Sprawozdanie

Obliczenia naukowe

Lista 2

Piotr Gałwiaczek

Numer indeksu: 221495

1. **Zadanie 1** – niewielkie zmiany danych początkowych dla iloczynu skalarnego

1.1. Opis problemu

W zadaniu tym mamy porównać zachowanie iloczynu skalarnego w przypadku, gdy spowodujemy drobne zmiany danych wejściowych.

1.2. Rozwiązanie

Do rozwiązania tego zadania użyto programu z zadania 5 z listy 1, oraz uruchomiono je na drobno zmodyfikowanych danych.

1.3. Wyniki oraz ich interpretacja

Tabela bez modyfikacji		
Kolejność sumowania	**Float32**	**Float64**
W przód	-0.4999442994594574	1.0251881368296672e-10
W tył	-0.454345703125	-1.5643308870494366e-10
Od najmniejszego do największego	-0.5	0.0
Od największego do najmniejszego	-0.5	0.0

Tabela z modyfikacjami		
Kolejność sumowania	**Float32**	**Float64**
W przód	-0.4999442994594574	-0.0042963427398915854
W tył	-0.454345703125	-0.0042963429987139534
Od najmniejszego do największego	-0.5	-0.0042963428422808647
Od największego do najmniejszego	-0.5	-0.0042963428422808647

Mimo tak małej zmiany danych wejściowych wyniki znacząco się różnią.

1.4. Wnioski

W arytmetyce Float64 małe zmiany danych wejściowych powodują duże zmiany wyników. W arytmetyce Float32 zmiany na dalekich pozycjach po przecinku nie mają znaczenia, dlatego nie odkształcają wyników. Obliczenie iloczynu skalarnego jest zadaniem źle uwarunkowanym, ponieważ względne zmiany danych powodują duże względne odkształcenia wyników.

2. Zadanie 2 – porównanie wykresów z granicą

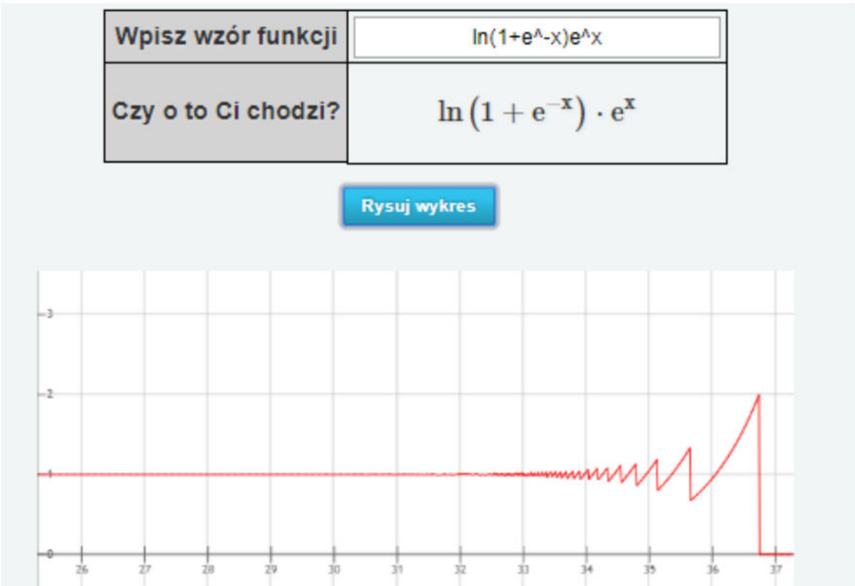
2.1 Opis problemu

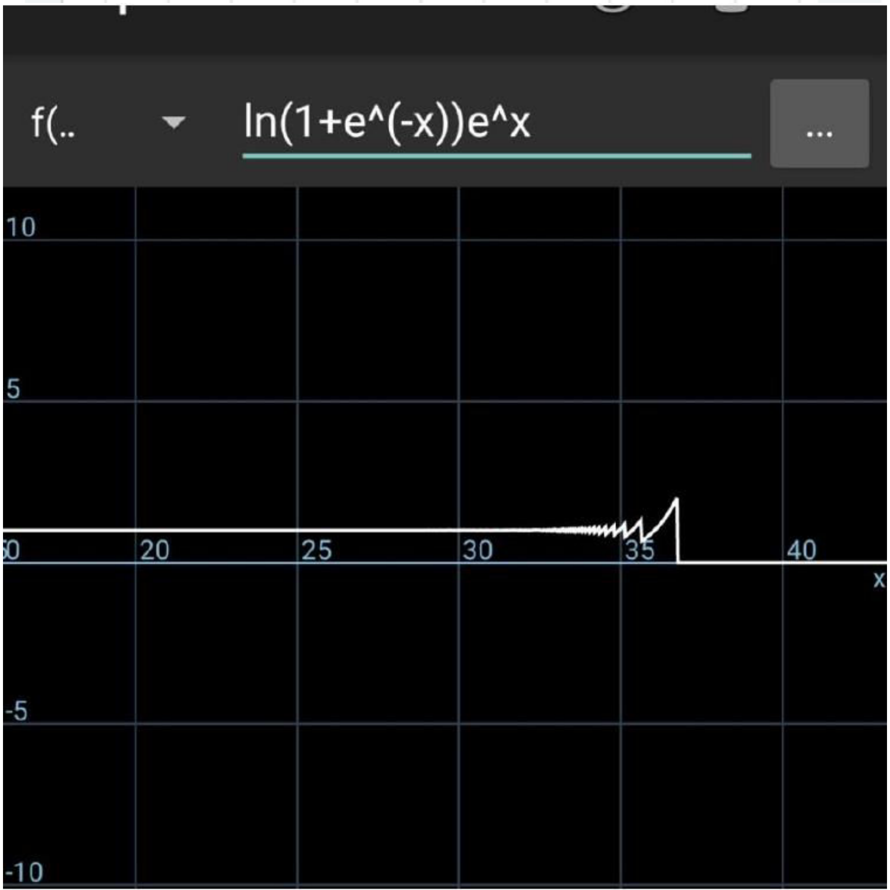
W zadaniu mamy sprawdzić zachowanie funkcji f(x) w dwóch programach do wizualizacji, następnie policzyć granicę funkcji i porównać wykres z policzoną granicą.

2.2 Rozwiązanie

Do rozwiązania zadania użyłem programu ze strony matemaks.pl oraz programu an androida Grapher free

2.3 Wyniki





2.4 Wnioski

Po obliczeniu granicy funkcji f(x) widzimy, że dąży ona do jedynki i z matematycznego punktu widzenia nigdy jej nie osiąga. Jednak w pewnym momencie e^-x będzie tak małe, że przy sumowani jego wartości z jedynką zostane pochłonięte. Z tego wynika, że 1+ e^-x =1, a logarytm z jednki to 0. Funkcja przed dotarciem do zera osiąga 2, ponieważ mnożenie bardzo dużej liczby(e^x) przez bardzo małą (logarytm) daje bardzo duży błąd.

3. **Zadanie 3** – rozwiązania układu równań liniowych dla macierzy

3.1. Opis problemu

W zadaniu mamy rozwiązać układ równań Ax=b, dla dwóch różnych macierzy A:

- Macierzy Hilberta
- Losowej macierzy z podanym wskaźnikiem uwarunkowania

Znając wartość x mamy obliczyć wartość b, a następnie za pomocą algorytmów:

- Eliminacji Gaussa (x=A\b)
- x = A^{-1}b

obliczyć wartość x znając wartości A oraz b.

Dzięki takiemu rozwiązaniu znamy prawdziwą wartość x, oraz możemy ją porównać z tą zwróconą z algorytmów.

Następnie należy policzyć błędy względne wartości x i $\tilde{x}$.

3.2. Rozwiązanie

Do rozwiązania zadania utworzono dwie funkcje:

1) Mającą wypisać dla ***n*** iteracji błędy względne x i $\tilde{x}$ dla A reprezentowanego przez macierz Hilberta, gdzie w każdej ***i***-tej iteracji macierz ta jest stopnia ***i***.
2) Mającą wypisać błędy względne x i $\tilde{x}$ dla macierzy losowych A stopnia kolejno 5, 10, oraz 20, gdzie w każdej iteracji ich wskaźnik uwarunkowania ***c*** jest równy odpowiednio 1, 10, 10^3, 10^7, 10^{12}, 10^{16}.

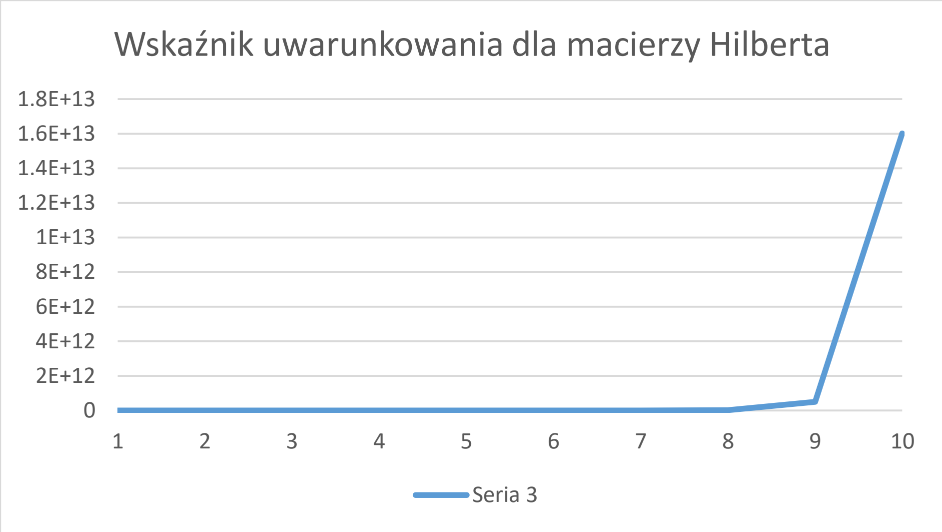
W obu funkcjach w każdej iteracji wartość $\tilde{x}$ jest obliczana dwa razy:

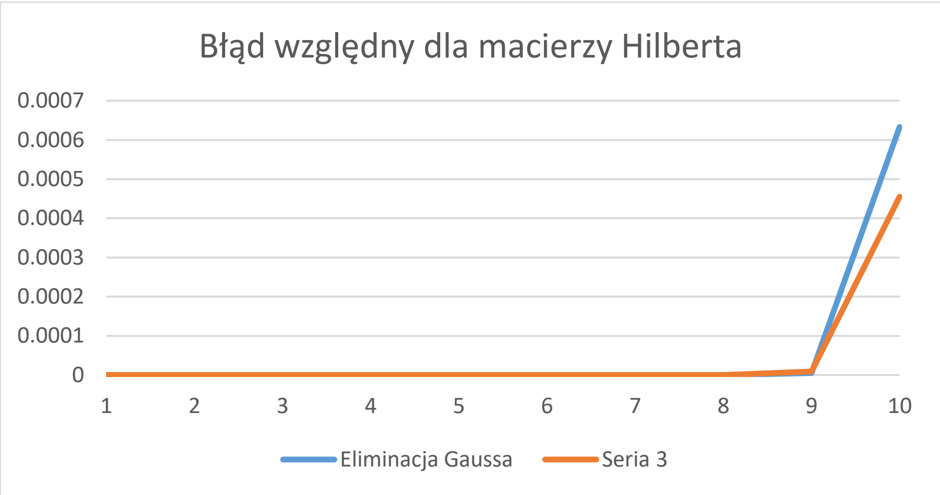
- za pomocą eliminacji Gaussa
- ze wzoru x = A^{-1}b

Dla każdej z tych obliczonych wartości wypisywany jest błąd względny.

3.3. Wyniki oraz ich interpretacja

Rozmiar	Wskaźnik uwarunkowania	Rząd	Błąd względny	
			Eliminacja Gaussa	Macierz odwrotna
1	1	1	0	0
2	19,281470067904	2	5,66104886700368e-16	1,1240151438117e-15
3	524,056777586064	3	8,02259377226773e-15	9,82552603818082e-15
4	15513,7387389292	4	4,45154596018121e-13	2,95047763728678e-13
5	476607,250242594	5	1,68284262992272e-12	8,5000557777533e-12
6	14951058,6422547	6	2,61891330231162e-10	3,34741350703617e-10
7	475367356,583129	7	1,26068672241715e-08	5,16395918357724e-09
8	15257575538,06	8	1,02654306568706e-07	2,69871507427682e-07
9	493153756446,876	9	4,83235712050215e-06	9,17584686861452e-06
10	16024416992541,7	10	0,000632915372298385	0,000455214225174089

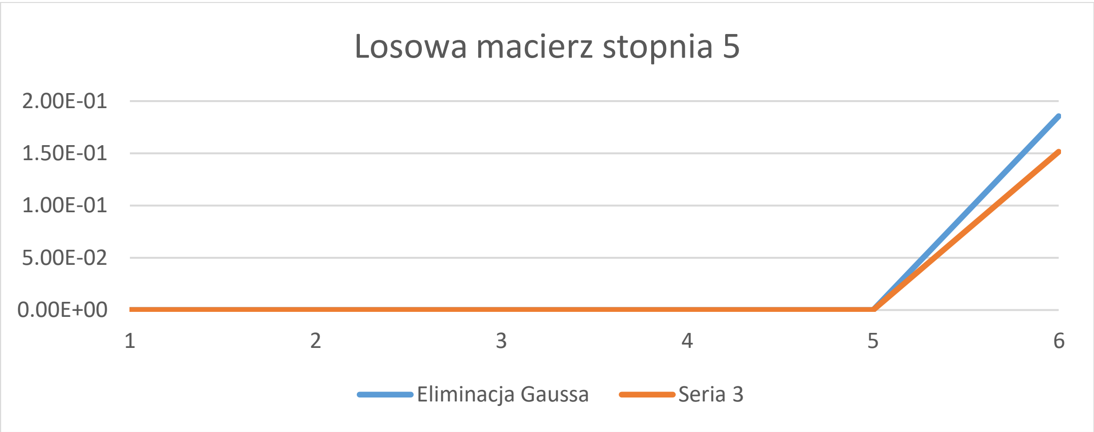


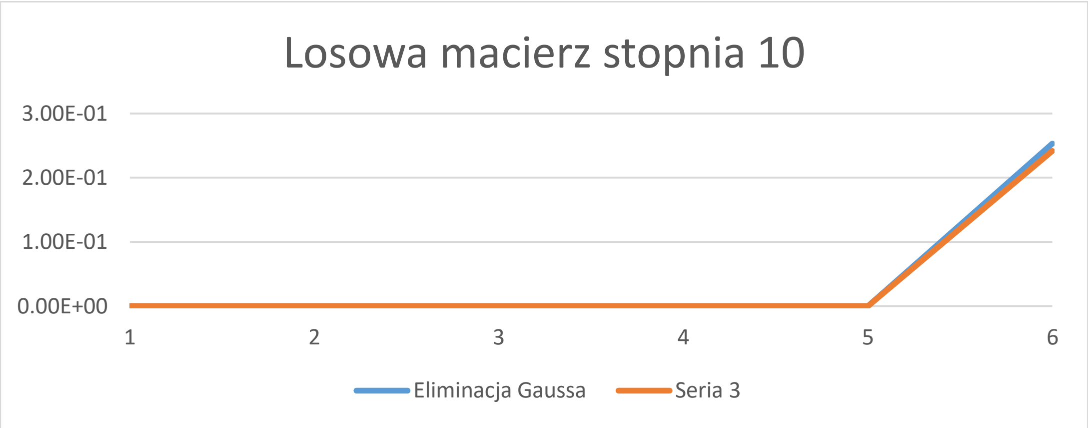


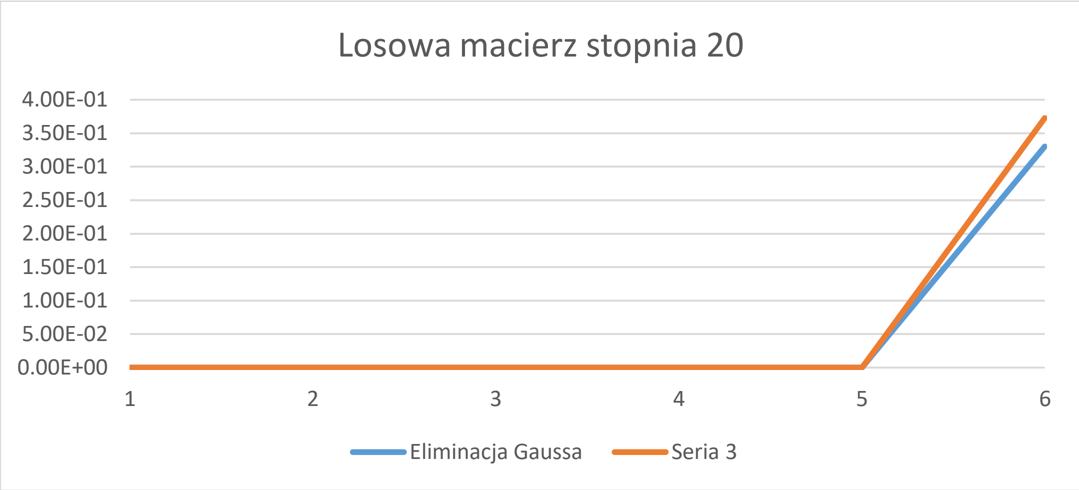
Wskaźnik uwarunkowania oraz błędy względne dla macierzy Hilberta rosną niesamowicie szybko wraz ze wzrostem jego stopnia.

Rozmiar	**Wskaźnik uwarunkowania**	**Rząd**	**Błąd względny**	
			Eliminacja Gaussa	**Macierz odwrotna**
5	1	5	1,11022302462516e-16	1,48952049194836e-16
5	10	5	2,67377111091533e-16	2,67377111091533e-16
5	10^3	5	2,11644736175299e-14	1,03763988358849e-14
5	10^7	5	2,49492786210351e-11	1,23466962808647e-10
5	10^{12}	5	1,98673432417387e-05	1,87468951177004e-05
5	10^{16}	4	0,18549335244033	0,151534546837672
10	1	10	2,89510744497907e-16	2,71947991102104e-16
10	10	10	1,95474934701723e-16	2,40690616200898e-16
10	10^3	10	3,56310681756815e-16	1,60886601221371e-15

10	10^7	10	2,13569305109663e-11	6,19140983792677e-11
10	10^{12}	10	4,42243612579543e-05	4,28403565699284e-05
10	10^{16}	9	0,253214368887547	0,241515846010459
20	1	20	8,54222650911012e-16	5,03287498638511e-16
20	10	20	4,67745274356022e-16	4,39205126597841e-16
20	10^3	20	1,2121581545673e-14	9,89033630608209e-15
20	10^7	20	6,75729838807358e-11	2,26023615207073e-11
20	10^{12}	20	3,83538047537461e-05	3,92202548322745e-05
20	10^{16}	19	0,330396575539836	0,372767606527931







Jak widzimy dla macierzy losowej wskaźnik uwarunkowania ma bardzo duży wpływ na jego błąd względny.

3.4. Wnioski

Macierz Hilberta jest przykładem macierzy źle uwarunkowanej. Wskaźnik uwarunkowania dla tej macierzy wynosi

$$\mathrm{cond}(H_n) = O\left(\frac{e^{3.5255n}}{\sqrt{n}}\right)$$

Wzrasta on bardzo szybko, i już dla małego ***n*** jest on bardzo duży, co mówi, że numeryczne rozwiązanie nawet niewielkich układów równań z tą macierzą jest praktycznie niemożliwe.
W przypadku macierzy losowej również widać, że wskaźnik uwarunkowania ma ogromny wpływ na wyniki zwracane przez algorytmy – czym jest on większy tym większe błędy względne otrzymujemy.

4. **Zadanie 4** – zera wielomianu Wilkinsona

4.1. Opis problemu

W tym zadaniu musimy wyznaczyć miejsca zerowe wielomianu Wilkinsona będącego w postaci naturalnej oraz iloczynowej, oraz porównać je z prawdziwymi wartościami jego pierwiastków. Należy również sprawdzić jakie wartości przyjmuje funkcja dla obliczonych miejsc zerowych.

Następnie mamy powtórzyć eksperyment, lecz zamienić współczynnik przy x^{19} z -210 na $-210-2^{23}$.

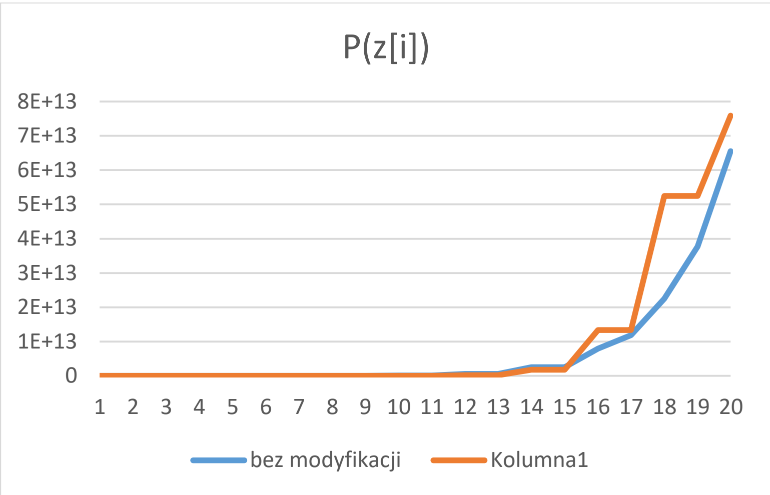
4.2. Rozwiązanie

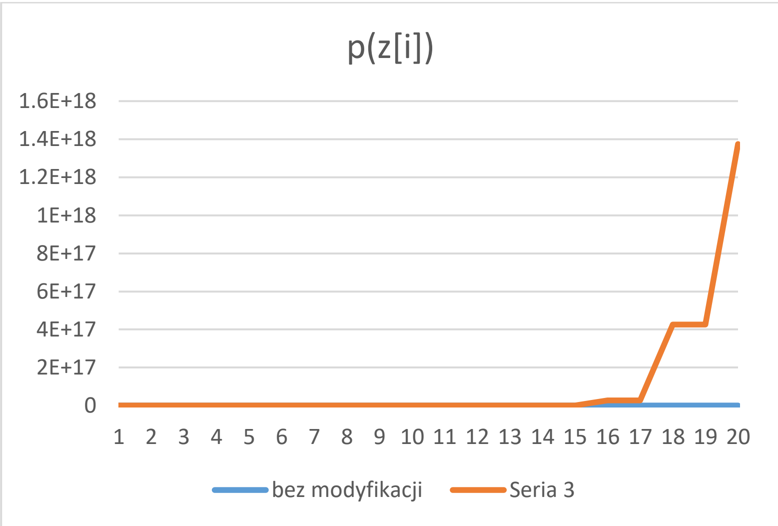
Do rozwiązania tego zadania utworzone zostały dwa wielomiany Wilkinsona: jeden w postaci iloczynowej, drugi w postaci naturalnej. Dla obu z tych funkcji obliczone zostały miejsca zerowe, to jakie funkcja przyjmuje w nich wartości, oraz różnica pomiędzy prawdziwymi miejscami zerowymi.

4.3. Wyniki oraz ich interpretacja

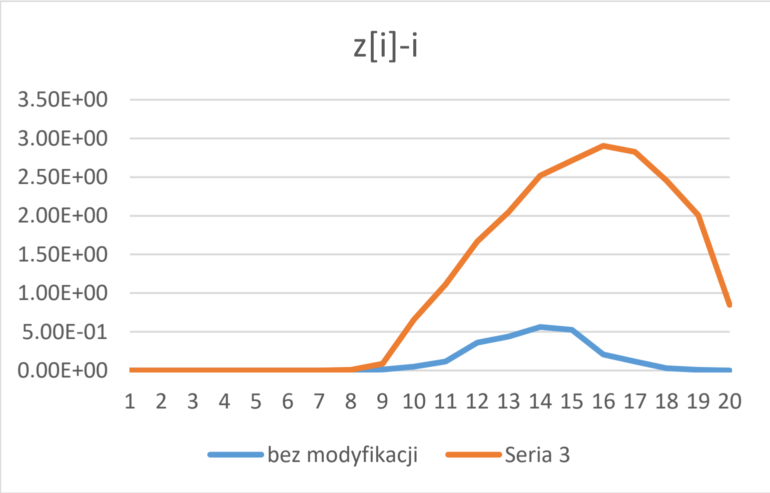
Wielomian Wilkinsona bez modyfikacji				
Nr iteracji	**z[i]**	**P(z[i])**	**p(z[i])**	**z[i]-i**
1	0,9999999999981168	229376,0	230400,0	1,8831602943691905e-12
2	2,0000000001891918	1,209856e6	1,22624e6	1,8831602943691905e-12
3	2,9999999926196894	5,04832e6	5,122048e6	7,380310584892413e-9
4	4,000000196012741	2,34368e7	2,3698944e7	1,9601274114933176e-7
5	4,999996302203527	1,1296512e8	1,1373312e8	3,6977964725792845e-6
6	6,000048439601834	5,03290368e8	5,04617472e8	4,8439601833649704e-5
7	6,999557630040994	1,968515584e9	1,97132544e9	0,0004423699590061503
8	8,002891069857936	6,86119424e9	6,86539008e9	0,0028910698579363014
9	8,986693042189247	2,1149393408e10	2,1155347456e10	0,013306957810753417
10	10,049974037139467	6,454844928e10	6,4558998016e10	0,04997403713946724
11	10,886016935269065	1,43521440768e11	1,43537213952e11	0,11398306473093456
12	12,358657519230299	5,32673099264e11	5,32696562176e11	0,35865751923029876
13	12,561193394139806	6,02947259392e11	6,02975376896e11	0,4388066058601936
14	14,52052610077266	2,4994706855470215e12	2,499521035474694e12	0,5610860887429185
15	14,52052610077266	2,4994706855470215e12	2,499521035474694e12	0,5262119169452244
16	16,206794587063147	7,95998408192e12	7,960054000128e12	0,2067945870631469
17	16,885716688231323	1,183580855552e13	1,1835897310208e13	0,11428331176867701
18	18,030097274474777	2,248445502208e13	2,2484563546624e13	0,030097274474776725
19	18,993902180590464	3,7714572212736e13	3,7714711862272e13	0,006097819409536243
20	20,000542093702702	6,5522804164608e13	6,5522968009216e13	0,0005420937027018624

Wielomian Wilkinsona z modyfikacją				
Nr iteracji	z[i]	P(z[i])	p(z[i])	z[i]-i
1	0,999999999998993	121344,0	123392,0	1,006972283335017e-12
2	2,000000000125739	798208,0	814592,0	1,2573897478773688e-10
3	2,999999997067533	2,075136e6	2,167296e6	2,932467157990004e-9
4	4,0000000239207445	1,92768e6	2,189824e6	2,3920744496308544e-8
5	4,9999998538783785	993280,0	1,56672e6	1,461216214693195e-7
6	6,000008306313138	464384,0	8,4470272e7	8,306313137751431e-6
7	6,999658994843255	1,60592384e8	1,514812416e9	0,0003410051567449557
8	8,007774348640831	1,254020096e9	1,853873408e10	0,007774348640831263
9	8,914607795492033	5,313501184e9	1,39331373056e11	0,0853922045079667
10	10,115234568432546	2,700928553743177e10	1,5061778806872131e12	0,655279113431731
11	10,115234568432546	2,700928553743177e10	1,5061778806872131e12	1,1137920590076884
12	11,910048132599037	1,982419589176788e11	3,3135452934365223e13	1,6669672077692326
13	11,910048132599037	1,982419589176788e11	3,3135452934365223e13	2,0468449137694735
14	14,217982615821274	1,7827980190248804e12	9,559711415363486e14	2,519231315803488
15	14,217982615821274	1,7827980190248804e12	9,559711415363486e14	2,7130223878029276
16	16,96573578747272	1,336962421751052e13	2,743093938691134e16	2,90608704457139
17	16,96573578747272	1,336962421751052e13	2,743093938691134e16	2,8254935594324038
18	19,598783279495287	5,250917494398969e13	4,2529636750149574e17	2,4540535405907598
19	19,598783279495287	5,250917494398969e13	4,2529636750149574e17	2,0043409942895165
20	20,846927301650048	7,5911013208576e13	1,3744570862014474e18	0,846927301650048





p(z[i])
1.6E+18
1.4E+18
1.2E+18
1E+18
8E+17
6E+17
4E+17
2E+17
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
bez modyfikacji
Seria 3



Jak widzimy mała modyfikacja jednego ze współczynników sprawiła, że wyniki znacząco się różnią.

4.4. Wnioski

Obserwując zachowanie wielomianu Wilkinsona na względne zmiany danych zadania nie mamy wątpliwości, że obliczenie jego pierwiastków jest zadaniem źle uwarunkowanym. Nawet niewielki błąd popełniony przy współczynnikach dla dużych wykładników x powoduje duże zniekształcenie wyników.
W pierwszym eksperymencie otrzymujemy rażące wyniki mówiące o tym, że wartości obliczonych przez nas miejsc zerowych są oddalone od 0 o nawet 6.5522804164608e13. Dzieje się tak, ponieważ arytmetyka Float64 posiada jedynie od 15 do 17 cyfr znaczących, gdzie niektóre

współczynniki wielomianu są większe, przez co dochodzi do nieuniknionej straty danych.

5. **Zadanie 5** – wpływ drobnej zmiany wartości wyrażenia w środkowej części iteracji na późniejsze wyniki dla pewnego równania rekurencyjnego

5.1. Opis problemu

W danym zadaniu mamy rekurencyjne równanie

$$P_{n+1} = p_n + rp_n(1 - p_n),\ \text{dla } n=0,1,\ldots,$$

Dla którego wykonać mamy dwa eksperymenty:

1) Dla p_0=0.01 i r=3 wykonać 40 iteracji wyrażenia naszego równania, a następnie znów wykonać 40 iteracji lecz w 10-tej iteracji mamy zastosować obcięcie dla aktualnej liczby po trzecim miejscu po przecinku. Mamy porównać otrzymane wyniki.
2) Dla p_0=0.01 i r=3 wykonać 40 iteracji wyrażenia naszego równania w arytmetyce Float32 a następnie Float64. Mamy porównać otrzymane wyniki.

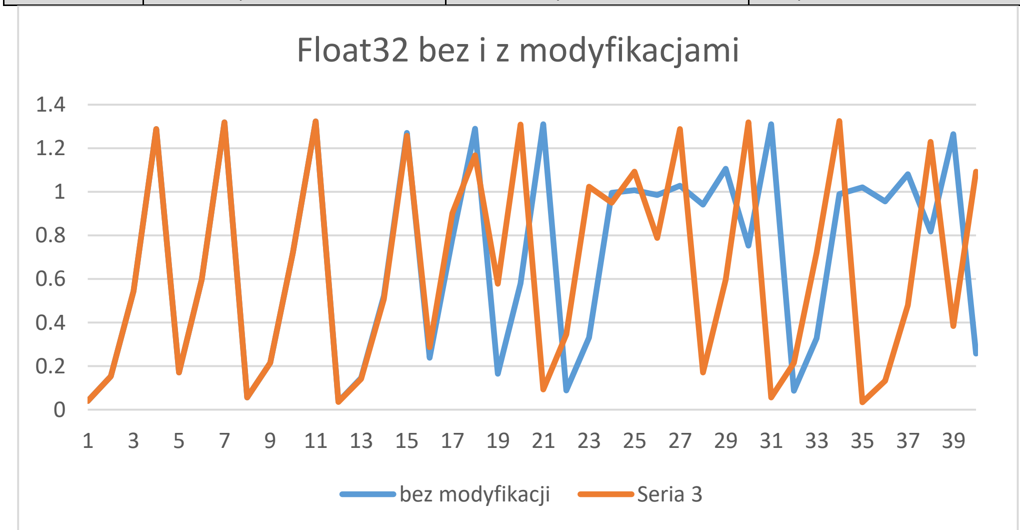
5.2. Rozwiązanie

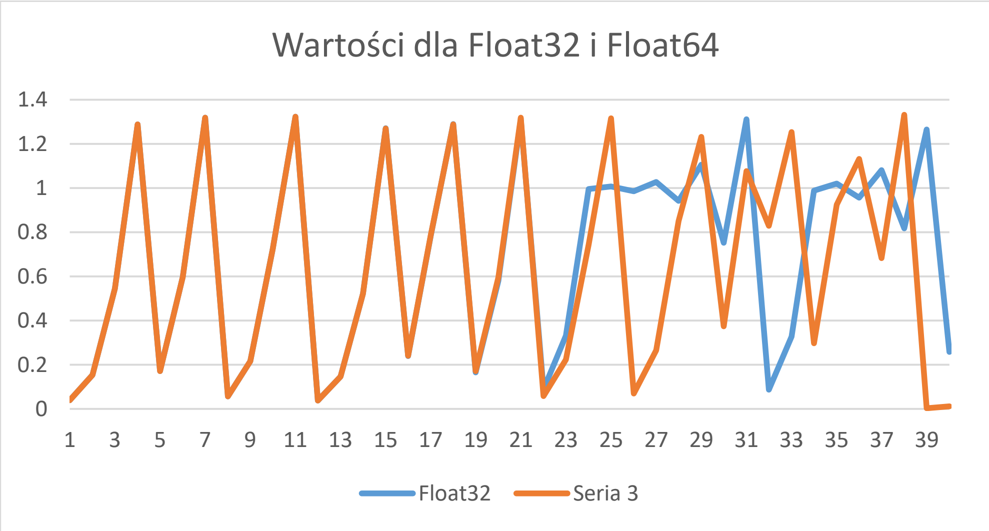
Do rozwiązania tego zadania utworzony został prosty program obliczający dane wyrażenie i wypisujący wyniki w każdej iteracji.

5.3. Wyniki oraz ich interpretacja

Nr iteracji	Float32 bez modyfikacji	Float32 z modyfikacjami	Float64 bez modyfikacji
1	0,0397	0,0397	0,0397
2	0,15407173	0,15407173	0,15407173000000002
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





5.4. Wnioski

Ten proces numeryczny jest niestabilny, ponieważ niewielki błąd popełniony w początkowym studium procesu kumuluje się w kolejnych etapach, przez co tracimy dokładność obliczeń.

W pierwszym eksperymencie celowo popełniony błąd stopniowo coraz bardziej zniekształca nam wyniki.

W drugim eksperymencie również są popełniane błędy które spowodowane są mniejszą precyzją arytmetyki Float32 w stosunku do Float64. Nawet jeśli w początkowej fazie iteracji wyniki nie różnią się znacząco, to i tak powodowane w trakcie błędy kumulują się i w końcowej fazie eksperymentu wyniki znacząco się różnią.

6. **Zadanie 6** – zachowanie generowanych ciągów dla różnych danych wejściowych

6.1. Opis problemu

W danym zadaniu mamy zaobserwować zachowanie generowanych ciągów dla równania rekurencyjnego postaci:

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

w zależności od podanych parametrów x_0 oraz c które są następujące:

1. c = -2 i x_o = 1
2. c = -2 i x_o = 2
3. c = -2 i x_o = 1. 99999999999999
4. c = -1 i x_o = 1
5. c = -1 i x_o = -1

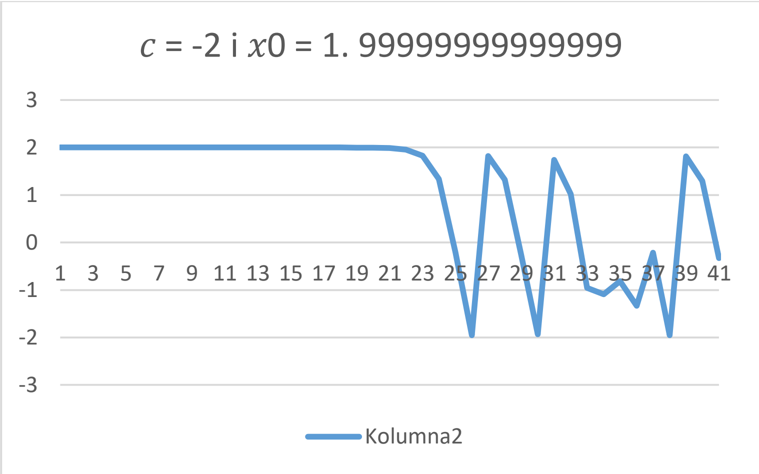
6. c = -1 i x_o = 0.75
7. c = -1 i x_o = 0.25

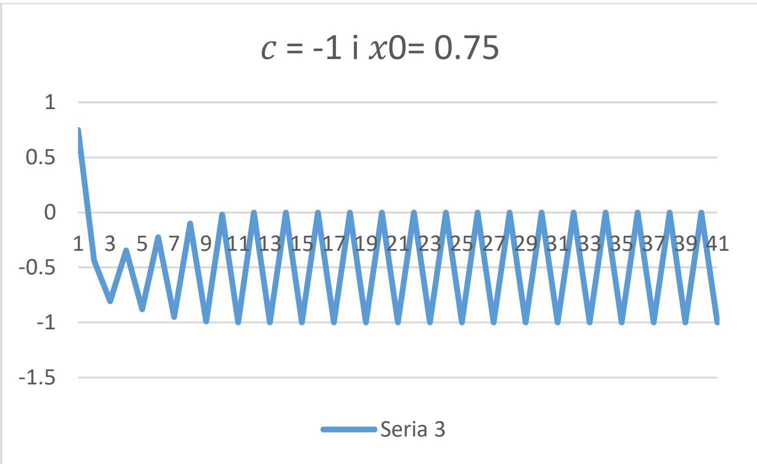
6.2. Rozwiązanie

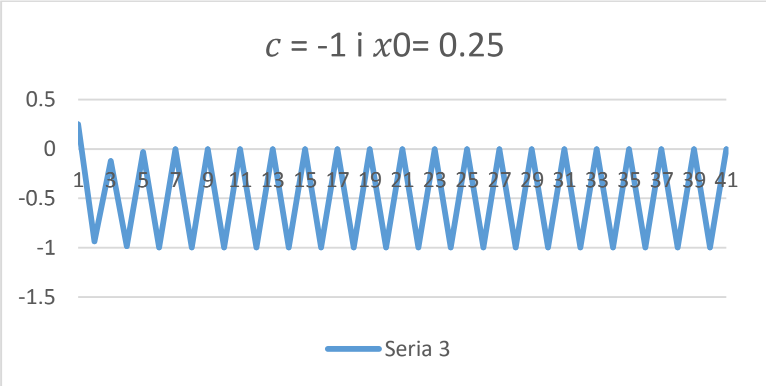
Do rozwiązania tego zadania utworzona została prosta funkcja uruchamiana z parametrami x_0 oraz c obliczającej podane wyrażenie. Wypisuje ona w każdej i-tej iteracji wartość x_i .

6.3. Wyniki oraz ich interpretacja

Nr iteracji	1	2	3	4	5	6	7
0	1	2	1,99999999999999	1	-1	0,75	0,25
1	-1	2	1,99999999999996	0	0	-0,4375	-0,9375
2	-1	2	1,99999999999984	-1	-1	-0,80859375	-0,12109375
3	-1	2	1,99999999999936	0	0	-0,346176147460938	-0,985336303710938
4	-1	2	1,99999999999744	-1	-1	-0,880162074929103	-0,0291123685892671
5	-1	2	1,99999999998977	0	0	-0,225314721856496	-0,999152469995123
6	-1	2	1,99999999995907	-1	-1	-0,94923327611473	-0,0016943417026456
7	-1	2	1,99999999983629	0	0	-0,0989561875164966	-0,999997129206195
8	-1	2	1,99999999934516	-1	-1	-0,9902076729522	-5,74157936927833e-06
9	-1	2	1,99999999738066	0	0	-0,0194887644265891	-0,999999999967034
10	-1	2	1,99999998952262	-1	-1	-0,999620188061125	-6,59314824957846e-11
11	-1	2	1,99999995809048	0	0	-0,000759479620641157	-1
12	-1	2	1,99999983236194	-1	-1	-0,999999423190706	0
13	-1	2	1,99999932944778	0	0	-1,15361825570037e-06	-1
14	-1	2	1,99999731779157	-1	-1	-0,999999999998669	0
15	-1	2	1,99998927117349	0	0	-2,66164867923635e-12	-1
16	-1	2	1,99995708480908	-1	-1	-1	0
17	-1	2	1,99982834107804	0	0	0	-1
18	-1	2	1,99931339377896	-1	-1	-1	0
19	-1	2	1,99725404654395	0	0	0	-1
20	-1	2	1,98902372643618	-1	-1	-1	0
21	-1	2	1,95621538432605	0	0	0	-1
22	-1	2	1,82677862987391	-1	-1	-1	0
23	-1	2	1,337120162564	0	0	0	-1
24	-1	2	-0,212109670864823	-1	-1	-1	0
25	-1	2	-1,95500948752562	0	0	0	-1
26	-1	2	1,82206209631517	-1	-1	-1	0
27	-1	2	1,31991028282844	0	0	0	-1
28	-1	2	-0,25783684528374	-1	-1	-1	0
29	-1	2	-1,93352016121413	0	0	0	-1
30	-1	2	1,73850021382151	-1	-1	-1	0
31	-1	2	1,02238299345744	0	0	0	-1
32	-1	2	-0,954733014689007	-1	-1	-1	0
33	-1	2	-1,08848487066284	0	0	0	-1
34	-1	2	-0,815200686338098	-1	-1	-1	0
35	-1	2	-1,33544784099389	0	0	0	-1
36	-1	2	-0,216579063984746	-1	-1	-1	0
37	-1	2	-1,95309350904349	0	0	0	-1
38	-1	2	1,81457425506782	-1	-1	-1	0
39	-1	2	1,29267972715492	0	0	0	-1
40	-1	2	-0,32897912300267	-1	-1	-1	0


c = -2 i x0 = 1. 99999999999999
3
2
1
0
-1
-2
-3
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41
Kolumna2


c = -1 i x0= 0.75
1
0.5
0
-0.5
-1
-1.5
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41
Seria 3



6.4. Wnioski

Dla danych początkowych 1, 2, 4, oraz 5 zwracane wyniki pośrednie są poprawne.

W przypadku 3 różnica między danymi początkowymi w porównaniu do podpunktu 2 jest minimalna, lecz jak się okazuje staje się ona kolosalna w późniejszym etapie obliczeń.

Jeżeli δ jest błędem względnym równym:

$$\delta = 2 - 1.99999999999999 \vee \frac{}{2\vee} = 5 * 10^{-15}$$

To otrzymujemy:

$$x_1 = x_0^2 - 2 = \left(2(1-\delta)\right)^2 - 2 = \left(4(1-2\delta+\delta^2)\right) - 2 \approx\approx 4(1-2\delta) - 2 = 4 - 8\delta - 2 = 2 - 8\delta = 2(1-4\delta)$$

$$x_2 = x_1^2 - 2 = \left(2(1-4\delta)\right)^2 - 2 = \left(4(1-8\delta+16\delta^2)\right) - 2 \approx\approx 4(1-8\delta) - 2 = 4 - 32\delta - 2 = 2 - 32\delta = 2(1-4^2\delta)$$

Ogólnie:

$$x_i \approx 2\left(1 - 4^i\delta\right) = 2\left(1 - 4^i * 5 * 10^{-15}\right)$$

co dokładnie pokazuje wpływ minimalnego odchylenia danych początkowych na zwracane wyniki. Analiza przypadku 1 obrazuje nam, że w podanym wzorze dla c=-2 gdy liczba x_i ma wartość bliską 1 to liczba x_{i+1} będzie miała wartość ujemną. Do takiego momentu dochodzimy w 23 iteracji. Od tej pory zwracane przez algorytm wyniki wpadają w pewien okres. Sytuacja ta pokazuje, jak bardzo niewielkie odchylenie danych początkowych w zadaniu tego typu może zmienić zwracane przez algorytm wartości.

Dla c=-1 i x_0=0.75 kolejne wartości zbliżają się do okresu identycznego jak w przypadku 5 – w iteracjach nieparzystych wartości przybliżają się do 0, zaś w parzystych do -1. Gdy w iteracji 16 zmiennej x_{16} udaje się uzyskać wartość -1, zaczynamy otrzymywać identyczne wyniki jak w przypadku 5.

Dla c=-1 i x_0=0.25 mamy identyczną sytuację, jednak tym razem wartości w iteracjach parzystych zbliżają się do 0, w nieparzystych do -1. Do wartości -1 udaje się dojść już w 11 iteracji.