Obliczenia naukowe. Lista 2.

Mateusz Walczak, nr indeksu 236775

21 października 2018

Wstęp

Celem listy jest zapoznanie się z pojęciem uwarunkowania zadania. Pokazuje, że przy pewnych problemach mała zmiana w danych może skutkować dużymi odchyleniami w wynikach.

1 Źle uwarunkowany iloczyn skalarny (zad. 1.)

W tym zadaniu miałem powrócić do obliczania iloczynu skalarnego z listy 1. Wykonać 4 sposoby sumowania.

- 'w przód': $\sum_{i=1}^{n} x_i y_i$
- 'w tył': $\sum_{i=n}^{1} x_i y_i$
- 'od największego do najmniejszego': od największego do najmniejszego (dodaję dodatnie liczby w porządku od największego do najmniejszego, dodaję ujemne liczby w porządku od najmniejszego do największego, a następnie dodaję do siebie obliczone sumy częściowe)
- 'od najmniejszego do największego': przeciwnie do metody 'od największego do najmniejszego'

Jedyną różnicą była niewielka zmiana w danych. We współrzędnych wektora x zmiennym x_4 i x_5 zostały obcięte odpowiednio ostatnia 9 i ostatnia 7. Stąd dane prezentowały się następująco.

x = [2.718281828, -3.141592654, 1.414213562, 0.577215664, 0.301029995]
y = [1486.2497, 878366.9879, -22.37492, 4773714.647, 0.000185049]

1.1 Wyniki dla poszczególnych typów

*Float*32

	Przed zmianą	Po zmianie
'w przód'	−0.4999443	−0.4999443
'w tył'	−0.4543457	−0.4543457
'od największego'	−0.5	−0.5
'od najmniejszego'	−0.5	−0.5

Tabela 1: Wartości iloczynu skalarnego dla *Float*32 przed i po zmianie danych

*Float*64

	Przed zmianą	Po zmianie
'w przód'	$1.0251881368296672 \cdot 10^{-10}$	-0.004296342739891585
'w tył'	$-1.5643308870494366 \cdot 10^{-10}$	-0.004296342998713953
'od największego'	0.0	-0.004296342842280865
'od najmniejszego'	0.0	-0.004296342842280865

Tabela 2: Wartości iloczynu skalarnego dla *Float*64 przed i po zmianie danych

1.2 Wnioski z zadania

Wartości iloczynu skalarnego dla typu *Float*32 pozostała bez zmian. Jest to związane z precyzją arytemtyki. Jest ona za mała. Obcięcie ostatnich liczb zmiennych x_4 i x_5 nie miało żadnego wpływu na ich zapis we *Float*32. Nie zmienia to faktu, że wartości obliczone wszystkimi czterema metodami są dalekie od prawidłwych. Z drugiej strony w arytmetyce *Float*64 zmiana jest znacząca. Niewielka różnia w danych wejściowych, bo rzędu 10^{-9}, rzutowała na bardzo dużą różnicę w wynikach. Stąd możemy wnioskować, że wszystkie 4 metody sumowania są źle uwarunkowane. Warto jest też zaznaczyć, że dla arytmetyki *Float*64 wyniki po modyfikacji są bardzo podobne. Jest to również skutek zmiany danych. Wektory nie są już tak blisko bycia ortagonalnymi, więc precyzja użytej arytmetyki wystarczyła na poprawne obliczenie (w granicy błędu) iloczynu skalarnego x i y

2 Granica funkcji(zad. 2.)

W tym zadaniu celem było narysowanie w dwóch programach graficznych funkcji danej wzorem:

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

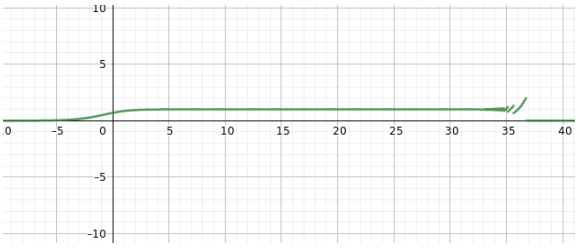
Następnie porównanie ich z granicą przy x dążącym do ∞. W moim przypadku, aby narysować wykresu użyłem programów, takich jak *FooPlot* i *Geogebra*, natomiast do obliczenia granicy skorzystałem z biblioteki *SymPy*.

2.1 Wyniki

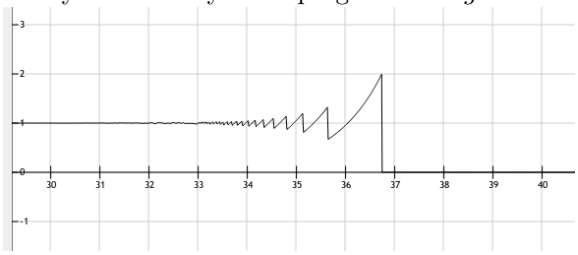
Granica wyliczona dzięki bibliotece *SymPy*:

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

Wykresy wygenerowane dzięki programom do wizualizacji.



Rysunek 1: Wykres z programu *Geogebra*.



Rysunek 2: Wykres z programu *Fooplot*.

2.2 Wnioski z zadania

Można łatwo sprawdzić, że granica obliczona za pomocą biblioteki *SymPy* jest poprawna. Na wykresach natomiast widać, że od $x \approx 30$ wartości funkcji zaczynają się wahać i osiągając liczby większe niż wartość granic, co jest błędne. Za takie zachowanie odpowiada mnożenie bardzo małej wartości logarytmu z wielką e^x. Co więcej dla $x \geqslant 37$ funkcja ewaluowana jest na 0. Dzieje się tak, ponieważ wartość e^{-x} jest tak mała, że podczas dodawania jej do 1 jest pochłaniana. Za tym idzie $\ln(1+e^{-x}) = \ln(1) = 0$. Warto dodać, że w niektórych programach graficznych wykres tej funkcji w pewnym momencie zostanie przerwany. Stanie się to wtedy, gdy e^x będzie równe ∞ (wystąpi tzw. $overflow$). Funkcja ta jest źle uwarunkowana o czym świadczą wahania na wykresach. Mała zmiana w danych oznaczała duże odchylenia w wyniku.

3 Rozwiązywanie równań liniowych (zad. 3.)

W tej części rozwiązywałem równania liniowe $\mathbf{Ax} = \mathbf{b}$. Wykorzystywałem do tego dwa algorytmy:

1. eliminację Gaussa (($x = A\ b$)
2. wykorzystujący macierz odwrotną ($x = A^{-1}b$)

To zadanie było podzielone na dwa podpunkty w zależności od danych wejściowych. $\mathbf{A}$ wyznaczane jest według:

a) *macierz Hilberta* $\mathbf{H}_n$ stopnia n,

b) *macierz losowa* $\mathbf{R}_n^c$ stopnia n o danym wskaźniku uwarunkowania c.

Gdzie $c = 1, 10^3, 10^7, 10^{12}, 10^{16}$, a $n = 5, 10, 20$. Wartość $\mathbf{b}$ wyliczana jest według $\mathbf{b} = \mathbf{Ax}$, gdzie $\mathbf{x} = [1, 1, ..., 1]^T$. Po rozwiązaniu równania miałem dodatkowo wyliczyć błąd bezwzględny wyniku.

3.1 Wyniki

Poniższa tabela pokazuje błedy względne dla obu algorytmów, gdzie **A** jest macierzą Hilberta rozmiaru nxn, uwarunkowanie i rząd macierzy.

n	rząd	uwarunkowanie	macierz odwrotna	Gauss
1	1	1.0	0.0	0.0
2	2	19.28147006790397	$1.4043333874306803 \cdot 10^{-15}$	$5.661048867003676 \cdot 10^{-16}$
3	3	524.0567775860644	0.0	$8.022593772267726 \cdot 10^{-15}$
4	4	15513.738738928929	$7.542470546988852 \cdot 10^{-13}$	$4.637277712035294 \cdot 10^{-13}$
5	5	476607.25024224253	$7.45602798259539 \cdot 10^{-12}$	$1.7697056701418277 \cdot 10^{-13}$
6	6	$1.495105864125091 \cdot 10^{7}$	$3.533151828962887 \cdot 10^{-10}$	$3.496491467713994 \cdot 10^{-10}$
7	7	$4.7536735637688667 \cdot 10^{8}$	$6.190844397992631 \cdot 10^{-9}$	$1.3175049864850338 \cdot 10^{-8}$
8	8	$1.5257575516147259 \cdot 10^{10}$	$3.775275483015941 \cdot 10^{-7}$	$2.487433466002445 \cdot 10^{-7}$
9	9	$4.9315408927806335 \cdot 10^{11}$	$1.1659486044133412 \cdot 10^{-5}$	$9.643625435772316 \cdot 10^{-6}$
10	10	$1.6024859712306152 \cdot 10^{13}$	0.0003357158826776558	0.00022035288727930986
11	11	$5.2210348947688544 \cdot 10^{14}$	0.01113776822564549	0.006022512934347414
12	11	$1.7255427417341868 \cdot 10^{16}$	0.16218620232347905	0.19509235225028912
13	11	$7.126491965424366 \cdot 10^{17}$	5.511855154155295	7.894191771622431
14	12	$6.101307732044041 \cdot 10^{17}$	3.3522039875276723	0.8270688593203056
15	12	$4.223311222761075 \cdot 10^{17}$	4.354299435453685	3.10349386243609
16	12	$3.535827507735838 \cdot 10^{17}$	54.189834405860445	9.083139658689422
17	12	$3.1182808742153696 \cdot 10^{17}$	5.786281231941037	4.24328971542452
18	12	$1.5639169583348145 \cdot 10^{18}$	5.7599951815224495	4.7860299021083
19	13	$1.3274441976880407 \cdot 10^{18}$	12.309212980457932	6.114994252530053
20	13	$2.2777635596453635 \cdot 10^{18}$	17.030822563878868	19.122235961045973

Tabela 3: Wartości błedów względnych dla konkretnych algorytmów i macierzy Hilberta

Poniższa tabela pokazuje błedy względne dla obu algorytmów, gdzie **A** jest macierzą generowaną losowo rozmiaru nxn o podanym uwarunkowaniu i rząd macierzy.

n	rząd	uwarunkowanie	macierz odwrotna	Gauss
5	5	1.0	$2.220446049250313 \cdot 10^{-16}$	$2.8522145930998397 \cdot 10^{-16}$
5	5	10.0	$4.440892098500626 \cdot 10^{-16}$	$8.992121150493565 \cdot 10^{-16}$
5	5	1000.0	$9.057678187205881 \cdot 10^{-15}$	$1.6647349085518926 \cdot 10^{-14}$
5	5	$1.0 \cdot 10^{7}$	$2.459230620213506 \cdot 10^{-10}$	$1.9761065371825815 \cdot 10^{-10}$
5	5	$1.0 \cdot 10^{12}$	$1.7324354290566396 \cdot 10^{-5}$	$1.8626817076587858 \cdot 10^{-5}$
5	4	$1.0 \cdot 10^{16}$	0.3071415593826404	0.2424294239433579
10	10	1.0	$3.080743865682491 \cdot 10^{-16}$	$3.1985215122904827 \cdot 10^{-16}$
10	10	10.0	$3.040470972244059 \cdot 10^{-16}$	$3.0606736594252445 \cdot 10^{-16}$
10	10	1000.0	$5.216866993716143 \cdot 10^{-15}$	$1.1008072638209573 \cdot 10^{-14}$
10	10	$1.0 \cdot 10^{7}$	$2.281044623918076 \cdot 10^{-10}$	$2.0304075589005433 \cdot 10^{-10}$
10	10	$1.0 \cdot 10^{12}$	$1.764684067721615 \cdot 10^{-5}$	$1.6446178932532505 \cdot 10^{-5}$
10	9	$1.0 \cdot 10^{16}$	0.05435164728414402	0.10520379840448996
20	20	1.0	$6.502154576691376 \cdot 10^{-16}$	$3.861916815434371 \cdot 10^{-16}$
20	20	10.0	$6.535244635568843 \cdot 10^{-16}$	$4.530199718267439 \cdot 10^{-16}$
20	20	1000.0	$4.994481756689637 \cdot 10^{-14}$	$4.909365541695919 \cdot 10^{-14}$
20	20	$1.0 \cdot 10^{7}$	$2.1301639636030697 \cdot 10^{-10}$	$2.3452420125960554 \cdot 10^{-10}$
20	20	$1.0 \cdot 10^{12}$	$5.054248948760549 \cdot 10^{-6}$	$2.716263247524948 \cdot 10^{-6}$
20	19	$1.0 \cdot 10^{16}$	0.06411487950107603	0.06508748486177346

Tabela 4: Wartości błedów względnych dla konkretnych algorytmów imacierzy generowanej losowo z zadanym uwarunkowaniem

3.2 Wnioski z zadania

Pierwszym wnioskiem, jaki można wyciągnąć z tego eksperymentu jest fakt, że im większy jest wskaźnik uwarunkowania macierzy, tym większy jest błąd w rozwiązaniu. Co więcej można zauważyć, że problemy, w których mamy do czynienia z macierzami o dużym wskaźniku uwarunkowania są źle uwarunkowane. Świadczą o tym przede wszystkim wartości błędów dla macierzy generowanej losowo (szczególnie gdzy przyjrzymy się błędom przy tym samym rozmiarze macierzy). Małe zmiany w macierzy generowanej losowo odbiły się mocno na rezultacie, przez co błąd jest znaczący. Warto również wspomnieć o tym, że macierz Hilberta jest wyjątkowo utrudnia poprawne rozwiązanie zadania. Wraz ze wzrostem jej rozmiaru znacznie rośnie wskaźnik jej uwarunkowania, co jak już wcześniej zauważyliśmy implikuje złe uwarunkowanie zadania. Ostatnim wnioskiem, jaki wyciągnąłem z tego zadania jest fakt, że z numerycznego punktu widzenia odwracanie macierzy nie jest rzeczą porządaną. Mówi o tym fakt, że metoda eliminacji Gaussa prawie zawsze osiągała mniejszy błąd niezależnie od uwarunkowania macierzy.

4 "Złośliwy wielomian" Wilkinsona (zad. 4.)

W tej sekcji zajmiemy się wielomianem Wilkinsona. Jest to wielomian postaci:

$$p(x) = (x-20)(x-19)...(x-1)$$

Można go też zapisać w tzw. postaci naturalnej (czyli takiej którą dostajemy po wymnożeniu nawiasów ze wzoru wyżej):

$$P(x) = x^{20} - 210x^{19} + ...$$

Zadaniem było wyliczyć pierwiastki dla obu sposobów zapisu wielomianu. Następnie wstawić je do wielomianów i policzyć jego wartość. Na końcu porównać pierwiastki z programu, z rzeczywistymi pierwiastkami, czyli $1, 2, 3, ..., 20$. Następnie powtórzyć eksperyment zmieniając wartość współczynnika przy x^{19} poprzez odjęcie od niego 2^{-23}.

4.1 Wyniki dla obu części zadania

Wyniki przed zmianą współczynnika i po zmianie

pierwiastek	wartość $P(x)$	wartość $p(x)$	błąd wyliczenia pierwiastka
0.9999999999996989	36352.0	38400.0	$3.0109248427834245 \cdot -13$
2.0000000000283182	181760.0	198144.0	$2.8318236644508943 \cdot -11$
2.9999999995920965	209408.0	301568.0	$4.0790348876384996 \cdot -10$
3.9999999837375317	$3.106816 \cdot 6$	$2.844672 \cdot 6$	$1.626246826091915 \cdot -8$
5.000000665769791	$2.4114688 \cdot 7$	$2.3346688 \cdot 7$	$6.657697912970661 \cdot -7$
5.999989245824773	$1.20152064 \cdot 8$	$1.1882496 \cdot 8$	$1.0754175226779239 \cdot -5$
7.000102002793008	$4.80398336 \cdot 8$	$4.78290944 \cdot 8$	0.00010200279300764947
7.999355829607762	$1.682691072 \cdot 9$	$1.67849728 \cdot 9$	0.0006441703922384079
9.002915294362053	$4.465326592 \cdot 9$	$4.457859584 \cdot 9$	0.002915294362052734
9.990413042481725	$1.2707126784 \cdot 10$	$1.2696907264 \cdot 10$	0.009586957518274986
11.025022932909318	$3.5759895552 \cdot 10$	$3.5743469056 \cdot 10$	0.025022932909317674
11.953283253846857	$7.216771584 \cdot 10$	$7.2146650624 \cdot 10$	0.04671674615314281
13.07431403244734	$2.15723629056 \cdot 11$	$2.15696330752 \cdot 11$	0.07431403244734014
13.914755591802127	$3.65383250944 \cdot 11$	$3.653447936 \cdot 11$	0.08524440819787316
15.075493799699476	$6.13987753472 \cdot 11$	$6.13938415616 \cdot 11$	0.07549379969947623
15.946286716607972	$1.555027751936 \cdot 12$	$1.554961097216 \cdot 12$	0.05371328339202819
17.025427146237412	$3.777623778304 \cdot 12$	$3.777532946944 \cdot 12$	0.025427146237412046
17.99092135271648	$7.199554861056 \cdot 12$	$7.1994474752 \cdot 12$	0.009078647283519814
19.00190981829944	$1.0278376162816 \cdot 13$	$1.0278235656704 \cdot 13$	0.0019098182994383706
19.999809291236637	$2.7462952745472 \cdot 13$	$2.7462788907008 \cdot 13$	0.00019070876336257925

Tabela 5: Wartości wielomianów $P(x)$ i $p(x)$, i błąd wyliczenia pierwiastka

pierwiastek	wartość $P(x)$	wartość $p(x)$	błąd wyliczenia pierwiastka
0.9999999999998357	20992.0	22016.0	$1.6431300764452317 \cdot 10^{-13}$
2.0000000000550373	349184.0	365568.0	$5.503730804434781 \cdot 10^{-11}$
2.99999999660342	$2.221568 \cdot 10^6$	$2.295296 \cdot 10^6$	$3.3965799062229962 \cdot 10^{-9}$
4.000000089724362	$1.046784 \cdot 10^7$	$1.0729984 \cdot 10^7$	$8.972436216225788 \cdot 10^{-8}$
4.99999857388791	$3.9463936 \cdot 10^7$	$4.3303936 \cdot 10^7$	$1.4261120897529622 \cdot 10^{-6}$
6.000020476673031	$1.29148416 \cdot 10^8$	$2.06120448 \cdot 10^8$	$2.0476673030955794 \cdot 10^{-5}$
6.99960207042242	$3.88123136 \cdot 10^8$	$1.757670912 \cdot 10^9$	0.00039792957757978087
8.007772029099446	$1.072547328 \cdot 10^9$	$1.8525486592 \cdot 10^{10}$	0.007772029099445632
8.915816367932559	$3.065575424 \cdot 10^9$	$1.37174317056 \cdot 10^{11}$	0.0841836320674414
$10.095455630535774 - 0.6449328236240688im$	$7.143113638035824 \cdot 10^9$	$1.4912633816754019 \cdot 10^{12}$	0.6519586830380406
$10.095455630535774 + 0.6449328236240688im$	$7.143113638035824 \cdot 10^9$	$1.4912633816754019 \cdot 10^{12}$	1.1109180272716561
$11.793890586174369 - 1.6524771364075785im$	$3.357756113171857 \cdot 10^{10}$	$3.2960214141301664 \cdot 10^{13}$	1.665281290598479
$11.793890586174369 + 1.6524771364075785im$	$3.357756113171857 \cdot 10^{10}$	$3.2960214141301664 \cdot 10^{13}$	2.045820276678428
$13.992406684487216 - 2.5188244257108443im$	$1.0612064533081976 \cdot 10^{11}$	$9.545941595183662 \cdot 10^{14}$	2.5188358711909045
$13.992406684487216 + 2.5188244257108443im$	$1.0612064533081976 \cdot 10^{11}$	$9.545941595183662 \cdot 10^{14}$	2.7128805312847097
$16.73074487979267 - 2.812624896721978im$	$3.315103475981763 \cdot 10^{11}$	$2.7420894016764064 \cdot 10^{16}$	2.9060018735375106
$16.73074487979267 + 2.812624896721978im$	$3.315103475981763 \cdot 10^{11}$	$2.7420894016764064 \cdot 10^{16}$	2.825483521349608
$19.5024423688181 - 1.940331978642903im$	$9.539424609817828 \cdot 10^{12}$	$4.2525024879934694 \cdot 10^{17}$	2.454021446312976
$19.5024423688181 + 1.940331978642903im$	$9.539424609817828 \cdot 10^{12}$	$4.2525024879934694 \cdot 10^{17}$	2.004329444309949
20.84691021519479	$1.114453504512 \cdot 10^{13}$	$1.3743733197249713 \cdot 10^{18}$	0.8469102151947894

Tabela 6: Wartości wielomianów $P(x)$ i $p(x)$, i błąd wyliczenia pierwiastka po zmianie współczynników

4.2 Wnioski z zadania

Pierwszym spostrzeżeniem jest to, że wraz ze wzrostem wielkości pierwiastka rośnie wartość błędu. Jest to przede wszystkim spowodowane niedokładnością w reprezentacji współczynników wielomianów w arytmetyce $Float64$. Bardziej zastanawiająca jest wartość obu wielomianów dla pierwiastka $x_0 \approx 1$. Tak duży wynik spowodowany jest faktem, że niedokładny pierwastek wielomianu mnożony jest przez współczynnik rzędu 19!. Spostrzeżenia względem wyników drugiej części eksperymentu ze zmienionym współczynnikiem są analogiczne. Jednak tutaj wart spojrzeć, że małą zmiana, bo odjęliśmy zaledwie 2^{-23} odbiły się diametralnie na wartościach wielomianów po ewaluacji. Stąd możemy wnioskować, że problem jest źle uwarunkowany. Ostatnim, co warto zauważyć w drugiej części eksperymentu to fakt, że od x_{10} w pierwiastkach pojawia się część urojona. To również wpłynęło na wartość błędu.

5 Obliczanie wzrostu populacji (zad. 5.)

Obliczanie wzorstu populacji według rekurencyjnego wzoru:

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

, gdzie r to pewna stała, a p_0 jest wielkością populacji stanowiącą procent maksymalnej populacji dla danego stanu środowiska. Zadanie to było powdzielone na dwa eksperymenty:

1. dla $p_0 = 0.01$ i $r = 3$ używając arytmetyki $Float32$ wykonać 40 iteracji powyższego modelu, a następnie ponownie wykonać 40 iteracji, tylko podczas 10 zastosować obcięcie zostawiające jedynie 3 cyfry po przecinku
2. wykonać 40 iteracji powyższego modelu, dla tych samych danych wejściowych, jednak raz zastosować arytmetykę $Float32$, a drugi raz $Float64$

5.1 Wyniki

W poniższej tabeli zawarte są wyniki obu eksperymentów naraz. Z tych danych można wygenerować

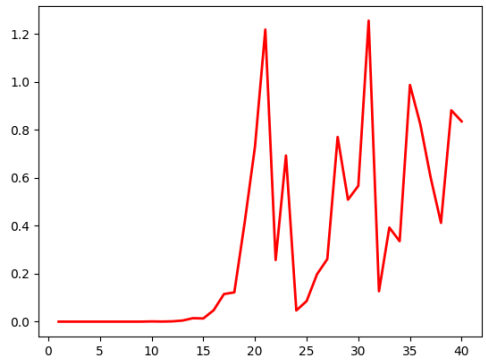
*Float*32	*Float*32 z obcięciem	*Float*64
0.01	0.01	0.01
0.0397	0.0397	0.0397
0.15407173	0.15407173	0.15407173000000002
0.5450726	0.5450726	0.5450726260444213
1.2889781	1.2889781	1.2889780011888006
0.1715188	0.1715188	0.17151914210917552
0.5978191	0.5978191	0.5978201201070994
1.3191134	1.3191134	1.3191137924137974
0.056273222	0.056273222	0.056271577646256565
0.21559286	0.21559286	0.21558683923263022
0.7229306	0.722	0.722914301179573
1.3238364	1.3241479	1.3238419441684408
0.037716985	0.036488414	0.03769529725473175
0.14660022	0.14195944	0.14651838271355924
0.521926	0.50738037	0.521670621435246
1.2704837	1.2572169	1.2702617739350768
0.2395482	0.28708452	0.24035217277824272
0.7860428	0.9010855	0.7881011902353041
1.2905813	1.1684768	1.2890943027903075
0.16552472	0.577893	0.17108484670194324
0.5799036	1.3096911	0.5965293124946907
1.3107498	0.09289217	1.3185755879825978
0.088804245	0.34568182	0.058377608259430724
0.3315584	1.0242395	0.22328659759944824
0.9964407	0.94975823	0.7435756763951792
1.0070806	1.0929108	1.315588346001072
0.9856885	0.7882812	0.07003529560277899
1.0280086	1.2889631	0.26542635452061003
0.9416294	0.17157483	0.8503519690601384
1.1065198	0.59798557	1.2321124623871897
0.7529209	1.3191822	0.37414648963928676
1.3110139	0.05600393	1.0766291714289444
0.0877831	0.21460639	0.8291255674004515
0.3280148	0.7202578	1.2541546500504441
0.9892781	1.3247173	0.29790694147232066
1.021099	0.034241438	0.9253821285571046
0.95646656	0.13344833	1.1325322626697856
1.0813814	0.48036796	0.6822410727153098
0.81736827	1.2292118	1.3326056469620293
1.2652004	0.3839622	0.0029091569028512065
0.25860548	1.093568	0.011611238029748606

Tabela 7: Wielkość populacji obliczana we *Float*32 z i bez obcięcia, i *Float*64

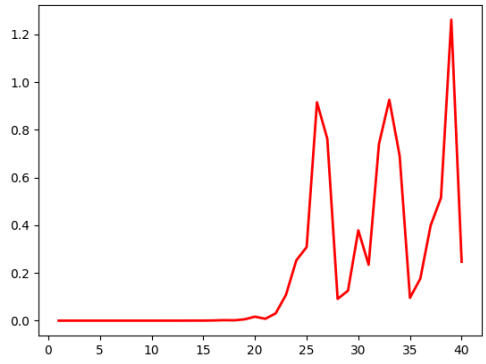
wykresy. Pierwszy odnosi się do różnicy pomiędzy liczebnością populacji z eksperymentu 1. Drugi odpowiednio do eksperymentu 2. gdzie mamy do czynienia z różnymi arytmetykami.

5.2 Wnioski z zadania

Oba wykresy pokazują nam bardzo ciekawą właściwość. Od pewnego momentu błąd pomiędzy obliczanymi wartościami początkowo wzrasta i zaczyna się wahać. Nie zostaje on w żaden sposób zniwelowany lub pochłonięty. Zaczynając od pierwszego eksperymentu. Wzrostu różnicy mogliśmy spodziewać się w okolicach 10 iteracji, gdzie obcinaliśmy obliczaną wartośc do 3 miejsc po przecin-



Rysunek 3: Różnica w liczebności od numeru iteracji dla eksperymentu 1.



Rysunek 4: Różnica w liczebności od numeru iteracji dla eksperymentu 2.

ku. Jednak takiej oscylacji już nie. Wnioskować stad możemy, że nie jesteśmy w stanie przewidzieć wzrostu populacji przy tak "dużej" zmianie, jak obcięcie do trzech liczb po przecinku. W tym momencie powinniśmy popatrzeć na wyniki drugiego eksperymentu. Niestety tam okazuje się, że mamy do czynienia z podobnym zjawiskiem pomimo, że nie wprowadzamy sami zakłóceń do danych. Zauważając fakt, że przy arytmetyce $Float64$ różnica pomiędzy wynikami wzrasta w dalszej iteracji niż w pierwszym eksperymencie pojawia się wniosek. Dysponując arytmetyką o nieskończonej precyzji moglibyśmy pozbyć się problemu wzrastającego błedu. Co więcej warto zaznaczyć, że błąd nie zostaje w żaden sposób pochłonięty, ponieważ mamy do czynienia z tzw. sprzężeniem zwrotnym - błąd, który raz się pojawił jest propagowany dalej i potęgowany. Podsumowując wcześniej nazwana "duża" zmiana z eksperymentu pierwszego jest w rzeczywistości mała. Stąd niewielka zmiana spowodowała duże odchylenie w wyniku. Stąd wnioskuję, że problem ten jest źle uwarunkowany.

6 Równań rekurencyjnych ciąg dalszy (zad. 6.)

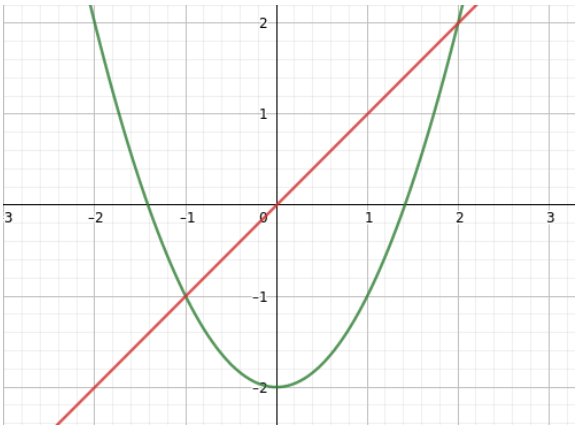
Następne spotkanie z równaniem rekurencyjnym. Tym razem jest ono dane wzorem:

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

, gdzie c jest bliżej niezidentyfikowaną zmienną. Używając arytmetyki $Float64$ przeprowadziłem 7 eksperymentów, wykonując po 40 iteracji dla każdego zestawu wejściowego:

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 $x_0 = 0.25$

6.1 Wyniki dla poszególnych eksperymentów



Rysunek 5: Wykres do iteracji graficznej.

Powyższy wykres pokazuje $\phi(x) = x^2 - 2$ or $f(x) = x$. Stosując iterację graficzną można zauważyć, że dla danych, np. $x_0 = 1 \vee x_0 = 2$ funkcja $\phi(x)$ jest zbieżna do odpowiednio -1 i 2. Natomiat np. dla 2.1 czy 1.99999999999999 już nie, ponieważ iteracja graficzna się nie kończy.

6.2 Wnioski z zadania

Wzór użyty w tej sekcji jest bardzo podobny zpunktu numerycznego do użytego w zadaniu poprzednim. Kwadrat pojawiający się w obu wzorach może być przyczyną złego uwarunkowania. Jednak wyniki z tego zadania mówiąc coś przeciwnego. Zarówno w pierwszym, jak i drugim eksperymencie można zaobserwować stabilność w obliczeniach. Z drugiej strony w trzecim eksperymencie, którego wartość początkowa jest prawie taka sama, jak drugiego, wyniki zachowują się totalnie nieprzewidywalnie. Stąd możemy wnioskować, że stabilne rozwiązania zależą od danych wejściowych. Jest to również wniosek z iteracji graficznej powyżej.

k	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	$-5.74157936927832710^{-6}$
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	$-6.59314824957846210^{-11}$
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	$-1.153618255700372710^{-6}$	-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	$-2.661648679236350310^{-12}$	-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

Tabela 8: Wyniki eksperymentu dla wszystkich 7 zestawów danych