

Sprawozdanie

Lista 2

Paweł Krzyszczak

Listopad 2024

1 Zadanie 1

1.1 Opis zadania

Celem zadania jest zbadanie wpływu drobnych modyfikacji danych wejściowych na wynik iloczynu skalarnego dwóch wektorów przy użyciu różnych metod sumowania. Zadanie wymaga modyfikacji elementów wektorów z zadania 5 z Listy 1 poprzez usunięcie ostatniej cyfry „9” z x_4 oraz ostatniej cyfry „7” z x_5 . Następnie, należy porównać wyniki uzyskane po zmianach z pierwotnymi wynikami, aby zaobserwować wpływ niewielkich zmian danych na obliczenia.

1.2 Rozwiązanie

Oryginalne wektory:

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

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

Zmodyfikowane wektory:

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

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

1.3 Wyniki

Poniższa tabela przedstawia wyniki dla zmodyfikowanych danych oraz porównanie z wartościami oryginalnymi. Wartość referencyjna wynosi $-1.00657107000000 \times 10^{-11}$.

Typ	"w przód"	"w tył"	od najw. do najmn.	od najmn. do najw.
Float32	-0.4999443	-0.4543457	-0.5	-0.5
Float64	1.0251881368296672e-10	-1.5643308870494366e-10	0.0	0.0

Tabela 1: Wyniki iloczynu skalarnego wektorów z zadania 5 z listy 1.

Typ	"w przód"	"w tył"	od najw. do najmn.	od najmn. do najw.
Float32	-0.4999443	-0.4543457	-0.5	-0.5
Float64	-0.00429634273	-0.00429634299	-0.00429634284	-0.00429634284

Tabela 2: Wyniki iloczynu skalarnego wektorów po niewielkiej zmianie.

Wyniki pokazują, że niewielkie zmiany w danych wejściowych mogą znacząco wpłynąć na wynik końcowy, zwłaszcza w obliczeniach z podwójną precyzją. Metody sumowania wykazują różną stabilność numeryczną, co wpływa na dokładność wyników.

1.4 Wnioski

- **Wpływ drobnych zmian danych:** Nawet niewielkie modyfikacje danych wejściowych mogą prowadzić do znaczących różnic w wynikach. Jednak przy niskiej precyzji błąd ten może być pomijalny ze względu na granice precyzji.
- **Wpływ precyzji obliczeń:** Obliczenia wykonane w podwójnej precyzji (Float64) są mniej stabilne za to bliższe wartości referencyjnej.

2 Zadanie 2

2.1 Opis zadanie

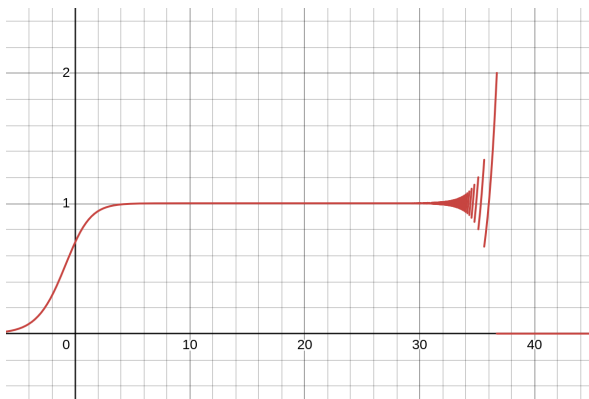
Zadanie polega na narysowaniu wykresu funkcji $f(x) = e^x \ln(1 + e^{-x})$ przy użyciu dwóch różnych narzędzi do wizualizacji. Następnie obliczona została granica funkcji dla $x \rightarrow \infty$ i porównano ją z wykresem funkcji w tym przedziale. Celem jest wyjaśnienie zachowania funkcji oraz przybliżenie jej wartości dla dużych x .

2.2 Rozwiązanie

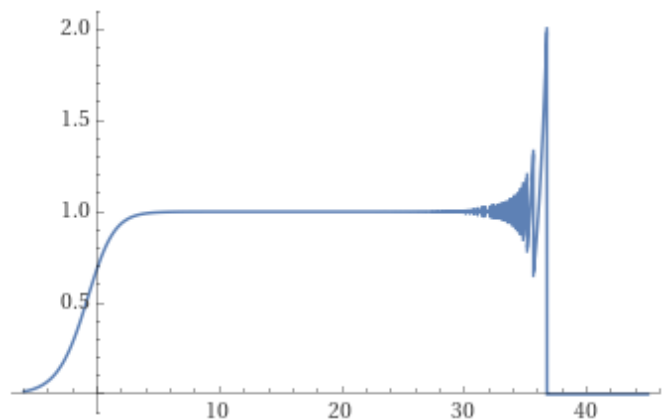
Do narysowania wykresu funkcji $f(x) = e^x \ln(1 + e^{-x})$ użyte zostały dwa programy do wizualizacji, tj. Desmos i Wolfram Alpha.

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

2.3 Wyniki



Rysunek 1:
Wykres funkcji otrzymany za pomocą Desmos.



Rysunek 2:
Wykres funkcji otrzymany za pomocą Wolfram Alpha.

Na wykresach (Rysunek 1 i 2) widać, że funkcja $f(x)$ początkowo zbliża się do wartości 1, co jest zgodne z wyznaczoną granicą. Jednak przy wartościach $x > 30$ granica zaczyna oscylować, aby w końcu przyjąć wartość 0.

2.4 Wnioski

Analizując wzór $f(x) = e^x \ln(1 + e^{-x})$ można zauważyć, iż wraz ze wzrostem argumentu funkcji czynnik e^x znacząco rośnie, natomiast czynnik $\ln(1 + e^{-x})$ wręcz przeciwnie. Efektem tego pojawiają się oscylacje wynikające z błędów idącego z mnożenia liczb o znacząco różniącym się rzędzie. Dodatkowo ze względu na arytmetykę stosowaną w użytych programach do wizualizacji od pewnego momentu $\ln(1 + e^{-x}) \approx 0$, dlatego funkcja się zeruje.

3 Zadanie 3

3.1 Opis zadania

Celem zadania jest analiza dokładności rozwiązywania układu równań liniowych $Ax = b$, gdzie A to macierz Hilberta lub macierz losowa o zadanym wskaźniku uwarunkowania c . Macierz A oraz wektor prawych stron b generowane są tak, aby dokładne rozwiązanie układu $Ax = b$ było wektorem $x = (1, \dots, 1)^T$. Następnie rozwiązanie numeryczne uzyskuje się za pomocą dwóch metod: eliminacji Gaussa i metody odwrotności macierzy. Dokładność wyników ocenia się przez porównanie obliczonego \tilde{x} z dokładnym x za pomocą względnego błędu.

3.2 Rozwiązanie

1. **Macierz Hilberta** Macierz Hilberta stopnia n , oznaczona jako H_n , jest generowana za pomocą funkcji $\text{hilb}(n)$. Jest ona przykładem macierzy słabo uwarunkowanej, co może prowadzić do dużych błędów przy rozwiązywaniu układu równań.
2. **Macierz losowa** Macierz losowa R_n jest generowana za pomocą funkcji $\text{matcond}(n, c)$, gdzie c to wskaźnik uwarunkowania. Dzięki tej funkcji można kontrolować kondycję macierzy, aby zbadać wpływ rosnącej wartości c na dokładność rozwiązań.
3. **Metody rozwiązywania** Do rozwiązania układu $Ax = b$ zastosowano dwie metody:
 - (a) **Eliminacja Gaussa**: rozwiązywanie przy użyciu operatora $A \setminus b$.
 - (b) **Metoda odwrotności macierzy**: obliczanie rozwiązania jako $x = \text{inv}(A) \cdot b$.

3.3 Wyniki

n	$\text{cond}(A)$	$\text{rank}(A)$	Błąd metody Gaussa	Błąd metody inwersji
1	1.0	1	0.0	0.0
2	19.28147006790397	2	5.661048867003676e-16	1.4043333874306803e-15
3	524.0567775860644	3	8.022593772267726e-15	0.0
4	15513.73873892924	4	4.137409622430382e-14	0.0
5	476607.2502425855	5	1.6828426299227195e-12	3.3544360584359632e-12
6	1.4951058642254734e7	6	2.618913302311624e-10	2.0163759404347654e-10
7	4.753673567446793e8	7	1.2606867224171548e-8	4.713280397232037e-9
8	1.5257575538060041e10	8	6.124089555723088e-8	3.07748390309622e-7
9	4.9315375594102344e11	9	3.8751634185032475e-6	4.541268303176643e-6
10	1.602441698742836e13	10	8.67039023709691e-5	0.0002501493411824886
11	5.222701316549833e14	10	0.00015827808158590435	0.007618304284315809
12	1.7515952300879806e16	11	0.13396208372085344	0.258994120804705
13	3.1883950689209334e18	11	0.11039701117868264	5.331275639426837
14	6.200786281355982e17	11	1.4554087127659643	8.71499275104814
15	3.67568286586649e17	12	4.696668350857427	7.344641453111494

Tabela 3: Wartości wskaźnika uwarunkowania i rzędu macierzy H_n oraz błędy względne rozwiązań układu równań metodami Gaussa i z macierzą odwrotną.

n	c	$rank(A)$	Błąd metody Gaussa	Błąd metody inwersji
5	1.0	5	3.0606736594252445e-16	2.275280134513746e-16
5	10.0	5	1.216188388976237e-16	1.4895204919483638e-16
5	1000.0	5	6.8830246068992106e-15	1.1655121101003682e-14
5	1.0e7	5	1.8433912490050538e-10	1.4317191526598063e-10
5	1.0e12	5	4.0271506137124706e-5	4.0414925152527825e-5
5	1.0e16	4	1.1102230246251565e-16	0.02115773708005347
10	1.0	10	2.4575834280036907e-16	3.0606736594252445e-16
10	10.0	10	3.4932351950072765e-16	3.665417751368233e-16
10	1000.0	10	1.8392579358722525e-14	2.6731071956605676e-14
10	1.0e7	10	1.672423868802004e-10	1.332910952506047e-10
10	1.0e12	10	1.4391864685145854e-5	1.0065760097123466e-5
10	1.0e16	9	0.01264020487412138	0.03992264919036816
20	1.0	20	6.483170143248366e-16	5.159850341939109e-16
20	10.0	20	5.347542221830666e-16	2.9893669801409083e-16
20	1000.0	20	3.1243567653618664e-14	2.9734251634904414e-14
20	1.0e7	20	3.967040176688922e-10	3.761993759834402e-10
20	1.0e12	20	1.650912083708681e-5	1.6620959308749998e-5
20	1.0e16	19	0.06915639868517645	0.06664947595151573

Tabela 4: Wartości wskaźnika uwarunkowania c i rzędu macierzy R_n oraz błędy względne rozwiązań układu równań metodami Gaussa i z macierzą odwrotną.

3.4 Wnioski

W przypadku macierzy Hilberta zarówno wskaźniki uwarunkowania, jak i błędy względne dla obu metod szybko osiągają wysokie wartości. Zauważalna jest zależność, że im wyższy wskaźnik uwarunkowania, tym większy błąd. W tej sytuacji metoda Gaussa wykazała się większą skutecznością – w większości przypadków dostarczała dokładniejsze wyniki.

Dla macierzy losowych o stałym wskaźniku uwarunkowania błędy obu metod są mniejsze i na tyle zbliżone, że trudno wskazać wyraźnego lidera. Ponownie, obserwujemy, że wyższy wskaźnik uwarunkowania prowadzi do większych błędów. Błędy dla macierzy o tym samym wskaźniku uwarunkowania, ale różnych rozmiarach, mają podobny poziom.

Z powyższego można wyciągnąć wniosek, że problem rozwiązywania układu równań $Ax = b$ dla macierzy Hilberta jest źle uwarunkowany. Wskazuje to na znaczenie wskaźnika uwarunkowania, który bezpośrednio wpływa na wielkość błędów.

4 Zadanie 4

4.1 Opis zadania

Problemem zadania jest złożliwy wielomian Wilkinsona $P(x) = \prod_{k=1}^n (x - k)$, który można zapisać w postaci:

- iloczynowej, tj. $P(x) = (x-1)(x-2)(x-3)(x-4)(x-5)(x-6)(x-7)(x-8)(x-9)(x-10)(x-11)(x-12)(x-13)(x-14)(x-15)(x-16)(x-17)(x-18)(x-19)(x-20)$;
- normalnej, tj. $P(x) = x^{20} - 210x^{19} + 20615x^{18} - 1256850x^{17} + 53327946x^{16} - 1672280820x^{15} + 40171771630x^{14} - 756111184500x^{13} + 11310276995381x^{12} - 135585182899530x^{11} + 1307535010540395x^{10} - 10142299865511450x^9 + 63030812099294896x^8 - 311333643161390640x^7 + 1206647803780373360x^6 - 3599979517947607200x^5 + 8037811822645051770x^4 - 12870931245150988800x^3 + 13803759753640704000x^2 - 8752948036761600000x + 2432902008176640000$.

Celem jest wyznaczenie w Julii pierwiastków z_k , gdzie $k \in [1, 20]$. Następnie sprawdzenie ich przez policzenie $|P(z_k)|$, $|p(z_k)|i|z_k - k|$ oraz wyjaśnienie rozbieżności.

Kolejno należy powtórzyć eksperyment Wilkinsona, tj. zmienić współczynnik -220 na $-210 - 2^{-23}$.

4.2 Rozwiązanie

Rozwiązanie opiera się na wykorzystaniu pakietu Polynomials w języku Julia. Najpierw wielomian Wilkinsona zapisano zarówno w postaci naturalnej, jak i iloczynowej (przy użyciu funkcji wbudowanej `fromroots()`). Następnie, za pomocą funkcji `roots()`, obliczono pierwiastki z_k tego wielomianu z jego postaci normalnej. Kolejno, obliczono błąd bezwzględny zarówno dla pierwiastków, jak i dla wyników wielomianu obliczonych w tych punktach. Na koniec, eksperyment powtórzono, zmieniając współczynnik -210 na $-210 - 2^{-23}$.

4.3 Wyniki

k	z_k	$ P(z_k) $	$ p(z_k) $	$ z_k - k $
1	0.9999999999996989	35696.50964788257	5.518479490350445e6	3.0109248427834245e-13
2	2.0000000000283182	176252.60026668405	7.37869762990174e19	2.8318236644508943e-11
3	2.9999999995920965	279157.6968824087	3.3204139316875795e20	4.0790348876384996e-10
4	3.9999999837375317	3.0271092988991085e6	8.854437035384718e20	1.626246826091915e-8
5	5.000000665769791	2.2917473756567076e7	1.8446752056545688e21	6.657697912970661e-7
6	5.999989245824773	1.2902417284205095e8	3.32039488870117e21	1.0754175226779239e-5
7	7.000102002793008	4.805112754602064e8	5.423593016891273e21	0.00010200279300764947
8	7.999355829607762	1.6379520218961136e9	8.262050140110275e21	0.0006441703922384079
9	9.002915294362053	4.877071372550003e9	1.196559421646277e22	0.002915294362052734
10	9.990413042481725	1.3638638195458128e10	1.655260133520688e22	0.009586957518274986
11	11.025022932909318	3.585631295130865e10	2.24783329792479e22	0.025022932909317674
12	11.953283253846857	7.533332360358197e10	2.886944688412679e22	0.04671674615314281
13	13.07431403244734	1.9605988124330817e11	3.807325552826988e22	0.07431403244734014
14	13.914755591802127	3.5751347823104315e11	4.612719853150334e22	0.08524440819787316
15	15.075493799699476	8.21627123645597e11	5.901011420218566e22	0.07549379969947623
16	15.946286716607972	1.5514978880494067e12	7.010874106897764e22	0.05371328339202819
17	17.025427146237412	3.694735918486229e12	8.568905825736165e22	0.025427146237412046
18	17.99092135271648	7.650109016515867e12	1.0144799361044434e23	0.009078647283519814
19	19.00190981829944	1.1435273749721195e13	1.1990376202371257e23	0.0019098182994383706
20	19.999809291236637	2.7924106393680727e13	1.4019117414318134e23	0.00019070876336257925

Tabela 5: Wyniki eksperymentu dla oryginalnego wielomianu.

k	z_k	$ P(z_k) $	$ p(z_k) $	$ z_k - k $
1	0.999999999998357 + 0.0im	20259.872313418207	19987.872313406842	1.6431300764452317e-13
1	0.999999999998357 + 0.0im	20259.872313418207	3.0131001276845885e6	1.6431300764452317e-13
2	2.0000000000550373 + 0.0im	346541.4137593836	7.37869763029606e19	5.503730804434781e-11
3	2.99999999660342 + 0.0im	2.2580597001197007e6	3.320413920110016e20	3.3965799062229962e-9
4	4.000000089724362 + 0.0im	1.0542631790395478e7	8.854437817429642e20	8.972436216225788e-8
5	4.99999857388791 + 0.0im	3.757830916585153e7	1.844672697408419e21	1.4261120897529622e-6
6	6.000020476673031 + 0.0im	1.3140943325569446e8	3.320450195282313e21	2.0476673030955794e-5
7	6.99960207042242 + 0.0im	3.939355874647618e8	5.422366528916004e21	0.00039792957757978087
8	8.007772029099446 + 0.0im	1.184986961371896e9	8.289399860984408e21	0.007772029099445632
9	8.915816367932559 + 0.0im	2.2255221233077707e9	1.160747250177049e22	0.0841836320674414
10	10.095455630535774 - 0.6449328236240688im	1.0677921232930157e10	1.7212892853670706e22	0.6519586830380407
11	10.095455630535774 + 0.6449328236240688im	1.0677921232930157e10	1.7212892853670706e22	1.1109180272716561
12	11.793890586174369 - 1.6524771364075785im	3.1401962344429485e10	2.8568401004080956e22	1.665281290598479
13	11.793890586174369 + 1.6524771364075785im	3.1401962344429485e10	2.8568401004080956e22	2.0458202766784277
14	13.992406684487216 - 2.5188244257108443im	2.157665405951858e11	4.934647147686795e22	2.518835871190904
15	13.992406684487216 + 2.5188244257108443im	2.157665405951858e11	4.934647147686795e22	2.7128805312847097
16	16.73074487979267 - 2.812624896721978im	4.850110893921027e11	8.484694713563005e22	2.9060018735375106
17	16.73074487979267 + 2.812624896721978im	4.850110893921027e11	8.484694713563005e22	2.825483521349608
18	19.5024423688181 - 1.940331978642903im	4.557199223869993e12	1.3181947820607215e23	2.4540214463129764
19	19.5024423688181 + 1.940331978642903im	4.557199223869993e12	1.3181947820607215e23	2.0043294443099486
20	20.84691021519479 + 0.0im	8.756386551865696e12	1.5911084081430876e23	0.8469102151947894

Tabela 6: Wyniki eksperymentu dla marginalnie zaburzonego wielomianu.

k	$ P(k) $	$ p(k) $
1	0.0	0
2	8192.0	0
3	27648.0	0
4	622592.0	0
5	2.176e6	0
6	8.84736e6	0
7	2.4410624e7	0
8	5.89824e7	0
9	1.45753344e8	0
10	2.27328e8	0
11	4.79074816e8	0
12	8.75003904e8	0
13	1.483133184e9	0
14	2.457219072e9	0
15	3.905712e9	0
16	6.029312e9	0
17	9.116641408e9	0
18	1.333988352e10	0
19	1.9213101568e10	0
20	2.7193344e10	0

Tabela 7: Wyniki wielomianów P (niezaburzony) i p dla dokładnych pierwiastków.

4.4 Wnioski

Wartości pierwiastków obliczone dla niezaburzonego wielomianu nie są identyczne z wartościami dokładnymi, ale są do nich bardzo zbliżone. Każdy błąd bezwzględny spełnia nierówność $|z_k - k| < 0.1$. Mimo to, wartości wielomianów dla niedokładnie obliczonych pierwiastków są niezwykle duże. Sugeruje to, że problem wyznaczenia pierwiastków dla wielomianu Wilkinsona jest źle uwarunkowany.

Arytmetyka `Float64` nie umożliwia dokładnego przechowywania współczynnika P ze względu na ograniczoną precyzję. Największe współczynniki nie mogą zostać dokładnie zapisane. W związku z tym, nawet w przypadku dokładnych pierwiastków, $P(k)$ nie zwraca wartości 0, w przeciwieństwie do $p(k)$.

Niewielka zmiana w P spowodowała znaczny wzrost błędu i pojawienie się rozwiązań zespolonych. To ponownie dowodzi, że problem wyznaczenia pierwiastków dla wielomianu Wilkinsona jest źle uwarunkowany.

5 Zadanie 5

5.1 Opis zadania

Problem dotyczy analizy równania rekurencyjnego, znanego jako model logistyczny, opisującego wzrost populacji w zależności od czasu. Równanie ma postać:

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

gdzie:

- p_n to wielkość populacji w czasie n ,
- r to stała tempo wzrostu,
- p_0 to początkowa wielkość populacji wyrażona jako procent maksymalnej wielkości populacji.

Celem zadania jest przeprowadzenie eksperymentów polegających na obliczeniu kolejnych iteracji tego równania przy różnych ustawieniach parametrów i porównaniu wyników uzyskanych w różnych środowiskach numerycznych.

5.2 Rozwiązanie

Rozwiązanie problemu polega na implementacji równania rekurencyjnego w języku Julia oraz przeprowadzeniu dwóch eksperymentów. W pierwszym eksperymencie wykonamy 40 iteracji równania, a następnie zmodyfikujemy wyniki przez obcięcie wartości po trzecim miejscu po przecinku i kontynuowanie obliczeń. W drugim eksperymencie porównamy wyniki uzyskane przy użyciu arytmetyki zmiennoprzecinkowej typu `Float32` i `Float64`.

5.3 Wyniki

n	Float32 p_n	Float32 z obciążeniem p_n	Float64 p_n
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
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

Tabela 8: Numer iteracji n oraz wartości p_n dla wszystkich typów eksperymentu.

5.4 Wnioski

Na początku obie arytmetyki generują podobne wyniki. W 19 i 9 iteracji po obciążeniu, wyniki uzyskane przy użyciu typu `Float32` z obciążeniem zaczynają się wyraźnie różnić od pozostałych. Kilka iteracji później, różnice między typami

`Float32` a `Float64` stają się również bardziej widoczne. Po 40 iteracjach otrzymujemy już całkowicie odmienne wyniki.

Pomimo że początkowe wyrazy są obliczane z dużą precyzją, w miarę wzrostu liczby iteracji błędy stają się coraz bardziej zauważalne. Jest to efektem kumulacji błędów. Jednym z powodów tej sytuacji jest podnoszenie wyrazów do kwadratu, co prowadzi do powstawania dużych błędów.

Proces obliczania kolejnych wartości równania rekurencyjnego p_n jest numerycznie niestabilny, ponieważ małe błędy popełnione na początku iteracji skumulowały się, prowadząc do poważnej utraty precyzji obliczeń.

6 Zadanie 6

6.1 Opis zadania

Rozważmy równanie rekurencyjne:

$$x_{n+1} := x_n^2 + c, \quad \text{dla } n = 0, 1, \dots$$

Wykonać 40 iteracji dla danych:

1. $c = -2, x_0 = 1$
2. $c = -2, x_0 = 2$
3. $c = -2, x_0 = 1.9999999999999999$
4. $c = -1, x_0 = 1$
5. $c = -1, x_0 = -1$
6. $c = -1, x_0 = 0.75$
7. $c = -1, x_0 = 0.25$

6.2 Rozwiązanie

Rozwiązanie polega na implementacji funkcji obliczającej rekurencyjnie następne czterdzieści wyników równania w arytmetyce `Float64`. Pomocne w rozwiązaniu jest przeprowadzenie iteracji graficznych $x_{n+1} := x_n^2 + c$.

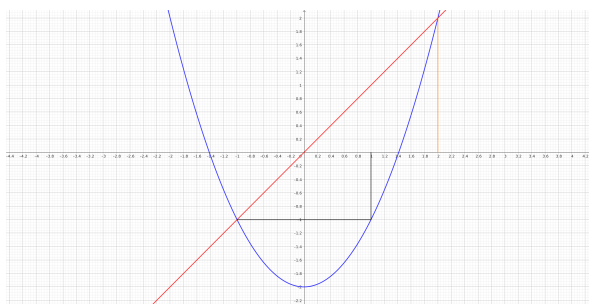
6.3 Wyniki

n	c	x_n z $x_0 = 1$	x_n z $x_0 = 2$	x_n z $x_0 = 1.9999999999999999$
0	-2	1.0	2.0	1.9999999999999999
1	-2	-1.0	2.0	1.9999999999999996
2	-2	-1.0	2.0	1.99999999999998401
3	-2	-1.0	2.0	1.9999999999993605
4	-2	-1.0	2.0	1.999999999997442
5	-2	-1.0	2.0	1.9999999999897682
6	-2	-1.0	2.0	1.9999999999590727
7	-2	-1.0	2.0	1.999999999836291
8	-2	-1.0	2.0	1.9999999993451638
9	-2	-1.0	2.0	1.9999999973806553
10	-2	-1.0	2.0	1.99999989522621
11	-2	-1.0	2.0	1.999999580904841
12	-2	-1.0	2.0	1.999998323619383
13	-2	-1.0	2.0	1.999993294477814
14	-2	-1.0	2.0	1.999973177915749
15	-2	-1.0	2.0	1.999892711734937
16	-2	-1.0	2.0	1.999570848090826
17	-2	-1.0	2.0	1.999828341078044
18	-2	-1.0	2.0	1.9993133937789613
19	-2	-1.0	2.0	1.9972540465439481
20	-2	-1.0	2.0	1.9890237264361752
21	-2	-1.0	2.0	1.9562153843260486
22	-2	-1.0	2.0	1.82677862987391
23	-2	-1.0	2.0	1.3371201625639997
24	-2	-1.0	2.0	-0.21210967086482313
25	-2	-1.0	2.0	-1.9550094875256163
26	-2	-1.0	2.0	1.822062096315173
27	-2	-1.0	2.0	1.319910282828443
28	-2	-1.0	2.0	-0.2578368452837396
29	-2	-1.0	2.0	-1.9335201612141288
30	-2	-1.0	2.0	1.7385002138215109
31	-2	-1.0	2.0	1.0223829934574389
32	-2	-1.0	2.0	-0.9547330146890065
33	-2	-1.0	2.0	-1.0884848706628412
34	-2	-1.0	2.0	-0.8152006863380978
35	-2	-1.0	2.0	-1.3354478409938944
36	-2	-1.0	2.0	-0.21657906398474625
37	-2	-1.0	2.0	-1.953093509043491
38	-2	-1.0	2.0	1.8145742550678174
39	-2	-1.0	2.0	1.2926797271549244
40	-2	-1.0	2.0	-0.3289791230026702

Tabela 9: Wartości x_n dla $c = -2$ w kolejnych iteracjach.

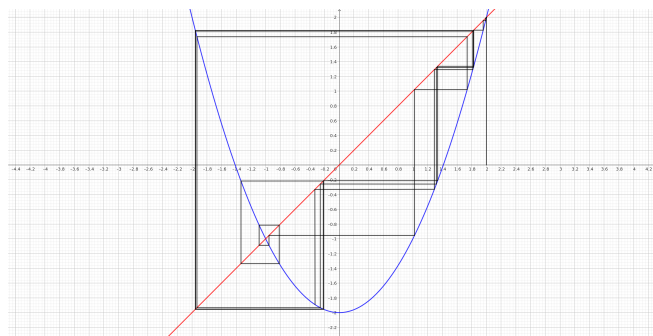
n	c	x_n z $x_0 = 1$	x_n z $x_0 = -1$	x_n z $x_0 = 0.75$	x_n z $x_0 = 0.25$
0	-1	1.0	-1.0	0.75	0.25
1	-1	0.0	0.0	-0.4375	-0.9375
2	-1	-1.0	-1.0	-0.80859375	-0.12109375
3	-1	0.0	0.0	-0.3461761474609375	-0.9853363037109375
4	-1	-1.0	-1.0	-0.8801620749291033	-0.029112368589267135
5	-1	0.0	0.0	-0.2253147218564956	-0.9991524699951226
6	-1	-1.0	-1.0	-0.9492332761147301	-0.0016943417026455965
7	-1	0.0	0.0	-0.0989561875164966	-0.9999971292061947
8	-1	-1.0	-1.0	-0.9902076729521999	-5.741579369278327e-6
9	-1	0.0	0.0	-0.01948876442658909	-0.999999999670343
10	-1	-1.0	-1.0	-0.999620188061125	-6.593148249578462e-11
11	-1	0.0	0.0	-0.0007594796206411569	-1.0
12	-1	-1.0	-1.0	-0.9999994231907058	0.0
13	-1	0.0	0.0	-1.1536182557003727e-6	-1.0
14	-1	-1.0	-1.0	-0.999999999986692	0.0
15	-1	0.0	0.0	-2.6616486792363503e-12	-1.0
16	-1	-1.0	-1.0	-1.0	0.0
17	-1	0.0	0.0	0.0	-1.0
18	-1	-1.0	-1.0	-1.0	0.0
19	-1	0.0	0.0	0.0	-1.0
20	-1	-1.0	-1.0	-1.0	0.0
21	-1	0.0	0.0	0.0	-1.0
22	-1	-1.0	-1.0	-1.0	0.0
23	-1	0.0	0.0	0.0	-1.0
24	-1	-1.0	-1.0	-1.0	0.0
25	-1	0.0	0.0	0.0	-1.0
26	-1	-1.0	-1.0	-1.0	0.0
27	-1	0.0	0.0	0.0	-1.0
28	-1	-1.0	-1.0	-1.0	0.0
29	-1	0.0	0.0	0.0	-1.0
30	-1	-1.0	-1.0	-1.0	0.0
31	-1	0.0	0.0	0.0	-1.0
32	-1	-1.0	-1.0	-1.0	0.0
33	-1	0.0	0.0	0.0	-1.0
34	-1	-1.0	-1.0	-1.0	0.0
35	-1	0.0	0.0	0.0	-1.0
36	-1	-1.0	-1.0	-1.0	0.0
37	-1	0.0	0.0	0.0	-1.0
38	-1	-1.0	-1.0	-1.0	0.0
39	-1	0.0	0.0	0.0	-1.0
40	-1	-1.0	-1.0	-1.0	0.0

Tabela 10: Wartości x_n dla $c = -1$ w kolejnych iteracjach.



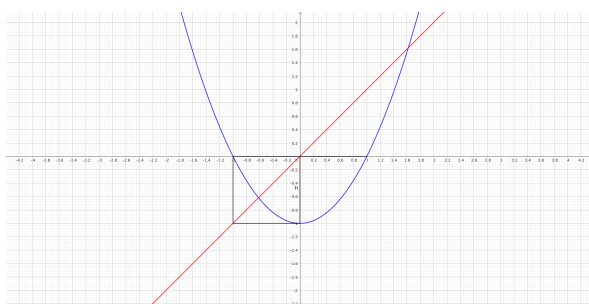
Rysunek 3:

Wykres dla $c = -2$ i $x_0 = 1$ (czarny) lub $x_0 = 2$ (czerwony). W obu przypadkach ciągi wędrują do punktów stałych, tj. odpowiednio -1 i 2, z których nie można wyjść, co obrazuje nam brak dalszych zmian w Tabeli 9.



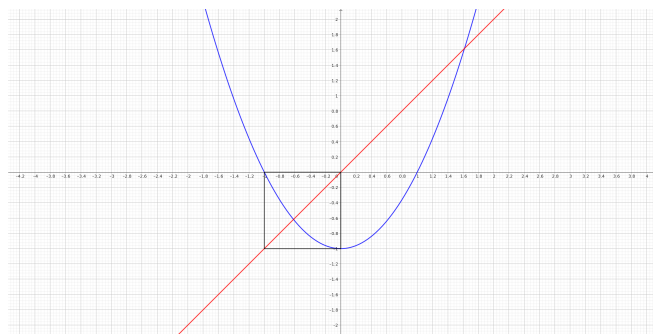
Rysunek 4:

Wykres dla $c = -2$ i $x_0 = 1.9999999999999999$ (czarny). W tym przypadku ciąg raczej się nie stabilizuje- wędruje w okolicach wartości 1.9, 1.7, -0.2, -1.9, co zgadza się z danymi Tabeli 9. Cykl się niestabilizuje.



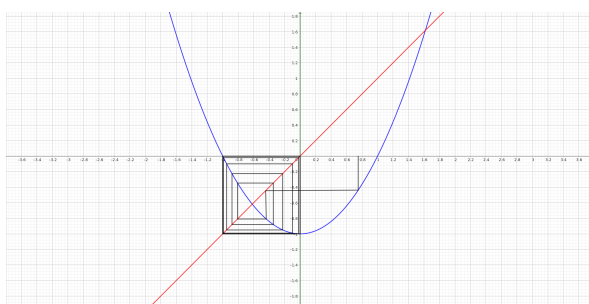
Rysunek 5:

Wykres dla $c = -1$ i $x_0 = 1$ (czarny). W tym przypadku widać zacyklenie się procesu, co odpowiada stabilizującym się wynikom zawartym w Tabeli 10, gdzie w kółko powtarzają się wartości 0 i -1.



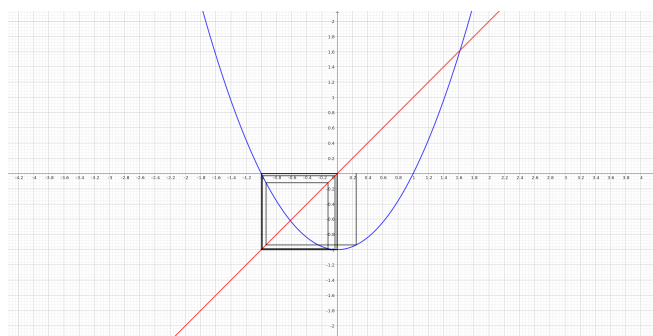
Rysunek 6:

Wykres dla $c = -1$ i $x_0 = -1$ (czarny). W tym przypadku również widać zacyklenie się procesu, co odpowiada stabilizującym się wynikom zawartym w Tabeli 10, gdzie w kółko powtarzają się wartości 0 i -1.



Rysunek 7:

Wykres dla $c = -1$ i $x_0 = 0.75$ (czarny). Podobnie jak w poprzednim przypadku ciąg osiąga stabilizację dopiero po pewnym czasie (z Tabeli 10 można odczytać, że dzieje się to szybciej, już po 10 iteracji). W kółko powtarzają się wartości 0 i -1.



Rysunek 8:

Wykres dla $c = -1$ i $x_0 = 0.25$ (czarny). W tym przypadku ciąg osiąga stabilizację dopiero po pewnym czasie (z Tabeli 10 można odczytać, że dzieje się to po 15 iteracji). W kółko powtarzają się wartości 0 i -1.

6.4 Wnioski

Rekurencyjne wyznaczanie ciągu może wykazywać różną stabilność w zależności od danych wejściowych. Może być całkowicie stabilne, stabilizować się w różnych tempach w zależności od parametrów lub być całkowicie niestabilne. Aby uzyskać wiarygodne wyniki, konieczne jest dobranie odpowiednich wartości parametrów oraz liczby iteracji.