia, że mała niepewność w danych może znacząco wpłynąć na wyniki obliczeń.

(b) Eksperyment Wilkinsona:

Dane zawarte w drugiej tabeli potwierdzają wcześniejsze przypuszczenia z podpunktu (a). Złe uwarunkowanie problemu, związanego z wyznaczaniem pierwiastków wielomianu Wilkinsona, prowadzi do sytuacji, w której niewielkie zakłócenie może skutkować pojawieniem się rozwiązań zespolonych. Co więcej, wartości funkcji dla tych "pierwiastków" są bardzo duże, co świadczy o znacznym wzroście wartości funkcji nawet przy niewielkich perturbacjach danych. Ten efekt wynika z niestabilności numerycznej spowodowanej złym uwarunkowaniem problemu, co sprawia, że nawet niewielkie zaburzenia danych wejściowych mogą wprowadzić duże błędy w wynikach obliczeń.

Zadanie 5.

Opis Zadania

Zadanie polegało na obliczeniu czterdziestu pierwszych wyrazów ciągu p_n zdefiniowanego równaniem rekurencyjnym:

$$p_{n+1} = p_n + r \cdot p_n \cdot (1 - p_n)$$

dla początkowego warunku $p_0 = 0.01$ i $r = 3$, przy użyciu trzech różnych reprezentacji arytmetycznych:

1. W precyzji Float32
2. W precyzji Float32, przy czym p_{10} był obcinany do trzech miejsc po przecinku, a następnie kolejny wyraz był obliczany na podstawie tej obciętej wartości
3. W precyzji Float64

Wyniki

n	Float32	Float32 z obcięciem	Float64
0	0.01	0.01	0.01
1	0.0397	0.0397	0.0397
2	0.15407173	0.15407173	0.154071730000000002
3	0.5450726	0.5450726	0.5450726260444213
4	1.2889781	1.2889781	1.2889780011888006
5	0.1715188	0.1715188	0.17151914210917552
6	0.5978191	0.5978191	0.5978201201070994
7	1.3191134	1.3191134	1.3191137924137974
8	0.056273222	0.056273222	0.056271577646256565
9	0.21559286	0.21559286	0.21558683923263022
10	0.7229306	0.722	0.722914301179573
11	1.3238364	1.3241479	1.3238419441684408
12	0.037716985	0.036488414	0.03769529725473175
13	0.14660022	0.14195944	0.14651838271355924
14	0.521926	0.50738037	0.521670621435246
15	1.2704837	1.2572169	1.2702617739350768
16	0.2395482	0.28708452	0.24035217277824272
17	0.7860428	0.9010855	0.7881011902353041
18	1.2905813	1.1684768	1.2890943027903075
19	0.16552472	0.577893	0.17108484670194324
20	0.5799036	1.3096911	0.5965293124946907
21	1.3107498	0.09289217	1.3185755879825978
22	0.088804245	0.34568182	0.058377608259430724
23	0.3315584	1.0242395	0.22328659759944824
24	0.9964407	0.94975823	0.7435756763951792
25	1.0070806	1.0929108	1.315588346001072
26	0.9856885	0.7882812	0.07003529560277899
27	1.0280086	1.2889631	0.26542635452061003
28	0.9416294	0.17157483	0.8503519690601384
29	1.1065198	0.59798557	1.2321124623871897
30	0.7529209	1.3191822	0.37414648963928676
31	1.3110139	0.05600393	1.0766291714289444
32	0.0877831	0.21460639	0.8291255674004515

n	Float32	Float32 z obcięciem	Float64
33	0.3280148	0.7202578	1.2541546500504441
34	0.9892781	1.3247173	0.29790694147232066
35	1.021099	0.034241438	0.9253821285571046
36	0.95646656	0.13344833	1.1325322626697856
37	1.0813814	0.48036796	0.6822410727153098
38	0.81736827	1.2292118	1.3326056469620293
39	1.2652004	0.3839622	0.0029091569028512065
40	0.25860548	1.093568	0.011611238029748606

Wnioski

Wyniki analizy wskazują, że w początkowych iteracjach obie arytmetyki generują zbliżone rezultaty. Jednak w miarę postępu iteracji, szczególnie w okolicach 19. kroku, model z obcięciem zaczyna wyraźnie odbiegać od pozostałych dwóch. Około 22. iteracji także modele bez przycięcia zaczynają się rozbiegać.

Zjawisko to jest rezultatem szybko narastającej kumulacji błędów podczas obliczania kolejnych wyrazów ciągu. Kumulacja ta wynika między innymi z operacji podnoszenia poprzednich wartości do kwadratu. Pierwsze wyrazy są obliczane z wysoką precyzją, ale późniejsze są zapamiętywane z uwzględnieniem zaokrąglenia wynikającego z ograniczonej precyzji. Dokładność kolejnych wyrazów staje się trudna do oszacowania ze względu na narastający błąd.

Proces wyznaczania kolejnych wyrazów ciągu p_n jest niestabilny, ponieważ drobne błędy początkowe mają tendencję do drastycznego wpływania na precyzję dalszych wyników. Obserwacja, że Float32 bez przycięcia pozostaje zgodny z Float64 dłużej niż jego wersja z obcięciem, sugeruje, że zarówno czas, w którym błąd zostaje popełniony, jak i jego wielkość, wpływają na tempo drastycznej utraty dokładności. Warto zauważyć, że nawet arytmetyka Float64 w pewnym momencie zaczyna manifestować błędy związane z precyzją, co oznacza, że wyniki dla większych n są bardziej godne zaufania, ale niestety nie można całkowicie wykluczyć wpływu błędów związanych z precyzją na ostateczne wyniki.

Zadanie 6.

Opis Zadania

Rozważane są ciągi rekurencyjne, których kolejne wyrazy są określone za pomocą wzoru rekurencyjnego:

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

Zbadajmy zachowanie tych ciągów dla siedmiu przypadków w arytmetyce Float64:

1. $c = -2$ i $x_0 = 1$

2. $c = -2$ i $x_0 = 2$

3. $c = -2$ i $x_0 = 1.9999999999999999$

4. $c = -1$ i $x_0 = 1$

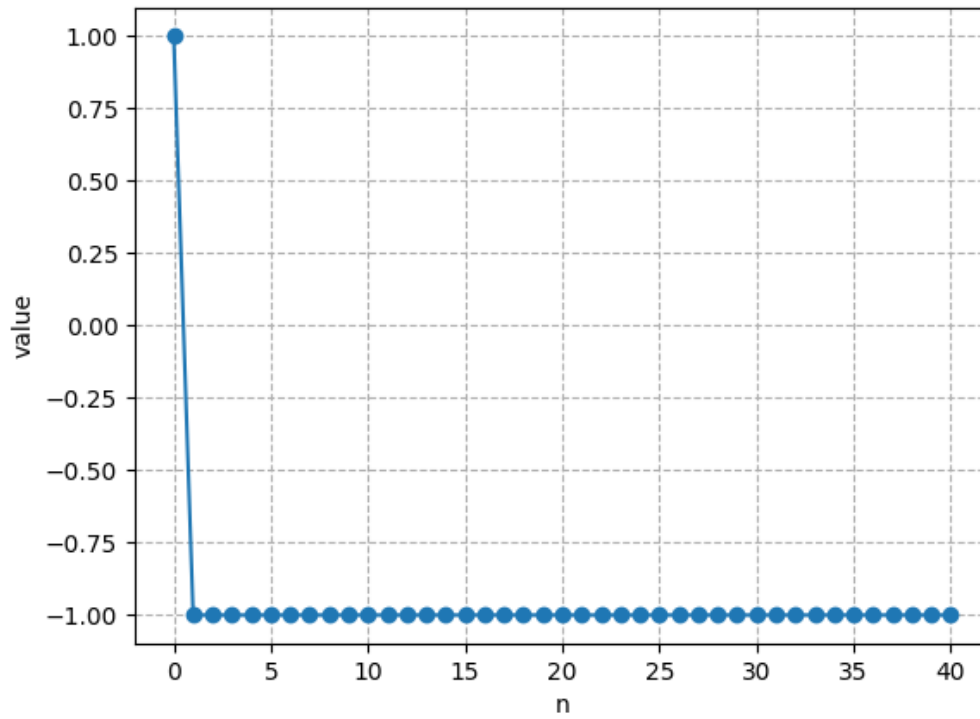
5. $c = -1$ i $x_0 = -1$

6. $c = -1$ i $x_0 = 0.75$

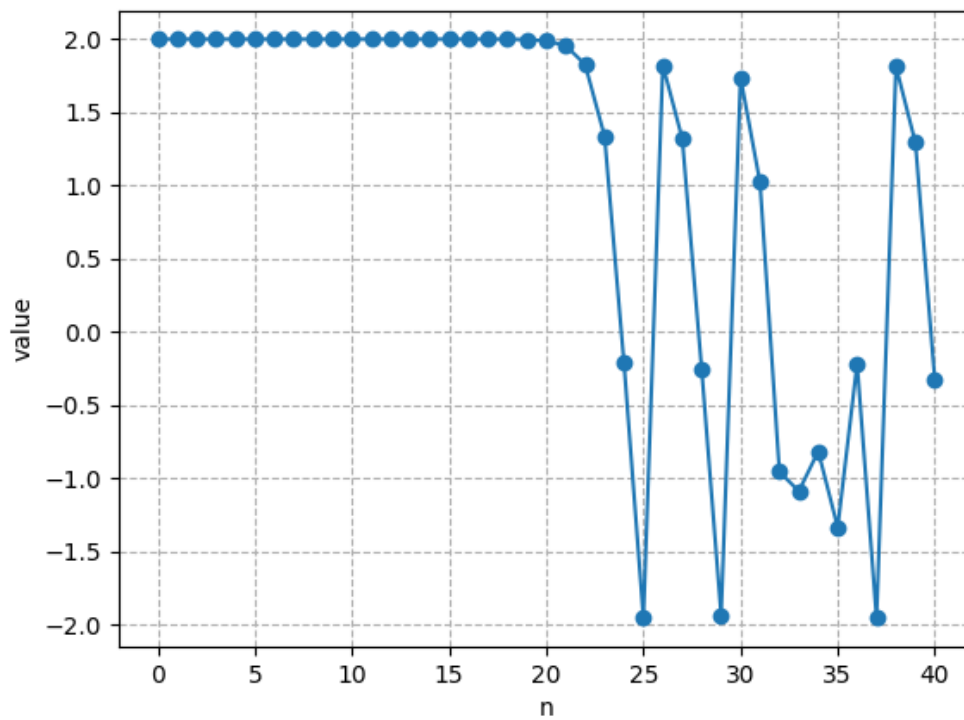
7. $c = -1$ i $x_0 = 0.25$

Wyniki

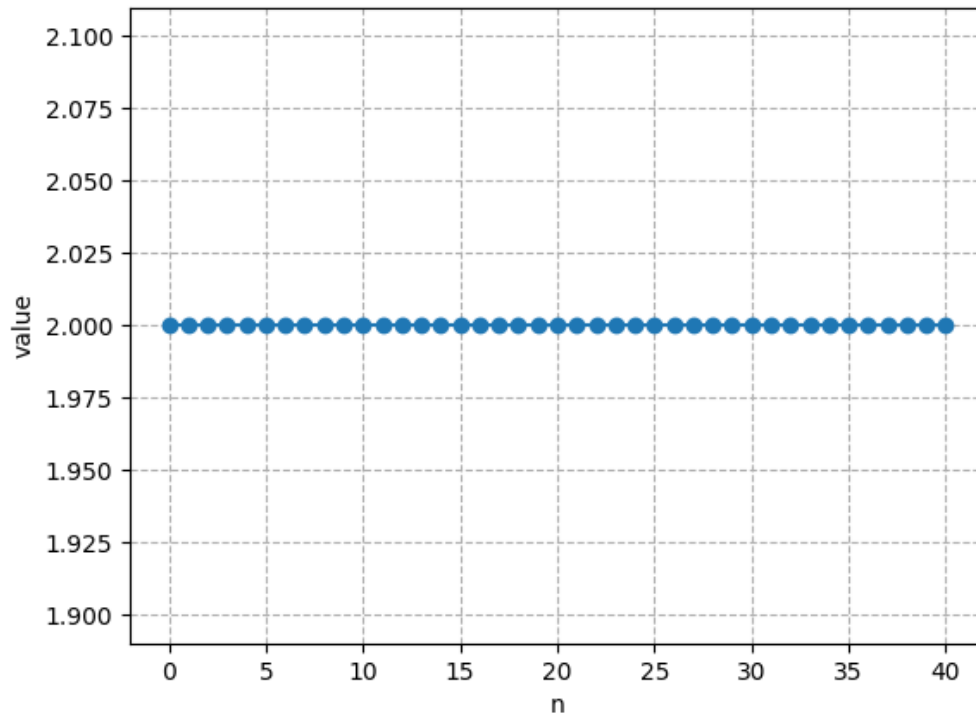
Results for ($c = -2.0$, $x = 1.0$):



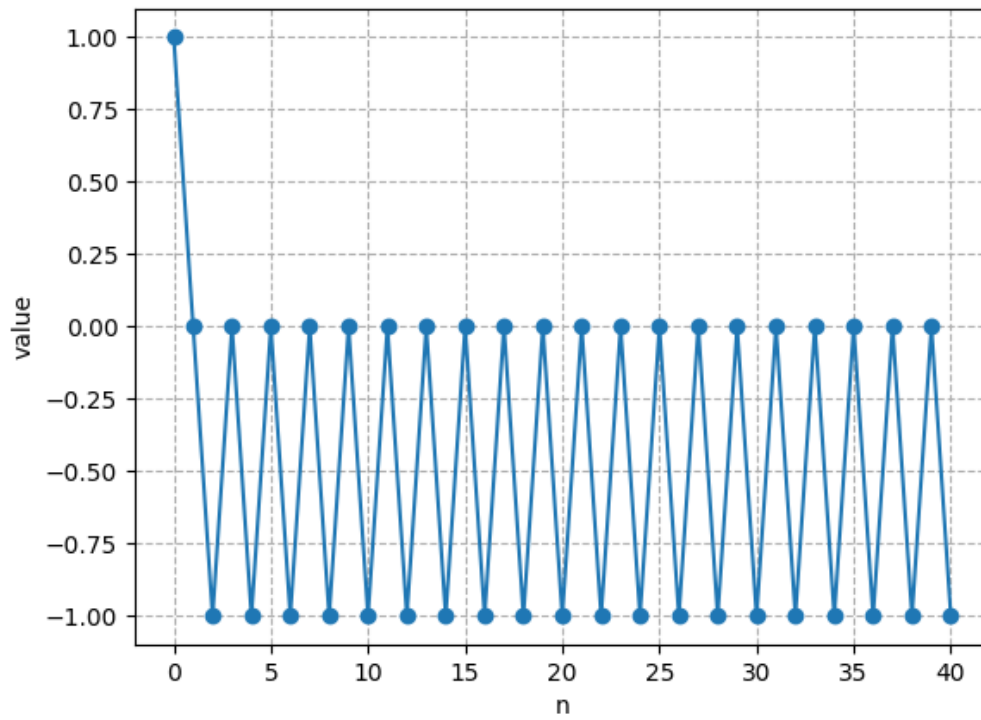
Results for ($c = -2.0$, $x = 1.999999999999999$):

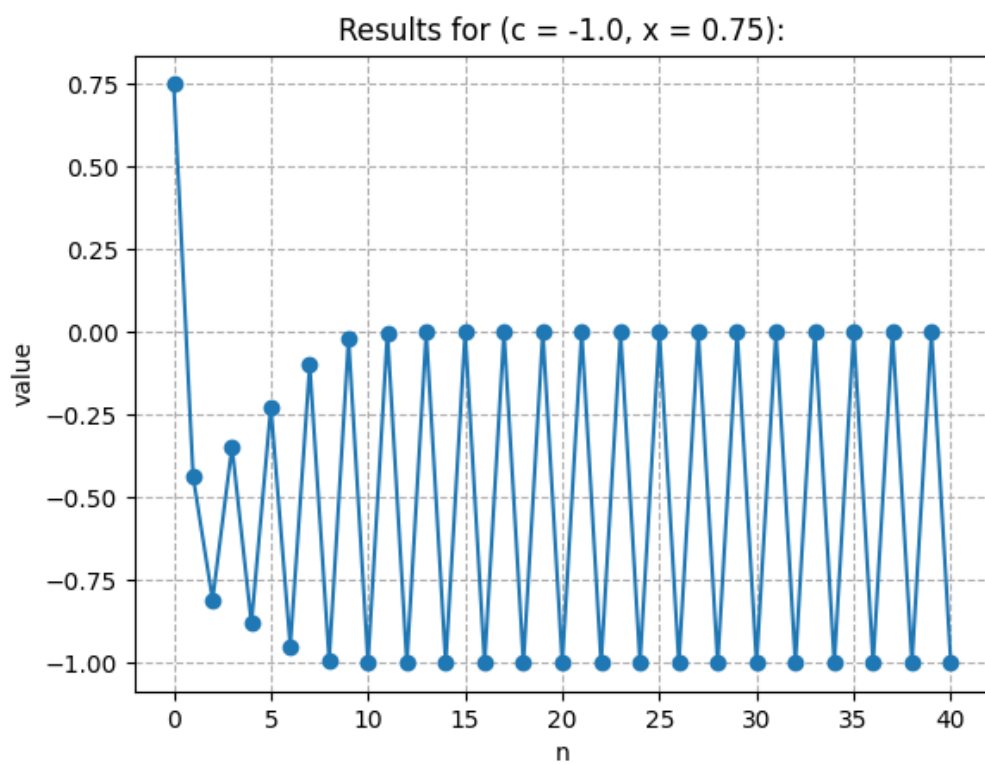
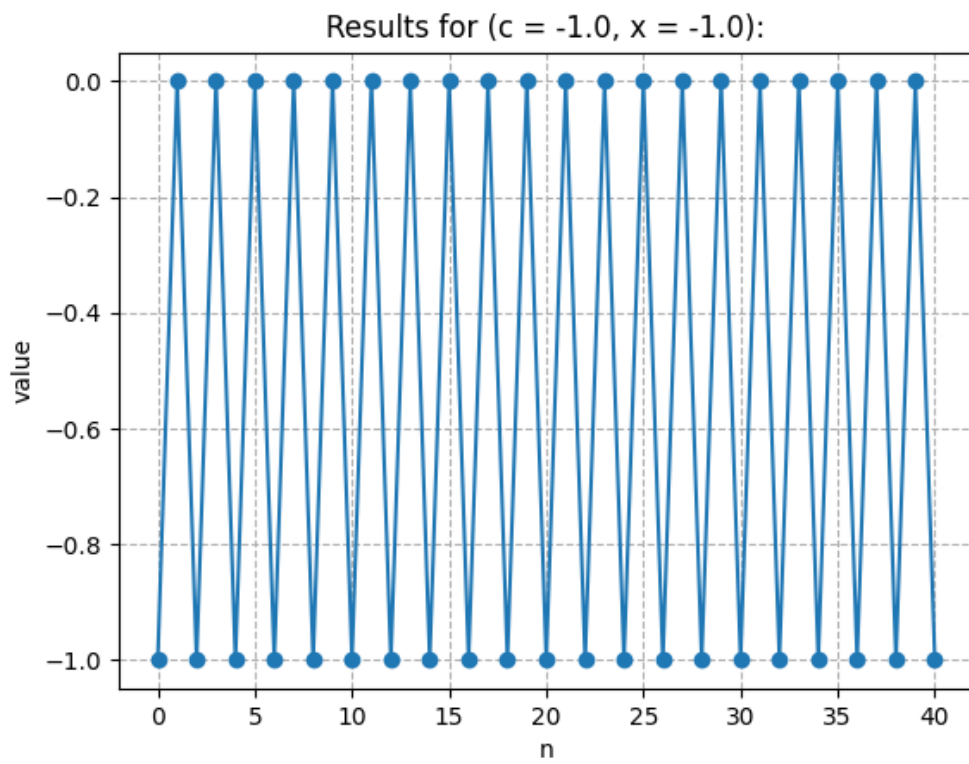


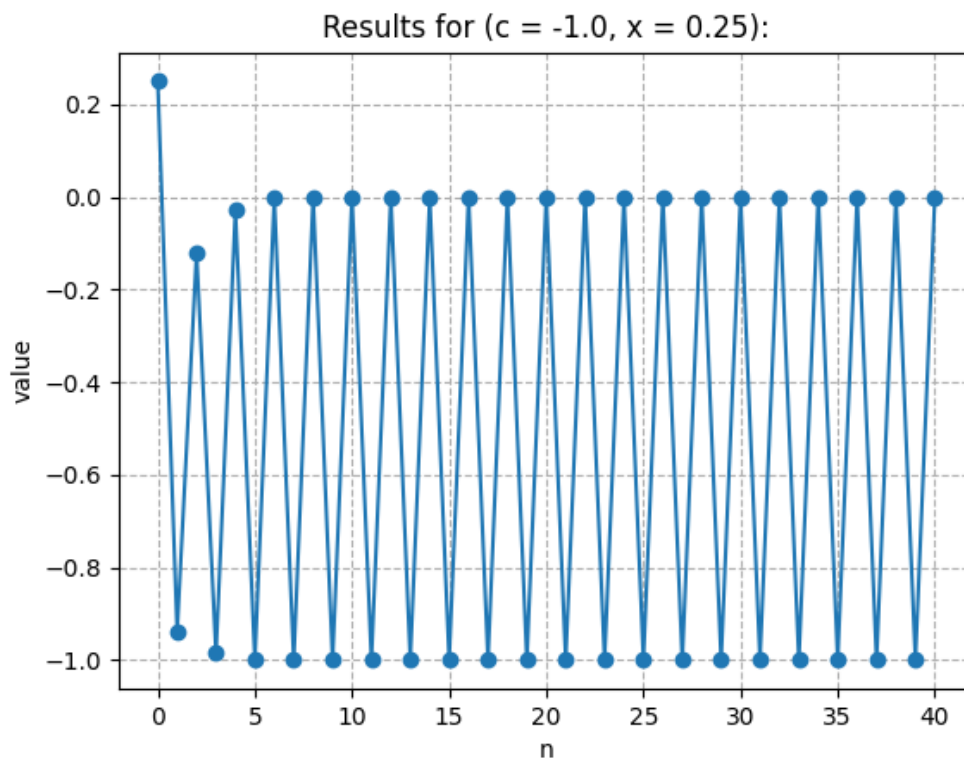
Results for ($c = -2.0$, $x = 2.0$):



Results for ($c = -1.0$, $x = 1.0$):







Wnioski

Proces generowania kolejnych elementów ciągów, opisanych za pomocą reguł rekurencyjnych, może wykazywać zróżnicowaną stabilność w zależności od przyjętych parametrów. Stabilność ta może przyjmować różne stopnie, począwszy od całkowitej stabilności (na przykład dla początkowej wartości $x_0 = 1$ przy obu wartościach c), poprzez stopniowe stabilizowanie się (np. dla $c = -1$ i x_0 równego 0.75 lub 0.25), aż do niestabilności (jak w przypadku $c = -2$ i $x_0 = 1.99\dots$, czy też w przypadkach opisanych w zadaniu 5).

Warto zauważyć, że proces stopniowej stabilizacji może również zależeć od konkretnych wartości parametrów, przykładowo w przypadku $c = -1$ stabilizacja może następować szybciej dla $x_0 = 0.25$ niż dla $x_0 = 0.75$. To oznacza, że istotny jest nie tylko sam fakt stabilizacji, lecz także tempo, z jakim to zachodzi, co wymaga dokładnego doboru odpowiednich wartości parametrów.

Podsumowując, dla uzyskania rzetelnych i praktycznych wyników w procesie generowania ciągów rekurencyjnych, istotne jest dokładne dostosowanie parametrów, biorąc pod uwagę ich wpływ na stabilność oraz tempo stabilizacji procesu.